Efficient Physics Signal Selectors for the First Trigger Level of the Belle II Experiment based on Machine Learning

Sebastian Skambraks

München 2020

Efficient Physics Signal Selectors for the First Trigger Level of the Belle II Experiment based on Machine Learning

Sebastian Skambraks

Dissertation
an der Fakultät für Physik
der Ludwig–Maximilians–Universität
München

vorgelegt von
Sebastian Skambraks
aus München

München, den Abgabedatum

Erstgutachter: Prof. Dr. Christian Kiesling
Zweitgutachter: Zweitgutachter
Tag der mündlichen Prüfung: Prüfungsdatum

Abstract

A neural network based z-vertex trigger is developed for the first level trigger of the upgraded flavor physics experiment Belle II at the high luminosity B factory SuperKEKB in Tsukuba, Japan. Using the hit and drift time information from the central drift chamber, a pool of expert neural networks estimates the 3D track parameters of the single tracks found by a 2D Hough finder. The neural networks are already implemented on parallel FPGA hardware for real time data processing and running pipelined in the online first level trigger of Belle II. Due to the anticipated high luminosity of up to $8 \times 10^{35}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$, Belle II will have to face severe levels of background tracks with vertices displaced along the beamline. The neural z-vertex algorithm presented in this thesis allows to reject displaced background tracks such that the requirements of the standard track trigger can be strongly relaxed. Especially for physics decay channels with a low track multiplicity in the final states, like τ pair production, or initial state radiation events with reduced center of mass energies, the trigger efficiencies can be significantly increased.

As an upgrade of the present 2D Hough finder in the neural network preprocessing, a model independent 3D track finder is developed that uses the additional stereo hit information of the drift chamber. Thus, the trigger efficiencies improve for tracks in the phase space of low transverse momenta and shallow polar angles. Since the cross sections of the physics signal events typically increase towards shallow polar angles, this enlarged acceptance of the track trigger provides a substantial gain in the signal efficiencies. By using an adapted pool of expert networks, the enlarged phase space provided by the 3D finder can be efficiently covered.

Studies on simulated MC background, on simulated initial state radiation events, and on recorded data from early Belle II runs demonstrate the high performance of the novel trigger algorithms. With the 3D finder an increase of the track finding rate of about 50 % is confirmed for signal tracks, while displaced background tracks are actively suppressed prior to the neural network. Based on z-vertex cuts on the tracks processed by the neural networks, a two track event efficiency of more than 99 % can be achieved with a purity of around 80 %.

Zusammenfassung

Für die Spurrekonstruktion auf der ersten Triggerstufe des neu aufgesetzten Flavor Physik Experiments Belle II an der, mit hoher Luminosität laufenden, B factory SuperKEKB in Tsukuba, Japan, ist ein auf neuronalen Netzen basierender z-Vertex Trigger entwickelt worden. Unter Verwendung der Hit- und Driftzeit-Informationen aus der zentralen Drahtkammer schätzt ein Ensemble aus neuronalen Expertennetzen die 3D Spurparameter jener Spuren, die vorab durch eine 2D Hough Transformation gefunden wurden. Die neuronalen Netze sind bereits auf paralleler FPGA Hardware zur Echtzeit-Datenverarbeitung implementiert sowie in der Pipeline der ersten Triggerstufe von Belle II in Betrieb. Aufgrund der angestrebten hohen Luminosität von $8 \times 10^{35}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$ wird Belle II mit großen Mengen an Untergrundspuren, mit entlang der Strahlspur versetzten Vertices, umgehen müssen. Der in dieser Arbeit vorgestellte, neuronale z-Vertex Algorithmus ermöglicht es, versetzte Untergrundspuren zurückzuweisen und somit die Bedingungen des konventionellen Spurtriggers stark zu lockern. Eine signifikante Steigerung der Trigger-Effizienzen kann insbesondere für Physik Kanäle mit einer niedrigen Spurmultiplizität in den Endzuständen erreicht werden, wie z.B. τ Paar Produktion oder Ereignisse mit niedrigerer Schwerpunktsenergie aufgrund von Photonabstrahlung in den Anfangszuständen.

Zur Verbesserung des bestehenden 2D Spurfinders in der Vorverarbeitung des neuronalen Netzes ist ein neuartiger, modellunabhängiger 3D Spurfinder entwickelt worden, welcher die zusätzliche stereo Hit-Information aus der Driftkammer verwendet. Dadurch wird die Effizienz des Triggers für Spuren im Phasenraumbereich kleiner Transversalimpulse sowie flacher Polarwinkel verbessert. Da die Wirkungsquerschnitte der Physik Signalereignisse typischerweise mit flachen Polarwinkeln zunehmen, trägt die erweiterte Akzeptanz des Spurtriggers substantiell zur Steigerung der Effizienzen von Signalereignissen bei. Durch die Verwendung einer angepassten Konfiguration der Expertennetze im Ensemble, kann der vom 3D Spurfinder bereitgestellte, vergößerte Phasenraumbereich effizient abgedeckt werden.

Experimentelle Studien mit simuliertem Monte Carlo Untergrund, mit simulierten Ereignissen mit Photonstrahlung im Anfangszustand, sowie mit aufgezeichneten Daten aus der frühen Datennahme bei Belle II zeigen die hohe Leistungsfähigkeit der neuen Trigger Algorithmen. Mit dem 3D Spurfinder wird eine Steigerung der Signalrate um etwa 50 % erreicht, während versetzte Untergrundspuren, noch vor dem neuronalen Netz, aktiv unterdrückt werden. Basierend auf Schnitten der von den neuronalen Netzen vorhergesagten z-Vertices kann eine Zwei-Spur Effizienz von über 99 % bei einer Reinheit von etwa 80 % erreicht werden.

Contents

Abstract v

Zusammenfassung vii

1. Introduction 1

2. Standard Model and Beyond 5
2.1. Standard Model 5
2.2. Electroweak Unification 11
2.2.1. Higgs Mechanism 12
2.2.2. CKM Mechanism 13
2.3. Flavor Physics 15
2.3.1. Meson Mixing 17
2.3.2. $\mathcal{CP}$ Violation 18
2.4. Beyond SM Physics 20
2.4.1. Signatures Towards New Physics 20
2.5. Physics Goals at Belle II - Physics Reach 22
2.5.1. Low Multiplicity Events 23
2.5.2. Muon $g-2$ 24

3. The Belle II Experiment 27
3.1. SuperKEKB Collider 29
3.1.1. Acceleration and Collisions 29
3.1.2. SuperKEKB Design 33
3.2. Belle II Geometry 36
3.2.1. Solenoid Magnetic Field 36
3.2.2. Material Budget Close to the IP 39
3.2.3. Track Parametrization 40
3.2.4. Beast Commissioning Phases 43
3.2.5. Subdetectors of Belle II 43
3.3. The Trigger System 49
3.3.1. L1 Subtriggers 51
3.3.2. CDC Trigger Pipeline 53
3.3.3. Track Segment Finder 54
3.3.4. Track Finding and Reconstruction 56
3.4. Belle II Simulation with basf2 58

4. Background at Belle II 61

4.1. Luminosity Background . . . 62

4.1.1. Radiative Bhabha Scattering . . . 62

4.1.2. Two Photon QED Events . . . 64

4.2. Machine Background . . . 65

4.2.1. Touschek Effect . . . 66

4.2.2. Beam Gas Scattering . . . 67

4.2.3. Synchrotron Radiation . . . 69

4.2.4. Further Machine Backgrounds . . . 69

4.3. Combinatorial Background Pileup . . . 70

4.4. Background Simulation . . . 71

4.4.1. Background Mixing . . . 72

4.4.2. Belle II Background Campaigns . . . 74

5. 3D Track Finder 81

5.1. Hough Transformations . . . 83

5.1.1. Straight Line 2D Hough Transformation . . . 84

5.1.2. Circular 2D Hough Transformation . . . 86

5.1.3. Helical 3D Hough Transformation . . . 91

5.2. 3D Track Finder Algorithm . . . 94

5.2.1. Probabilistic Track Finding . . . 95

5.2.2. Structure of the Hough Space . . . 99

5.2.3. CDC Symmetric Hit Representations . . . 102

5.2.4. Training of the Hit Representations . . . 109

5.2.5. Clustering and Track Estimation . . . 117

5.3. Track Finding Performance . . . 129

5.3.1. Train and Test Data . . . 130

5.3.2. Analysis Method . . . 137

5.3.3. Study of the Training Target Tracks . . . 140

5.3.4. Performance at the CDC Acceptance Boundaries . . . 148

5.3.5. Summary of the Track Finding Performance . . . 158

5.4. Optimization of the 3D Finder Parameters . . . 159

5.4.1. Properties of the Track Clusters in the 3D Hough Space . . . 160

5.4.2. Parameter Variation . . . 163

5.4.3. Summary of the Parameter Optimization . . . 172

6. Neural Network Trigger 175

6.1. Least Square Track Fit . . . 179

6.1.1. Method of Least Squares . . . 180

6.1.2. Linear Least Squares . . . 180

6.1.3. Linear 2D Circle Fits . . . 182

6.1.4. 3D Helix Fit . . . 184

6.2. Neural Network Model . . . 186

6.2.1. Supervised Machine Learning . . . 187

6.2.2. Single Perceptron Neurons 189
6.2.3. Multi Layer Perceptron Network 190
6.2.4. Input and Output Features 191
6.2.5. Training of the Network 194
6.3. Neuro Trigger Setup 198
6.3.1. Expert Neural Networks 198
6.3.2. Preprocessing 201
6.3.3. Implementation 204
6.4. Performance of the Neural Network Trigger 207
6.4.1. Tested Event Types 207
6.4.2. Single Track Simulation 208
6.4.3. Basic Network Training 217
6.4.4. Single Track Analysis 221
6.4.5. Basic Neural Network Performance 223
6.4.6. Advanced Single Track Features 232
6.5. Benefits of the 3D Finder 238
6.5.1. Comparison of the 2D Finder with the 3D Finder 239
6.5.2. Neural Network Short Track Experts 255
6.5.3. Neural Networks Theta Sector Experts 261
6.5.4. Summary of the 3D Finder Benefits 267

7. Physics Measurements 269
7.1. Initial State Radiation Events 269
7.1.1. Physics Motivation 271
7.1.2. Analysis Method 273
7.1.3. Final State $\mu^+\mu^-$ 276
7.1.4. Final State $\pi^+\pi^-$ 282
7.1.5. Final State $\pi^+\pi^-\pi^0$ 290
7.1.6. Summary of the ISR Analysis 295
7.2. Belle II Run Data 298
7.2.1. Collision Events 299
7.2.2. Background Events 313
7.2.3. Summary of the Belle II Data Analysis 323

8. Conclusion 325

A. Error Estimation 327
A.1. Statistical Errors 327
A.1.1. Gaussian Error Propagation 327
A.1.2. Moments and Variance 328
A.1.3. Standard Error 328
A.1.4. Root Mean Square 329
A.1.5. Efficiency and Rate Errors 331
A.2. Histogram Binning Error 332

A.3. Track Matching 334
A.3.1. Matching Algorithm 335

B. Appendix Background 343
B.1. Review of the Background Simulation 343
B.1.1. SAD Machine Simulation 343
B.1.2. Large Scale SAD Simulation 345
B.1.3. Background Topology 346

C. Appendix 3D Finder 349
C.1. Pseudo Code 349
C.1.1. Training 349
C.1.2. Track Finding 350
C.2. Resolution Plots at the Separate Boundary Regions 352

D. Appendix Neural Network 355
D.1. Studies of the Neural Network Training 355
D.1.1. Sample Size of the Training Data 355
D.1.2. Training Target with Reconstruction vs MC 359
D.1.3. Background in the Training and Testing 363
D.1.4. Track Segment Finder Left/Right Information 364

Bibliography 369

Chapter 1.

Introduction

The Standard Model of particle physics (SM) is the successful theoretical model of the observable elementary particles and their interactions. To this day, its validity has been verified by all experiments, even though it is known that the SM cannot be complete. Limits of the SM comprise the missing explanation of the fermion mass hierarchies and the complete absence of a mechanism generating mass terms for the neutrinos, although the existence of neutrino masses is experimentally confirmed via neutrino oscillations. Additionally, the SM fails to include a theory of gravity, the fourth force, as well as descriptions of dark matter and dark energy, which are postulated in order to explain cosmological observations. Furthermore, the SM lacks a sufficient description of the observed asymmetry between matter and anti-matter in our matter dominated universe. Given that matter and anti-matter will most likely annihilate into photons upon collisions, the matter dominance is a fundamental requirement for the formation of stable galaxies, stars, planets and our life on earth.

Three necessary conditions to create a matter / anti-matter asymmetry in the baryogenesis of the universe were postulated by Sakharov [1]: 1.) Baryon number violation, 2.) $\mathcal{C}$ asymmetry and $\mathcal{CP}$ violation, and 3.) thermal non equilibrium. The concept of $\mathcal{CP}$ violation refers to a non-invariance of the equations of motions under the combined application of the discrete charge operator $\mathcal{C}$, exchanging particles into anti-particles, and the discrete parity operator $\mathcal{P}$, acting like a point symmetric mirror that inverts the directions of vectors while it preserves the direction of axial vectors. The only observed source of $\mathcal{CP}$ violation, which is described in the SM, is the CKM mechanism of weak interactions [2]. After its experimental confirmation by the Belle and BaBar experiments, Kobayashi and Masukawa were awarded with the 2008 Nobel prize. However, the size of $\mathcal{CP}$ violation in the CKM model is not sufficient to describe the $\mathcal{CP}$ asymmetry in our observable universe. Discrepancies to the established SM must exist which manifest as new sources of $\mathcal{CP}$ violation. Several tensions with the SM exist, where small observed discrepancies indicate search directions for future experiments. However, the significances are still too low to claim new discoveries. Hence, the goal in particle physics experiments is to probe the limits of the known theory in order to find discrepancies and to establish new models of physics beyond the SM.

The Belle II detector at the SuperKEKB flavor factory in Tsukuba, Japan is an upgrade of the successful precision experiment Belle at KEKB. Luminosities of up to $8 \times 10^{35}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$ in $\mathrm{e}^+\mathrm{e}^-$ collisions at the center of mass energy of the $\Upsilon(4S)$ resonance are anticipated, a factor 40 larger than the world record luminosity reached by its predecessor Belle. A caveat to this luminosity upgrade is that it is accompanied by huge increases in the background hit and track rates, which makes noise robust 3D track triggering techniques mandatory. Background types scaling directly with the luminosity are collision events at the IP with large cross sections relative to the interesting physics decay channels. The main luminosity background tracks are backscattered particles from $\mathrm{e}^+\mathrm{e}^- \to \mathrm{e}^+\mathrm{e}^-$ collisions, the Bhabha scattering, and two photon interactions ($\mathrm{e}^+\mathrm{e}^- \to \mathrm{e}^+\mathrm{e}^-\mathrm{e}^+\mathrm{e}^-$). Additionally, severe levels of single beam background are created by lost beam particles hitting the beam wall. The main candidates are intra-beam scattering including the Touschek effect, which is in particular enhanced by the squeezing of the bunches at the IP, and beam gas scattering due to imperfect vacuum conditions. In the course of the Belle II runtime, the background conditions are changing due to the slow increase of the luminosity.

Novel triggering techniques coping with the severe background conditions are crucial to maximize the signal to background ratio in the recorded Belle II data. The maximum first level trigger rate at Belle II of 30 kHz is far below the expected total rate of signal and background events. The track trigger algorithms developed in this thesis introduce additional means for signal selection and background suppression. Thus, the remaining trigger conditions can be relaxed to increase the signal efficiency. In particular for signal events with a low track multiplicity, as expected in rare new physics scenarios like events including dark matter particles, or lepton flavor violation in τ decays. Furthermore, low multiplicity triggers allow for precision measurements of hadronic cross sections in initial state radiation events which are a crucial input for the clarification of a possible discrepancy of the anomalous magnetic moment of the muon.

The machine learning based track trigger algorithms presented in this thesis allow a significant improvement of the signal efficiencies for events with a low track multiplicity and enlarge the acceptance region of the online first level track trigger. In the course of this thesis, the main project is a neural network trigger for robust 3D single track reconstruction. To date the neural network trigger is already operational in the L1 trigger pipeline where it is implemented in FPGA hardware. Using the hit and drift time information from the central drift chamber (CDC), this supervised machine learning algorithm enriches the 2D tracks found by a Hough track finder with precise 3D track parameter estimates. Since the collision point is well known, tracks with displaced z-vertices can clearly be rejected as background. Given the limited computing resources available at the first trigger level and the fuzziness of the input data due to the presence of background noise, the neural networks are beneficial in contrast to a 3D track fit, which is optimal under ideal conditions only.

In order to optimize the neural network input, a new 3D track finder is developed as an alternative to the existing 2D track finder. By using additional stereo wire hits from the CDC, the track finding efficiency for low momentum tracks and for tracks with shallow polar angles can be significantly improved. At the same time, background tracks displaced from the IP are actively suppressed prior to the neural network. The additionally estimated 3D track parameters of the found tracks can be used in the neural network preprocessing for an improved 3D track reconstruction.

The first chapters of this thesis are dedicated to theoretical and technical details concerning Belle II. Chapter 2 introduces the flavor physics sector within the SM which motivates the construction of the Belle II experiment. It closes with a summary of important low multiplicity event types where significant efficiency improvements can be achieved with the novel algorithms. In Chap. 3 the Belle II experiment at the SuperKEKB collider are described from a technical viewpoint as required for the simulation studies. This includes an introduction to the concepts for the analysis of the trigger tracking. The expected background processes at Belle II and their simulation are described in Chap. 4. In preparation for the 3D finder and neural network studies, the trigger track rates and hit rates in the subsequently used background samples are studied.

The core algorithm of the novel 3D track finder algorithm as well as basic performance studies are presented in Chap. 5. After a review of the analytic Hough transformation method, the model independent construction of hit patterns from a training data set and the track finding with a novel density based clustering algorithm are described. This includes studies of the newly introduced hyperparameters and the selection of the default training setup of the 3D finder used in the integrated studies with the neural network.

Chapter 6 is dedicated to the neural network algorithm and its main achievements. An introduction to the least square fitting of helical tracks in 3D motivates the linearization of the single track hit information in the neural network preprocessing, by using the 2D track parameters provided by the track finder. Initial studies on the integrated performance of the neural network trigger and the 3D finder are carried out with simulated single track events. In particular, the adapted pool of expert networks is developed, which allows to make full use of the improved 3D finder efficiency.

Studies with realistic physics test cases are presented in the last Chap. 7, demonstrating the significant improvement in the track finding efficiency of the 3D finder as well as its capabilities for 3D background suppression prior to the neural network. A comparison with the 2D finder as input to the neural network validates the 3D finder as a viable alternative to the existing 2D finder. Additionally, based on the neural network estimated, efficient event trigger logics for signal selection and background suppression are studied in combination with both track finders.

Chapter 2.

Standard Model and Beyond

The Standard Model of particle physics (SM) is a gauge theory of the elementary fermionic particles and their interactions via gauge bosons. It covers three out of the four known forces, the electromagnetic force, the weak force and the strong force. Gravity, the fourth force, is not included in the SM, since it is so much weaker compared to the other forces (hierarchy problem) and therefore has no measurable effect on the traditional scattering processes in High Energy Physics (HEP). The theory of weak interactions with the $V - A$ (vector minus axial vector) structure was developed in 1958 [3] in order to explain the observed parity violation in weak interactions. The electromagnetic and weak interactions were unified into the electroweak interaction via the Glashow-Weinberg-Salam model (GWS) [4, 5, 6]. In 1973 the strong interactions were described by an octet of gluons coupling to color charges, within the framework of Quantum Chromo Dynamics (QCD) [7].

The predictions of the SM are tested by many experiments, which can be grouped into two main strategies: high energy experiments aiming at the direct production of particles in the center of mass system and high precision experiments that allow an indirect measurement of virtual particles also in higher order correction terms. Although the SM is successful in predicting most of the physics interactions of the elementary particles, it is known to be incomplete.

In Sec. 2.1, the main parts of the SM are reviewed from a phenomenological and from a theoretical viewpoint. In Sec. 2.4 follows a discussion of the limits of the SM and possible New Physics (NP) scenarios that aim to fill in the missing gaps. This chapter closes with Sec. 2.5 where the experimental approaches to test the physics beyond the SM are discussed, with emphasis on the flavor physics sector accessible with the Belle II experiment.

2.1. Standard Model

The SM is a renormalizable, relativistic and gauge invariant Quantum Field Theory (QFT). It describes elementary particle fields and their interactions in a Lagrangian formalism. The gauge invariance describes the invariance of the Lagrangian under phase transformations in the symmetry group $\mathcal{SU}(3) \times \mathcal{SU}(2)_L \times \mathcal{U}(1)$ [8]. $\mathcal{SU}(3)$ (Special Unitary group of order 3) is the symmetry group of the strong interaction coupling to the three color charges of the quarks. $\mathcal{SU}(2)_L \times \mathcal{U}(1)$ is the unified

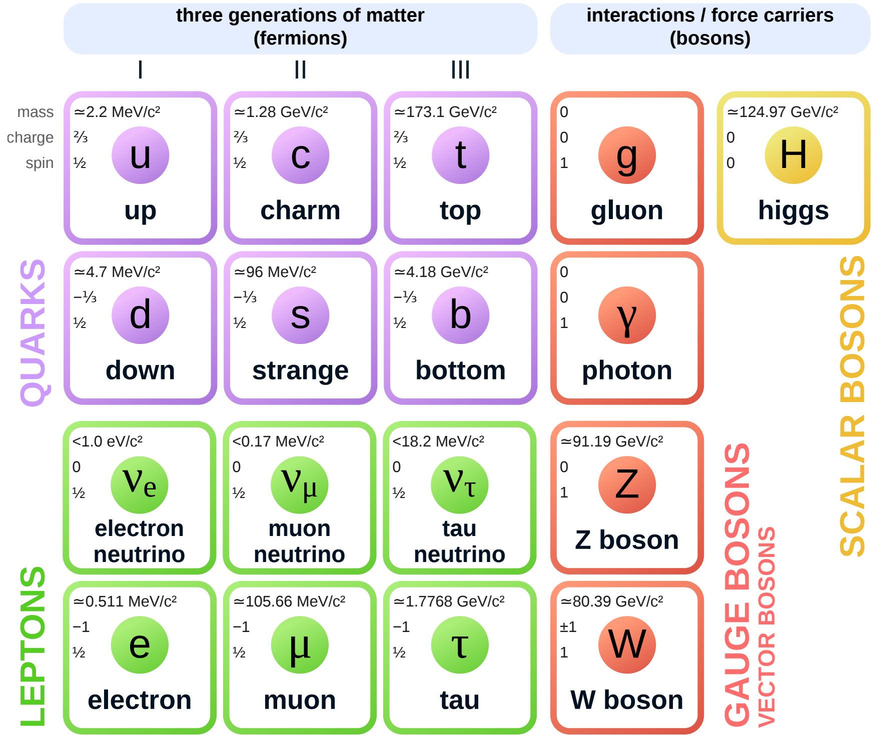


Figure 2.1.: The SM particles with three generations of the elementary fermions (leptons and quarks), the gauge bosons as force carriers of the electroweak and strong interaction and the Higgs boson generating the mass terms [9].

symmetry group of the electroweak interactions coupling to the weak charge Y_w, composed of the electric charge Q and the third component I_3 of the weak isospin I_w. The abelian $\mathcal{U}(1)$ (Unitary group) is the electromagnetic part, $\mathcal{SU}(2)$ and $\mathcal{SU}(3)$ are non-abelian. The $\mathcal{SU}(2)$ symmetry is spontaneously broken by the Higgs mechanism. It generates the masses of the gauge bosons and of the elementary particles by coupling to the Higgs field. An overview of the elementary particles in the SM is shown in Fig. 2.1.

The elementary constituents of matter are two classes of fermions (spin $\frac{1}{2}$ particles): the quarks and the leptons. While leptons[1] are observed freely, quarks are always confined into hadrons[2]. They are bound together via the strong force to mesons (quark-antiquark states) or to baryons (three quark states). There are 12 el-

[1] Greek $\lambda\epsilon\pi\tau o\varsigma$ „thin”

[2] Greek ὰδρος „thick”

ementary fermions: the quarks and leptons come in three generations of doublets, leading to a total of six so called flavors for each fermion class. Additionally, each elementary fermion has its corresponding anti-particle leading to a total of 24 elementary fermions. All ordinary matter elements is constructed of particles from the first generation. The hull electrons (e^-) in the atoms are leptons of the first generation. The atomic nuclei are composed of protons (p) and neutrons (n), which are baryons composed of up quarks (u) and down quarks (d) from the first generation. The quark content of the proton is (uud), the neutron has (udd).

From the gauge symmetries of the Lagrangian, the interactions of the free fields can be derived by requiring the global symmetries to hold also locally [8]. This formally introduces the gauge fields which restore the invariance of the Lagrangian under local phase transformations. The field quanta of the gauge fields are the gauge bosons, the carriers of the force in the SM. According to the Noether theorem each symmetry results in a conserved current. Here each gauge boson only couples to a force specific charge, a conserved quantum number emerging from the symmetry.

From the unified $\mathcal{SU}(2)_L \times \mathcal{U}(1)$ symmetry follow the electroweak gauge bosons, the weak hypercharge Y_w and the electric charge Q. The weak hypercharge Y_w is related to the electric charge Q via the third component of the weak isospin I_3. The electric charge is given by the relation

$$Q = I_3 + \frac{1}{2} Y_w. \tag{2.1}$$

The massless photon (γ) is the carrier of the electromagnetic force, coupling to the electric charge of the particles. The massive W^+, W^-, and Z^0 bosons are the three carriers of the weak force coupling to the weak hypercharge Y_w and the weak isospin (I_w). The charged currents W^+, W^- (with an electric charge of ±1) couple only to left-handed doublets of the weak isospin. The neutral current Z^0, like the photon γ, does not distinguish between left and right handed particles and couples to I_w singlet states. Therefore, the weak interaction arranges the elementary fermions into left handed doublets and right handed singlet. In leading order, only the charged currents can mediate a flavor change between an I_w up-type and an I_w down-type fermion. Flavor changing neutral currents mediated by the neutral Z^0 boson are forbidden in lowest order in the SM by the GIM mechanism (Glashow, Iliopolus, Maiani) [10]. In the case of leptons, the coupling to the gauge bosons is independent of the flavor, a feature known as lepton universality.

From the $\mathcal{SU}(3)$ symmetry follow the eight gluon fields, with massless gluon gauge bosons as field quanta, coupling to the three color charges of the quarks. Since leptons do not carry a color charge, they do not couple to the gluons and therefore do not interact via the strong interaction. The gluon potential increases with the distance of the quarks, with the result that the colored quarks are always confined within colorless hadrons. If quarks are pulled apart until they are separated, new quark-antiquark-pairs are created from the potential energy of the gluon field to confine the separated quarks into new hadrons.

Without further couplings, the gauge bosons are all massless. In order to explain the massive weak gauge bosons, the SM includes the Higgs mechanism [11] with a scalar Higgs boson (spin 0). It was the last missing piece of the SM which is responsible for the generation of all mass terms.

Fields and Interactions

The SM is a renormalizable relativistic QFT, where the elementary particle fields appear as wave functions of the 4-dimensional spacetime x^μ with the Lorentz index μ. There are three main field types in the SM:

$\phi(x)$:	scalar fields, spin 0, (Higgs boson)
$\psi(x)$:	Dirac fields, spin $\frac{1}{2}$ (fermions)
$A_\mu(x)$:	vector fields, spin 1, (gauge bosons)

The Lagrangian densities for the free fields are given by [8]

$$\mathscr{L}_{\text{scalar}} = \frac{1}{2}(\partial_\mu \phi)^2 - m^2\phi^2 \tag{2.2}$$

$$\mathscr{L}_{\text{Dirac}} = \overline{\psi}(\mathrm{i}\gamma^\mu\partial_\mu - m)\psi \tag{2.3}$$

$$\mathscr{L}_{\text{vector}} = -\frac{1}{4}F_{\mu\nu}F^{\mu\nu} - \frac{1}{2}m^2 A_\mu A^\mu \tag{2.4}$$

where the mass of the particles corresponding to each field, is described by m. $F_{\mu\nu} = \partial_\mu A_\nu - \partial_\nu A_\mu$ is the field strength tensor of the free vector field. Formally the gauge interactions are derived, by requiring gauge invariance of the fermionic Lagrangian $\mathscr{L}_{\text{Dirac}}$ under local phase transformations. After applying a phase transformation of the fermionic wave function within a symmetry group ($\mathcal{SU}(3)$, $\mathcal{SU}(2)$, $\mathcal{U}(1)$), the invariance of the Lagrangian needs to be restored. This is achieved by introducing covariant derivatives, containing the generators of the respective symmetry groups and a gauge vector field.

$$\psi(x) \rightarrow \psi'(x) = \mathrm{e}^{\mathrm{i}\alpha_j\sigma_j}\psi \qquad\qquad \partial_\mu \rightarrow \mathrm{D}_\mu = \partial_\mu - \mathrm{i}g\sigma_j W^j_\mu \tag{2.5}$$

where $\psi'(x)$ is the fermionic wave function after a phase transformation with phases α_j in a symmetry group with generators σ_j. D_μ is the covariant derivative, where the vector field W^j_μ couples with strength g in order to restore gauge invariance of the transformation. In the case of the $\mathcal{U}(1)$ symmetry, it is only a single phase α with the photon γ as vector boson. In the case of $\mathcal{SU}(2)$ and $\mathcal{SU}(3)$ the gauge terms are non-abelian which introduces the possibility of self couplings for the weak and strong gauge bosons.

The gauge bosons introduced with this formalism are strictly massless, since the explicit mass term in Eq. 2.4 would break the gauge invariance. Instead, the mass terms of the massive weak gauge bosons are introduced via a coupling to the Higgs field.

Feynman diagrams

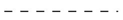
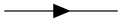
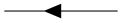
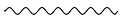
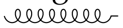
field	particles	diagram
scalar	Higgs	ϕ^0 
dirac	fermion $(\mathrm{u}, \mathrm{e}^-, \ldots)$	$\psi^\pm$ 
dirac	anti fermion $(\bar{\mathrm{u}}, \mathrm{e}^+, \ldots)$	$\bar{\psi}^\pm$ 
vector	electroweak $(\gamma, \mathrm{Z}^0, \mathrm{W}^+, \mathrm{W}^-)$	$\gamma, \mathrm{W}^\pm, \mathrm{Z}^0$ 
vector	strong (gluon g)	g 

Table 2.1.: Feynman diagram notation for the propagators of the fields.

Using the Euler Lagrange equation

$$\partial_\mu \frac{\partial \mathscr{L}}{\partial(\partial_\mu \phi)} = \frac{\partial \mathscr{L}}{\partial \phi} \tag{2.6}$$

the equations of motion of a field ϕ can be calculated. A solution to the equations of motions is given by the Feynman diagrams, with an intuitive visual representation and an exact mathematical expression. The diagrams have a horizontal time axis and a vertical space axis. External lines are the incoming and outgoing observable particles. Internal lines are virtual particles representing the propagators. They are connected with interaction vertices where the gauge bosons couple to the interaction specific charges. The Feynman notation for the different field propagators is shown in Tab. 2.1. A list of important interaction vertices is shown in Tab. 2.2. For Dirac particles, the arrow points into the direction of time and for Dirac anti-particles the arrow points in the opposite direction of time. Formally they appear in the Dirac equation as particles with negative energy traveling backwards in time.

The diagrams can be used to calculate amplitudes of the interactions, the transition rates and cross sections. Using the Feynman rules, for each diagram a matrix element $\mathscr{M}_{fi}$ for the transition from an initial state i to a final state f can be written down. The total amplitude for a transition $i \to f$ is given by the square of the sum of all possible matrix elements $|\sum \mathscr{M}_{fi}|^2$. To calculate an amplitude, at first, one has to draw all possible Feynman diagrams with the same initial and final state configuration. In general, an infinite number of diagrams is contributing to each transition. But since each vertex enters with a specific coupling strength, the higher order diagrams have a smaller contribution. The renormalizability of the SM guarantees the convergence

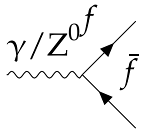
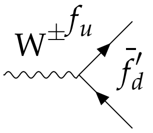
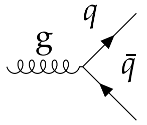
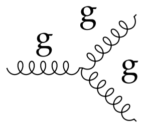
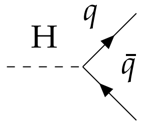
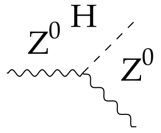
vertex	diagram
electroweak neutral current	
electroweak charged current (for weak isospin doublets $(f_u, f_d'))$	
gluon quark coupling	
gluon self coupling	
Higgs quark coupling	
Higgs radiation	

Table 2.2.: Feynman diagram notation for some important interaction vertices.

upon the infinite sum of higher order contributions. In practical calculations only the largest contributions are considered, i.e. the first terms of the perturbative series: Leading Order (LO), Next to Leading Order (NLO), Next to Next to Leading Order (NNLO), ... Using Fermi's golden rule, transition rates can be calculated from the matrix elements.

Discrete Symmetries

Discrete symmetries of the Lagrangian interaction are important in the description of flavor physics. The $\mathcal{C}$ (charge) operator inverts all the charge quantum numbers of the particles. This means it converts particles into their anti-particles. The $\mathcal{P}$ (parity) operator acts like a point symmetry, changing the sign of all vectors and pseudoscalars, while the sign of scalars and pseudovectors is preserved. Inversion of the direction of time is described by the time operator $\mathcal{T}$. While energy conservation implies invariance under the combined symmetry $\mathcal{CPT}$ violations of the $\mathcal{CP}$ symmetry are possible in the SM.

2.2. Electroweak Unification

The Higgs mechanism was proposed by Peter Higgs in 1964 [11] in order to generate the mass terms for the fermion and vector boson fields. It describes the coupling of a scalar field ϕ, the Higgs field, which is a doublet under the $\mathcal{SU}(2)$ symmetry. Using the Higgs mechanism, the theoretically associated vector bosons of the unified electroweak interactions are rotated into the physically observed bosons and the CKM matrix can be described as a rotation of the quark mass eigenstates into their weak eigenstates.

The complete electroweak Lagrangian is a sum of the Lagrangians for: the gauge bosons $\mathscr{L}_\mathrm{G}$ (free vector field, Eq. 2.4), the Higgs $\mathscr{L}_\mathrm{H}$ (free scalar field, Eq. 2.2), the fermion $\mathscr{L}_\mathrm{F}$ (free Dirac field, Eq. 2.3 and the Yukawa interaction $\mathscr{L}_\mathrm{Y}$ to generate the fermion mass terms via a coupling to the Higgs field (see Sec. 2.2.1, Eq. 2.26) [8].

$$\mathscr{L}_\mathrm{EW} = \mathscr{L}_\mathrm{G} + \mathscr{L}_\mathrm{H} + \mathscr{L}_\mathrm{F} + \mathscr{L}_\mathrm{Y} \tag{2.7}$$

The gauge field interaction term for the combined non-abelian $\mathcal{SU}(2) \times \mathcal{U}(1)$ group has the three isospin generators σ_i for the $\mathcal{SU}(2)$ weak isospin doublets I_w and the weak hypercharge Y_w for the $\mathcal{U}(1)$ group. The associated vector fields are the vector field triplets W^i_μ for the $\mathcal{SU}(2)$ group and the vector singlet B_μ for the $\mathcal{U}(1)$ group.

To define the interaction terms of the associated gauge bosons with the fermions, left handed and right handed fermions have to be distinguished. While the W^i_μ bosons fields only couple to left handed doublets of weak isospin, the B_μ boson field couples to the weak hypercharge of left and right handed particles. Right handed particles are singlets under the weak interaction. Using the minimal substitution the combined covariant derivative is

$$\mathrm{D}^L_\mu = \partial_\mu - \frac{\mathrm{i}}{2} g_2 I^i W^i_\mu + \frac{\mathrm{i}}{2} g_1 Y_w B_\mu \tag{2.8}$$

$$\mathrm{D}^R_\mu = \partial_\mu + \frac{\mathrm{i}}{2} g_1 Y_w B_\mu \tag{2.9}$$

This leads to the full interaction term of the fermions with associated vector fields B_μ, W^i_μ [8]

$$\mathscr{L}_\mathrm{F} = \overline{\Psi}^a_L \mathrm{i}\gamma^\mu \mathrm{D}^L_\mu \Psi^a_L + \overline{\Psi}^a_{R\sigma} \mathrm{i}\gamma^\mu \mathrm{D}^R_\mu \Psi^a_{R\sigma} \tag{2.10}$$

where the index a denotes a summation over the three flavor generations. The left handed fermion fields are doublets of weak isospin, while the right handed fermions are singlets. Therefore, the right handed fields have the additional index σ for the summation over up- and down-type fermions. Again, mass terms are not present here as they would break gauge symmetry and the associated vector fields are not yet the observed gauge boson fields. After introducing the Higgs mechanism to generate masses for the electroweak bosons, the rotation of the fields B_μ, W^i_μ into the observed gauge bosons $\gamma, \mathrm{Z}^0, \mathrm{W}^+, \mathrm{W}^-$ (with the Weinberg angle θ_W) will be explained.

2.2.1. Higgs Mechanism

The Higgs mechanism spontaneously breaks the $\mathcal{SU}(2) \times \mathcal{U}(1)$ symmetry via a coupling to a doublet of scalar fields with hypercharge $Y_w = 1$.

$$\Phi = \begin{pmatrix} \phi^+ \\ \phi^0 \end{pmatrix} \tag{2.11}$$

The explicit mass term in the scalar Lagrangian is replaced by the Higgs potential for Higgs self coupling including tertiary and quartic couplings:

$$V(\Phi) = -\mu^2 \Phi^\dagger \Phi + \frac{\lambda}{4} (\Phi^\dagger \Phi)^2 \tag{2.12}$$

where two new constants are introduced: λ, μ^2. The Lagrangian is

$$\mathscr{L}_\mathrm{H} = (\mathrm{D}_\mu \Phi)^\dagger (\mathrm{D}_\mu \Phi) - V(\Phi) \tag{2.13}$$

For non-zero fields Φ, the minimum of the potential is found at

$$\Phi^\dagger \Phi = \frac{2\mu^2}{\lambda} = \frac{v^2}{2} \tag{2.14}$$

Using the electric charge operator $Q = \frac{1}{2}(\sigma_z + \sigma_0)$ to select the electrically neutral Higgs field, the vacuum expectation value is

$$\langle \Phi \rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ v \end{pmatrix} \tag{2.15}$$

In order to expand the Higgs field around this vacuum expectation value, it can be rewritten as [8]

$$\Phi = \begin{pmatrix} \phi^+ \\ \frac{1}{\sqrt{2}}(v + H + \mathrm{i}\chi) \end{pmatrix} \tag{2.16}$$

The fields ϕ^+ and χ are unphysical degrees of freedom which can be eliminated by a gauge transformation (Goldstone bosons). In the unitarity gauge, the Higgs doublet is

$$\Phi = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 \\ v + H \end{pmatrix} \tag{2.17}$$

where H describes the real massive Higgs boson. The Higgs potential becomes [8]

$$V(\Phi) = 2\mu^2 \left(\frac{H^4}{8v^2} + \frac{H^3}{2v} + \frac{H^2}{2} \right) \tag{2.18}$$

where $2\mu^2 = M_\mathrm{H}^2$ is the squared mass of the Higgs boson. The couplings to the gauge bosons can be obtained from the kinetic part in the Lagrangian $(\mathrm{D}_\mu \Phi)^\dagger (\mathrm{D}_\mu \Phi)$ yielding HWW, HZZ, $HHWW$, $HHZZ$ vertices.

The Higgs interaction term with the gauge fields has the following mass term

$$\frac{v^2}{8}g_2^2(W_1^2 + W_2^2)(W_\mu^3, B_\mu)\begin{pmatrix} g_2^2 & g_1g_2 \\ g_1g_2 & g_1^2 \end{pmatrix}\begin{pmatrix} W^{3,\mu} \\ B^\mu \end{pmatrix} \tag{2.19}$$

A diagonalization yields the mass eigenstates of the gauge bosons.

$$W_\mu^\pm = \frac{1}{2}(W_\mu^1 \mp \mathrm{i}W_\mu^2) \tag{2.20}$$

$$\begin{pmatrix} Z_\mu \\ A_\mu \end{pmatrix} = \begin{pmatrix} \cos(\theta_\mathrm{W}) & \sin(\theta_\mathrm{W}) \\ \sin(\theta_\mathrm{W}) & \cos(\theta_\mathrm{W}) \end{pmatrix}\begin{pmatrix} W_\mu^3 \\ B_\mu \end{pmatrix} \tag{2.21}$$

where $\cos(\theta_\mathrm{W})$ is the Weinberg angle. It rotates the gauge fields W_μ^3 and B_μ^3 to the physical fields A_μ (massless photon field) and Z_μ (massive neutral current of the weak interaction). The diagonalized mass term becomes

$$M_\mathrm{W}^2 W_\mu^+ W^{-\mu} + \frac{1}{2}(A_\mu Z_\mu)\begin{pmatrix} 0 & 0 \\ 0 & M_\mathrm{Z}^2 \end{pmatrix}\begin{pmatrix} A^\mu \\ Z^\mu \end{pmatrix} \tag{2.22}$$

with the observable masses M_W and M_Z of the weak gauge bosons

$$M_\mathrm{W} = \frac{1}{2}g_2 v \qquad\qquad M_\mathrm{Z} = \frac{1}{2}\sqrt{g_1^2 + g_2^2}\,v \tag{2.23}$$

The Weinberg angle θ_W can be measured by measuring the mass of the $W^\pm$ and Z^0 bosons

$$\cos(\theta_\mathrm{W}) = \frac{g_2}{\sqrt{g_1^2 + g_2^2}} = \frac{M_\mathrm{W}}{M_\mathrm{Z}} \tag{2.24}$$

Identifying A_μ with the photon field also provides a relation for the electric charge

$$e = \frac{g_1 g_2}{\sqrt{g_1^2 + g_2^2}} = g_1 \cos(\theta_\mathrm{W}) \tag{2.25}$$

2.2.2. CKM Mechanism

The foundation of the CKM mechanism was given by the Cabibbo angle θ_C. It was proposed by Cabibbo in 1963 [12], at a time when only three quarks were known $(\mathrm{u}, \mathrm{d}, \mathrm{s})$. With θ_C the observed smallness of flavor changing weak interactions involving strange quarks $(\mathrm{s} \to \mathrm{u})$ compared to transitions within the first generation $(\mathrm{d} \to \mathrm{u})$ could be explained. Later θ_C turned out to be the mixing angle between the first two generations (u, d) with (c, s) within the CKM mechanism.

$\mathcal{CP}$ violation in weak interactions was first observed in the Kaon system by Cronin and Fitch in 1964 [13], who were rewarded with the Nobel prize in 1980. Later this observation of $\mathcal{CP}$ violation in weak interactions was the motivation for Kobayashi and Masukawa [2] to postulate the CKM mechanism in weak interactions. At least three quark generations are required to allow for $\mathcal{CP}$ violating terms. At that time, still only three quarks were discovered $(\mathrm{u}, \mathrm{d}, \mathrm{s})$. The second generation with a fourth quark c was postulated by Glashow, Iliopolus, Maiani in 1970 in order to explain the suppression of $\mathrm{K}^0 \rightarrow \mu^+\mu^-$ decays via a destructive interference term [10].

The CKM matrix is a complex 3×3 unitary matrix describing the mixing between the quark weak eigenstates and the mass eigenstates. The mass terms of the fermions are obtained from a Yukawa coupling of the fermions to the Higgs field. With the left handed lepton and quark doublets ($L_L = (\nu_L, l_L)^T, Q_L = (u_L, d_L)^T$), the right handed quark and lepton singlets (u_R, d_R, l_R) and the color conjugate of the Higgs field ($\Phi^c = i\sigma_2\Phi$) the Yukawa interaction of the scalar Higgs field with the massive fermions is expressed by

$$\mathscr{L}_Y = -g_l \overline{L}_L \Phi l_R - g_d \overline{Q}_L \Phi d_R - g_u \overline{Q}_L \Phi^c u_R + h.c. \tag{2.26}$$

In the SM, neutrinos are assumed to be massless. Therefore, a Yukawa term coupling right handed neutrinos to the Higgs field is omitted. The fermion masses are proportional to fermion-specific coupling constants g_f:

$$m_f = g_f \frac{v}{\sqrt{2}} \tag{2.27}$$

and the Yukawa Lagrangian in the unitarity gauge becomes

$$\mathscr{L}_Y = -m_f \overline{\psi}_f \psi_f - \frac{1}{v} \overline{\psi}_f \psi_f H \tag{2.28}$$

With three generations of quarks, flavor mixing needs to be considered. It is achieved by replacing the Yukawa couplings with matrices $g_u \rightarrow g^u_{ij}, g_d \rightarrow g^d_{ij}$. The quark Yukawa Lagrangian with three generations is then

$$\mathscr{L}^q_Y = -g^d_{ij} \overline{Q}^i_L \Phi d^j_r - g^u_{ij} \overline{Q}^i_L \Phi^c u^j_R + h.c. \tag{2.29}$$

Using the vacuum expectation value of Φ, leads to the mass terms of the fermions.

$$\mathscr{L}_Y = -\frac{v}{\sqrt{2}} \left(g^d_{ij} \overline{d}^i_L d^j_R + g^u_{ij} \overline{u}^i_L u^j_R \right) + h.c. \tag{2.30}$$

The mass eigenstates of the fermions are obtained by diagonalizing this term with the help of unitary matrices $V^{u,d}_{L,R}$.

$$\tilde{u}^i_{L,R} = (V^u_{L,R})_{ik} u^k_{L,R} \qquad \tilde{d}^i_{L,R} = (V^d_{L,R})_{ik} d^k_{L,R} \tag{2.31}$$

The fermion masses of the diagonalized mass matrices are

$$\mathrm{diag}(m_q) = \frac{v}{\sqrt{2}} V^q_L g_q V^{q\dagger}_R \tag{2.32}$$

The flavor diagonal interactions with the neutral gauge bosons as well as the Yukawa Higgs coupling to the mass eigenstates remain unchanged. But the flavor changing interaction with the left handed fermion mass eigenstates leads to the CKM matrix

$$\mathbf{V}_{\mathrm{CKM}} = V_L^u V_L^{d\dagger} \tag{2.33}$$

It is a unitary complex 3×3 matrix. Following from the unitarity condition, it has 4 free real parameters: three mixing angles and one complex phase. The CKM matrix rotates the mass eigenstates of the down-type quarks $(\mathrm{d}, \mathrm{s}, \mathrm{b})$ to the weak states $(\mathrm{d}', \mathrm{s}', \mathrm{b}')$. These couple via the weak charged currents exactly to the mass eigenstates of the up-type quarks $(\mathrm{u}, \mathrm{c}, \mathrm{t})$:

$$\begin{pmatrix} \mathrm{d}' \\ \mathrm{s}' \\ \mathrm{b}' \end{pmatrix} = \underbrace{\begin{pmatrix} V_{\mathrm{ub}} & V_{\mathrm{us}} & V_{\mathrm{ub}} \\ V_{\mathrm{cd}} & V_{\mathrm{cs}} & V_{\mathrm{cb}} \\ V_{\mathrm{tu}} & V_{\mathrm{ts}} & V_{\mathrm{tb}} \end{pmatrix}}_{\mathbf{V}_{\mathrm{CKM}}} \begin{pmatrix} \mathrm{d} \\ \mathrm{s} \\ \mathrm{b} \end{pmatrix} \tag{2.34}$$

The complex phase in the CKM matrix is responsible for the $\mathcal{CP}$ violation in the weak interactions.

2.3. Flavor Physics

Flavor physics is concerned with the different flavors of the elementary particles where flavor refers to the generations of particles. In the quark sector, there are three generations of weak $\mathcal{SU}(2)$ doublets of up- and down-type quarks. While the mass eigenstates of the quarks coincide with the eigenstates of the strong and the electromagnetic interaction, the weak eigenstates are an admixture of the mass eigenstates. Thus, flavor violating interactions are possible in weak interactions.

The CKM matrix is a rotation matrix with three angles and a complex phase. The standard parametrization used in the PDG [14] is

$$\mathbf{V}_{\mathrm{CKM}} = \begin{pmatrix} c_{12}c_{13} & s_{12}c_{13} & s_{13}e^{-i\delta} \\ -s_{12}c_{23} - c_{12}s_{23}s_{13}e^{i\delta} & c_{12}c_{23} - s_{12}s_{23}s_{13}e^{i\delta} & s_{23}c_{13} \\ s_{12}s_{23} - c_{12}c_{23}s_{13}e^{i\delta} & -c_{12}s_{23} - s_{12}c_{23}s_{13}e^{i\delta} & c_{23}c_{13} \end{pmatrix} \tag{2.35}$$

A common representation of the CKM matrix is the Wolfenstein parametrization. It is a Taylor expansion of the CKM matrix around the Cabibbo angle $\lambda = \theta_{\mathrm{C}}$ (mixing angle of the first two generations [12]) up to the third order.

$$\mathbf{V}_{\mathrm{CKM}} = \begin{pmatrix} 1 - \frac{1}{2}\lambda^2 & \lambda & A\lambda^3(\rho - i\eta) \\ -\lambda & 1 - \frac{1}{2}\lambda^2 & A\lambda^2 \\ A\lambda^3(1 - \rho - i\eta) & -A\lambda^2 & 1 \end{pmatrix} \tag{2.36}$$

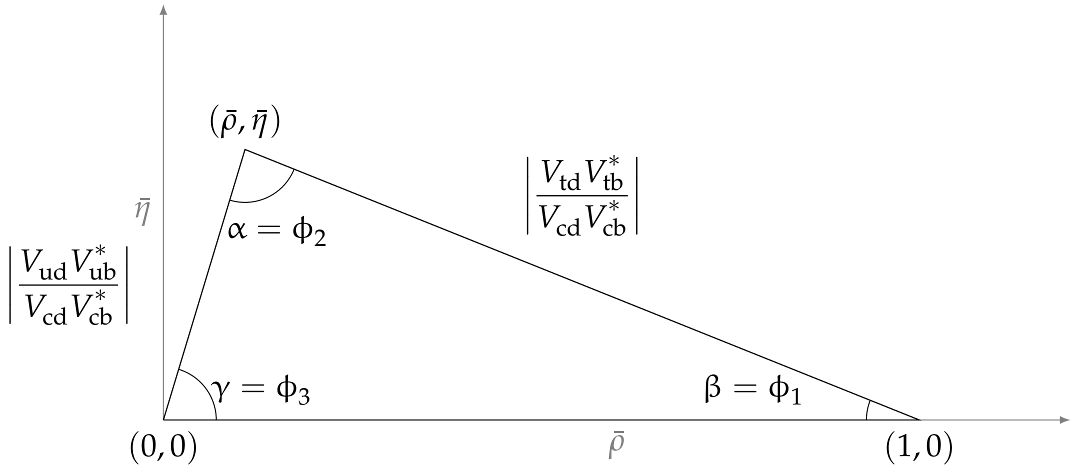


Figure 2.2.: Unitarity triangle of the first and third column of the CKM matrix.

The unitarity of the CKM matrix can be expressed as [14]

$$V_{ij}V_{ik}^* = \delta_{jk} \qquad V_{ij}V_{kj}^* = \delta_{ik} \tag{2.37}$$

where δ_{ij} is the Kronecker delta. This leads to six vanishing equations which can be represented as triangles in the complex $\rho\eta$ plane. All the triangles have the same area, $\frac{1}{2}J$, with the Jarlskog determinant J [15]. The Jarlskog determinant is defined by

$$\mathrm{Im}[V_{ij}V_{kl}V_{il}^*V_{kj}^*] = J\epsilon_{ikm}\epsilon_{jln}. \tag{2.38}$$

It is a measure of the $\mathcal{CP}$ violation in the weak mixing sector. Within the weak quark interactions, $\mathcal{CP}$ is violated if and only if

$$\Delta m_{tc}^2 \Delta m_{tu}^2 \Delta m_{cu}^2 \Delta m_{bs}^2 \Delta m_{bd}^2 \Delta m_{sd}^2 J \neq 0 \tag{2.39}$$

where $\Delta m_{ij} = m_i^2 - m_j^2$ [16]. So the quark masses must not be generate and the mixing angles as well as the complex phase must be $\neq 0$ and $\neq \pi$.

The interesting triangle in B physics is constructed from the first and third column of the CKM matrix.

$$V_{ud}V_{ub}^* + V_{cd}V_{cb}^* + V_{td}V_{tb}^* = 0 \tag{2.40}$$

Dividing the terms in this triangle with $V_{cd}V_{cb}^*$ leads to the common representation of the unitarity triangle in the $\bar\rho, \bar\eta$ plane

$$\frac{V_{ud}V_{ub}^*}{V_{cd}V_{cb}^*} + \frac{V_{td}V_{tb}^*}{V_{cd}V_{cb}^*} + 1 = 0 \tag{2.41}$$

This triangle is shown in Fig. 2.2. The three angles have two different naming schemes, one was commonly used in the BaBar experiment (α, β, γ), the other one

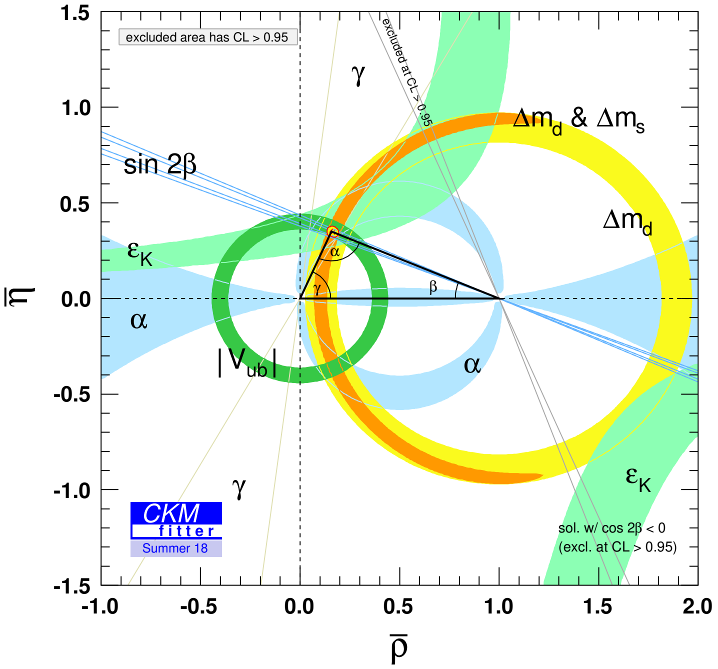


Figure 2.3.: Combined fit of the unitarity triangle published by the CKMfitter group in summer 2018 [17].

in the Belle experiment (ϕ_1, ϕ_2, ϕ_3). They are defined by [16]

$$\alpha = \arg\left(-\frac{V_{td}V_{tb}^*}{V_{ud}V_{ub}^*}\right) \qquad \beta = \arg\left(-\frac{V_{cd}V_{cb}^*}{V_{td}V_{tb}^*}\right) \qquad \gamma = \arg\left(-\frac{V_{ud}V_{ub}^*}{V_{cd}V_{cb}^*}\right) \tag{2.42}$$

The side lengths in the normalized unitarity triangle (Fig. 2.2) are

$$\left|\frac{V_{td}V_{tb}^*}{V_{cd}V_{cb}^*}\right| = \sqrt{\rho^2 + \eta^2} \qquad \left|\frac{V_{ud}V_{ub}^*}{V_{cd}V_{cb}^*}\right| = \sqrt{(1-\rho)^2 + \eta^2} \tag{2.43}$$

Fig. 2.3 shows the latest combined fit of the unitarity triangle, published by the CKMfitter group [17].

2.3.1. Meson Mixing

Neutral meson mixing can be mediated via the weak interaction. It describes the oscillation between weakly decaying neutral mesons and their antiparticles. Mixing was first discovered in the neutral kaon sector (mixing of $\overline{\mathrm{K}}^0 \leftrightarrow \mathrm{K}^0$ with quark content $\mathrm{K}^0 = (\mathrm{d}, \bar{\mathrm{s}})$) through the $\mathcal{CP}$ violating decay of assumed $\mathcal{CP}$ odd state of long

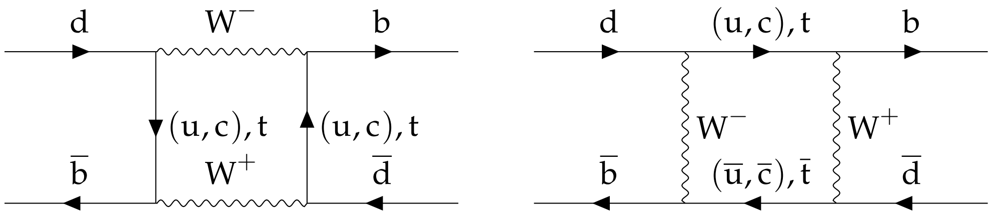


Figure 2.4.: The two Feynman box diagrams contributing in lowest order to neutral $B^0\overline{B}^0$ meson mixing with the quark contents $B^0 = (d,\overline{b})$ and $\overline{B}^0 = (\overline{d},b)$.

lived kaons (K^0_L) into the $\mathcal{CP}$ even final state of two pions [13]. At B-Factories, the neutral B meson mixing ($\overline{B}^0 \leftrightarrow B^0$ with quark content $B^0 = (d,\overline{b})$) can be studied. In the process $\Upsilon(4S) \to \overline{B}^0B^0$ the B meson pair is produced in a coherent $\mathcal{C}$ odd state [18, p. 18] [19, p. 186], i.e. $|B^0,\overline{B}^0\rangle = (|B^0\overline{B}^0\rangle - |\overline{B}^0B^0\rangle)/\sqrt{2}$. Before the decay of the first B meson, the state consists of a B^0 and its $\mathcal{C}$-conjugate $\overline{B}^0$ oscillating coherently. At the time when the first B meson decays via the weak interaction the coherent wave function collapses. The decaying B meson is observed in an eigenstate of the weak interaction (one of B^0, $\overline{B}^0$) and the remaining B meson is designated to have the $\mathcal{C}$-conjugate eigenstate. From that moment on, the second B meson oscillates freely, which allows a measurement of the time dependent mixing induced $\mathcal{CP}$ violation.

The Feynman diagrams for B^0 meson mixing is shown in Fig. 2.4. The mixing can be described by an effective, non-Hermitian Hamiltonian $\mathscr{H}$.

$$\mathscr{H} = \mathbf{M} - \frac{i}{2}\mathbf{\Gamma} \tag{2.44}$$

This Hamiltonian can be decomposed into two Hermitian matrices: a mass matrix $\mathbf{M}$ and a decay width matrix $\mathbf{\Gamma}$. From the $\mathcal{CPT}$ conservation follows $M_{11} = M_{22}$ and $\mathit{\Gamma}_{11} = \mathit{\Gamma}_{22}$ [16]. The diagonal elements describe the mass and decay rate of the flavor eigenstates, the off-diagonal elements describe the interference between the decays of the two flavor eigenstates. If $\mathscr{H}$ contains non-zero off-diagonal elements, the mass eigenstates differ from the flavor eigenstates $B^0,\overline{B}^0$. The mass eigenstates (B_H heavy, B_L light) are the eigenstates of the Hamiltonian. They can be described as superposition of the flavor eigenstates, with mixing parameters p and q

$$|B_{L,H}\rangle = p|B^0\rangle \pm q|\overline{B}^0\rangle \tag{2.45}$$

2.3.2. $\mathcal{CP}$ Violation

Three different types of $\mathcal{CP}$ violation have to be distinguished:

1. direct $\mathcal{CP}$ violation in a decay

2. $\mathcal{CP}$ violation in the mixing

3. mixing induced $\mathcal{CP}$ violation (interference of direct $\mathcal{CP}$ violation and $\mathcal{CP}$ violation in the mixing)

Direct $\mathcal{CP}$ violation is observed if the modulus of the amplitude for a particle i decaying to a final state f differs from the modulus of the amplitude of the antiparticle $\bar{i}$ decaying to the anti final state $\bar{f}$.

$$|A(i \to f)| \neq |A(\bar{i} \to \bar{f})| \tag{2.46}$$

The $\mathcal{CP}$ violation is measured with the asymmetry

$$a_{\mathcal{CP}} = \frac{\Gamma(\bar{i} \to \bar{f}) - \Gamma(i \to f)}{\Gamma(\bar{i} \to \bar{f}) + \Gamma(i \to f)} \tag{2.47}$$

$\mathcal{CP}$ violation in the mixing of neutral mesons can be described in terms of the mixing parameters p and q (see Eq. 2.45). The decay amplitudes from the initial states $\bar{i}$ and i into the final state f are

$$A_f = A(i \to f) \qquad\qquad \bar{A}_f = A(\bar{i} \to f) \tag{2.48}$$

An observable to measure the $\mathcal{CP}$ violation in the mixing is

$$A_f(t) = \frac{\Gamma(\bar{i}(t) \to f) - \Gamma(i(t) \to f)}{\Gamma(\bar{i}(t) \to f) + \Gamma(i \to f)} \tag{2.49}$$

$\mathcal{CP}$ in the mixing is violated if

$$\left|\frac{p}{q}\right| \neq 1 \tag{2.50}$$

If $\left|\frac{p}{q}\right| = 1$ and $\left|\frac{A_f}{\bar{A}_f}\right| = 1$, $\mathcal{CP}$ violation can only occur in the interference of decay and mixing (mixing induced $\mathcal{CP}$ violation). Defining the parameter λ_f is useful in the description of $\mathcal{CP}$ violation in the interference of mixing and decay [16]

$$\lambda_f = \frac{q}{p}\frac{\bar{A}_f}{A_f} \tag{2.51}$$

In the case of $|\lambda_f| = 1$, an observable $\mathcal{CP}$ asymmetry is [16]

$$A_f(t) = \sin(\arg(\lambda_f))\sin(\Delta m t) \tag{2.52}$$

where $\Delta m = m_H - m_L$ is the mass difference between the heavy and the light states $\mathrm{B}_{\mathrm{H,L}}$. A measurement of λ_f provides a constraint on the CKM matrix parameters.

2.4. Beyond SM Physics

There are still phenomena that cannot be described within the SM and open questions remain that need to be answered by New Physics models (NP) [20, 21, 18]:

- The coupling strengths of the forces are vastly different, although no mechanism is known to explain the difference (hierarchy problem).
- To explain the dominance of matter over antimatter in the baryogenesis phase of our universe, the known $\mathcal{CP}$ violating phases in the SM are too small.
- The masses and mixing parameters of the SM particles can only be determined experimentally.
- The CKM matrix has a diagonal hierarchy which is an indication of a flavor symmetry at higher energy scales.
- The SM neutrinos are massless as the Higgs mechanism does not contain coupling terms of the neutrinos to the Higgs field.
- $\mathcal{CP}$ violation is not observed in the strong interactions (strong $\mathcal{CP}$ problem), although strong $\mathcal{CP}$ violation is plausible within the SM.
- The observed gravitational force in galactic length scales is much larger than the visible mass. Up to 85 % of the gravitational interacting mass is expected to be dark matter which is not described by the SM.
- A Grand Unified Theory (GUT) unifying the electroweak interactions with QCD is still missing.

2.4.1. Signatures Towards New Physics

There are many theoretical models for new physics that could solve the mentioned problems with the standard model. Although no evidence for any particular NP model has been found yet, the models indicate search directions for future experiments.

There are several cosmological observations of the large scale structure of the universe indicating that dark matter has to be cold (CDM - Cold Dark Matter) [22]. Among these observations are: gravitational lensing, fluctuations in the cosmic microwave background (CMB) and the cosmic web. This means the gravitational interacting particles have to be non-relativistic massive particles, which excludes neutrinos as dark matter candidates. A viable candidate for dark matter are WIMPs (Weakly Interacting Massive Particles) interacting via the weak force. Several experiments are built to search for this type of dark matter candidates [23], for example CRESST [24].

The axion, a new particle with a Yukawa coupling to the fermions (proposed in 1978 [25, 26]), could explain the strong $\mathcal{CP}$ problem and it is also a viable candidate

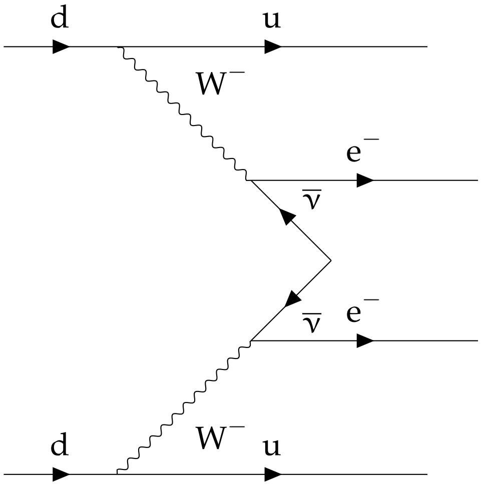


Figure 2.5.: Feynman diagram for the hypothetical neutrinoless double beta decay. It can only occur if neutrinos are their own anti-particles i.e. Majorana neutrinos.

to explain the dark matter in the universe. The additional coupling to the axion would cancel the $\mathcal{CP}$ violating strong phase [27]. Dedicated experiments are built to search for axions and Axion Like Particles (ALP), like the Axion Dark Matter eXperiment (ADMX) [28] and the newly proposed MadMax experiment [29].

Supersymmetry introduces a symmetry between elementary fermions and bosons. Each elementary fermion gets a bosonic partner, each elementary boson a fermionic partner. In the MSSM (Minimal Supersymmetric Standard Model), no new particles are introduced but the supersymmetric partners to the known particles. MSSM could explain the hierarchy problem in the SM [18] and the lightest supersymmetric particle (LSP) is a viable dark matter candidate [22].

Within the SM, neutrinos are described as massless particles, so their are no coupling terms of neutrinos to the Higgs field. But recent measurements of neutrino flavor oscillations (Experiments: SuperKamiokande [30], SNO [31]) indicate that neutrinos have mass. The necessary extension of the SM is the PMNS matrix, a mixing matrix of the neutrino mass eigenstates to their flavor eigenstates in analogy to the CKM mechanism. It was suggested in 1962 to explain neutrino oscillations within the first two generations [32].

With massive neutrinos, the next open question is to unravel their nature: are they Dirac particles (particles and anti-particles are not identical $\nu \neq \overline{\nu}$ or Majorana neutrinos (i.e. neutrinos are their own anti-particles $\nu = \overline{\nu}$). If neutrinos were Majorana neutrinos, a neutrinoless double beta decay would be observable (see Fig. 2.5). It describes the simultaneous decay of two down quarks, where the two emitted anti neutrinos in the beta decays are connected as internal lines.

2.5. Physics Goals at Belle II - Physics Reach

The experiments presently running, aim to measure the detailed structure of the SM with the hope to find NP in deviations from the theoretical predictions. The LHC at CERN with the ATLAS, CMS, LHCb and ALICE detectors are the current high energy experiments, where NP is hoped to be found as direct production of new particles. The LHC collides protons instead of electrons with the advantage that much higher kinetic energies can be achieved. Due to their huge mass ($m_\mathrm{p} \approx 1836\,m_\mathrm{e}$) the energy loss in an accelerator ring due to synchrotron radiation is strongly suppressed ($\Delta E_\mathrm{sync} \propto m^{-4}$). The downside in using protons is that they are not elementary. A proton is a state composed of three particles (uud): 2 up quarks (u) and a down quark (d), bound together by the gluons, the mediators of the strong force. In a pp collision only some parts of the quarks and gluons within the protons interact, while the rest can be seen as unaffected spectators. Which interaction occurred can only be inferred from the reconstruction. The precisely known CM energy can not be used directly as an input parameter. In this way, the Higgs boson was found in 2012 via direct production in the CM system [33]. If new physics particles exist at the reachable energy scale of the LHC, they can be directly produced.

SuperKEKB with the Belle II detector is a flavor physics factory colliding e^+ and e^- at the $\Upsilon(4S)$ resonance, the fourth radially excited state of the $\mathrm{b\overline{b}}$ meson. The $\Upsilon(4S)$ mass of $m_{\Upsilon(4S)} = 10.58\,\mathrm{GeV}$ is slightly larger than the masses of two B mesons ($m_{\Upsilon(4S)} > m_{\overline{\mathrm{B}}^0} + m_{\mathrm{B}^0}$ and $m_{\Upsilon(4S)} > m_{\mathrm{B}^+} + m_{\mathrm{B}^-}$). Hence, the production of coherent B meson pairs is energetically allowed. Since electrons are elementary, they always interact completely and the CM energy can be used directly as input parameter in the reconstruction. This gives rise to the analysis parameters ΔE and m_bc [18, p. 102ff]

$$\Delta E = E_\mathrm{B} - E_\mathrm{beam} \qquad\qquad m_\mathrm{bc} = \sqrt{E_\mathrm{beam}^2 - p_\mathrm{B}^2} \tag{2.53}$$

where E_B and p_B are the reconstructed energy and the reconstructed momentum of a B meson and E_beam is $\frac{1}{2}$ of the CM energy. Thus, a very high precision can be achieved and the occurrence of NP in higher order corrections can be tested.

In the previous B-Factories Belle and BaBar many observables of the SM could already be measured, most importantly the angles in the unitarity triangle. The discovery of the CKM mechanism as the main source of time dependent $\mathcal{CP}$ violation in the B meson system, was honored with the 2008 Nobel prize for Kobayashi and Masukawa [34]. Also unexpected effects were observed, like the discovery of the X(3872) particle [35].

In a typical $\Upsilon(4S)$ event, there are on average 10 tracks visible in the detector with an average transverse momentum of $p_\mathrm{T} = 500\,\mathrm{MeV}$. Precise vertex reconstruction is important to study the time dependent $\mathcal{CP}$ violation. Using flavor tagging, the flavor of the coherently produced $\mathrm{B}^0\overline{\mathrm{B}}^0$ mesons is designated at the time of the decay of the first B meson. The second B meson can oscillate freely until it decays. Due

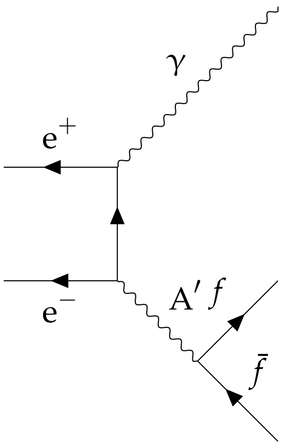


Figure 2.6.: Feynman diagram for the production of a dark photon A' in e^+e^- collisions. Under the assumption of a dark photon decay to the fermion pair $f\bar{f}$, the dark photon A' could be measured by reconstructing the displaced decay vertex position of these decay products.

to the boost of the CM system, the vertex displacement of the second decay can be translated into a time difference.

2.5.1. Low Multiplicity Events

The e^+e^- collisions around the $\Upsilon(4S)$ resonance provide a broad spectrum of final states which are relevant in the NP searches at Belle II. Besides the typical $\Upsilon(4S)$ events, the clean environment supports the analysis of processes with a low track multiplicity in the final state. Typical candidates with a low track multiplicity are invisible B decays, τ physics and initial state radiation events [20].

In invisible B decays one B meson decays into an invisible final state. This would be the case in e.g. $B^0 \to \nu\nu$ transitions, which are highly forbidden in the SM. In an $\Upsilon(4S)$ event where two B mesons are produced ($\Upsilon(4S) \to B\overline{B}$), invisible B decays can in principle be reconstructed by a full reconstruction of the other side B meson.

Belle II has a large cross section for τ pair production, comparable to the cross section for $B\overline{B}$ production. Therefore, the τ physics sector can be explored as well [20]. This includes the search for the lepton number violating decays $\tau \to \ell\ell\ell$ with three leptons ℓ in the final state and the search for $\tau \to \mu\gamma$ where the τ would be an excited state of the μ. If one of these processes is observed, it is a clear indication of NP.

Another interesting low multiplicity NP scenario is that of dark photons A'. At Belle II they could theoretically be produced in the process $e^+e^- \to \gamma A'$. These dark photons would correspond to an additional $\mathcal{U}(1)'$ symmetry coupling to the electric charge e and thus mix with the normal photon [36]. The sensitivity of Belle II allows to set new limits on such a new symmetry in both cases: invisible A' observable only as missing energy and visible A' through decays to final state particles with

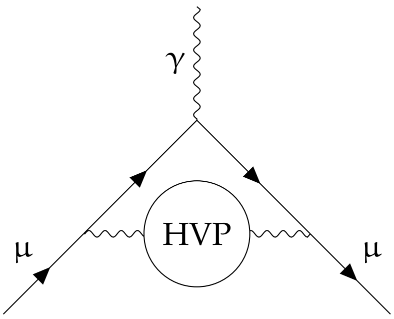


Figure 2.7.: Feynman diagram for the Hadronic Vacuum Polarization (HVP) of the muon. The loop process (HVP) has to be taken into account to obtain the correct magnetic moment of the muon.

displaced vertex positions. The Feynman diagram for a visible production of dark photons A' is shown in Fig. 2.6.

Via initial state radiation events Belle II has access to a broad hadron production continuum in addition. In the process $\mathrm{e}^+\mathrm{e}^- \to \gamma\mathrm{e}^+\mathrm{e}^- \to \gamma q\bar{q}$, the additional photon radiated in the initial state effectively lowers the CM energy.

2.5.2. Muon $g-2$

The SM expectation for the magnetic moment of the muon has a tension ($> 3\,\sigma$) between the theoretical SM expectation of the gyromagnetic ratio g_μ and its measured experimental value [37]. The cross sections $\mathrm{e}^+\mathrm{e}^- \to$ hadrons are especially important for the measurement of the anomalous magnetic moment of the muon the $g-2$ problem. The theoretical error of the muon $g-2$ is dominated by the Hadronic Vacuum Polarization (HVP) which relies on experimentally measured hadronic cross sections. Thus, the theory error is owed to the large experimental error on the hadronic cross section and the pion form factor [20]. With the muon $g-2$ experiment presently running at Fermilab [38], the measurement error of the asymmetry will be substantially reduced. An improved measurement of the hadronic cross section at Belle II will be essential to clarify NP in the $g-2$ sector.

From the Dirac equation follows a magnetic moment of the muon of $\vec{M} = g_\mu \frac{e}{2m_\mu}\vec{S}$ with the gyromagnetic ratio $g_\mu = 2$ [37]. Higher order corrections result in small deviations from this value which is commonly expressed by:

$$a_\mu = \frac{g_\mu - 2}{2} \tag{2.54}$$

where in the lowest order SM prediction $g_\mu = 2$ so $a_\mu = 0$.

The theoretical estimate a_μ^{SM} is split in three parts:

$$a_\mu^{\mathrm{SM}} = a_\mu^{\mathrm{QED}} + a_\mu^{\mathrm{EW}} + a_\mu^{\mathrm{Had}} \tag{2.55}$$

a QED contribution a_μ^{QED}, an electroweak contribution a_μ^{EW} and a hadronic contribution a_μ^{Had}. Presently the largest theoretical error enters through the hadronic part $a_\mu^{\mathrm{Had}} = a_\mu^{\mathrm{HVP}} + a_\mu^{\mathrm{HO\ Had}}$, where a_μ^{HVP} is the leading order contribution due to hadronic vacuum polarization (largest error) and $a_\mu^{\mathrm{HO\ Had}}$ are higher order hadronic contributions including the hadronic light-by-light scattering (second largest error) [37]. The Feynman diagram for LO HVP is shown in Fig. 2.7.

Since the lattice QCD calculations are still too imprecise, a_μ^{HVP} is obtained from a dispersion relation using the experimentally measured hadronic cross section of $\mathrm{e}^+\mathrm{e}^- \to q\bar{q}$ [37]

$$a_\mu^{\mathrm{HAD}} = \frac{1}{3}\left(\frac{\alpha}{\pi}\right)^2 \int_{m_\pi^2}^{\infty} \mathrm{d}s \frac{K(s)}{s} R(s) \tag{2.56}$$

where $K(s) \propto s^{-1}$ is a QED kernel function, $R(s)$ is the ratio of the bare hadron cross section $\mathrm{e}^+\mathrm{e}^- \to$ hadrons to the muon cross section $\mathrm{e}^+\mathrm{e}^- \to \mu^+\mu^-$ for a CM energy $\sqrt{s}$, and α is the fine structure constant [37].

The present experimental and theoretical values for a_μ are [37]

$$a_\mu^{\mathrm{SM}} = (116591823 \pm 1_{\mathrm{EW}} \pm 34_{\mathrm{HVP}} \pm 26_{\mathrm{HO\ Had}}) \times 10^{-11} \tag{2.57}$$

$$a_\mu^{\mathrm{exp}} = (116592091 \pm 54_{\mathrm{stat}} \pm 33_{\mathrm{syst}}) \times 10^{-11} \tag{2.58}$$

The currently observed discrepancy between the SM and the experiment is [37]

$$\Delta a_\mu = a_\mu^{\mathrm{exp}} - a_\mu^{\mathrm{SM}} = (268 \pm 63_{\mathrm{experiment}} \pm 43_{\mathrm{theory}}) \times 10^{-11} \tag{2.59}$$

which is a $3.5\,\sigma$ effect. This ratio $R(s)$ can be measured precisely at Belle II in initial state radiation events in order to reduce the theoretical error. With new measurements of a_μ^{exp} at the $g-2$ experiment at Fermilab [38], the experimental error will be reduced and the significance of the discrepancy will hopefully be unraveled.

Chapter 3.

The Belle II Experiment

The SuperKEKB collider with the Belle II detector [21] is an upgrade of the successful B-Factory KEKB with Belle [39] at the KEK research center in Tsukuba, Japan. The predecessor Belle was running at the same time as the BaBar [40] experiment at the PEP II accelerator at SLAC, which both contributed to the confirmation of the CKM mechanism as the main source of $\mathcal{CP}$ violation in weak interactions in the SM. SuperKEKB has the goal to achieve an instantaneous luminosity of $8 \times 10^{35}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$ in $\mathrm{e}^{+}\mathrm{e}^{-}$ collisions at the $\Upsilon(4S)$ resonance, which is 40 times larger than the peak luminosity of its predecessor KEKB. The Belle II detector is constructed surrounding the Interaction Region (IR) in the typical layered structure of a general purpose detector; with vertex detectors in the vicinity of the Interaction Point (IP), followed by tracking and Particle IDentification detectors (PID), a calorimeter and a muon tracker on the outside. The big structures like the solenoid magnet and the iron return yoke at the endcaps are reused from Belle. The Belle II detector has to be capable of recording a huge data rate, while coping with severely increased background rates compared to the predecessor experiment Belle.

In order to provide a high signal efficiency, a powerful online trigger is crucial for fast background rejection. However, due to their readout times, not all sub-detectors are suited for the trigger which leaves only a limited event information as input for the trigger algorithms. The first level (L1) trigger requirements are a trigger rate of 30 kHz at a minimum two event separation of 200 ns within a maximum latency of 5 µs [21, 41]. A following online High Level Trigger (HLT) further reduces the trigger rate to 10 kHz of event data which is finally stored for offline analysis. The L1 trigger system runs fully pipelined and is processing new data every 32 ns in a dead-time-free setup on parallel FPGA hardware, while the longer time scales of the HLT allow the use of computer farms for this second level trigger.

A schematical illustration of the Belle II detector is shown in Fig. 3.1 and a technical drawing of the detector including labels of the subdetectors and the supporting structure elements in Fig. 3.5. The innermost detectors are the PiXel Detector (PXD) and the Silicon Vertex Detector (SVD), which are used for precise vertex reconstruction. They are followed by the Central Drift Chamber (CDC), the main tracking device in Belle II. The next detector shells are the Electromagnetic CaLorimeter (ECL), the Time Of Propagation (TOP) detector, the Aerogel CHerencov counter (ARICH) and the Kaon Long and Muon detector ($\mathrm{K}_{\mathrm{L}}^{0}$ and μ – KLM). A complete overview of

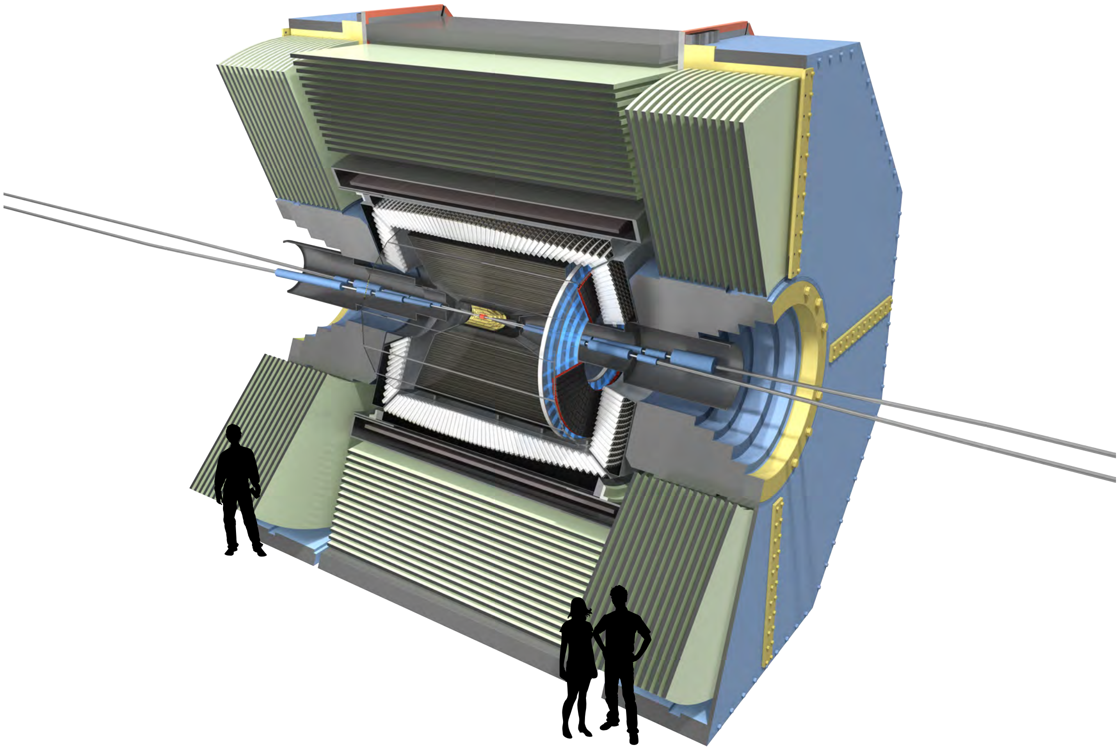


Figure 3.1.: Perspective illustration of the Belle II detector [42].

the subdetectors and components in Belle II can be found in the Belle II Technical Design Report [21].

The improved precision of the silicon detectors, especially the PXD, is essential for a precise vertex reconstruction. One measure in the luminosity increase is the reduced asymmetry of the colliding beams resulting in a smaller boost of the CM system. The new PXD detector provides high precision vertex measurements in the laboratory system. Despite the smaller boost, in the CM system a vertex resolution comparable with the predecessor Belle can be achieved. This is especially important for the study of time dependent $\mathcal{CP}$ violation in $\mathrm{B}^0\overline{\mathrm{B}}^0$ oscillations, where distance measurements of the decay vertices in the laboratory frame can be translated into decay time differences of the B mesons in the CM frame.

Belle II starts collecting data at a small luminosity provided by SuperKEKB and the luminosity will be increased slowly over a long period of time to gently reach the anticipated world record peak luminosity. Two important measures are taken to increase the luminosity. By slowly increasing the beam current, more collision partners are available at the IP. A fine tuning of the final focusing magnets reduces the beam size at the IP such that the density of beam particles is increased during the bunch crossing.

This chapter starts in Sec. 3.1 with an overview of the SuperKEKB accelerator machine, with a focus on the relevant properties in the design and evaluation of the 3D track triggers using neural networks. Section 3.2 introduces the Belle II detector geometry including the solenoid magnetic field, the material in the vicinity of the IR, and the used coordinate system. In Sec. 3.2.5 follows an overview of all the Belle II subdetectors. The full Belle II trigger system is presented Sec. 3.3. Special focus is put on the L1 CDC subtrigger system in Sec. 3.3.2, where the proposed machine learning algorithms (including the neural network trigger) will be integrated. The simulation of the Belle II detector is shortly introduced in Sec. 3.4.

3.1. SuperKEKB Collider

SuperKEKB is installed in the existing KEKB tunnel, where an e^- beam with 7 GeV is maintained in the high energy ring (HER) and an e^+ beam with 4 GeV in the low energy ring (LER). The beam energies are chosen such that the energy in the center of mass system of an e^+e^- collision is close to the mass of the $\Upsilon(4S)$ meson, the fourth radially excited state of the bound $b\bar{b}$ system (hence the name B-Factory). The CM energy at this resonance ensures a high production rate of the $\Upsilon(4S)$ in inelastic e^+e^- scattering events. Due to the beam energy asymmetry, a scattering process of an e^- with an e^+ has a boost with respect to the laboratory frame of $\beta\gamma = 0.28$. With respect to the previous experiment, the asymmetry of the beam energies was reduced in order to reduce the Touschek background rate in the LER (see Sec. 4.2.1 for details on the Touschek background). A new target for e^+ production was installed and the e^+ damping ring was renewed. Furthermore, new focusing quadrupole magnets were installed in the collision area. In order to achieve the high luminosity, the nano beam scheme [43, 44] is adopted and a large crossing angle of the beams is used. The main structure of the SuperKEKB machine is illustrated in Fig. 3.2, which shows the placement of the Belle II detector surrounding the IR where the two beams are brought to collision.

A review of basic accelerator physics principles and the luminosity at SuperKEKB is given in Sec. 3.1.1. Section 3.1.2 shows the main components of SuperKEKB and the design of the interaction region from a technical viewpoint. The key features of SuperKEKB are summarized to allow a discussion of their impact on the expected background in Belle II. More details on SuperKEKB can be found in the references [45, 46, 47].

3.1.1. Acceleration and Collisions

Electrons and positrons are injected into the SuperKEKB HER and LER after they are accelerated to the design energies of 4 GeV (e^+) and 7 GeV (e^-). After injection, the particle beams are kept on their nominal trajectory in the rings via a lattice of magnets. Solenoid fields arrange the bending to keep the bunches on the circular track and the transverse focusing of the beam is realized by multipole magnets.

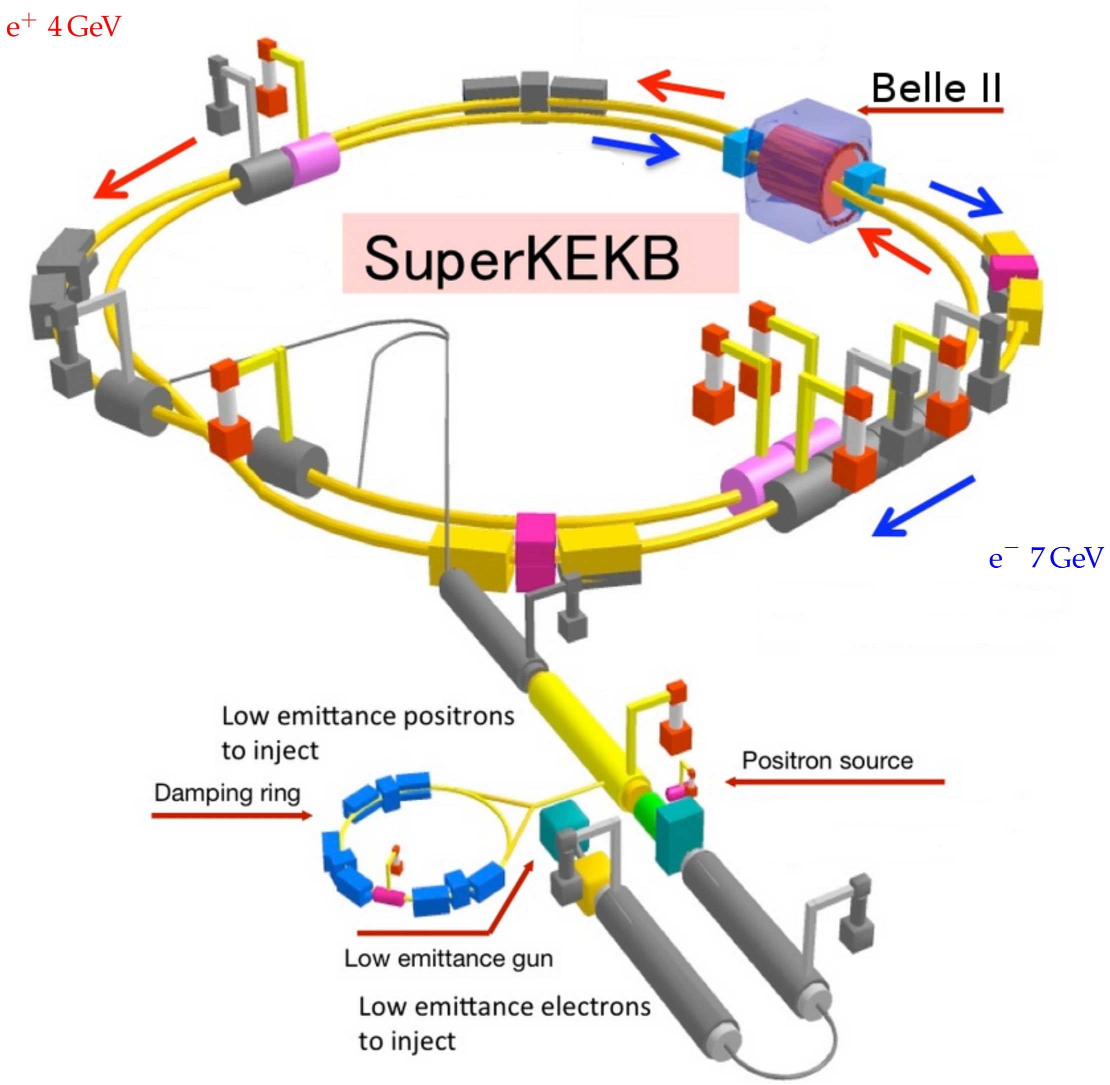


Figure 3.2.: Main structure of the SuperKEKB machine [42]. The e^- and e^+ particles are injected into two separate rings and brought to collision in the interaction region, which is surrounded by the Belle II detector.

In order to keep the beams on their nominal beam energies, Radio Frequency (RF) cavities are installed. There an oscillating electric field accelerates particles deviating from their nominal position on the beam trajectory.

In the interaction region the final focusing magnets bring the two beams to collision. A quadrupole is focusing in one direction, and defocusing in the other direction. By arranging several quadrupoles with alternating orientations rotated by 90° a total net focusing of the beam can be achieved, which is commonly referred to as FODO setup (focusing-, zero-, defocusing-, zero-field). However, the focal length of a quadrupole is dependent on the energy of the particle that is focused. Therefore, higher order multipoles are used to prevent a blowup of the beam. Inclined sextupole magnets correct focusing errors arising from pure quadrupole focusing [21].

Beam Optics

In an electron positron collider charged particles move in bunches in separate storage rings, before they are brought to collision. Each bunch can be approximated as a 3D Gaussian distribution with extensions (RMS) $\sigma_x, \sigma_y, \sigma_z$. However, only the center of the bunch is located ideally on the correct reference trajectory. The focusing magnetic quadrupole fields are arranged such that the bunch particles with a transverse displacement are accelerated towards the reference trajectory with a force proportional to their deviation, which reduces the bunch extension in x- and y-direction (σ_x, σ_y). The electric fields in the RF cavities accelerate the particles longitudinally in order to reduce the bunch extension in z-direction (σ_z). The related transverse oscillations around the nominal trajectory ar called betatron oscillations, the longitudinal oscilations are the synchrotron oscillations.

A characteristic quantity of a particle accelerator is its emittance ϵ, which describes the volume of the particle bunches in the phase space. In linear focusing it is a conserved quantity. Therefore, it is constant at all positions s, where s is the parameter along the beamline around the SuperKEKB ring. The transverse size of the beam is described by the beta function $\beta(s)$ at a position s. Combined with the emmitance ϵ, the transverse width of the Gaussian bunches is given by the product [48]

$$\sigma(s) \propto \sqrt{\epsilon\beta(s)}. \tag{3.1}$$

Due to scattering processes within a bunch, with residual gas in the beam line, and due to energy loss via synchrotron radiation the beam particles get displaced from the nominal beam trajectories. The synchrotron radiation is a result of the acceleration of the charged particles by the bending magnets, which keep the bunches on a circular path. If the displacement of a particle becomes too large, the particle gets lost from the beam. This introduces an effective lifetime of the beam. In order to maintain constant beam currents, a continuous injection of new particles is required.

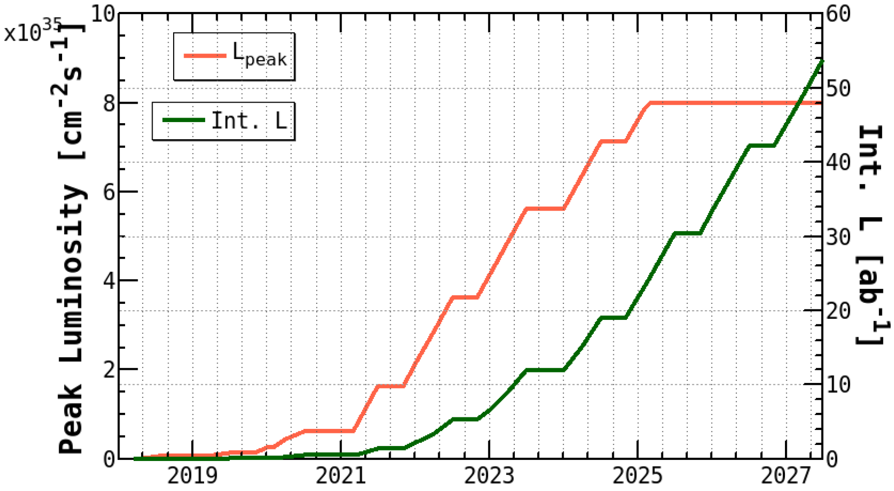


Figure 3.3.: Peak value of the instantaneous luminosity and the time integrated luminosity anticipated with SuperKEKB [49].

Luminosity

The figure of merit for a particle accelerator is its luminosity $\mathcal{L}$ [cm^{-2} s^{-1}]. It provides a relation of the cross section σ [cm^2] of the physics interactions to their interaction rate $\frac{dN}{dt}$ [s^{-1}]. The general definition of the number of interactions N given the luminosity $\mathcal{L}$ and the cross section σ is [37, 47]:

$$N = \sigma \int \mathcal{L}(t)\mathrm{d}t \tag{3.2}$$

Assuming equally Gaussian shaped bunches, a basic term for the luminosity $\mathcal{L}$ is given by [37]

$$\mathcal{L} = \frac{f N_+ N_-}{4\pi \sigma_x^* \sigma_y^*} \cdot R_{\mathcal{L}} \tag{3.3}$$

where N_+, N_- are the number of particles in each bunch, $\sigma_x^* = \sqrt{\epsilon_x \beta_x^*}$, $\sigma_y^* = \sqrt{\epsilon_y \beta_y^*}$ are the bunch sizes at the IP with $\beta_{x,y}^* = \beta_{x,y}(0)$ and f is the collision frequency. For ideal head on collisions the reduction factor is $R_{\mathcal{L}} = 1$. Due to the geometry of the interaction the luminosity term is reduced in reality; the field of one bunch is acting on the other bunch via the hourglass effect and the crossing angle at SuperKEKB is non-zero. The reduction factor $R_{\mathcal{L}} \leq 1$ takes these corrections into account. A convenient luminosity definition for SuperKEKB is given by [47]

$$\mathcal{L} = \frac{\gamma_\pm}{2er_\mathrm{e}} \left(1 + \frac{\sigma_y^*}{\sigma_x^*}\right) \left(\frac{I_\pm \xi_{y\pm}}{\beta_y^*}\right) \left(\frac{R_{\mathcal{L}}}{R_{\xi_{y\pm}}}\right) \tag{3.4}$$

parameter	value	description
I_{HER}	2.62 A	HER current
I_{LER}	3.6 A	LER current
N	2503	number of bunches
p	10^{-7} Pa	average vacuum pressure

Table 3.1.: Design parameters of SuperKEKB.

where + identifies the positron beam in the LER, − the electron beam in the HER, I is the beam current, β_y^* is the vertical beta function at the IP, r_e is the classical electron radius, and γ is the Lorentz factor. The parameter $\xi_{y\pm}$ is the vertical beam-beam parameter that includes information on the horizontal emittance and on the beta function, the bunch length, and the crossing angle. The parameter $R_{\xi_{y\pm}}$ is the reduction factor related to the beam-beam parameter. An expression for the beam-beam parameter is given by [47]

$$\xi_{y\pm} = \frac{r_\mathrm{e}}{2\pi\gamma_\pm} \frac{N_\mp \beta_y^*}{\sigma_y^*(\sigma_x^* + \sigma_y^*)} R_{\xi_{y\pm}} \tag{3.5}$$

where N denotes the number of particles per bunch. The beam-beam is a dimensionless parameter representing the interaction of the field of one beam disturbing the other one. At SuperKEKB a value of $\xi_{y\pm} \approx 0.09$ can be achieved for the beam-beam parameter [21]. Note that inserting $\xi_{y\pm}$ into Eq. 3.4 exactly gives the definition of the luminosity (Eq. 3.3).

At SuperKEKB, the anticipated instantaneous luminosity of $8 \times 10^{35}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$ is not collected directly after the start. Instead, it will be solwly increased during the first years of the running experiment. Figure. 3.3 shows the anticipated timeline for the continuous increase of the instantaneous luminosity as well as the expected integrated luminosity.

3.1.2. SuperKEKB Design

The main structure of SuperKEKB is shown in Fig. 3.2. The electrons and positrons are accelerated to the nominal beam energies prior to the injection into the HER and the LER. These two rings maintain the particle beams at the design energies. At the IP, which is surrounded by the Belle II experiment, the two beams are brought to collision. A list of SuperKEKB design parameters is shown in Tab. 3.1.

Injection

The particles need to be continuously injected on energy into the HER and the LER by the injector linac [45], which is shown in the bottom part of Fig. 3.2. This linear accelerator has a length of 600 m and it is composed of 54 accelerator modules. For the electron acceleration, the full length of the linac is used. In order to generate positrons, electrons are accelerated up to 4 GeV and shot onto a converter target. Here the positrons are caught in the damping ring, before injecting the positron beams into the SuperKEKB LER. Due to the damping, the emittance of the injected positron beam is smaller than the emittance of the electron beam.

This results in larger injection background in the HER where no damping ring is used, because the injection background scales with the emittance at the interaction point. The injection background affects only the measurements shortly after the injection. At the predecessing collider KEKB, where beams were injected about 20 times per day, injection background was not an issue. In contrast, at SuperKEKB the linac has the capability to continuously inject beam particles at 50 Hz resulting in large levels of injection background. Therefore, injection vetos are realized as a countermasure at the trigger level of Belle II to avoid the readout of the noisy data shortly after the injections.

Masks

At several locations along the ring so called masks collimate the beam to reduce the background in the IR. There are horizontal and vertical masks to cut off particles with a huge betatron amplitude. Due to slight parameter changes affecting the beta functions, updated mask setups might be requred to cope with the new background expectations. The intention of the mask setup is to reduce the background in the IR by a factor of ≈ 1000. However, masks result in shorter beam lifetimes. Therefore, the concurring objective is to keep the particle loss rate smaller than the injection rate. Furthermore, the Transverse Mode Coupling (TMC) instability has to be considered in the mask optimization, which can be caused by the vertical collimators [50]. More information on the development of the mask setup can be found in [51].

Interaction Region

In SuperKEKB the so called nano beam scheme [43, 44] is used in the beam collisions. It was originally developed for the now discontinued Italian B-Factory SuperB [43]. In the nano beam scheme the beam is squeezed at the IP to have a very low beta function while at the same time having a very small emittance. For this purpose, the final focusing quadrupole magnets (QCS magnets) have to be close to the IP.

The QCS magnets at the IP are shown in the technical drawing of the Belle II detector in Fig. 3.5. The QCS are doublets of vertical focusing quadrupoles (QC1S) and doublets of horizontal focusing quadrupoles (QC2S) for both rings. Located closest to the IP are the QC1S magnets, with QC1LP and QC1RP for the LER and the QC1LE and QC1RE for the HER. The QC2S magnets are QC2LP and QC2RP for the LER and QC2LE and QC2RE for the HER. Tab. 3.2 shows the design parameters

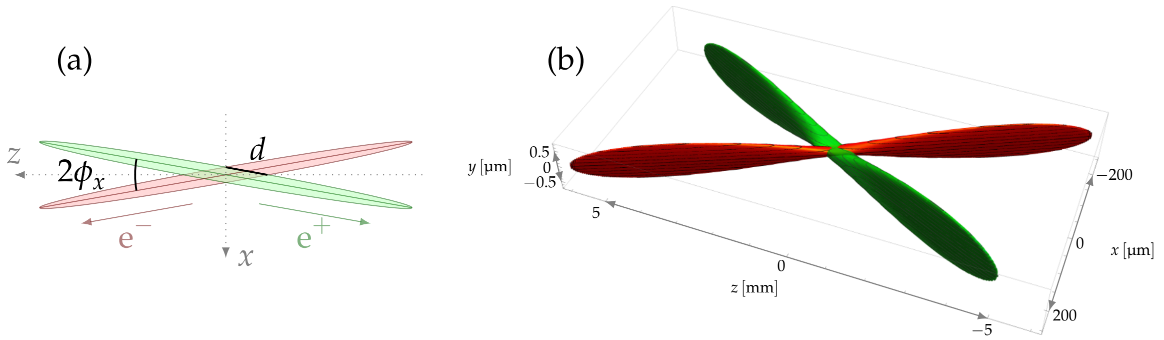


Figure 3.4.: (a) Bunch crossing at the Interaction Point (IP) of Belle II with the nano beam scheme in the xz-plane with the crossing angle $2\phi_x = 83\,\text{mrad}$ and the effective bunch length $d \approx 0.2\,\text{mm}$. (b) Perspective view of the 3D bunch shape at the bunch crossing [52].

design parameter	description	value
$\sigma_{x,y,z}$	beam size	$\sigma_y \approx 60\,\text{nm}$
$2\phi_x$	crotch angle	$2\phi_x = 83\,\text{mrad}$
$d \approx \dfrac{\sigma_x}{\phi_x}$	eff. bunch length	$d \approx 0.2\,\text{mm}$
β_x/β_y LER	beta function at IP	32 mm / 270 µm
β_x/β_y HER	beta function at IP	25 mm / 410 µm
ϵ_x LER / HER	horizontal emittance	3.2 nm / 2.4 nm
ϵ_y/ϵ_x LER / HER	emittance ratio	0.4 %/0.35 %

Table 3.2.: Technical design parameters for the bunch collisions using the nano beam scheme [21].

of the IR with the bunch crossing in the nano beam scheme including the small emittance in the order of $\mathcal{O}(\text{nm})$.

To achieve a high luminosity with the nano beam scheme and to cope with the hourglass effect [53] despite a long bunch length, an effective small beam size at the IP is achieved via a large Piwinski angle [47] which reduces the overlap length d. The overlap length d is

$$d = \frac{\sigma_x^*}{\sin(\phi_x)} \tag{3.6}$$

where ϕ_x is the half-crossing angle and σ_x^* is the horizontal beam size. The crossing angle at SuperKEKB is $2\phi_x = 83\,\text{mrad}$ and the vertical beta function at the IP will be squeezed to 270-300 µm. The bunch width in z is $\sigma_z \approx 5\,\text{mm}$. The sketch of the bunch crossing in Fig. 3.4 illustrates the parameters d and ϕ_x in Fig. 3.4 (a) and the 3D bunch shapes at the collision point in Fig. 3.4 (b). In the right handed coordinate system illustrated in Fig. 3.4, the z-axis is almost parallel to the s-parameter around the beamline; the x-axis points outwards of the accelerator plane.

3.2. Belle II Geometry

A technical drawing of the placement of the Belle II subdetectors is show in Fig. 3.5. It shows a cut through the detector in the horizontal xz-plane, in which the bunch crossing is implemented. In the Cartesian right handed coordinate system used in Belle II, the z-axis is the central axis of the cylindrical detector. It is approximately the direction of the e$^-$ beam in the HER at the IR which is oriented in parallel to the magnetic field and the boost direction of the collisions. In the transverse xy-plane, the axis pointing outwards of the SuperKEKB collider ring is the x-axis (see Fig. 3.2) and the y-axis is pointing upwards into the normal direction to the SuperKEKB ring. The two beams are crossed in the horizontal xz-plane, where the crotch angle of 83 mrad is used (see Sec. 3.1.2). The origin of the coordinate system $(x, y, z) = (0, 0, 0)$ is placed at the IP, where the intended collisions occur. In polar coordinates, ϕ is the azimuthal angle in the xy-plane with the x-axis and θ is the polar angle with the z-axis. The required tracking acceptance for tracks from the IP is defined to be $\theta \in [17, 150]^\circ$, with the smaller angle of 17° in the forward boost direction due to the larger energy in the HER.

3.2.1. Solenoid Magnetic Field

The solenoid magnet is placed surrounding the ECL and inside of the KLM (see super conducting coil in Fig. 3.5). It creates an almost uniform magnetic field of $B_z \approx 1.5\,\text{T}$, $\vec{B} \approx (0, 0, 1.5)\,\text{T}$, penetrating the central area of Belle II. Due to the focussing magnets small corrections have to be considered. Especially, towards the endcaps and in the crotch part region between the two beam-lines, the magnetic field cannot be considered uniform. Fig. 3.6 (a) and (b) show the x and z components

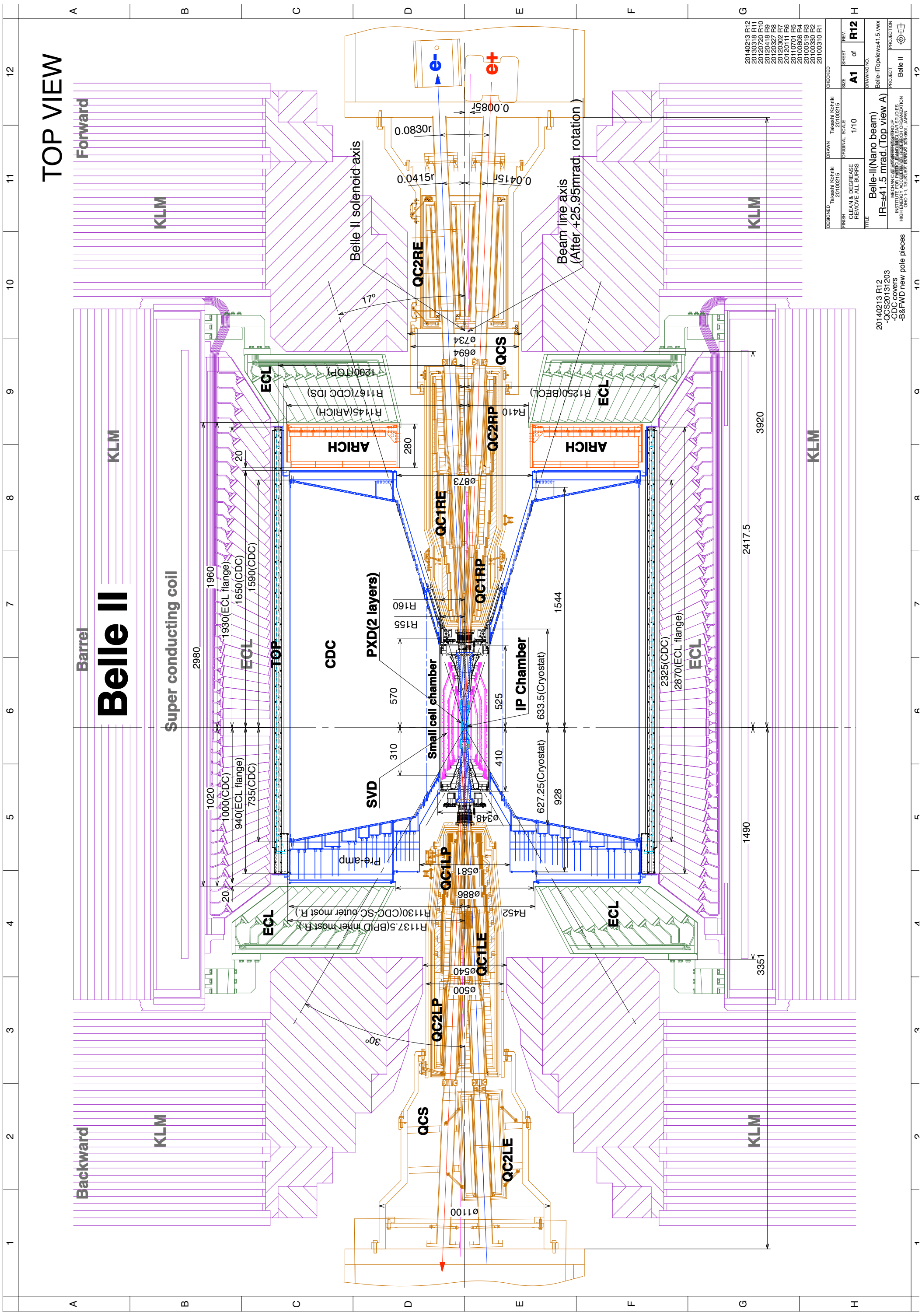
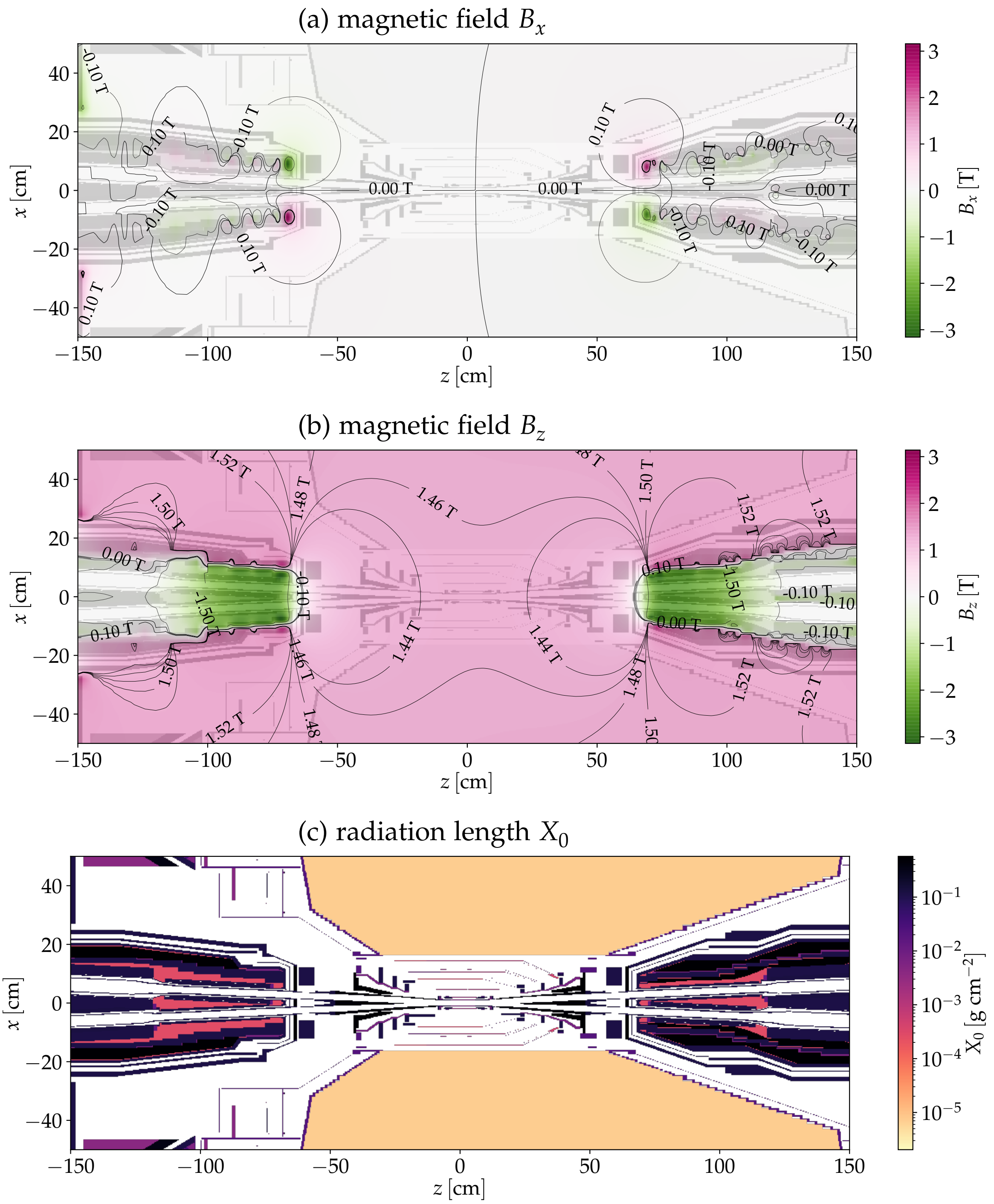


Figure 3.5.: Technical drawing of the Belle II detector viewed in a cut in the horizontal *xz*-plane [42].



(a) magnetic field B_x

(b) magnetic field B_z

(c) radiation length X_0

Figure 3.6.: (a) and (b) show a simulation of the Belle II 3D magnetic field in x-direction and in z-direction illustrating the distortions from the uniform 1.5 T solenoid field in z-direction. (c) shows the simulated radiation length of the material budget. The magnetic field an radiation length correspond to a cut through the Belle II detector in the xz-plane.

of the simulated 3D magnetic field in Belle II in units of Tesla. Only within the volume of the tracking and vertexing detectors of Belle II (CDC, SVD, PXD), the magnetic field is approximately constant in z-direction and vanishing in x-direction. The focusing quadrupole and beam line magnets disturb the uniformity of the field in the vicinity of the beam line in the forward and backward region. For values of $|z| \geq 50\,\text{cm}$ the field in x-direction deviates from zero in Fig. 3.6 (a) and it is actually inverted in the z-direction as can be seen in Fig. 3.6 (b).

These field nonlinearities and the inversion of the field in the range of $|z| \in [50, 100]\,\text{cm}$ in the crotch part region between the two beam lines on the z-axis are important for the simulation studies in the subsequent chapters of this thesis. In those regions, the tracks cannot be idealized as helices anymore and thus the usual track reconstruction method breaks down. Therefore, in machine background simulations and in single track simulations with track vertices with larger displacements along the z-axis the MC track parameter values are expected to differ from the reconstructed track parameter values. This is especially important for the neural network training and evaluation (see Sec. 6), where usually simulated single tracks are used with a uniform distribution along the z-axis. In order to cope with the expected deterioration of track parameter values due to the inverted magnetic fields and due to the crotch part material for $|z| \geq 50\,\text{cm}$, the neural network trainings presented in this thesis are limited to the region $|z| \leq 50\,\text{cm}$. If larger training regions are used, an advantage can be expected from using reconstructed tracks over MC true value as training target.

3.2.2. Material Budget Close to the IP

The particles propagating through the detector do not only hit the active detector components, but also the support structure in between them. This is especially dangerous for the innermost material, where all of the physics events pass through. Fig. 3.6 (c) shows the simulated radiation length in the material close to the IP in a xz-plane cut through the Belle II detector. Clearly, the subdetectors and the material of the support structure are visible. Since backscattered background tracks have their vertices at locations with thick material where the scattering is more likely, the thick material in the crotch part between the two beam-lines will appear as a large background source in particular ($20\,\text{cm} < |z| < 60\,\text{cm}$ and $x = 0\,\text{cm}$).

The beam pipe is the innermost material surrounding the IP [21] with an inner radius of 10 mm and an outer radius of 12 mm. All outgoing particles from the physics collisions should be able to pass through the material, so they can be reconstructed with the detector information. On the other hand, a first filter for background radiation, like photons from synchrotron radiation, is desirable. Therefore, it is constructed of the materials beryllium, paraffin, and gold; a 1 mm layer of paraffin serving as a coolant, is sandwiched between two layers of beryllium, with 0.6 mm on the inside and 0.4 mm on the outside. Since the beryllium of the beam pipe is reactive, paraffin is chosen as coolant instead of water [21]. From the inside the

Phase 3 beam pipe is layered with a 10 µm layer of gold functioning as a filter for synchrotron radiation. Due to difficulties in the production process, during Phase 2 the gold layer was applied to the outside of the beam pipe instead.

With the very small inner radius of the beam pipe of only 10 mm, the vacuum conditions at the IP are difficult to be maintained. Therefore, the achievable vacuum in the vicinity of the IP might be worse than expected. This requires a careful considerations of the beam gas background.

3.2.3. Track Parametrization

Without energy loss, the idealized motion of a charged particle in the approximately constant uniform Belle II magnetic field is completely determined by its initial conditions. Generally, the initial conditions are described by the charge q of the particle, its initial position usually referred to as the vertex $\vec{v} = (v_x, v_y, v_z)$, its momentum $\vec{p} = (p_x, p_y, p_z)$ at this vertex position. The Lorentz force law for a charged particle in a magnetic field is

$$\vec{F} = \frac{\mathrm{d}\vec{p}}{\mathrm{d}t} = \frac{q}{m} \cdot \vec{p} \times \vec{B} \tag{3.7}$$

where $\vec{F}$ is the force on the particle, q is its charge, m its mass, $\vec{p}$ its momentum and $\vec{B}$ the magnetic field. In a constant magnetic field in z-direction ($\vec{B} = (0, 0, B)$), the z component of the Lorentz force is $F_z = 0$. Therefore, the particle trajectory can be separated in its transverse motion perpendicular to the magnetic field and in its longitudinal motion parallel to the z-axis.

At Belle II such a helix is commonly described in the Perigee track parametrization, a local parametrization around the fixed perigee point [54]. This perigee point is the position of closest approach of the track to the z-axis and is referred to as vertex position in the trigger applications throughout this thesis. In the Belle II tracking algorithms the following five free parameters describe the helical tracks:

$$\text{helix} = (\omega, \phi_0, \theta, z_0, d_0). \tag{3.8}$$

The parameter d_0 is the signed distance of the vertex point to the z-axis and z_0 is its position along the z-axis. The azimuthal angle ϕ_0 and the polar angle θ describe the direction of the track momentum at the vertex. The parameter $\omega = q/r$ is the signed curvature of the track in the transverse plane, with a radius r. For the single charged particles at Belle II, $q \in [-1, 1]$ is simply the sign of the charge. This radius r is proportional to the transverse momentum p_T and the charge q identifies the direction of the propagation along the circle. The magnitude p_T is a constant of motion and can be related to the radius r of the circle by:

$$r[\mathrm{m}] \approx \frac{p_\mathrm{T}[\mathrm{GeV}]}{0.3 \cdot B[\mathrm{T}]} \tag{3.9}$$

where the factor of 0.3 approximates the speed of light c.

Since the Lorentz force changes the transverse momentum direction but not its magnitude, the particle is kept on this circular trajectory. At the vertex position, the Cartesian transverse momentum vector $\vec{p}_\mathrm{T} = (p_x, p_y)$ is related to p_T and ϕ_0 by

$$\vec{p}_\mathrm{T}(p_\mathrm{T}, \phi_0) = p_\mathrm{T} \begin{pmatrix} \cos(\phi_0) \\ \sin(\phi_0) \end{pmatrix} \qquad p_\mathrm{T} = \sqrt{p_x^2 + p_y^2} \qquad \phi_0 = \arctan\left(\frac{p_y}{p_x}\right). \tag{3.10}$$

Unaffected by the magnetic field, the longitudinal momentum p_z stays constant. Using the polar angle θ it is related to the transverse momentum by

$$\cot(\theta) = \frac{p_z}{p_\mathrm{T}}. \tag{3.11}$$

The transverse momentum direction at the transverse vertex position $\vec{v}_\mathrm{T} = (v_x, v_y)$ is perpendicular to the distance of closest approach of the track to the beamline. Hence, the direction of d_0 is given by ϕ_0 shifted by $\frac{1}{2}\pi$, where the sign of d_0 describes the shift direction. In the Belle II coordinates, the sign of d_0 is defined as the sign of $\vec{v}_\mathrm{T} \times \vec{p}_\mathrm{T}$: In this way the vector $\vec{v}_\mathrm{T}$ can be defined by a fixed shift $\phi_0 - \frac{1}{2}\pi$.

$$\vec{v}_\mathrm{T}(d_0, \phi_0) = d_0 \begin{pmatrix} \sin(\phi_0) \\ -\cos(\phi_0) \end{pmatrix} \qquad d_0^2 = v_x^2 + v_y^2 \qquad \tan\left(\phi_0 \pm \frac{\pi}{2}\right) = \frac{v_y}{v_x} \tag{3.12}$$

Since $|d_0|$ is much smaller than the track radius ($|d_0| \ll r$), at the trigger level setting d_0 to zero is usually a valid assumption. This means the tracks have their vertices at the IP in the transverse plane. In contrast, d_0 becomes quite important in the precise offline vertex reconstruction.

Track Propagation

With respect to the magnetic field, the track momentum is a superposition of a transverse component $|\vec{p}_\perp| = p_\mathrm{T}$ and a parallel component $|\vec{p}_\parallel| = p_z$. The transverse circular part of the helix trajectory is illustrated in Fig. 3.7. In order to describe the motion, the transverse position of the central helix axis is defined at $\vec{v}_\mathrm{T} + \vec{m}$, where the sign of the particle charge determines the sign of $\vec{m}$.

$$\vec{m}(\omega, \phi_0) = \begin{pmatrix} m_x \\ m_y \end{pmatrix} = \omega^{-1} \begin{pmatrix} \sin(\phi_0) \\ -\cos(\phi_0) \end{pmatrix} \tag{3.13}$$

where $\omega = q/r$ is used. In order to describe the points along the 2D track in the transverse plane, the 2D arc length s along the track is used. This arc length is expressed as $s = r \cdot |\psi|$, where ψ is the angle along the 2D projection of the track and r is the radius of the track circle in the 2D plane (see Fig. 3.7). The sign of ψ corresponds to the charge q and designates the direction of the rotation. In the definition used here, positive ψ describe clockwise rotations and negative ψ counterclockwise rotations.

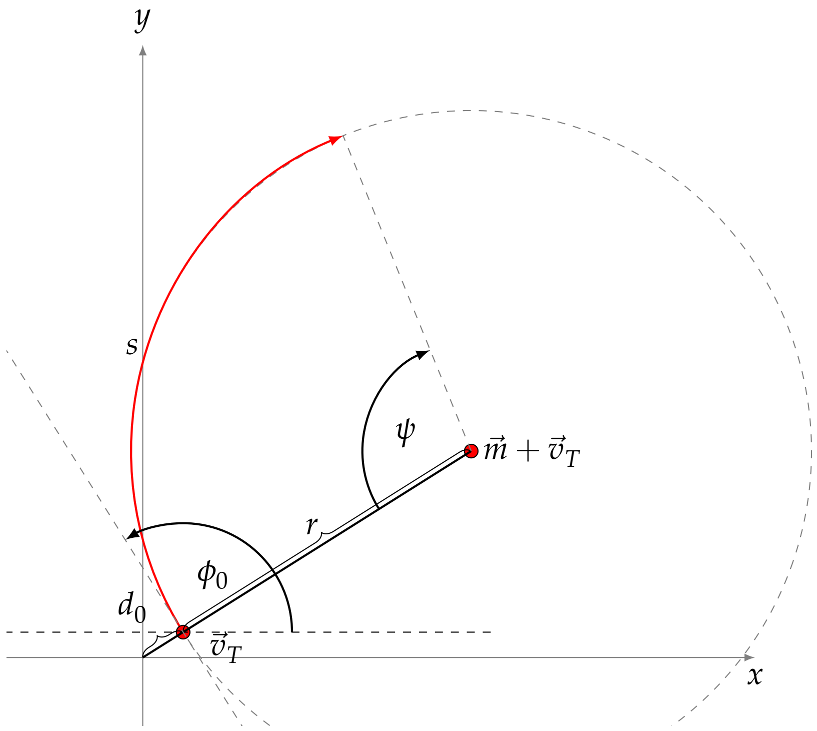


Figure 3.7.: Transverse parameters of a track displaced from the IP. The transverse vertex position $\vec{v}_{\mathrm{T}}$ is the position of closest approach to the z-axis with the displacement $|d_0| = |\vec{v}_{\mathrm{T}}|$. The track radius is $r = |\vec{m}| \propto p_{\mathrm{T}}$ and its center is located at $\vec{m} + \vec{v}_{\mathrm{T}}$. The angle ϕ_0 defines the tangential direction of the track momentum at $\vec{v}_{\mathrm{T}}$. The circle center $\vec{m}$ and the vertex position $\vec{v}_{\mathrm{T}}$ are perpendicular to the tangent vector, described by the shift $\phi_0 \pm \frac{1}{2}\pi$. The sign of the charge q identifies the sign of the track curvature and of the sign of the ϕ-shift for $\vec{m}$. The sign of d_0 describes the ϕ-shift for $\vec{v}_{\mathrm{T}}$. Along its path in the xy-plane the track covers an angle ψ or equivalently an arc length $s = r|\psi|$.

Starting from the vertex position $\vec{v}_{\mathrm{T}}$ the transverse motion is described by a rotation

$$R[-\psi](-\vec{m}) + \vec{m} = \omega^{-1} \begin{pmatrix} \sin(\psi - \phi_0) + \sin(\phi_0) \\ \cos(\psi - \phi_0) - \cos(\phi_0) \end{pmatrix} \tag{3.14}$$

where $R[-\psi]$ denotes the rotation matrix, rotating by an angle $-\psi$ around the origin. After rotating the position $-\vec{m}$ around the origin by an amount $-\psi$, the resulting vector is shifted to its supposed position by adding $\vec{m}$. By adding $\vec{v}_{\mathrm{T}}$ (Eq. 3.12) the offset of the initial vertex is reinstalled.

Since p_{T} and p_z are constant, the polar angle θ also describes the relation of the transverse path length s to the longitudinal path length $z - z_0$

$$\cot(\theta) = \frac{z - z_0}{s} \qquad\qquad z = z_0 + \frac{\psi}{\omega}\cot(\theta) \tag{3.15}$$

Putting the pieces together, the full helix equation in terms of the perigee track parameters with the rotation angle ψ as free parameter becomes

$$\vec{h}(\psi) = \omega^{-1}\begin{pmatrix} \sin(\psi - \phi_0) + \sin(\phi_0) \\ \cos(\psi - \phi_0) - \cos(\phi_0) \\ \psi\cot(\theta) \end{pmatrix} + \begin{pmatrix} d_0\sin(\phi_0) \\ -d_0\cos(\phi_0) \\ z_0 \end{pmatrix}. \tag{3.16}$$

3.2.4. Beast Commissioning Phases

The upgraded acceleration and beam injection methods and the completely new collision mechanism with the nano beam scheme [43, 44] need to be commissioned first. Accidents like unintended loss of the beam or heavily increased background rates can be expected in the early runs. The methods need to be tested and corrections need to be applied in order to avoid radiation damage to the sensitive detectors. To this end, the start up of Belle II is carried out in three main commissioning phases.

The first two commissioning phases for Belle II run under the name BEAST II, with Phase 1 in 2017 and Phase 2 in 2018; the name BEAST II is adopted from the predecessor experiment "Belle and Beast", where it refers to the acronym (Beam Exorcism for A STable belle II experiment) and to the fairytale "Beauty and the Beast". During BEAST II, special commissioning detectors are installed to monitor the backgrounds at Belle II. Especially the sensitive silicon detectors, the PXD and the SVD, are only installed completely in Phase 3.

During Phase 1 no collisions occurred, since the runs were carried out without the final focussing magnets. A main goal was to study the single beam conditions, test the injection, and monitor the single beam backgrounds. Details on the beast commissioning phases and the main results of Phase 1 are published in [55]. Due to the lower beam current and luminosity expectations for Phase 2, different background simulations were carried out for Phase 2 (see Sec. 4). Since the goal luminosity will only be reached in the late Phase 3, the background levels from Phase 2 are similar to the background levels in the present early Phase 3. Therefore, the neural network trigger (see Sec. 6) running in hardware during the early Phase 3 is still trained with the Phase 3 background expectations.

3.2.5. Subdetectors of Belle II

A short review on all the subdetectors involved in Belle II is given in this subsection. Special focus is set on their readout times, precision and relations to the trigger system. Furthermore, the risk of background and possible countermeasures for the subdetectors are discussed.

K Long and Muon detector - KLM

The K Long (K^0_L) and Muon (μ) detector (KLM) is the outmost detector of Belle II. It consists of 14 layers of iron plates with a thickness of 4.7 cm with an interaction length of $\lambda = 3.9\,\text{cm}$ sandwiched with 15 layers of glass electrode Resistive Plate Chambers (RPC) as active detector material [21]. The iron plates allow hadronic showering of the K^0_L. The KLM is split up into a barrel part with a polar angle coverage of 45 degree to 125° and endcap parts extending the full coverage to 20° to 155°.

Electromagnetic Calorimeter - ECL

The Electromagnetic CaLorimeter (ECL) is the second last detector shell. Like the KLM, it is divided into a barrel part and an endcap part with an inner radius of $r = 1.25\,\text{m}$ in the barrel region and endcaps placed at $z = 1.96\,\text{m}$ in the forward region and $z = -1.02\,\text{m}$ in the backward region [21]. Its active material are scintillating CsI(Tl) crystals covering a total polar angle of 12.4° to 155.1°. However, there are two small gaps of $\approx 1^\circ$ between the barrel and endcap parts [21]. Photodiodes are used to detect the scintillation light of the 6624 crystals in the barrel region and the 2212 crystals in the endcaps. Its main purposes are the detection of electromagnetically interacting particles, which create electromagnetic showers in the crystals. Most importantly, photons and electrons can be identified. Due to its fast readout time, the ECL signals are used in the L1 trigger. Especially, a Bhabha veto is realized by a matching of the ECL clusters to the CDC tracks for electron particle identification.

Particle Identification detectors - PID: ARICH and TOP

For Particle IDentification (PID) two different detectors are used in the barrel and in the endcap region: ARICH and TOP which are both based on the Cherenkov effect. The Time Of Propagation (TOP) detector is installed in the barrel region of Belle II and the Aerogel Ring Imaging Cherenkov (ARICH) detector covers the forward endcap region (see Fig. 3.5). The PID detectors allow to discriminate pions and kaons in particular [21].

In the ARICH detector, particles passing through the aerogel layer of 2 cm thickness generate Cherenkov photons. At a distance of 20 cm the photon detectors detect the rings of Cherenkov photons. The ARICH detector allows for pion, muon, and electron identification for particles with energies below 1 GeV in addition [21].

The TOP counter is also based on the Cherenkov effect; the Cherenkov photons inside the quartz are reflected until they reach one end of the crystal where the signal is measured with Photo Multiplier Tubes (PMT). Due to the different Cherenkov angles, the time of propagation for the photons is different. Thus, the Cherenkov angles can be measured by measuring the time differences of the Cherenkov photons at the PMTs [21]. The TOP detector is used in the L1 trigger for precise timing estimates.

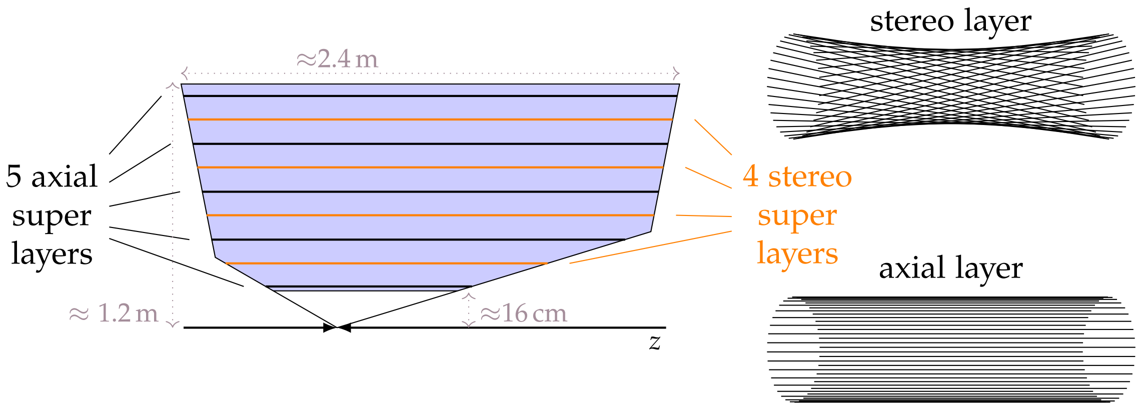


Figure 3.8.: Illustration of the alternating SL configuration in the CDC. Five axial SL are inclined with four stereo SL with a twist w.r.t. the z-axis.

Central Drift Chamber - CDC

The Central Drift Chamber (CDC) is the main tracking device of Belle II [21, 56]. It is a wire chamber containing 14336 sense wires oriented closely parallel to the z-axis with 42240 field wires in between, creating a high voltage electric field. Charged particles traversing the CDC volume ionize the gas molecules, where the volume is filled with a special gas mixture of 50 % Helium and 50 % Ethan. The electric fields accelerate the free electrons from the field wires towards the sense wires, where the constant pressure in the gas admixture assures an almost constant drift velocity. In contrast to a Geiger counter, a complete ionization of the gas is avoided by operating the CDC at a lower voltage; only in the close vicinity of the sense wires, the electrons pick up sufficient energy to ionize further gas molecules on their way. Due to this so called Townsend avalanche [57], a measurable amount of electrons arrives at the sense wires.

The drift time is the difference of the time of the measured signal at a sense wire relative to the time of the initial ionization. Due to the almost constant drift velocity of the electrons in the gas admixture, the drift time is an important distance measure of a track to a wire. This xt-relation is illustrated in Fig. 3.9.

The sense wires are arranged in 56 layers, where six to eight neighboring layers of sense wires with the same orientation are combined into nine Super Layers (SLs). Five axial SLs (A) consisting of wires parallel to the z-axis, are inclined with four stereo SLs of wires with different stereo angles with the z-axis (U, V). In short, the orientation of the SLs is AUAVAUAVA [21]. Fig. 3.8 shows a sketch of the alternating SLs and their orientation. The stereo angle ranges of the wires in the stereo SLs are listed in Tab. 3.3. An illustration of the ϕ-shifted stereo wire mounting positions and the relation to the stereo angle is illustrated in Chapter 5 in Fig. 5.6. These stereo angles enable a 3D track reconstruction with the CDC. Further discussions on the geometrical symmetry of the wires can be found in Sec. 5.2.3.

The innermost axial SL (A) contains eight axial layers, all other SLs contain six

SL-id	SL	`nshift`	stereo angle [mrad]
1	U	18	$[45.4, 45.8]$
3	V	-18	$[-55.3, -64.3]$
5	U	19	$[63.1, 70.0]$
7	V	-20	$[-68.5, -74.0]$

Table 3.3.: Configuration of the stereo SLs. The U SLs have positive stereo angles, the V SLs negative stereo angles. Within a stereo SL the mounting position of the wires at the endplates has a fixed offset in the number of holes (`nshift`). Therefore, the stereo angle within a stereo SL slightly increases with the radius. The offset `nshift` corresponds to a ϕ-difference $\Delta\phi = \texttt{nshift} \cdot \pi/\texttt{nwires}$ with `nwires` the number of wires in the SL. Since sense wires are inclined with field wires the number of holes per layer is twice the number of sense wires.

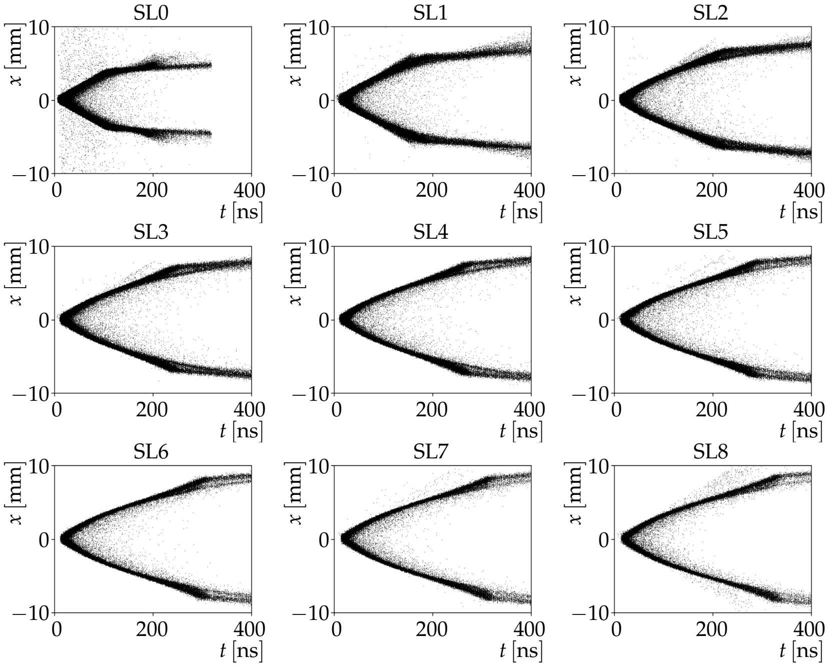


Figure 3.9.: Simulated xt-relation in the nine CDC SLs. The approximately linear regime is proportional to the cell sizes. The times are calculated from the simulated TDC counts after subtracting the t^0_{wire} of the wires.

layers of wires with the same orientation. In order to achieve a higher tracking precision close to the IP, the innermost SL has a smaller spacing between the wires, the small cell chamber [21]. The The simulated drift time to distance relation, a.k.a. xt-relation, is shown in Fig. 3.9 separately for the 9 SLs. In all SLs, an approximately linear regime can be seen. However, the length of the linear regime differs. It is shortest in the innermost SL 0, where the drift cells have a smaller spacing. In SLs 1-8, the maximum drift times in the outer layers are slightly larger. Beneficially, the slope of the xt-relation in the linear regime allows an easy estimation of the drift length using the drift times as input.

In the CDC readout electronics, the time of the signals t_{tdc} is described by a 13 bit number in units of TDC (Time to Digital Converter) counts, the clock cycles of the readout system with a clock frequency of $f_{\text{tdc}} = 1.017\,774\,\text{GHz}$. Hence, one TDC count is approximately one nano second

$$t\,[\text{tdc clock}] = t\,[\text{ns}] \cdot f_{\text{tdc}}\,[\text{GHz}]. \tag{3.17}$$

Since the TDC counts in the experiment are continuous clock counts, the actual drift times can only be obtained as a difference to an zero timing t^0.

In the simulation, where the true drift times $t'_{\text{drift}}\,[\text{ns}]$ are known, the tdc count $t_{\text{tdc}}\,[\text{tdc counts}]$ is calculated inversely from the drift time relative to a zero timing of the wire $t^0_{\text{wire}}\,[\text{ns}]$ and relatively small corrections δt^0

$$t_{\text{tdc}}\,[\text{tdc clock}] = \left\lfloor \left(t^0_{\text{wire}} - \delta t^0 - t'_{\text{drift}}\right) \cdot f_{\text{tdc}} + 0.5 \right\rfloor \tag{3.18}$$

where the value of 0.5 [tdc clock] is added to obtain a rounding with the floor function $\lfloor\rfloor$. In the present CDC simulation, the zero value of each wire t^0_{wire} is globally defined and has the same value for each wire of $t^0_{\text{wire}} \cdot f_{\text{tdc}} = 4900\,[\text{tdc clock}]$.

The additional correcting delays δt^0 included in the TDC calculation are small compared to the drift time. The following terms are included in the simulation of this correction term:

$$\delta t^0\,[\text{ns}] = t^0_{\text{evt}} + t_{\text{tof}} + t_{\text{prop-delay}} + t_{\text{time-walk}} \tag{3.19}$$

where t^0_{evt} is an event timing of the event, t_{tof} is the time of flight of a track which increases with the SL number, $t_{\text{prop-delay}}$ is the time required for the signal propagation from the wire hit position to the readout electronics and $t_{\text{time-walk}}$ is a time shift dependent on the pulse height of the signal in the wire. For hits related to a simulated event the event timing is zero ($t^0_{\text{evt}} = 0$), while this value is chosen randomly in the background simulations (see Sec. 4.4).

The realistic timing t_{drift} shown in Fig. 3.9 is calculated from the simulated TDC count t_{tdc} by subtracting t^0_{wire}:

$$t_{\text{drift}}\,[\text{ns}] = t^0_{\text{wire}} - t_{\text{tdc}}/f_{\text{tdc}} = t'_{\text{drift}} + \delta t. \tag{3.20}$$

Therefore, the time smearing in Fig. 3.9 corresponds to the expected realistic drift time smearing of the CDC which is also seen at the L1 trigger level (see Sec. 3.3.3).

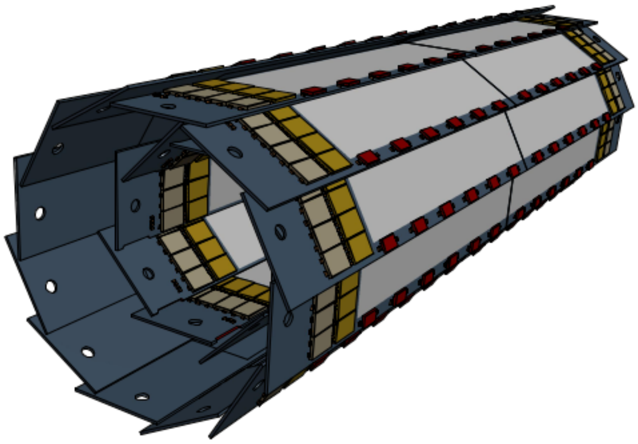


Figure 3.10.: Schematic picture of the Belle II PiXel Detector (PXD). The gray area contains the active detector area consisting of the sensitive pixels with the DEPFET [58] technology.

Silicon Vertex Detector - SVD

Inside of the CDC is the Silicon Vertex Detector (SVD) [21]. It consists of four layers of double-sided silicon strip detectors (DSSD), with perpendicular strip orientations on the up- and downside for stereo reconstruction. Its inner radius is 38 mm and its outer radius is 140 mm. In the polar angle θ it covers the region between 17° to 150°. In order to avoid multiple scattering, the material budget is kept low. The SVD improves the tracking efficiency for low momentum tracks. Additionally, it is used in combination with the PXD for precise vertex reconstruction.

The readout time of the SVD is too slow for the L1 trigger, but SVD information is included in the HLT trigger. During the readout, the SVD signals are stored in a ring buffer, which is only read out upon a positive L1 trigger signal. This ring buffer can maximally store the signals for $\approx 5\,\mu s$, which sets a limit for the maximum latency of the L1 trigger.

Pixel Detector - PXD

The silicon pixel detector (PXD) is the innermost detector of Belle II. It is a silicon detector based on the DEPFET [58] (DEpleted P-channel Field Effect Transistor) semiconductor technology. Ionizing radiation creates free charges in the silicon layer which are accumulated by the DEPFET pixels in an internal gate (for a detailed description on the PXD see [21, 59]). Its sensitive areas are arranged in two layers of ladders, shown in Fig. 3.10. Each ladder consists of two sensor modules glued together, where each module is a flat, planar detector. In order to cover the full azimuthal angle ϕ the ladders are arranged in a windmill structure with little overlaps.

layer	pixel	size $R\phi \times z$
1	256 close to IP	$50\,\mu m \times 55\,\mu m$
1	away from IP	$50\,\mu m \times 60\,\mu m$
2	256 close to IP	$50\,\mu m \times 70\,\mu m$
2	away from IP	$50\,\mu m \times 85\,\mu m$

Table 3.4.: The different pixel sizes in the PXD. In both layers, the 256 pixels closest to the IP have a smaller size.

The first layer has a radius of 14 mm and consists of 8 ladders and the second layer has a radius of 22 mm and consists of 12 ladders.

The active area in the sensors has a thickness of 75 μm and contains 768 × 250 pixels of varying size. Close to the IP, where tracks are expected to traverse the material perpendicular, smaller pixel sizes are used. Pixels displaced from the IP in the z-direction, where signal tracks from the IP have a flat angle with the detector, have a larger pixel size. The different pixel sizes are listed in Tab. 3.4.

The mean energy loss of charged particles traversing the silicon material is approximately constant during its path through the material.

$$\left\langle \frac{\mathrm{d}E}{\mathrm{d}x} \right\rangle \approx const. \tag{3.21}$$

For this reason, a single charged particle usually creates a signal in several neighboring pixels. To measure the total amount of charge deposited by a single particle, the pixel information is clustered. The analysis methods then operate on PXD cluster information. In the vertexing algorithms, the PXD clusters are matched to tracks and the full cluster information is used to achieve a higher precision.

For the clustering procedure to be functional, the overall occupancy in the PXD must not get too large. With a higher occupancy, the probability of distinct clusters to be connected increases. To avoid this combinatory problem, a maximum occupancy of 3 % is required, but ideally the occupancy should not exceed 2 % [60, 59].

The readout time for the PXD is quite low and the amount of produced data is huge (10× more data than all other detectors combined). For this reason, the PXD is not used in any trigger (L1 and HLT). The information is only stored for offline analysis upon a positive signal from both triggers.

3.3. The Trigger System

Early data reduction by the trigger system in Belle II is crucial: not all collected information can actually be read out due to bandwidth limitations and long readout times in several subdetectors. Additionally, the readout data rate is huge, which makes data reduction desirable to minimize the storage space.

physics process	cross section [nb]	rate [Hz]
$\Upsilon(4S) \to B\overline{B}$	1.2	960
hadron production from continuum	2.8	2200
$\mu^+\mu^-$	0.8	640
$\tau^+\tau^-$	0.8	640
Bhabha ($\theta_{\mathrm{lab}} \geq 17^\circ$)	44	350 $^{(a)}$
$\gamma\gamma$ ($\theta_{\mathrm{lab}} \geq 17^\circ$)	2.4	19 $^{(a)}$
2γ ($\theta_{\mathrm{lab}} \geq 17^\circ$, $p_{\mathrm{T}} \geq 0.1\,\mathrm{GeV}$)	≈ 80	≈ 15000
total	≈ 130	≈ 20000

(a) rate is prescaled by a factor 1/100 due to distinct signature

Table 3.5.: Cross sections and expected trigger rates for various physics processes at Belle II [21]. The rates are the expected output rates of the L1 trigger.

With the anticipated luminosity at Belle II, high levels of background can be expected (see Chap. 4). Everything that is not an interesting physics signal from an intended collision, can be considered background. In the case of collision events at the IP, especially the QED Bhabha channel ($e^+e^- \to e^+e^-$) has a very large cross section. In favor of the interesting flavor physics channels, these Bhabha events have to be detected and vetoed at the L1 trigger. The expectations for cross sections and trigger rates for important physics processes are shown in Tab. 3.5 [21]. The rates for processes with a high cross section that can be identified by their distinct signature are prescaled by a factor of 1/100 [21].

Background tracks not pointing to the IP can be reduced by carrying out a 3D track reconstruction of the tracks. This is the main goal of the 3D track triggers developed in this thesis, which are described in detail in Chapter 6 on the neural network trigger. Details of the expected background types and their rates are presented in Chapter 4.

An ideal trigger would select all the interesting physics events, while rejecting all background events and uninteresting physics events with a large cross section. In reality, a tradeoff on the trigger cuts has to be made: most importantly, a high efficiency is required for the desired physics events. In order to avoid a rejection of the signal events, the trigger cuts should be maximally relaxed. However, loose trigger cuts degrade the signal to background ratio and increase the total trigger rate which is required to stay below 30 kHz.

The Belle II trigger system consists of two online trigger steps: a first level (L1) trigger [41, 61] with a 30 kHz output rate at a maximum latency of 5 μs and a following High Level Trigger (HLT) with a 10 kHz output rate [62] which is based on a software reconstruction with basf2. The software basf2 is the Belle II analysis software framework used for simulation and analysis of the Belle II events (see Sec. 3.4). Only upon a signal from the L1 trigger, the full information is read out of all subdetectors (except the PXD) and fed through the HLT for further background rejection.

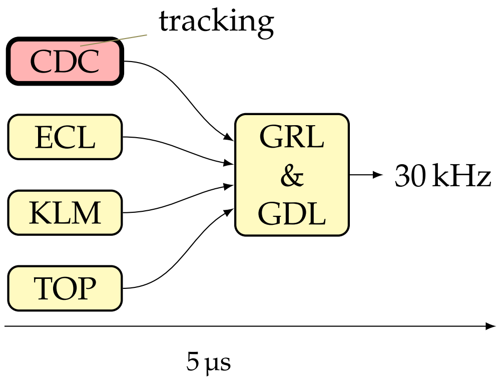


Figure 3.11.: Data flow in the L1 trigger. Within a maximum latency of 5 µs the subtrigger results are combined in the GDL with a trigger rate of 30 kHz.

The online L1 trigger algorithms are implemented in FPGA (Field Programmable Gate Array) parallel hardware. Since the runtime optimized L1 trigger results cannot reach the full offline precision, the L1 triggered events still contain a substantial amount of background. To this end, before the events are stored for offline analysis via the Data AQuisition system (DAQ), a further reduction of the trigger rate to 10 kHz is achieved by the HLT. The HLT uses the full offline reconstruction algorithms within basf2, including all detectors except the PXD. With 1 MB per event at the expected occupancy of 2 % the PXD data rate is too large [63] for the trigger system. Therefore, the HLT results are used to find Regions of Interest (RoI) on the PXD [64] such that only the reduced hit information in the RoIs are read out from the PXD for offline analysis.

3.3.1. L1 Subtriggers

The L1 trigger is required to have a high efficiency for hadronic signal events from $\Upsilon(4S) \to B\overline{B}$ and continuum events [41, 21]. Additionally, the L1 trigger has to be dead-time free and fully pipelined processing new data every 32 ns while minimizing the total latency. Hence, all components are implemented on the Universal Trigger Boards 3 (UT3), a specialized FPGA board designed for the application in the Belle II trigger system. An improved FPGA board, the Universal Trigger Board 4 (UT4) housing a newer FPGA with more memory and computation and IO capabilities, is currently under development.

Within a maximum latency of 5 µs the L1 subtrigger pipelines process the detector signals to allow a final trigger decision in the Global Decision Logic (GDL) of the L1 trigger. Due to the tight timing constraints, only subdetectors with a fast readout time can be used by the online L1 trigger algorithms. The four L1 subtrigger pipelines illustrated in Fig. 3.11, where the CDC is the only tracking device.

The ECL subtrigger provides important information for triggering neutral and

Category	Item
Physics	$\Upsilon(4S)$ / continuum τ pair
Calibration	Bhabha $\gamma\gamma$ μ pair Random Trigger
Veto	Beam injection two-photon events

Table 3.6.: Exemplary list of basic L1 trigger logics [21].

charged particles [41]. It measures the energy depositions in so called Trigger Cells (TC), a combination of 4 × 4 crystals. By measuring the total energy deposition in all TCs, events with a large electromagnetic energy deposition are detected. Counting the number of isolated clusters allows to recognize multi hadron events. The ECL trigger is well suited to detect Bhabha and $\gamma\gamma$ events. The online Bhabha measurement is the well suited method to measure the luminosity. Additionally, it can be used in the identification of cosmic events. The ECL trigger has a good timing resolution of 20 ns [41], which is especially relevant in the endcap region to complement the timing information of the TOP trigger.

The KLM trigger provides online muon detection. Due to its high efficiency, it can be used for efficiency measures of the other subdetectors. The TOP trigger in the barrel region provides precise information on the timing of $\mathcal{O}(1\,\mathrm{ns})$ and on the hit topology. At each clock cycle the arriving photons are sampled from the 60×8 detectors. A queuer orders the hits and encodes them with 9 bit resolution [41]. The CDC trigger is the sole tracking device at the L1 trigger. It has a Hough track finder in the $r\phi$-plane followed by 3D track reconstruction. In addition it provides event time information, but with less precision than the TOP and ECL timing. The details of the CDC subtrigger pipeline are described in Sec. 3.3.2.

GRL & GDL

The information from all the L1 subtrigger pipelines is combined in the Global reconstruction Logic (GRL) and Global Decision Logic (GDL) to issue the trigger decision. The GRL improves the trigger decision by carrying out a low level event reconstruction. It matches information from several subtriggers which allows the detection of special event types (e.g. Bhabha events). A matching of CDC and ECL is done by calculating Δr and Δz values of the ECL cluster positions with the extrapolated CDC tracks [65]. It allows the distinction of ECL clusters related to neutral or charged particles and thus the identification of electrons, hadrons and photons. Matching with the KLM is used for particle identification and the detection of cosmic tracks.

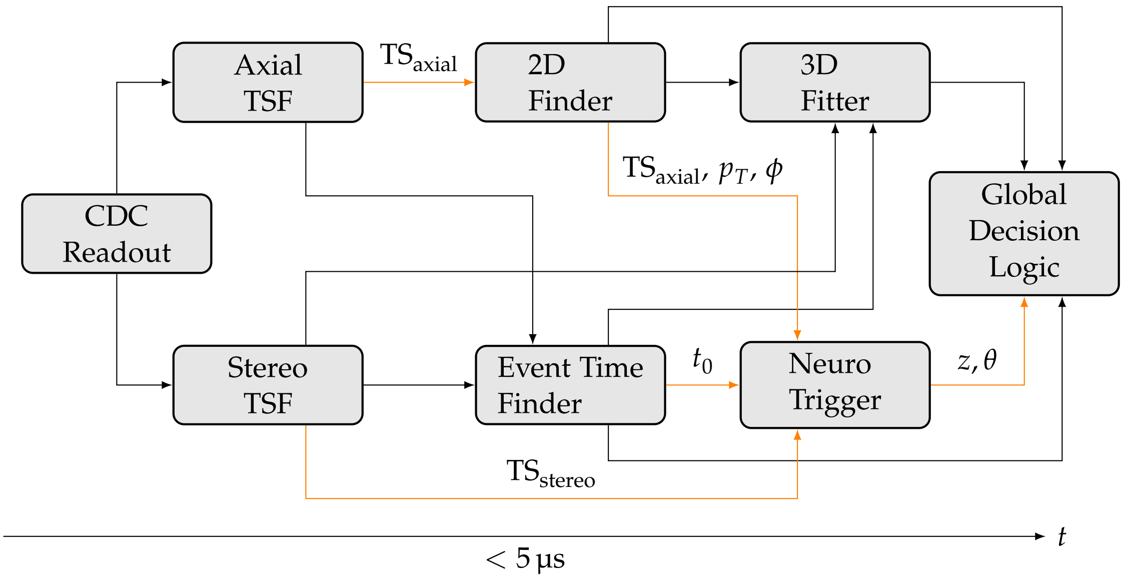


Figure 3.12.: Components in the L1 CDC trigger pipeline. Each consecutive component is implemented on several parallel FPGA boards (UT3 - Universal Trigger Board 3), each covering a different sector of the phase space.

Additionally, the final event time is calculated by matching the ECL and TOP timing.

The GDL directly receives summary information from each subtrigger pipeline and the results from the GRL in addition. Logical combinations are calculated in order to identify three main trigger categories: physics triggers, calibration triggers and veto triggers. An exemplary list of basic trigger logics is shown in Tab. 3.6. Physics triggers are optimized for event signatures like $\Upsilon(4S)$ or τ-pair events. The calibration triggers comprise for example μ-pair events and Bhabha events which can be used for luminosity measurements and random triggers. Typically event signatures recognized as calibration triggers have large rates and therefore a prescaling factor is used for these events [41]. The veto triggers stop data taking during injection to avoid injection background and reject two-photon events. The trigger signals from the GDL are finally sent to a sequence controller within the maximum latency of $5\,\mu\mathrm{s}$ after the collision.

3.3.2. CDC Trigger Pipeline

The internal data flow in the L1 CDC trigger pipeline, where charged tracks are reconstructed, is illustrated in Fig. 3.12. The first components are the merger boards collecting the information from the CDC readout system and preparing it for the Track Segment Finder (TSF) boards. This TSF is a first noise suppression and data reduction step, where the wire hit information from CDC sense wires in 56 layers are combined to Track Segment (TS) information in the 9 CDC SLs. The TSF is followed

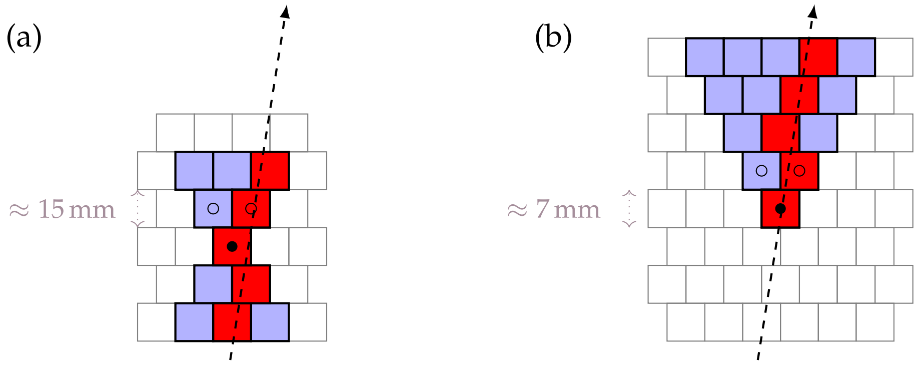


Figure 3.13.: The shapes of the Track Segment (TS) with inactive wires shown in blue and the active wire hits in red caused by an ionizing track shown as dashed arrow. (a) The hourglass shape of the outer TSs used in eight outer SLs. (b) The shape of the TSs in the innermost SL within the small cell chamber

by the 2D Hough track finder. Using only the axial TS hit information, it finds 2D tracks parametrized with (p_{T}, ϕ) in the transverse plane. In parallel, an Event Time Finder (ETF) is foreseen, which estimates the event time using the fastest hit information of the TSs. The last components are the 3D track reconstruction boards. They receive the combined information from the 2D finder, the ETF, and all TS information (axial and stereo) as input and enrich the found 2D tracks with the precise 3D track information on, parametrized by (z, θ). The estimated 3D track parameter information $(p_{\mathrm{T}}, \phi, \theta, z)$ of all found tracks, is finally used in the GRL/GDL to make the global trigger decision. Details on the Hough track finding method are presented in Chapter 5, where an improved 3D track finder is proposed as an upgrade of the existing 2D track finder. The methods for precise 3D track reconstruction, namely the LS fit and the neural network trigger, are described in Chapter 6 in detail.

3.3.3. Track Segment Finder

The Track Segment Finder (TSF) combines several wire hits in the CDC layers to the compressed Track Segment (TS) hits within each SL. In total, nine TSF boards are used with one TSF for each SL. In every 32 ns cycle of the trigger pipeline, each of these TSF boards sends up to 10 TSs to the following trigger boards. An upgrade to a 15 TSs version is presently under development.

The TSF searches for geometrical patterns of wire hits in close vicinity, with the shapes shown in Fig. 3.13. Each box denotes a single drift cell, with a sense wire in the center and field wires at the boundaries. The blue and red areas are the cells within the TS pattern, where only the red wire cells have a hit. In the eight outer SLs, the TSs have the hourglass shape shown in Fig. 3.13 (a), while in the innermost SL the pyramid shape shown in Fig. 3.13 (b) is used. As shown in Fig. 3.13, both TS

var	description	interpretation	values
TS-ID	id of the TS	transverse position of the TS (ϕ, r)	[0..2336]
prio	priority wire	reference position within the TS	$[0, 1, 2, 3]]$
t	drift time	distance of the track to the piro wire	$[0..t_{max}]$ ns
LR	left right info	relative location (track to prio wire)	[l, r, 0]
t_{fast}	fastest time	smallest drift time at any wire	$[0..t_{max}]$ ns

Table 3.7.: The compressed information contained in the Track Segment (TS) hits. The 2D Finder uses only the position information of the axial TS as input (TS-ID). The 3D Trackers (neural network and fitter) use all but the t_{fast} time information as input. Only the ETF uses the t_{fast} information.

patterns consist of only five wire layers; the sixth sense layer in the outer SLs and the inner three sense layers in the innermost SL are left out. A TS is counted as active TS hit, if at least four out of five layers of sense wires within the TS pattern have a hit.

The information provided for each active TS is listed in Tab. 3.7. Only the drift time t at a single reference wire, the priority wire, is used for the following track reconstruction. By TS-ID a single TS can be uniquely identified and prio is the position of a single priority wire within the TS. The first priority wire is the layer where the TS pattern contains only a single wire, shown as filled circle in Fig. 3.13. The second priority wires are the two wires in the next outer layer, shown as loose circles in Fig. 3.13. If the first priority wire is hit, it is used as reference wire for the further processing. Otherwise, one of the two second priority wires is used. If both second priority wires have a hit, the wire with the shorter drift time is selected. Additionally, a Left/Right information (LR) is extracted from the hit pattern using a lookup table. This LR information resolves the drift time ambiguity of the track with respect to the priority wire. In order to allow for an optimal event time estimation by the ETF, the TS information includes the drift time of the fastest hit within the TS in addition. However, this information is not available as input for the neural network trigger. Using the geometry information of the detector (the wire positions, the nonlinear xt-relation, stereo angles of the stereo wires, etc.) this TS hit information is crucial for the subsequent track finding an track reconstruction algorithms.

The drift times provided by the TSF are calculated as counts of a trigger clock with a resolution of approximately 2 ns in units of [trg clocks], where one [trg clock] corresponds to two [tdc clock] (see Eq. 3.18 in Sec. 3.2.5). In the simulation, the drift times t_{trg} in units of trigger clocks are calculated from the TDC counts t_{tdc} in units of tdc clocks

$$t_{\text{trg}} = \left\lfloor \frac{t^0_{\text{wire}} \cdot f_{\text{tdc}} - t_{\text{tdc}} + 0.5}{2} \right\rfloor \tag{3.22}$$

where t^0_{wire} is a globally defined zero value of each wire and f_{tdc} is the fixed clock

parameter	description
ω	signed curvature of the found track (34 Hough bins)
ϕ	azimuthal track angle at the IP (160 Hough bins)
TS	list of maximally five related axial TS

Table 3.8.: Track information provided by the 2D finder for each found track as input to the 3D fitter and to the neural network trigger. In the 2D Hough plane with a size 34 × 160, clusters identify found tracks and the 2D track parameter estimates ω and ϕ are calculated as cluster centers.

rate of the TDC readout. In order to obtain a drift time which can be used as a distance information of the track to the wire, an additional event time t^0_{event} is required. Provided an event time estimate t^0_{event} [trg clock], the drift time proportional to the distance corresponds to the difference $t_{\text{trg}} - t^0_{\text{event}}$.

As an example of a physics event as seen by the L1 CDC trigger system, a single simulated $\Upsilon(4S)$ event with its CDC trigger response is shown in Fig. 3.14. This illustrates the noise suppression capabilities of the TSF, since single background hits (orange) are less likely to produce TS than signal hits (red) from the $\Upsilon(4S)$ event. Furthermore, Fig. 3.14 illustrates a limit of the TSF algorithm; the dependence of the TSF algorithm on the crossing angle of the track can be seen for the single curl-back track with a low transverse momentum p_{T} track, where only the innermost three TS are found. At large crossing angles, the wire hits related to a track do not fill a single TS pattern.

3.3.4. Track Finding and Reconstruction

The development of novel track finding and track reconstruction algorithms for the L1 CDC trigger are the main topics of this thesis. They are briefly summarized here, a detailed description of the underlying algorithms can be found in the subsequent chapters. In the introduction to the improved 3D finder algorithm including the stereo TSs in the track finding (Chapter 5, the Hough transformation method pursued by the 2D finder is described in Sec. 5.1.2.

In the CDC trigger pipeline, the TSF is followed by the 2D track finder receiving axial TS information from the axial TSF boards only. The wires in the axial TS hits appear as single points in the transverse xy-plane. Due to the constant magnetic field oriented in z-direction, charged tracks can be described by circles in the transverse plane. These track circles are assumed to originate from the IP such that they can be described by two parameters: the signed curvature ω, where the sign denotes the charge of the track, and the azimuthal angle ϕ. After transforming the axial hits into a binned $\omega\phi$-space, tracks are found as clusters (see Sec. 5.1.2). Track estimates are obtained by calculating the cluster centers by averaging the bins of the diagonal opposing cluster corners. The size of the Hough plane is listed in Tab. 3.8. The extent

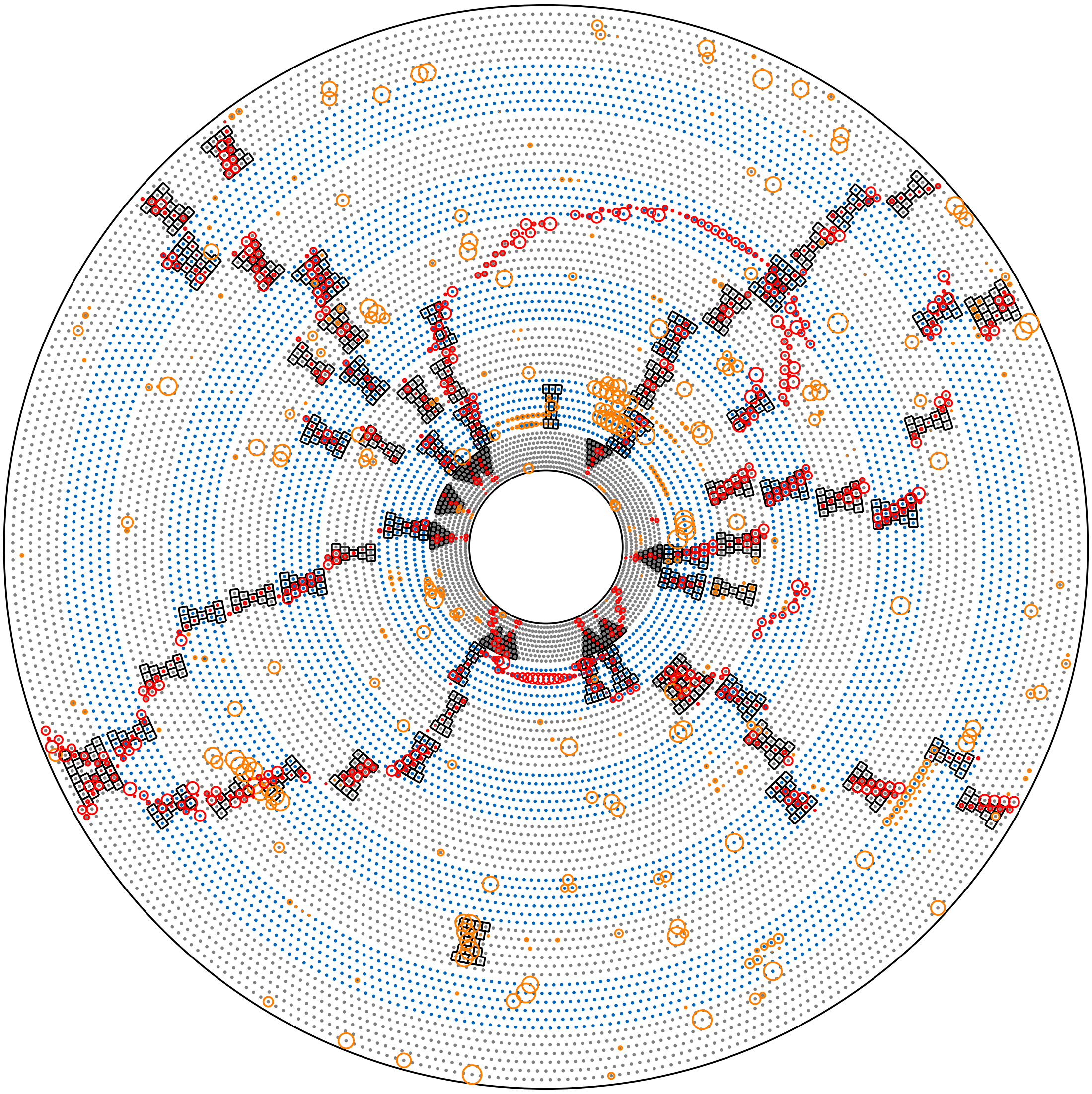


Figure 3.14.: Example event display of an $\Upsilon(4S)$ event as seen after the TSF of the L1 CDC trigger [66]. CDC wires in axial layers are shown in gray, wires in stereo layers in blue. The CDC hits from the $\Upsilon(4S)$ event are shown as red circles, background hits as orange circles. The found TSF are shown in black.

in ω is defined by a minimum p_{T} value, which is set to $p_{\mathrm{T}}^{\mathrm{min}} = 0.3\,\mathrm{GeV}$, while in ϕ the full 360° are covered. In this configuration, the maximum value of p_{T} estimated by the 2D finder is $p_{\mathrm{T}}^{\mathrm{max}} = 10.2\,\mathrm{GeV}$.

In addition to the track parameters, the 2D finder provides information on the related axial TSs for each found track. With the five axial SLs in the CDC, a fully described 2D track with one TS per SL has five related axial TSs. In order to improve the track finding efficiency, the 2D finder requires at least four axial TSs per track. The information provided by the 2D finder for each found track is listed in Tab. 3.8. Further implementation details on the 2D finder can be found in [67].

The 2D finder is realized in four separate boards, each covering one quadrant of the CDC. One of these boards can find up to six 2D tracks simultaneously in each 32 ns time interval of the trigger pipeline. This split into quadrants is adopted by the 3D track reconstruction algorithms. Two alternative methods are already implemented in hardware for the 3D track reconstruction; a 3D track fitter [68] using a linear Least Squares (LS) fit and the Neural Network (NN) trigger. Both these methods are presented in Chapter. 6. Especially the combination of the stereo TSs with the found 2D tracks is important for the 3D track reconstruction of the polar track angle θ and the longitudinal vertex position z. By using an event time estimate t_{event}^{0} and the `LR` information in the TSs, the drift times can be used as a precise, directed distance measure between the track and the wire. Since the foreseen Event Time Finder is not yet ready in hardware, a track timing method was developed in the preprocessing of the neural network trigger (see Sec. 6.3.2).

3.4. Belle II Simulation with basf2

The Belle II detector with all its subdetectors is simulated in basf2 [69], the Belle II analysis software framework. In a modularized structure [70], it combines all the steps involved in Monte Carlo (MC) event simulation as well as the analysis of real data recorded from Belle II into as single framework. Especially, the simulation modules of the CDC trigger and the precise offline reconstruction of charged particle tracks in the detector are important for the experiments presented in this thesis. The newly developed 3D finder (Chap. 5) and the neural network trigger (Chap. 6) are both integrated into the basf2 trigger simulation and can be used to carry out performance measurements and optimization studies of the trigger algorithms. The concrete configuration options used in the experiments in this thesis are explained later in the respective chapters.

In the MC based generation of physics processes, a particle physics generator simulates the interaction after the colliding $\mathrm{e}^{+}\mathrm{e}^{-}$ particles according to a physics model. Typically, Feynman diagrams can be used to represent the lowest order interactions of the generated physics processes. The cross sections and probability density functions are obtained from a physics model implemented by the generator. After the generator, each event is given by a set of final state particles, with their production vertices and four-momenta. A following simulation of the material interaction of these particles, when traveling through Belle II, is carried out with Geant4 [71]. To this end, the complete Belle II geometry is implemented in basf2. Geant4 simulates the interaction with the detector material resulting in a simulated cascade of produced secondary particles. This enables a following simulation of the detector response in the digitizer modules of each subdetector, called digitization, where the realistic detector hits are formed.

Besides the physics signal simulation, also single track simulations and background simulations are carried out in order to obtain the primary four-vectors of the particles propagating through the detector. In the single track simulations, no physics interactions are simulated at the generator level. Instead, a chosen particle type is simulated within defined track parameter ranges and distributions. This is especially important in order to obtain the uniform track parameter distributions used for the training of the neural network and the 3D finder (see Sec. 5.3.1 for the 3D finder and Sec. 6.4.2 for the neural network). For the simulation of machine background events the simulation of the SuperKEKB collider, the SAD [72] software, is used as generator. The output of SAD are lost particles from the single beams, which are used as input for the following detector simulation. Details on the machine background simulation with SAD is described in the Appendix B.1.1.

Starting from the hit level, i.e. after the digitization, the MC simulation and recorded Belle II data can ideally be used in the same way. This means, recorded events from Belle II can be unpacked into basf2 in order to obtain the same hit classes as obtained from the simulation. This is advantageous for physics signal analysis, since the measurements can be fully developed using MC. To this end, the hits from all subdetectors are used as input to the precise offline reconstruction, where tracks are fitted with different mass hypothesis. In order to reconstruct the underlying processes based on the track reconstruction results, the physics analysis code can be developed for explicit physics signal signatures. Once the analysis code is ready and Belle II has collected a sufficient amount of data, the input to the analysis can be switched from MC to real data in order to obtain the actual results in a so called "box opening".

Chapter 4.

Background at Belle II

All types of signals in the detector not related to an interesting physics event created by an intended e^+e^- collision at the IP can be considered as background. This comprises any radiation sources increasing the occupancies in the subdetectors, either by single background hits or by full reconstructable background tracks. It is indispensable for Belle II to carry out detailed simulation studies of those background types with an expectedly substantial background event rate. In order to guarantee a high efficiency for physics signal events, background tracks have to be recognized and suppressed ideally at the Belle II first level trigger. This is the goal of the neural network trigger project, where a significantly improved background suppression is enabled by reconstructing the 3D track parameters of single tracks using neural networks. Tracks with vertices displaced from the IP can clearly be identified as background and rejected already at the first trigger level.

In this section, the nature of important background types is presented and studies of their expected rates and topologies are carried out using simulated background samples. In general, two main classes of relevant background can be distinguished: luminosity background and machine background. Luminosity background refers to QED events with large cross sections compared to the interesting flavor physics channels at Belle II. The main luminosity background types (Bhabha scattering and two photon interactions) and their simulation are discussed in Sec. 4.1. Additionally, the machine itself produces large amounts of background in the detector via radiating particles from the single beams. The most important machine background types in Belle II are inner beam scattering processes (Touschek effect [73]), elastic scattering at residual gas molecules in the beamline (Coulomb) and synchrotron radiation. The machine background types and their simulation are discussed in Sec. 4.2. Background sources like cosmic radiation or radioactive decays only contribute with relatively small rates and can therefore be neglected in the presence of particle beams in SuperKEKB. Further background types and combinatorial background pileup effects are discussed in Sec. 4.2.4. As a groundwork for the trigger studies in the subsequent chapters (Chapter 5, Chapter 6, Chapter 7), this section closes in Sec. 4.4 with a measurement of the hit and trigger track rates in the official Belle II background campaigns. A connection to the high statistics background simulation published in [74] can be found in the Appendix B.1.

4.1. Luminosity Background

Physics processes initiated by the intended e^+e^- collisions at the IP scale with the luminosity. Therefore, luminosity background refers to inevitable physics QED processes with a huge cross section which are not subject of the research goals at Belle II as the underlying physics model is already well established. The two main luminosity background candidates considered here are (radiative) Bhabha events with two electrons in the final state ($e^+e^- \to (\gamma)e^+e^-$) and the two photon QED background with four electrons in the final state ($e^+e^- \to e^+e^-e^+e^-$)). Considering only the background tracks from the IP, the early background and signal rate expectations at the L1 trigger are listed in Tab. 3.5 (from the Belle II TDR [21]).

Via backscattering of the primary background particles at the detector material, large rates of secondary luminosity background tracks with vertices displaced from the IP are created in addition. Here resides the advantage of the z-vertex trigger with neural networks. Based on the estimated displacements of the z-vertices from the IP ($z = 0\,\text{cm}$), the neural network trigger introduces new measures for the suppression of background tracks. The primary luminosity background tracks have to be suppressed by other trigger measures. Based on electron identification by matching of ECL clusters with CDC tracks, so called Bhabha veto triggers are installed. Due to the anticipated high luminosity in Belle II, those background types scaling with the luminosity need to be suppressed efficiently already at the L1 trigger level.

4.1.1. Radiative Bhabha Scattering

Bhabha events are scattering events where the initial and final state particles are electrons and positrons $e^+e^- \to e^+e^-(\gamma)$. In the inelastic case, an additional photon is created as initial state or final state radiation.

Typical Bhabha events with large scattering angles have sufficiently large momenta within the acceptance region of the CDC for the tracks to become visible in the detector. Fortunately their unique signature can be used to reject these large angle Bhabha events efficiently with Bhabha veto trigger. The events with back-to-back electron tracks can clearly be identified as Bhabha events.

Bhabha events with small scattering angles of the e^+e^-, or hard photons created in radiative Bhabha scattering events, might still create additional background without the unique Bhabha signature: instead backscattered tracks become visible from positions with large displacements from the IP ($z > 50\,\text{cm}$).

In first order QED, the scattering process is described by the exchange of a virtual photon. Fig. 4.1 shows the ISR and FSR Feynman diagrams for the radiative Bhabha process: there a photon is emitted in the initial state or in the final state, while the two electrons participate in an elastic interaction.

Considering only the elastic scattering ($e^+e^- \to e^+e^-$), the t-channel diagram for the exchange of a virtual photon is only possible because the initial state as well as the final state consist of an e^+e^- pair. Additional e^+e^- annihilation diagrams

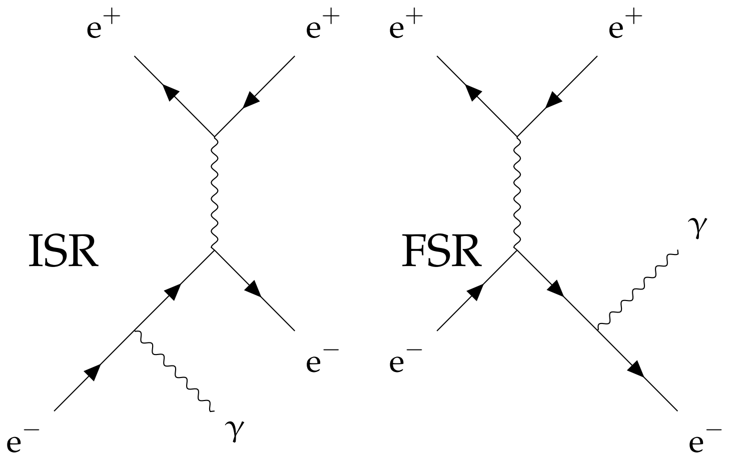


Figure 4.1.: Initial state radiation (ISR) and final state radiation (FSR) t-channel Feynman diagrams of the radiative Bhabha interaction.

are allowed in the s-channel, where the e^+e^- annihilate into a virtual photon. In the t-channel diagrams the exchange of very soft photons is possible resulting in only minimal momentum transfer and thus minimal deflections of the e^+e^- in the interaction. Therefore, the cross section of the (radiative) Bhabha process strongly depends on the scattering angles θ_{e^+} and θ_{e^-} of the outgoing e^+ and e^-. The Bhabha cross section is proportional to the inverse square of the scattering angle θ

$$\sigma \propto \frac{1}{\theta^2}. \tag{4.1}$$

Hence, the cross section actually diverges in the limit of $\theta_{e^+} = \theta_{e^-} = 0$, which corresponds to limit of minimal momentum transfer by the virtual t-channel photon.

In radiative Bhabha scattering, additionally photons are emitted in the e^+e^- scattering process from the initial or from the final state particles. These photons are emitted along the beam axis and interact with the magnet material where neutrons and protons are produced via the giant photo nuclear resonance effect. Therefore, inspecting radiative Bhabha background means inspecting the neutron flux as well. Since the neutron production rate R_N is proportional to the luminosity [50] as a countermeasure, neutron shields are installed. The protons may directly become visible as background tracks in the CDC.

The Bhabha generator BBBrem [75] is used to simulate single Bremsstrahlung in the process $e^+e^- \to e^+e^-\gamma$. It is well suited to study Bhabha events at very small to vanishing scattering angles. With the limits shown in Fig. 4.5 medium and large scattering angle θ Bhabha events are simulated with BHWide [76]. The distinction between medium and large θ is done to account for the different cross sections while assuring a sufficient statistics for each case.

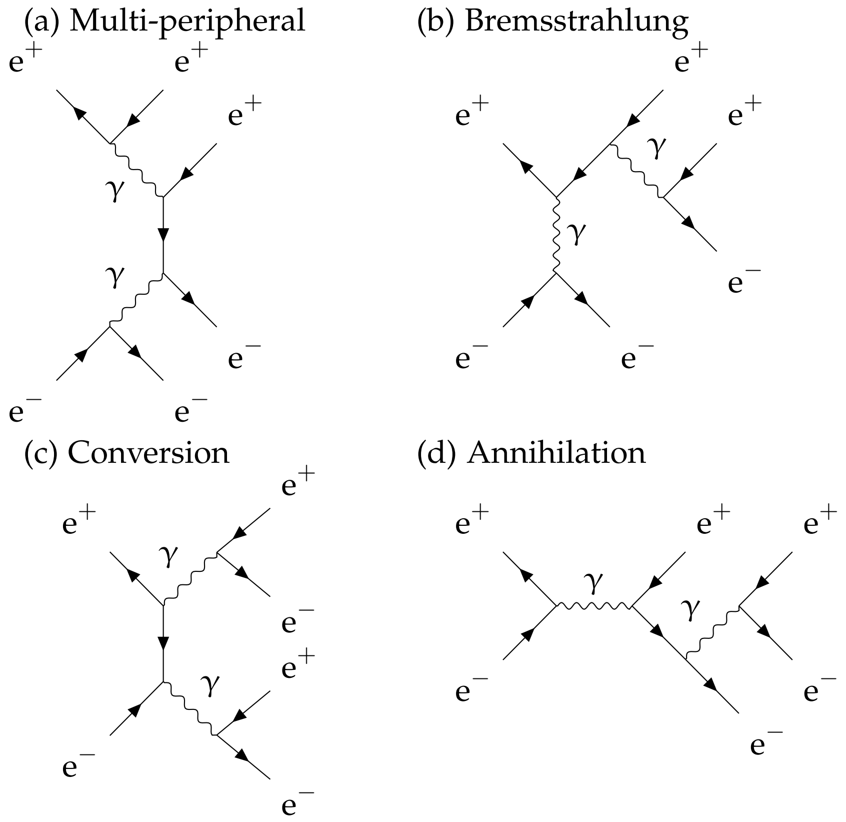


Figure 4.2.: Feynman diagram examples of the four different cases of the two photon interaction in the process $e^+e^- \rightarrow e^+e^-e^+e^-$.

Since the cross section for Bhabha scattering is inverse proportional to the scattering angle θ, it is largest for the lowest scattering angles. In order to simulate similar amounts of background events with the different Bhabha scattering angles, usually smaller real time equivalents are simulated for the small scattering angles than for the larger scattering angles.

4.1.2. Two Photon QED Events

In the two photon QED process, the initial e^+e^- pair emits two photons, which in turn create an additional e^+e^- pair in the final state. Therefore, the total two photon process from the initial state to the final state is $e^+e^- \rightarrow e^+e^-e^+e^-$. In general, there are many combinations in which the two photons can interact in order to create the additional e^+e^- pair. For the four different categories, one Feynman diagram each is shown in Fig. 4.2.

Although the two photon process has a large cross section for the creation of e^+e^- in the acceptance region of the detector, these produced e^+e^- typically have extremely low transverse momenta such that the tracks are not visible in the detector [77]. Especially the multi-peripheral interaction shown in Fig. 4.2 (a) has the largest contribution to low energetic e^+e^- pairs produced from the $\gamma\gamma$, as no annihilation is involved and the interaction is not suppressed by photon propagators [60]. There is no annihilation involved, since the initial e^+e^- pair are still found in the final state. Instead, the two radiated photons interact to create the additional e^+e^- pair.

The two photon background at Belle II is simulated using the Aafh [78] generator. It is supposed to calculate the complete matrix element of the $e^+e^- \to e^+e^-\ell^+\ell^-$ final state with the general lepton pair $\ell^+\ell^-$ created in the two photon interaction. Due to the relatively small mass of the electrons compared to muons or taus, in the case of small energy transfers the produced lepton pair is almost exclusively an e^+e^- (the cross section is proportional to the inverse squared mass m_ℓ^{-2} of the produced lepton [60]).

4.2. Machine Background

Machine background is caused by the loss of beam particles from the single beams running in the magnetic guiding field of the SuperKEKB accelerator and by photons from synchrotron radiation. The machine background strongly depends on the structure and design parameters of SuperKEKB. Especially the mask settings are relevant: these mask settings describe the collimators at several positions around the ring, which are installed to reduce the rate of machine background tracks in the vicinity of the IP. The main machine background contribution is due to inner beam scattering (called Touschek effect [73]) and beam gas scattering, where the beam particles hit residual gas molecules in the beamline. In the beam gas collisions, usually the elastic beam gas interactions (Coulomb scattering) and inelastic beam gas scattering (Bremsstrahlung scattering) are distinguished in the simulation. Further information on the SuperKEKB machine background can be found in [50, 47, 60].

The machine background types Touschek and the elastic and inelastic beam gas backgrounds (Coulomb and Bremsstrahlung) are generated with the simulation software SAD [72], a simulation of the SuperKEKB magnetic field developed by KEK. To this end, the physics models of the machine background interactions are used to estimate their probability distributions as input for the SAD generator. Then SAD is used to track bunch particles of $e^+(e^-)$, after the background interaction, for several turns in the magnetic guiding field in the SuperKEKB HER and LER until they eventually get lost. Details on the background simulation using SAD are given in the Appendix B.1.1.

The collimators are included in the SAD simulation and can thus be optimized to reduce the particle loss in the vicinity of the Belle II detector. In the Belle II experiment, a reduction of the machine background by a factor of ≈ 1000 is anticipated by the use of the collimators [51]. Although at this point the limits of the simulation are met and a careful experimental fine-tuning of the collimators is required in the real experiment.

The machine backgrounds affect the lifetime τ of the beam. Thus, the particle number N in the beam at time t can be described by the exponential

$$N(t) = N_0 \mathrm{e}^{-\frac{t}{\tau}} \tag{4.2}$$

where N_0 is the number of initial particles in the beam and τ is determined by the machine background interaction. For each machine background the loss rate R can be expressed as the derivative of the particle number N at time t [47]:

$$R = \frac{\mathrm{d}N}{\mathrm{d}t} = -\frac{N_0}{\tau}\mathrm{e}^{-\frac{t}{\tau}} = -\frac{1}{\tau}N(t). \tag{4.3}$$

4.2.1. Touschek Effect

The Touschek effect [79] is the elastic coulomb scattering of two particles within a bunch of the particle beam; $\mathrm{e}^+\mathrm{e}^+ \to \mathrm{e}^+\mathrm{e}^+$ in the LER or $\mathrm{e}^-\mathrm{e}^- \to \mathrm{e}^-\mathrm{e}^-$ in the HER. Such a coulomb scattering process between two bunch particles results in a deviation of these two particles from the nominal bunch trajectory. With small momentum transfers, the scattering positions do not coincide with the loss position. Instead, the deviation causes betatron and synchrotron oscillations of the particles within the beam. Only some turns in the SuperKEKB ring later the scattered particles will eventually get lost from the beam by hitting the beampipe. Initially, the Touschek effect addressed the lifetime of a particle beam. However, with the high luminosity machine SuperKEKB the Touschek loss rate in the vicinity of the Belle II detector becomes a serious source of background events.

The general derivation of the beam lifetime due to Touschek scattering particle loss is derived from the relativistic kinematics of the two particle interaction in [73]. To overcome computational problems with the general formula [60], in the simulation a non-relativistic flat beam approximation is used [47]. This approximation of the local Touschek loss rate r is known as Brück's formula. The total loss rate R can be calculated by integrating the local Touschek loss rate r around the SuperKEKB ring

$$R = \frac{1}{L}\int r\mathrm{d}s \tag{4.4}$$

where L is the Length of the ring and s is the parameter along the beamline. In the simulation, the tracking using SAD is combined with a stepwise evaluation of the approximated scattering formula [60].

Qualitatively, the Touschek scattering rate R is proportional to [60, 50]:

$$R \propto \sigma_{xy}^{-1} \cdot E^{-3} \cdot N_b \cdot I_b^2 \tag{4.5}$$

where E is the respective beam energy (in the HER or in the LER), σ_{xy} is the bunch size, N_b is the number of bunches and I_b the bunch current. Due to the dependence on E^{-3}, the Touschek scattering rate is larger in the LER. With increasing bunch densities of the beam particles the probability for a collision increases. Thus, the Touschek scattering rate depends on the number of particles per bunch (current I) and on the bunch size (σ_{xy}^{-1}). Especially, the small bunch sizes in the final focussing for the collisions achieved by the nano beam scheme [43, 44] result in a large Touschek scattering rate.

4.2.2. Beam Gas Scattering

Beam gas scattering describes the interactions of beam particles with residual gas molecules in the beamline due to an imperfect vacuum. An ideal vacuum, with the complete absence of any particles, cannot be realized in reality within the SuperKEKB beamline. However, very low pressure levels can be achieved which still leave the chance of beam gas scattering events. With constantly running vacuum pumps, the vacuum is improving over a longer run time period. Therefore, after the beamline has been closed, later physics runs face better vacuum conditions. Vacuum pumps are constantly working to improve the vacuum in the long term. Therefore, after the beamline has been closed, later physics runs face better vacuum conditions.

The scattering of beam particles on the residual gas can be distinguished into two main background categories: processes with small and large momentum transfer q^2. Small deflections of the beam particles from the nominal design orbit might result in loss of those beam particles after several turns in the SuperKEKB collider ring, if the deflected particles eventually hit the beam wall. These low q^2 interactions are interactions with the electric field of the gas nuclei, with a large contribution from elastic coulomb scattering without energy change. A small correction due to inelastic Bremsstrahlung can be considered where an additional photon carries away a fraction of the energy. With a much smaller rate, the beam particles may directly hit the nuclei, which results in nuclear spallation. In this case, the hadronic background particles have large transverse momenta and thus can directly be observed at the position of the background interaction.

Coulomb Scattering

Beam gas coulomb scattering is the elastic scattering process of a beam particle with a residual gas molecule in the beamline (illustrated in Fig. 4.3 (a)). The elastic beam gas scattering rate can be derived from the cross section σ of a lepton scattering on a nucleus. While the Mott cross section correctly describes the scattering of a spin $\frac{1}{2}$ particle on a nucleus [60], in the limit of small scattering angles, the Rutherford cross section is sufficient (used in [47]). This cross section is given by:

$$\frac{\mathrm{d}\sigma}{\mathrm{d}\Omega} = \left(\frac{Zr_e}{2\gamma}\right)^2 \frac{1}{\sin^4\left(\frac{1}{2}\theta\right)} \tag{4.6}$$

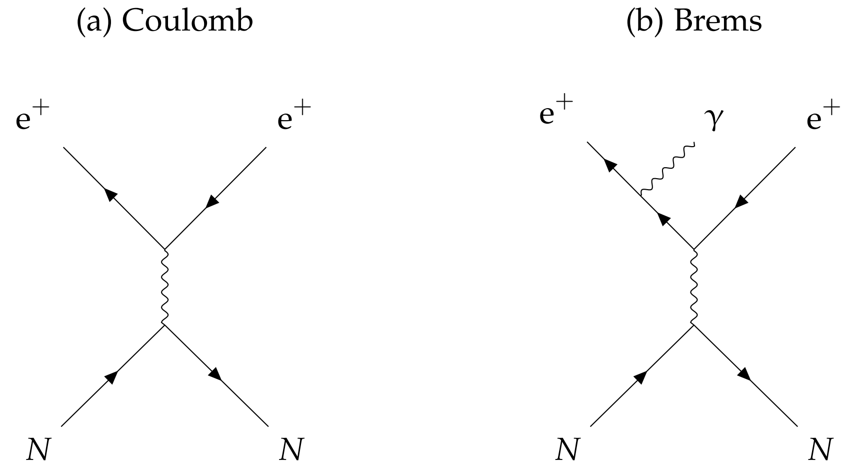


Figure 4.3.: (a) Coulomb: elastic scattering of a beam particle (here e$^+$) on a residual gas nucleus N. Only the beam particles momentum direction is changed, but not its energy. (b) Brems: inelastic scattering of a beam particle on a residual gas nucleus N. The photon γ carries away a fraction of the energy of the beam particle.

where Z is the charge of the gas nucleus, r_e is the electron radius, γ is the Lorentz factor, and θ is the scattering angle. An integration of this differential cross section in the small angle approximation for small values of θ, gives a cross section dependent on the inverse square of the scattering angle. Therefore, the cross section can be approximated by an expression depending on the minimal scattering angle θ_{min}:

$$\sigma = 4\pi \left(\frac{Zr_e}{\gamma}\right)^2 \frac{1}{\theta_{\mathrm{min}}^2} \tag{4.7}$$

The lifetime associated with elastic beam gas scattering can now be expressed as

$$\frac{1}{\tau} = cn_g\sigma \tag{4.8}$$

where c is the speed of light and n_g is the density of the residual gas [47]. With the gas pressure p, temperature T and the Boltzmann constant k_{B} the density n_g is expressed as

$$n_g = \frac{2p}{k_{\mathrm{B}}T} \tag{4.9}$$

where the factor 2 accounts for two-atomic nuclei. Therefore, the loss rate due to beam gas scattering is proportional to

$$R \propto Z^2 \cdot p \cdot I \tag{4.10}$$

where Z is the charge of the gas nuclei, p denotes the gas pressure, and I is the beam current.

The Coulomb background is simulated with SAD, by tracking deviated beam particles in the SuperKEKB ring. Depending on the mask settings, the Coulomb background can contribute with similar occupancies in the Belle II detector as the Touschek background. Methods for the mask optimizations to cope with the Coulomb background are presented in [51, 47].

Bremsstrahlung

Bremsstrahlung is an inelastic scattering process of a beam particle with a residual gas molecule (illustrated in Fig. 4.3 (b)). In the electric field of the molecule the particle scatters and a fraction of the momentum is carried away by the photon. The particle loss rate due to Bremsstrahlung is expected to be small compared to the machine background types Touschek and Coulomb [47, 60]. Therefore, this background is usually neglected in the official Belle II background campaigns. Although Bremsstrahlung background can in principle be simulated with SAD as well.

4.2.3. Synchrotron Radiation

Synchrotron radiation is emitted from the charged beam particles due to the bending acceleration in the magnetic field. The synchrotron radiation power is proportional to the beam energy squared and to the magnetic field squared [50]. Therefore, the HER is the main source for synchrotron radiation.

The synchrotron radiation corresponds to single photons with keV energies, which do not create tracks in the detector. The photons can only create single hits, with the effect of an increased occupancy in the detectors. Since the intensity decreases with the inverse square of the distance from the beamline especially the innermost detectors are affected by this background. In order to protect the inner detectors (PXD / SVD), the inner wall of the beryllium beam pipe is coated with gold. Additionally, direct synchrotron radiation hits are suppressed by the shape of the IP chamber [50, 47]. Therefore, synchrotron background is not included in the official Belle II background simulation in the background campaigns. A discussion of the synchrotron radiation at Belle II can be found in [60].

4.2.4. Further Machine Backgrounds

In beam-beam interactions, the field of one beam interacts with the other beam, with the effect of a beam blowup. The result are non-Gaussian shapes of the beam, which can increase the other background types [50].

For SuperKEKB continuous injection of new particles is foreseen to maintain the anticipated high luminosity [47]. But newly injected particles always have small deviations from the nominal trajectory. Therefore, after the injection of new particles into the beam, the new particles have to stabilize on the beam trajectory (e.g. by synchrotron radiation). The deviations of the injected particles also cause background directly, which is especially seen shortly after the new injections.

4.3. Combinatorial Background Pileup

Pileup is a combinatorial background where hits and tracks from one event overlap with the next event. With large background rates, even pileup of single track events might result in apparent multi track vents. Therefore, it is not a background type itself, but an effect that modifies the appearance of background as well as signal events.

At Belle II, the trigger will most likely trigger only events with at least two tracks, while the single track trigger approach is expected to be too loose to meet the requirement of a 30 kHz total trigger rate. In addition to background events with several tracks, apparent multi track background events could be caused by pileup of single track background events as well. In this section, the rate of multi track events caused by pileup of single track events will be calculated.

The pileup depends on the total rate of tracks R and on the maximum pileup time interval T for two tracks to be recognized as one event. Here, the pileup rate is modeled via a Poisson distribution. The probability mass function of the Poisson distribution is

$$P(n=k) = \frac{\lambda^k e^{-\lambda}}{k!} \tag{4.11}$$

It tells the probability to have $n = k$ discrete occurrences in an interval, where the mean number of occurrences in this interval is λ. So for a rate R [Hz] and a time interval T [s] the mean number of events is $\lambda = R \cdot T$. Since it is only interesting whether there is a pileup or not k is set to $k = 1$ and the pileup probability is given by $P(n > 1) = 1 - P(n = 1) - P(n = 0)$.

$$P(n \geq k) = 1 - e^{-\lambda} \sum_{i=0}^{k} \frac{\lambda^i}{i!} \tag{4.12}$$

$$P(n \geq 2) = 1 - \exp\left(-R \cdot T\right) \left(\frac{R^0 T^0}{0!} + \frac{R^1 \cdot T^1}{1!} \right) \tag{4.13}$$

$$P(n \geq 2) = 1 - \exp\left(-R \cdot T\right) \left(1 + R \cdot T\right) \tag{4.14}$$

The measured background rates in the simulation experiments and in the early Belle II data taking phase provide an estimate for R in the order $\mathcal{O}(10^2\,\text{kHz})$. The time interval T in the order $\mathcal{O}(10^2\,\text{ns})$ can be estimated from the requirements of the trigger system and the properties of the CDC: in [21] the trigger requirement for a "minimum two event separation" is defined as 200 ns; in the CDC, the maximum drift time is ≈ 500 ns.

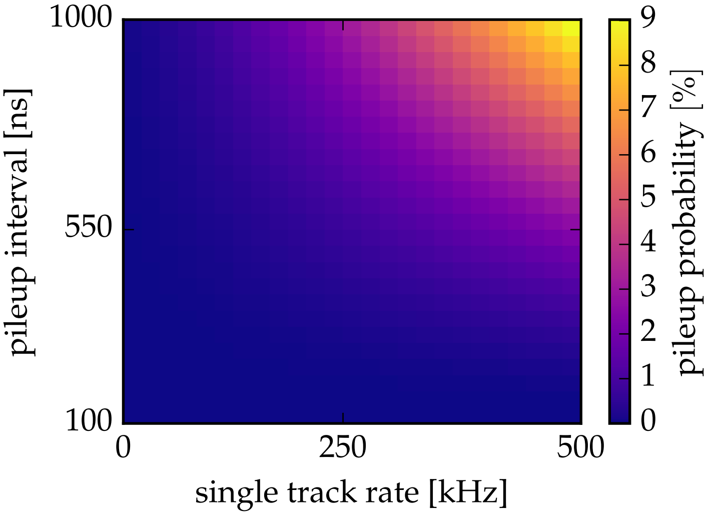


Figure 4.4.: Probability of pileup events for an interval in the single track rate and the overlap time interval.

In order to study flexible value ranges for R and T, here an interval is chosen for both parameters and the resulting pileup rates are shown. The plot in Fig. 4.4 shows the probability of a pileup event (indicated by colors) for a range in the track rate R and the desired overlap interval T.

In a scenario with a background expectation of $R = 200\,\mathrm{kHz}$ single tracks and a pileup interval of $T = 500\,\mathrm{ns}$ the pileup probability is almost 1 %. This corresponds to an additional 1 kHz rate of two track background events. In a almost worst case scenario, with the same time interval $T = 500\,\mathrm{ns}$ but a larger single track background rate of $R = 500\,\mathrm{kHz}$ the pileup probability reaches $\approx 2.25\,\%$. This creates an additional 13 kHz two track background rate. Hence, depending on the actual observed background level in the late experiment at full luminosity, the pileup may become a serious problem for two track triggers. Although in the early experiment these effects will not be visible.

4.4. Background Simulation

The analysis of the expected background involves a generation of the primary background events and a simulation of the detector response to obtain the particles actually seen in the detector. Background simulation takes place in a similar way to the simulation of a physics signal event, starting with the four-vectors of the particles created in the background processes. In the Belle II collaboration, so called background campaigns are regularly carried out to provide simulated background samples using the latest updates of the collider and detector settings. These background campaigns are used by the sub-detector groups to study the background hit occupancies in the detector and readout electronics under the present background expectations. Furthermore, the campaigns are used for mixing background into the large scale MC physics signal event production, which are used for the preparation of the upcoming physics analysis with the real data.

As a prerequisite to the studies with the novel trigger algorithms in the subsequent chapters, the background types used therein are studied here. The background types and their topology as seen by the CDC trigger are analyzed and the hit and track occupancies are measured and compared to the requirements of the L1 trigger Chapter 3. To this end, the full CDC trigger is simulated in background only events, where the method of background mixing is used as explained in Sec. 4.4.1. The differences in the background properties for several recent background campaigns are addressed in Sec. 4.4.2. Limitations due to the low statistics in the official campaigns discussed and a solution to increase the sample size is presented in Appendix B.1.2.

4.4.1. Background Mixing

At Belle II two main methods are implemented for the combination of the separate background samples: background mixing and background overlay. The background mixing is carried out directly after the Geant4 simulation, however, before the digitizers where the detector efficiencies are included to simulate the readout of realistic hits. In contrast, background files for overlay already contain the digitized hits. In order to distinguish the realistic hits obtained from the detector readout simulation with the digitizers, the hits created by the Geant4, prior to digitization, are called SimHits here.

In theory, with a perfect background MC, mixing is the preferred method, as it simulates the detector response to the simultaneous occurrence of physics signal hits and background hits in the detector. However, the MC background simulation is not perfect, the plan for the future of Belle II is to use real measured background, once a sufficient amount of background is collected. Obviously with real data SimHits do not exist and the combination of recorded background hits onto simulated MC physics signal events is only possible after the digitization. This combination of background at the hit level is called background overlay. In the studies throughout this thesis, the method of background mixing is used.

The background mixer uses the real time equivalents of each background type in order to randomly select simulated hits, prior to the readout, from each background sample. In the simulation, the method of background mixing is also used to create the background overlay files via mixing background into empty events without signal tracks. Hence, effects of multiple simulated background hits leading to single readout hits is included in background mixing. Since the background is already mixed, using the overlay files no more randomizers are involved. Instead, the hits of the background overlay files are simply added on top of the simulated signal hits. Hence, larger real time equivalents are required for the background overlay files.

time window	min	max
Background mixer	$-1000\,\text{ns}$	$800\,\text{ns}$
CDC drift time (inner SL)	$-100\,\text{ns}$	$300\,\text{ns}$
CDC drift time (outer SLs)	$-100\,\text{ns}$	$500\,\text{ns}$
CDC TDC count	4384 [tdc clocks]	5024 [tdc clocks]
CDC trigger	-62 [trg clock]	258 [trg clock]

Table 4.1.: Relevant time windows of the CDC digitizer, the background mixer, and the trigger in the simulation. The time windows for the background mixer and for the CDC drift times are used as input to the simulation, while the time windows in TDC clocks are the result of the simulation after readout effects are included by the CDC digitizer. Using the default zero timing of the wires ($t^0_{\text{wire}} \cdot f_{\text{tdc}}$, see Sec. 3.2.5) of the CDC wires in the simulation of $t^0_{\text{wire}} \cdot f_{\text{tdc}} = 4900$ [tdc clock], the CDC trigger time window can be calculated from the CDC TDC count time window.

Several relevant time windows are involved in simulations with the background mixer, which are shown in Tab. 4.1. By selecting a proper readout time window for each event, the background hit- and track-rates can be calculated. In the background mixer the SimHits are added to the events within the mixing time window prior to the simulation of the detector readout in the digitizer. The default value for background mixing used here, is $t \in [-1000, 800]\,\text{ns}$. The following digitizer for the CDC wire hits has a smaller readout time window such that the background hits are uniformly distributed in time and not correlated to the event time in particular. The digitizer combines the CDC SimHit information to form the realistic CDC hits. After the calculation of the drift times of each possible CDC hit, a time window cut is applied. Only CDC hits with drift times in the time windows $t_{\text{inner}} \in [-100, 300]\,\text{ns}$ for the innermost SL (small cell chamber) and $t_{\text{outer}} \in [-100, 500]\,\text{ns}$ for SLs one to eight are accepted by the digitizer. In order to simulate realistic timings, a smearing is applied and offsets are added to mimic the readout delays of the hardware. Finally, the drift times are not stored directly in units of [ns], but as a TDC clock counter value in units of [tdc clock], where the clock frequency of the TDC clock counter is $f_{\text{tdc}} = 1.017\,774\,\text{GHz}$ (see Sec. 3.2.5). The total width of the TDC time window in TDC clocks is $T = 640$ [tdc clock] corresponding to a time window of $T \approx 628.823\,\text{ns}$.

At the trigger level the timings are represented in trigger clocks, where one trigger clock corresponds to two TDC clocks (about 2 ns). In the TSF simulation, the trigger clocks are related to the time in nano seconds by Eq. 3.22. Accordingly, with a given drift time in units of [trg clock], the corresponding drift time in [ns] can be calculated:

$$t[\text{ns}] = \frac{t[\text{trg clock}] \cdot 2}{1.017774[\text{GHz}]}. \tag{4.15}$$

The effective simulated time window of the TS drift times at the trigger level is

$t \in [-62, 258]$ [trg clock]. Using the scale in Eq. 4.15, the time window of the trigger in nano seconds is about $t \in [-121.83, 506.99]$ ns.

4.4.2. Belle II Background Campaigns

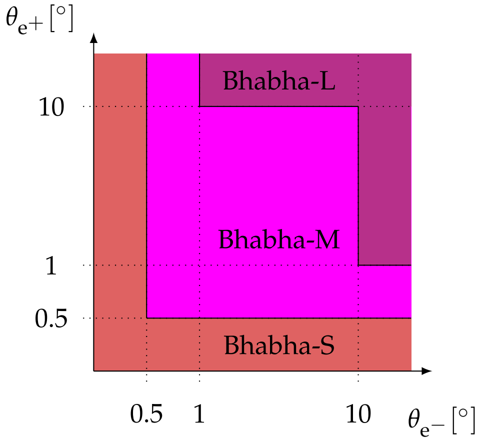
A Belle II background group of collaborators from different subdetectors, regularly carries out MC background campaigns. Using the latest collider and detector settings, background files are provided for further studies via background mixing or via background overlay. This comprises the generation of physics background processes and the simulation of the interaction with the detector material in order to provide the SimHits for background mixing. Campaign overlay files are produced by carrying out the mixing, followed by the readout simulation by the digitizers.

The subdetector groups of Belle II use the background campaign files to evaluate the hit occupancies and radiation doses in their subsystems. By comparing the measured background from the campaigns to the requirements of the components, the necessity of additional background countermeasures can be detected. Besides occupancy studies in the subdetectors, the background campaigns are used in the MC signal production as well, where huge data sets of physics processes are simulated which are the subject of research at Belle II. To mimic realistic conditions at Belle II, the files from the background campaigns merged into the physics signal events via mixing or via overlay.

In new background campaigns the detailed settings of the simulation method are optimized and the latest machine parameters are included. Although the same physics processes are expected to result in observable background, the background rates and topologies as well as the simulated sample sizes change in between campaigns. Furthermore, in earlier campaigns only the phase 3 luminosity background was simulated, while for the Phase 2 background the same files were used with a scale factor of 40^{-1} to account for the lower luminosity. The latest campaigns can be seen as the state of the art background expectations.

Typically the observable background is caused by secondary physics interactions and thus strongly depends on the simulated machine and detector settings. Therefore, the simulated background is always expected to differ from the real observed background. Furthermore, a comparison of the occupancies and topologies in different background campaigns forms a useful basis for the performance studies of the CDC trigger.

An important change in the machine background simulation are e.g. the mask (collimator) settings (see Sec. 3.1.2). The masks are installed at positions of a large beta function of the beam. The objective in the mask optimization is a minimizing the beam loss close to the IP while at the same time minimizing the total particle loss around the ring (i.e. strongly reduce the background in the detector while only minimally reducing the beam lifetime). Using the collimators, reduction by a factor of ≈ 1000 is anticipated for the particle loss rate in the vicinity of the IP. Therefore, different collimator settings may drastically change the background conditions of



Bhabha-L	"large angle Bhabha"
	either track $\theta > 10^\circ$
	both tracks $\theta > 1^\circ$
Bhabha-M	"medium angle Bhabha"
	both tracks $\theta > 0.5^\circ$
	not Bhabha-L
Bhabha-S	"small angle Bhabha"
	either track $\theta < 0.5^\circ$

Figure 4.5.: Distinction of the three different Bhabha classes in background campaign 15, based on the scattering angle θ of the outgoing particles. Small, medium and large scattering angles are distinguished. The generator BBBrem [75] is used for the small angle scattering Bhabha-S and BHWide [76] is used for the medium and large scattering angle samples Bhabha-M and Bhabha-L.

the machine background types. In the case of the luminosity backgrounds, the used generators, the configuration parameters, and the method of simulation is subject to change in different campaigns.

Background Campaign Comparison

Background campaign 15 (15p2 and 15p3) are the latest official background campaigns used in the experimental studies throughout this thesis. The background types included in campaign 15 are the QED luminosity backgrounds TwoPhoton and Bhabha as well as the machine backgrounds Touschek and Coulomb. The generator Aafh [78] is used for the generation of the TwoPhoton background. Because the Bhabha cross section strongly depends on the scattering angle of the outgoing particles, three phase space sectors are treated separately for the Bhabha scattering in campaign 15. Figure 4.5 shows the separation of the three classes Bhabha-S, Bhabha-M, Bhabha-L based on the scattering angle of both, the outgoing e^+ and e^-. Two different Bhabha generators are used: BHWide [76] is used for the medium and large angle Bhabha events in Bhabha-M and Bhabha-L, while BBBrem [75] is used for the small angle Bhabha events in Bhabha-S.

In the earlier background campaign 12 (12p3), Bhabha-M and Bhabha-L were not distinguished, but simulated in a single simulation run with the generator BHWide. Although the background TwoPhoton is included in campaign 12, the SimHits are only provided for the innermost detectors, the PXD and the SVD. Since no CDC SimHits are stored, TwoPhoton contributions to the CDC track trigger studies in this

Campaign	12p2		15p2		15p3	
	t [ms]	10^6 events	t [ms]	10^6 events	t [ms]	10^6 events
Coulomb	1	0.16	20	0.76	12	1.39
Touschek	1	0.11	20	1.46	12	2.42
Bhabha-S	1	2.61	140	13.25	12	44.61
Bhabha-M	1	0.13	140	0.34	12	1.15
Bhabha-L	0	0	1400	0.01	140	0.06
TwoPhoton	1	0.21	140	10.44	12	34.64

Table 4.2.: Real time equivalents and number of events in the official background mixing files of the background campaigns 12p3, 15p2 and 15p3.

thesis cannot be observed with 12p3. An distinctive feature in 12p3 is the tracking of the luminosity backgrounds: after the generation of the four-vectors of the QED backgrounds and prior to the detector simulation with Geant4, the machine simulation SAD was used for the tracking of the luminosity background particles inside the beam-pipe, from the IP up to the loss position at the beam-pipe. In contrast, in campaign 15 the luminosity background types are directly simulated with Geant4, which is reasonable as long as particle loss after several turns is not considered.

The two machine background types included in all three campaigns (12p3, 15p2, 15p3) are the inner bunch scattering (Touschek effect) and the elastic and inelastic beam gas scattering processes (Coulomb and Brems). Although the inelastic beam gas bremsstrahlung scattering background was generated with SAD, it is not included in the background mixing samples because the total rate is considered negligible.

Each background sample provided for mixing has a corresponding real time interval. An overview of the real time equivalents and number of events of the used background samples is shown in Tab. 4.2. For the luminosity background types, the real time equivalent is directly related to the cross section and to the luminosity. In contrast, for the generation of the machine background types rated events are used as input to the tracking with SAD. These rates have to be properly translated to yield the unrated machine background events for mixing with a corresponding real time equivalent. In the normal setup, large rated SAD events are repeated according to their rate, while low rated events are selected randomly. Hence, the real times and event numbers of the machine background types in Tab. 4.2, are not a direct indicator of the underlying population of simulated events. A problematic effect with this approach is a strong reduction of the actually used sample size of SAD events, as many of the low rated events are totally skipped. This also means, the real time equivalent is not a direct measure of the simulated sample size, because it is only drawn from a set of rated events. To overcome this implicit significance reduction and to create a large sample of background events, an alternative simulation mechanism has been developed in the course of this thesis, during the development of background campaign 15. The results of this background study can be found in [74]. A summary of the method for the high statistics machine background simulation is given in the Appendix B.1.

Selected Campaigns for the Trigger Studies

In the experimental studies with the neural network trigger and the 3D finder in the subsequent chapters, the method of background mixing is used with background files from three background campaigns. The background samples used in the studies with the neural network trigger in Chapter 6, Chapter 7, and Appendix D.1 are from background campaign 12 and background campaign 15 (15p3, 15p2, 12p3). In the studies with the 3D finder in Chapter 5, only the single background type 15p3 is used. Due to the lower Phase 2 beam currents and luminosities, the background rate in 15p2 is much lower than in the Phase 3 background 15p3. In order to match the early running conditions in Belle II, the neural network presently running in the hardware is trained with this background 15p2. For this reason, this Phase 2 background is included in the subsequent studies with the neural network trigger. In order to have an alternative Phase 3 background, 12p3 is included as third background type in the studies in the neural network studies.

These three background campaigns were also used in the large scale Belle II MC signal productions. In the Belle II MC signal productions "MC7" and "MC8", background campaign 12 was used for the Phase 3 backgrounds 12p3, while an intermediate background campaign 14 was used for the Phase 2 background simulation. In MC productions "MC9", "MC10", "MC11" the Phase 2 and Phase 3 files from background campaign 15 were used (15p2, 15p3). During the work on the high statistics background simulation [74] within the scope of this thesis, bugs in campaign 14 were successfully discovered and fixed, which contributed a successful production of the background campaign 15 with a larger sample size.

Occupancy Study with Background Mixing

In the background mixing samples, each background type in each campaign has a real time equivalent. During the mixing, this real time equivalent is used to select the appropriate amount of background to be mixed into the simulated signal. The real time equivalents and the number of events in each background sample are shown in Tab. 4.2. In principle, larger real time equivalents correspond to a higher statistics. However in the case of a rated simulation, as it is the case for the machine background types, additionally the number of simulated rated events has to be considered [74]. Table 4.2 shows campaign 15 to have much larger real time equivalents and numbers of background samples than campaign 12. Furthermore, Bhabha-L is not simulated separately in campaign 12, but included in Bhabha-M.

In this section, the essential properties of these background campaigns are studied. To this end, the background events are mixed into 10^5 events without any signal, which is similar to the creation of overlay files. The width of the simulated CDC trigger time window of $T = 628.82\,\text{ns}$ is used to calculate the rates in this study, by counting the CDC hits, TSs, 2D finder tracks and reconstructed tracks. At the single wire hit level, the rate of CDC hits is calculated. At the trigger level, at first the TS hit occupancies are measured at the TSF level. The following simulation of the 2D finder provides a measurement of the expected trigger rate of background tracks. The additional offline reconstruction is carried out and the rate of reconstructed tracks is calculated.

In this study, the background rates of each background type (shown in Fig. 4.6) are obtained via counting the CDC SimHit contributions. During the digitization, each created CDC hit is related to the SimHits used as input to create that hit. Since for each mixed background SimHit the background type is known, this can be used to read the background type of each CDC hit. In the case of the TSF, the 2D finder, and the reconstruction, the studied objects are composed of several CDC hits. Due to the background mixing, usually multiple different background types contribute to the single TSs, 2D tracks, and reconstructed track. In order to determine the fractional background composition for an object with N_{sim} related SimHits, the number of occurrences n_b of each background type b is counted ($N_{\text{sim}} = \sum_b n_b$). The fractional background contribution f_b of each background type is

$$f_b = \frac{n_b}{N_{\text{sim}}} \tag{4.16}$$

such that each TS and track in total is counted as one ($\sum_b f_b = 1$).

After filling a histogram with the fractional background counts, the background rates r_b are calculated by the each background count f_b by the number of events N_{events} and by the window T:

$$r_b = \frac{f_b}{N_{\text{events}} \cdot T} \tag{4.17}$$

These r_b are the total rates of CDC hits, TSs, 2D finder tracks and reconstructed track. The relative hit rates in Fig. 4.6 are calculated by dividing the total hit rates r_b by the total number of respectively CDC sense wires (15336) or TS hits (2336).

The observed hit and track rates in the background only events for the campaigns 12p3, 15p2, 15p3 are shown in Fig. 4.6. The hit rates in Fig. 4.6 (a) and (b) are relative rates in [kHz]; the CDC hit rates in Fig. 4.6 (a) are the averaged hit rates at each of the 15336 sense wires; the TS hit rates in Fig. 4.6 (b) are the averaged rates at each of the 2336 TSs. The noise suppression of the TSF becomes visible in the lower TS hit rate in Fig. 4.6 (b) compared to the CDC hit rate in Fig. 4.6 (a).

The track rates in Fig. 4.6 (c) and (d) are the total rates r_b (Eq. 4.17). For the background 15p3, 2D trigger rate in Fig. 4.6 (c) is about 136 kHz and the rate of reconstructed tracks in Fig. 4.6 (d) is about 190 kHz. These track rates are far beyond the maximum L1 trigger rate of 30 kHz, which clearly signals the necessity of a novel track triggers. The 2D track rate observed here via mixing is slightly larger than the 2D track rate of 121.5 kHz measured in [74]. The reason is the additional background pileup through the mixing, which results in fake tracks composed of

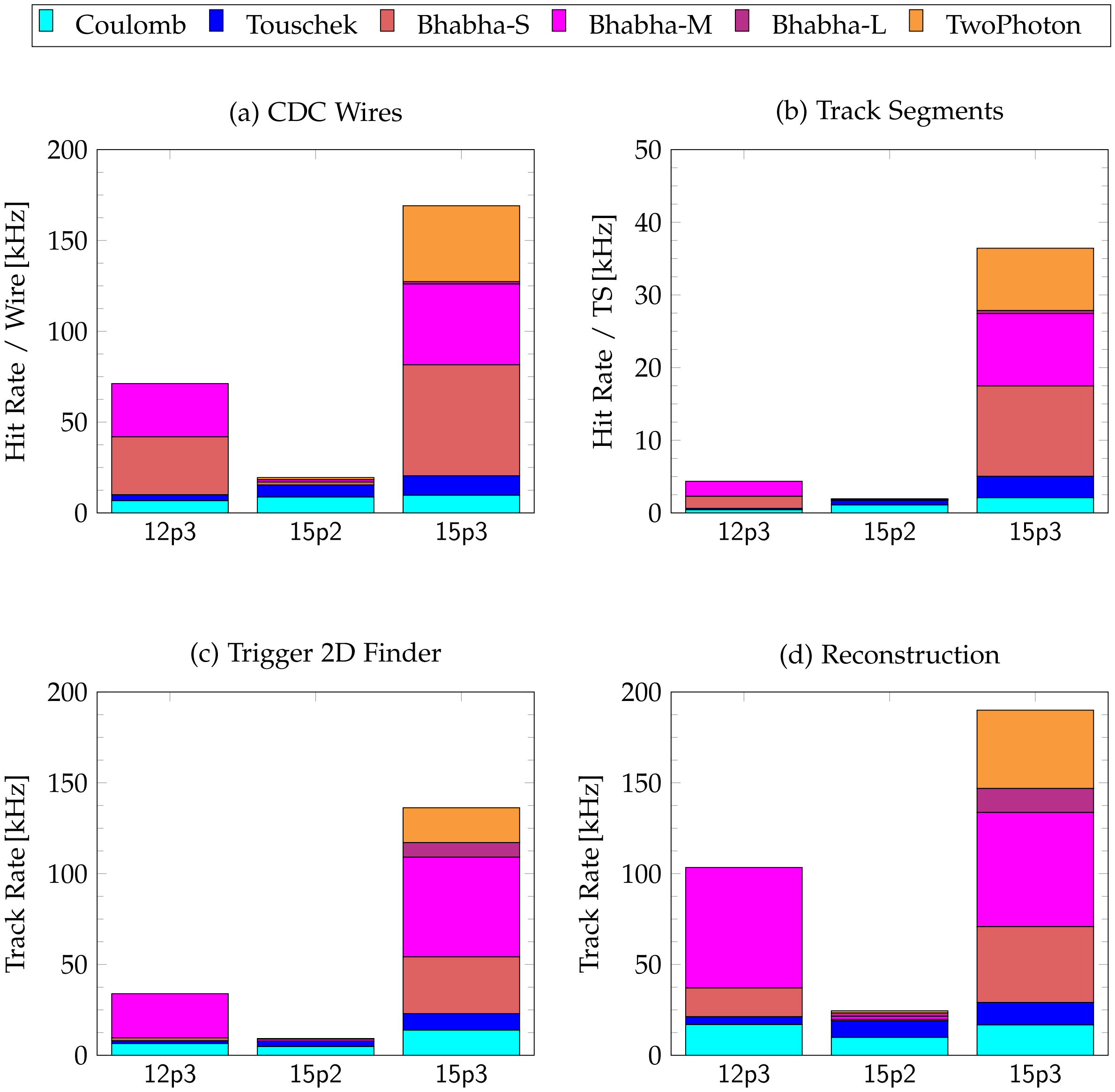


Figure 4.6.: Rates of the CDC hits, TS hits, 2D finder tracks, and reconstructed tracks in the background campaign 12 Phase 3 (12p3), campaign 15 Phase 2 (15p2), and campaign 15 Phase 3 (15p3). The hit rates are averaged relative rates (CDC hit rate per each of the 15336 sense wires; TS hit rate per each of the 2336 TSs). The rates are calculated from 10^5 background only events simulated with the background mixer in basf2, by assuming a fixed time window T for each event. Corresponding to the difference of the maximum and minimum CDC hit timings provided by the CDC digitizer in the simulation, the selected time window is $T = 640\,[\text{tdc clock}] \approx 628.83\,\text{ns}$.

several background types. In contrast, in [74] each background type was simulated separately (see Appendix B.1).

Chapter 5.

3D Track Finder

At the time of the launch of Belle II, the scheduled approach for track finding in the CDC first level trigger system is based on a 2D Hough transformation (see Sec. 3.3). It receives its input from the Track Segment Finder (TSF) and sends its output to the 3D track reconstruction methods, including the neural network trigger presented in Chapter 6. This 2D track finder processes only Track Segments (TSs) information from the five axial Super Layers (SLs), while the information from the four stereo SLs is skipped for track finding at this stage. The stereo TS information is only used in the following 3D track reconstruction methods. Combining the already found 2D tracks with the stereo TSs provides the information required for the precise 3D track reconstruction (see Chapter 6).

The main advantage in using only axial TSs information in the 2D finder is a reduction of the input data size and a complexity reduction of the 2D Hough algorithm. This simplifies the hardware (HW) implementation, which is beneficial considering the limited resources of the L1 trigger. However, the disadvantage in discarding the additional information from the stereo TSs is a significant loss in the track finding efficiency. Tracks with more than one missing axial TSs are lost, although they could be recognized by including the additional stereo TS information. Especially tracks with shallow polar angles of $\theta \lessapprox 35^\circ$ or $\theta \gtrapprox 123^\circ$ and with low transverse momenta of $p_\mathrm{T} \lessapprox 350\,\mathrm{MeV}$ are in danger to get lost (see Sec. 5.3.1).

Using a 3D finder that includes the stereo TSs can in particular improve the efficiency for physics signal events with a low track multiplicity. Several channels important in the searches for physics beyond the standard model are affected. This includes searches for lepton flavor violating $\tau \to \mu\gamma$ decays, dark matter searches and rare events with invisible final states. Furthermore, hadronic cross section measurements in initial state radiation events are an important low multiplicity channel where precision measurements are required to unravel the nature of the muon $g-2$. Further examples of low multiplicity events are discussed in Sec. 2.5.1. Experiments with simulated initial state radiation using the 3D finder in combination with the neural network are presented in Sec. 7.1.

The wires in the axial SLs processed by the 2D finder are parallel to the magnetic field and oriented along the z-axis. For these axial wires the transverse mounting positions at the forward and backward endplate coincide. For this reason, the axial TSs appear as single points in the transverse projection. Owing to their stereo angle

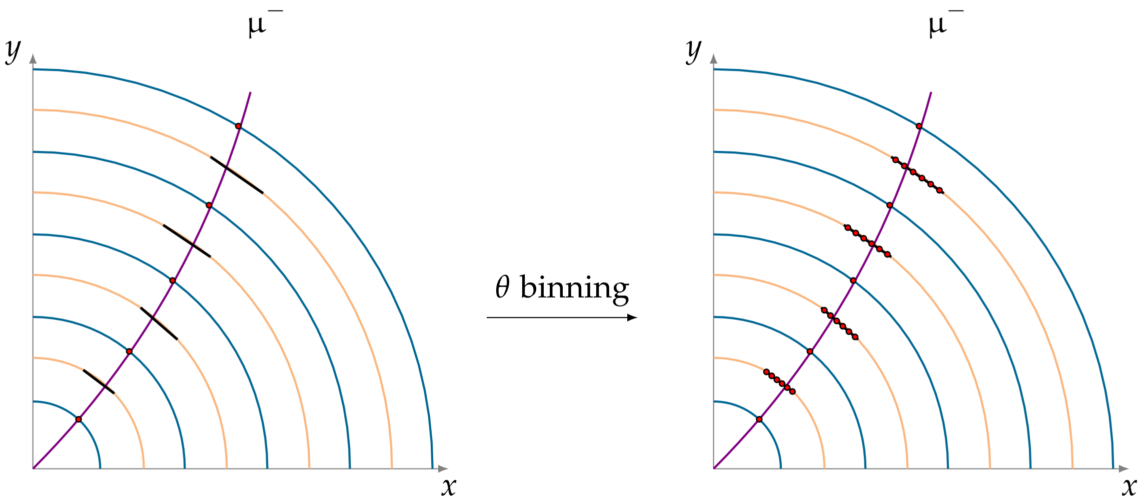


Figure 5.1.: Priority wires of the TSs related to an exemplary simulated charged track (here a μ^- with $p_\mathrm{T} = 1\,\mathrm{GeV}$) projected into the transverse plane. The reference wire positions of the axial TSs, which are parallel to the z-axis, are represented as single points. Due to their stereo angle with respect to the z-axis, the stereo TSs appear as short line segments connecting the mounting positions of the wires at the forward and backward endplates. A binning in the polar angle θ provides single point reference positions of the stereo wires as well.

with the z-axis, the stereo wires in the transverse plane appear as short line segments (see Fig. 5.1). The endpoints of these stereo line segments are given by the mounting positions of the stereo wires at the forward and backward endplate (see Sec. 3.2.5).

In this chapter, a new 3D track finder is proposed which can make use of the stereo TSs. By simultaneously processing the TS information from all nine SLs, axial and stereo, an improved track finding efficiency can be expected. The basic idea of this 3D finder algorithm is to apply a binning along the polar angle θ as illustrated in Fig. 5.1. Thus, single point representations of the stereo TSs in the transverse plane are obtained, which can be processed similarly as the axial TSs.

Benefits with the 3D finder are expected especially for signal tracks with shallow-θ angles, which leave the CDC volume before passing through all SLs. Here shallow-θ angle refers to tracks with a small angle of the momentum vector with the z-axis in either the for forward or backward direction. If the outer axial TSs are missing due to the shallow-θ angles, these tracks are lost. However, they do have a sufficiently large transverse momentum p_T to be easily reconstructed by using the additional stereo TSs. Similarly, the efficiency for low-p_T tracks can be improved. Because the crossing angle of the track with the SLs is too large, low-p_T tracks with their large curvature do not create TS hits in the outer SLs. A further benefit of the 3D track finder is its ability to provide estimates of the 3D track parameter θ which enables refined 3D track reconstruction algorithms at the trigger level.

The track finding algorithm of the existing 2D finder and of the new 3D finder is based on the Hough transformation [80]: by parametrizing a track in geometrical space, the hit positions along the track can be transformed to curves in the track parameter space. Parameters of the actual tracks are found at the intersections of the hit curves in the parameter space. This general Hough transformation was developed at the time of bubble chambers. Tracks can be found analytically by calculating the intersections of the hit curves or by binning the parameter space and using clustering and peak finding algorithms. While the analytical solution is feasible in software implementations, in hardware typically the clustering in a binned parameter space is used.

The already installed 2D finder in the Belle II trigger (see Sec. 3.3.4) assumes that all tracks are coming from the IP in the transverse plane with vertices at $(x_0, y_0) = (0, 0)$. With the constant and homogeneous magnetic field in z-direction, charged tracks appear as circles in the transverse x, y-plane passing through the origin (see Sec. 3.2.3). These circular 2D tracks can be parametrized by two track parameters: p_{T} and ϕ, the transverse momentum of the track and the azimuthal angle of the track momentum at the IP. In the extended 3D finder the vertices are also constrained the IP as well, which means to set the 3D vertex position to $(x_0, y_0, z_0) = (0, 0, 0)$. The tracks are parametrized by $(p_{\mathrm{T}}, \phi, \theta)$, with an additional 3D track parameter θ, the polar angle of the track.

Details on the mathematics involved in the track finding algorithms with a Hough transformation are described in Sec. 5.1. The implemented 3D finder algorithm is presented in Sec. 5.2. Performance measurements of the proposed 3D finder setup are presented in Sec. 5.3 and optimization studies of the 3D finder setup are discussed in Sec. 5.4. Integrated performance measurements with the neural network on tracks displaced from the IP are presented later in Chapter 6 and applications to physics test cases in Chapter 7.

5.1. Hough Transformations

The main ingredients used in the development of the 3D finder are the Hough transformation and Bayesian parameter estimation. This section introduces the underlying mathematical concepts used in the track finding algorithms. The basic linear Hough transformation is introduced in Sec. 5.1.1, followed by the description of a conformal transformation which allows to use the linear Hough transformation with circular charged particle tracks in a constant magnetic field in Sec. 5.1.2. This 2D circular Hough transformation is the method used in the existing 2D finder. The concept of model independent Bayes track finding, which is used in the proposed 3D finder, is introduced in Sec. 5.2.1. The analytical calculation of the 3D Hough transformation is discussed and compared to the model independent Bayesian track parameter estimation in Sec. 5.1.3. Such an analytical 3D Hough transformation is equivalent to the Bayesian parameter estimation, under the condition that the same track model – helical tracks in a constant magnetic field – and the same track parametrization is used. However, different track model corrections including energy loss or field non-linearities can easily be realized with the proposed 3D finder setup. By training the hit representations on real data from the Belle II detector or on MC data including advanced track models, these corrections are effectively included in the 3D finder track estimations.

5.1.1. Straight Line 2D Hough Transformation

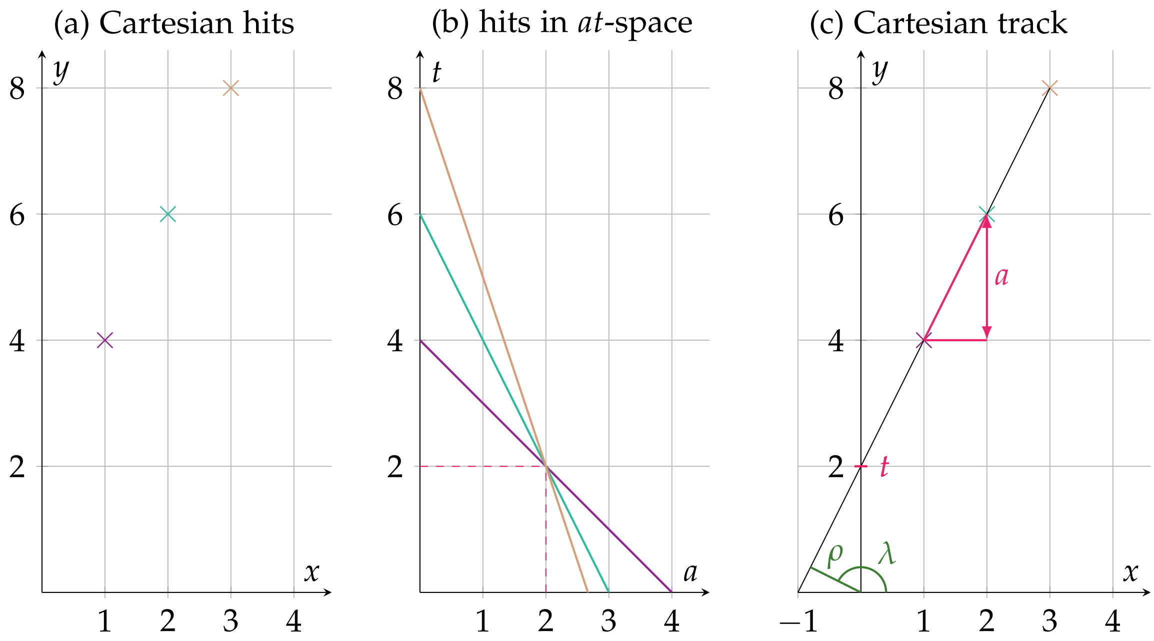


Figure 5.2.: Basic Hough transformation with a linear track model. The hits shown in (a) in a geometrical space are transformed to straight lines in the a, t-parameter space in (b). The parameters of the track responsible for the hits are found at the intersection point of the hit-lines in (b); (c) shows the resulting track found after a back-transformation from the parameter space. In (c) both track parameterizations are illustrated, a, t and the normal parametrization λ, ρ.

The first linear version of the Hough transform was originally invented by Paul V. C. Hough [80]. Given a set of measured hit points in a detector and a physical model for a track leaving those hits in a detector, the Hough transformation is a method to obtain the track parameters form the set of hits. As an example, the 2D projection of a straight track traversing a detector volume can be described by a straight line in a Cartesian 2D coordinate system with the geometrical detector x, y-axes:

$$y = a \cdot x + t \tag{5.1}$$

The track propagation is fully described by the slope parameter a and the y-intercept t of the line. Given a set of N hits $(x_i, y_i)_{i\in[1,N]}$ in the detector plane, as illustrated in Fig. 5.2 (a), the task of a track finder is to find the parameters (a, t). In the linear Hough transformation, the hits (x_i, y_i) in geometrical space are represented as lines in parameter space, as shown in Fig. 5.2 (b). The (a, t) points on the hit curves in the parameter space correspond to all the tracks that could possibly create that hit:

$$t = -x_i \cdot a + y_i \tag{5.2}$$

The real track parameters (a, t) of the track being responsible for all the hits can then be found at the intersection point of the hit lines in the parameter space. A back-transformation of those intersections in Fig. 5.2 (b) into the geometrical space yields the found tracks illustrated in Fig. 5.2 (c).

Normal Parametrization

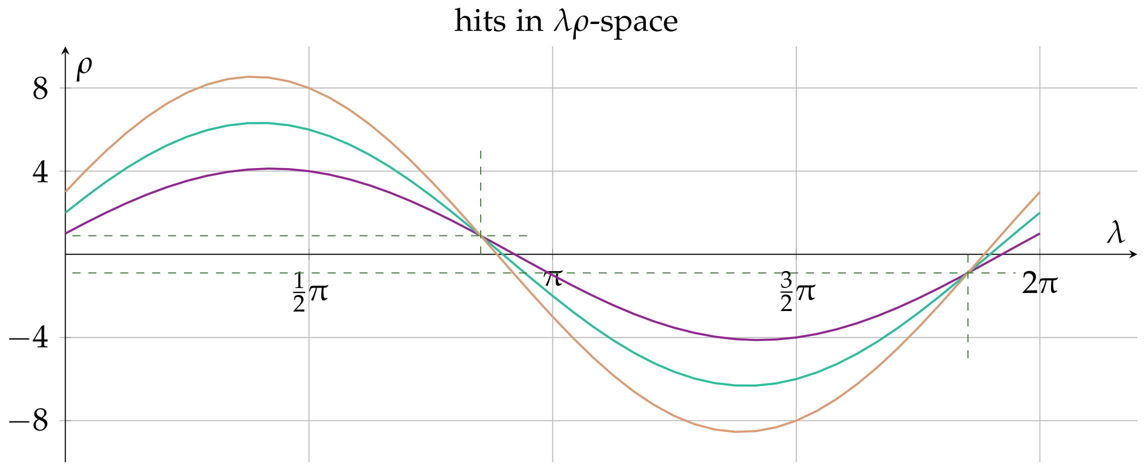


Figure 5.3.: Hit curves in the λ, ρ-space using the normal parametrization of straight lines. The hits are shown in the previous Fig. 5.2 (a). The intersections of the sine curves identify the same track as found with the slope intercept parametrization, as illustrated in Fig. 5.2 (c).

This linear track model was extended by using the normal parametrization of a straight line, which is well suited for the Hough transformation [81]. A straight line is described by the length of its normal vector ρ to the origin and the angle of this normal vector λ, as illustrated in Fig. 5.2 (c) in addition to the a, t-parametrization. In the normal parametrization, the equation for the straight line becomes [81]:

$$y = -x\cot(\lambda) + \frac{\rho}{\sin(\lambda)}. \tag{5.3}$$

where a comparison of the coefficients illustrates the relation to the slope a and the y intercept t. In the normal parametrization, all the hits (x_i, y_i) are mapped to the sinusoidal curves in the λ, ρ-parameter space shown in Fig. 5.3:

$$\rho = x_i \cos(\lambda) + y_i \sin(\lambda). \tag{5.4}$$

In this parametrization, the orientation of the line is defined by the angle λ rather than by the slope a. Since the angle λ is continuous, this parametrization has the advantage that numerical instabilities due to infinite slopes a are avoided.

The inverse relations between the a, t-space and the λ, ρ-space can be derived geometrically for a straight line $\vec{l}(x)$ and its normal line $\vec{n}(x)$. The equations for the straight line and its normal vector in terms of the a, t-parameters are

$$\vec{l}(x) = \vec{l}_0 + x \cdot \vec{l}_1 = \begin{pmatrix} 0 \\ t \end{pmatrix} + x \cdot \begin{pmatrix} 1 \\ a \end{pmatrix} \qquad \vec{n}(x) = x \cdot \vec{n}_1 = x \cdot \begin{pmatrix} 1 \\ -a^{-1} \end{pmatrix} \tag{5.5}$$

where $\vec{n}(x)$ is defined by the condition $\vec{l}_1 \cdot \vec{n}_1 = 0$. Calculating the intersection x_{cross} of $\vec{l}(x)$ and $\vec{n}(x)$ yields the distance of closest approach ρ:

$$\rho_{1,2} = \pm \frac{t}{\sqrt{a^2 + 1}} \qquad \lambda_{1,2} = \text{arctan}_2 \left(-a^{-1} \right) \tag{5.6}$$

where the two solutions describe two intersections of the sine curves shown in Fig. 5.3. The second solution for λ is obtained from the arctan_2 function, which extends the arctan function to the full 2π region.

Note that in principle a half-space is sufficient to describe all straight lines; since the second solution redundantly describes the same straight line with λ shifted by π and negated ρ. Although in the case of a charged particle with its vertex position located at the distance of closest approach ρ, this corresponds to the two possible directions of the particle along the straight line.

5.1.2. Circular 2D Hough Transformation

Charged particles moving in a uniform, constant magnetic field $\vec{B}$ are bent by the Lorentz force $\vec{F}$ perpendicular to their flight direction and perpendicular to the magnetic field (see Sec. 3.2.3). With the small material budget of the CDC (see Sec. 3.2.5), the multiple scattering energy loss can be neglected such that the ideal track model can be used as a good approximation. Ideal particle tracks without energy loss due to interactions with material, can be represented by circles in the transverse plane, as illustrated in Fig. 5.4.

Circle Model

Generally, three values are required to describe a circle in a 2D plane: in Cartesian coordinates, these parameters are the center of the circle $\vec{m} = (m_x, m_y)$ and the radius of the track r. In the used physical model the track parameters are (see Sec. 3.2.3):

the transverse displacement d_0 of the track vertex (point of closest approach to the IP), p_T - the transverse momentum of the track which is proportional to the radius r of the circle and ϕ_0 - the tangential angle of the track direction at the vertex position. However, in the setup of a particle physics collider experiment, all tracks are assumed to originate from the IP, where the anticipated interactions occur. This condition fixes one parameter on the circle to $(x, y) = (0, 0)$, i.e. the tracks have $d_0 = 0\,\mathrm{cm}$ and the only free parameters left are p_T, ϕ_0. In Cartesian coordinates, the free parameters are the 2D position of the center $\vec{m} = (m_x, m_y)$ whose absolute value is then identical to the track radius

$$r = \sqrt{m_x^2 + m_y^2}. \tag{5.7}$$

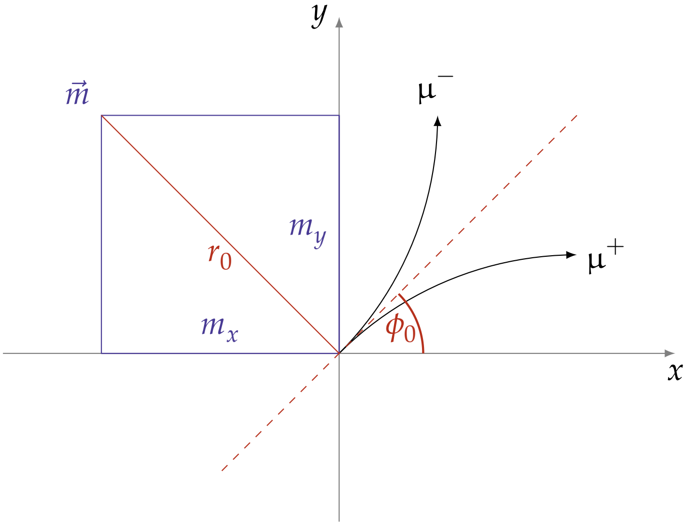


Figure 5.4.: Cartesian and polar track representations in the transverse plane of charged particles in the magnetic field $\vec{B} = (0, 0, B)$. Particles starting at the IP, $(x, y) = (0, 0)$, in the direction ϕ_0 propagate along a circular path with radius $r \propto p_\mathrm{T}$. The charge determines the direction: negatively charged particles propagate counter-clockwise around the center $\vec{m}_- = (m_x, m_y)$, positively charged particles clockwise around $\vec{m}_+ = (-m_x, -m_y)$.

Although the circular 2D track model itself is independent of the charge, the charge of the particle determines the direction of the track along the circle. Both track parametrizations are visualized in Fig. 5.4 for two particles of opposite charge. The important equation for $\vec{m}(\omega, \phi_0)$ was already defined in Eq. 3.13

$$\vec{m}(\omega, \phi_0) = \begin{pmatrix} m_x \\ m_y \end{pmatrix} = \omega^{-1} \begin{pmatrix} \sin(\phi_0) \\ -\cos(\phi_0) \end{pmatrix} \tag{5.8}$$

Tracks with transverse momenta of $p_{\mathrm{T}} \gtrapprox 350\,\mathrm{MeV}$ leave the CDC volume (see Sec. 5.3.1). Since they are stopped by interactions with the outer detector components, only a fraction of the circular track is seen in the detector. Using the assumption that the tracks originate from the IP this can beneficially be used to determine the charge of the found particle tracks.

All possible hit points along the track have to fulfill the following circle condition:

$$(x_i - m_x)^2 + (y_i - m_y)^2 = m_x^2 + m_y^2 \tag{5.9}$$

which is a quadratic model equation in the hit positions (x_i, y_i). By inserting the polar track representation of the track center $\vec{m}$ (Eq. 5.8) into (Eq. 5.9), the circle equation can be rewritten in the following form:

$$\omega = \frac{2 \cdot y_i}{x_i^2 + y_i^2} \sin(\phi_0) - \frac{2 \cdot x_i}{x_i^2 + y_i^2} \cos(\phi_0). \tag{5.10}$$

Beneficially, such a circle equation can be linearized using a conformal mapping [82]

$$x_i' = \frac{2 \cdot x_i}{x_i^2 + y_i^2} \qquad\qquad y_i' = \frac{2 \cdot y_i}{x_i^2 + y_i^2} \tag{5.11}$$

which transforms the hit points (x_i, y_i) on a circle through the origin to the points (x', y') on a straight line. A reconstruction of the track parameters of this straight line provides the 2D track parameters.

The final equation for each hit curve in the $\omega\phi_0$-parameter space used for the Hough transformation is:

$$\omega = x' \sin(\phi_0) - y' \cos(\phi_0) \tag{5.12}$$

which is now linear in the transformed hit positions (x', y'). This is the track model used by the 2D finder in the Belle II trigger [67].

Linearized Parameter Space

Due to the linearization, it is essentially the same Hough equation as obtained for the normal parametrization of straight lines in Eq. 5.4. However, the track parameters in Eq. 5.12 have a different interpretation; $\rho \equiv \omega \propto q \cdot p_{\mathrm{T}}$ becomes the signed curvature of the track and $\lambda \equiv \phi_0$ becomes the track direction at the IP.

As in the case of the normal track parametrization, the tracks are found as intersections of sinusoidal curves, as shown in Fig. 5.3 in the λ, ρ-space. Due to the different interpretation of the parameters λ and ρ, the sign of ρ is now important to determine the charge of the particles. The two solutions in Fig. 5.3 correspond to a positively charged track propagating in ϕ-direction and to a negatively charged track propagating in $\phi + \pi$-direction. As mentioned earlier, the ambiguity of the two solutions can be resolved for found tracks in the normal CDC acceptance region (see Sec. 5.3.1). Since these tracks are stopped in the outer detector layers, usually only one quarter of the circular track creates TS hits in the CDC. Technically, the two intersections are resolved in the 2D finder by taking only those intersection points where all the hit curves have a positive slope. Using the above sign conventions in the definition of the circle center $\vec{m}$, a positive slope corresponds to outwards going tracks.

The steps of the 2D circular Hough transformation carried out by the 2D finder are illustrated in Fig. 5.5. The axial hits in Fig. 5.5 (a) lie on a circular path along a track through the IP. With the conformal mapping, the hits are transformed onto the straight line in Fig. 5.5 (b). These hits are represented as sinusoidal curves in the track parameter space in Fig. 5.5 (c). The sine curves for the outwards going track is shown as solid line, the curl-back region closing the full 2D Hough space as dotted line. In (c), the track parameters can be identified at the solid intersection point: it is a negatively charged track with $\phi = 45^\circ$ and $p_\mathrm{T} = 1\,\mathrm{GeV}$.

In Fig. 5.5 (c) the low-p_T region outside of the CDC acceptance region is marked gray. In the red area, tracks do not create TSs in all 9 SLs (see Sec. 5.3.1) which results in a significant loss in track finding efficiency of the 2D finder. In this region, benefits can be expected from the 3D finder. In the central region, for tracks with $p_\mathrm{T} \gtrapprox 0.35\,\mathrm{GeV}$ the 2D finder reaches almost 100 % track finding efficiency.

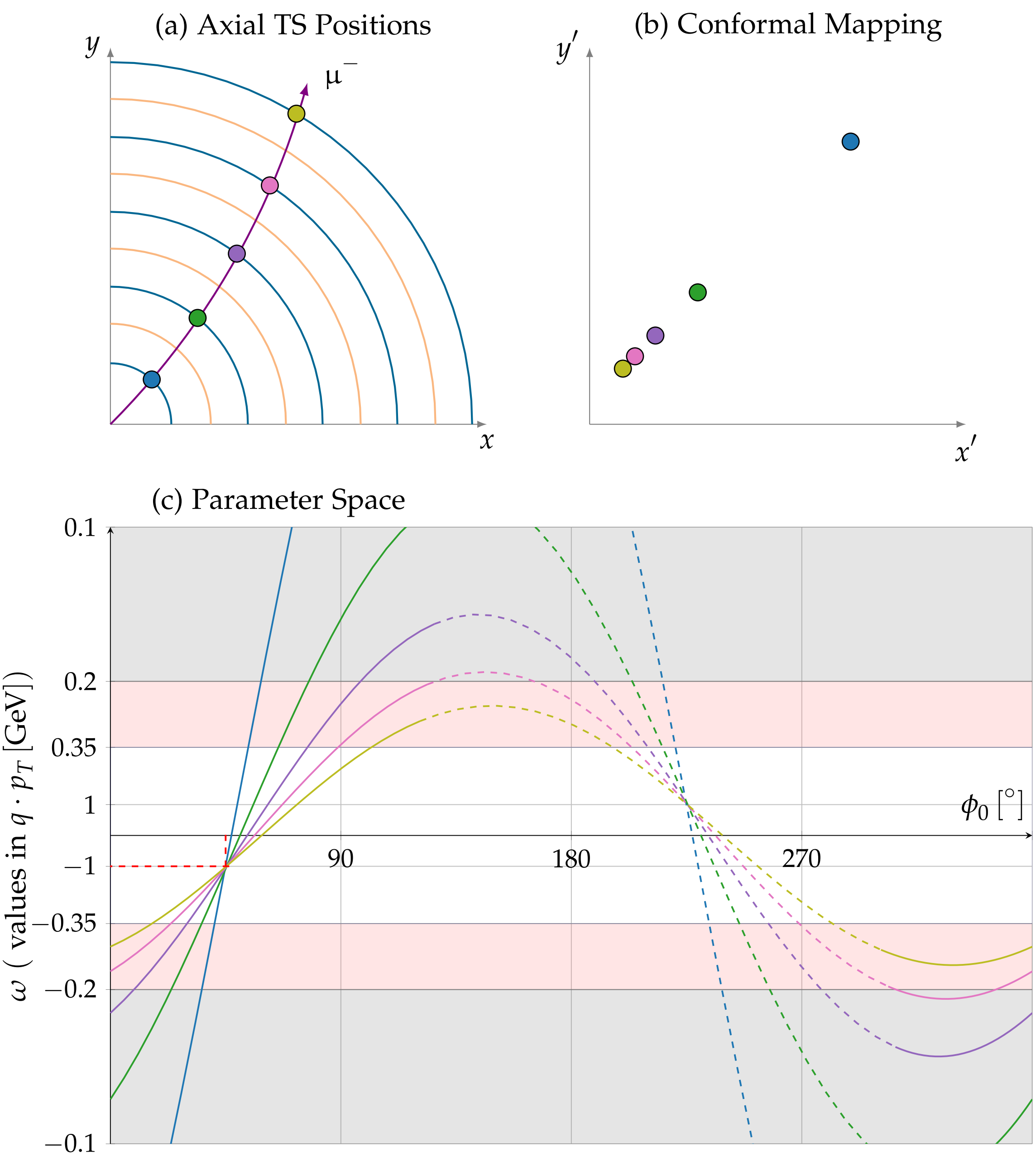


Figure 5.5.: Circular Hough transformation. The axial hits on the track circle in (a) are conformally mapped onto the straight line in (b). In (c) the track parameters are found as intersections of the sine curves using Eq. 5.12. Note that the $\omega \propto qp_{\mathrm{T}}^{-1}$ axis in (c) is scaled uniformly in ω. However, the values on the axis are shown in qp_{T}. Only in the central region ($p_{\mathrm{T}} > 0.35\,\mathrm{GeV}$) the 2D finder has a high efficiency. In the region marked red, where tracks are still visible in the CDC but do not create hits in all 9 SLs, benefits can be expected from the 3D finder

5.1.3. Helical 3D Hough Transformation

This subsection discusses the analytical calculation of the 3D Hough transformation. It addresses the actual 3D model, which is calculated by the model independent Bayesian approach of the proposed 3D finder (see Sec. 5.2.1), under the condition that the same track- and hit-parametrizations are used. As previously discussed, the axial wires appear as single points in the transverse plane (see Fig. 5.1) and can be processed by the previously described 2D Hough transformation. In contrast, since the stereo wires appear as short line segments in the transverse plane, it is necessary to find the 3D hit positions along the stereo wires.

Here it is assumed that tracks have their vertices at the IP, which means $(d_0, z_0) = (0, 0)$ cm. In this way, the stereo hit positions can be expressed as functions of ω, ϕ, θ. In principle, this is done by searching for the intersection points of the helical tracks with the wires. To this end, a 3D wire parametrization is introduced which allows to calculated intersections with the helix track parametrization in Eq. 3.16.

A stereo wire can be described by a straight line $\vec{w} + \lambda \vec{d}$ with a reference position $\vec{w}$ and a direction vector $\vec{d}$. This 3D parametrization of the stereo wires is illustrated in Fig. 5.6. It can be calculated from the ϕ-shift $\Delta\phi$ of the wire mounting positions at the forward and backward CDC endplates.

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} w_x \\ w_y \\ w_z \end{pmatrix} + \lambda \cdot \begin{pmatrix} d_x \\ d_y \\ d_z \end{pmatrix} \tag{5.13}$$

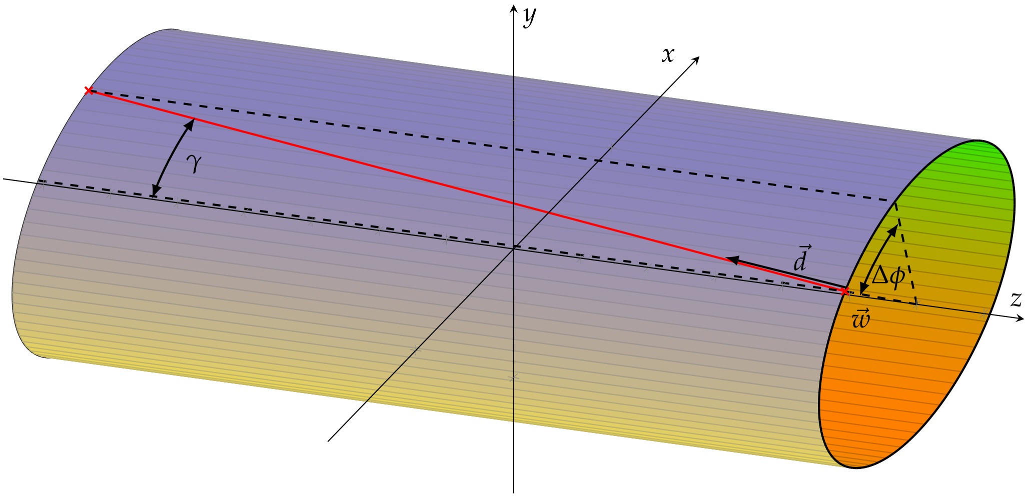


Figure 5.6.: Parametrization of a stereo wire as a straight line with a reference position $\vec{w}$ and a direction vector $\vec{d}$. A stereo angle γ and the direction vector $\vec{d}$ can be calculated from the ϕ-shift $\Delta\phi$ of the wire mounting positions at the forward and at the backward endplate.

3D Hit Curves

Extending the hit curves in the parameter space to 3D can be done by adding an expression for z', which denotes the z-position of the hit. For axial TSs the x', y'-positions are constant for all values of θ, i.e. it does not change with the z'-position of the hit. Hence, the 2D sine curves in the ω, ϕ-space described by Eq. 5.12 is also valid in the 3D extension of the sine shaped planes.

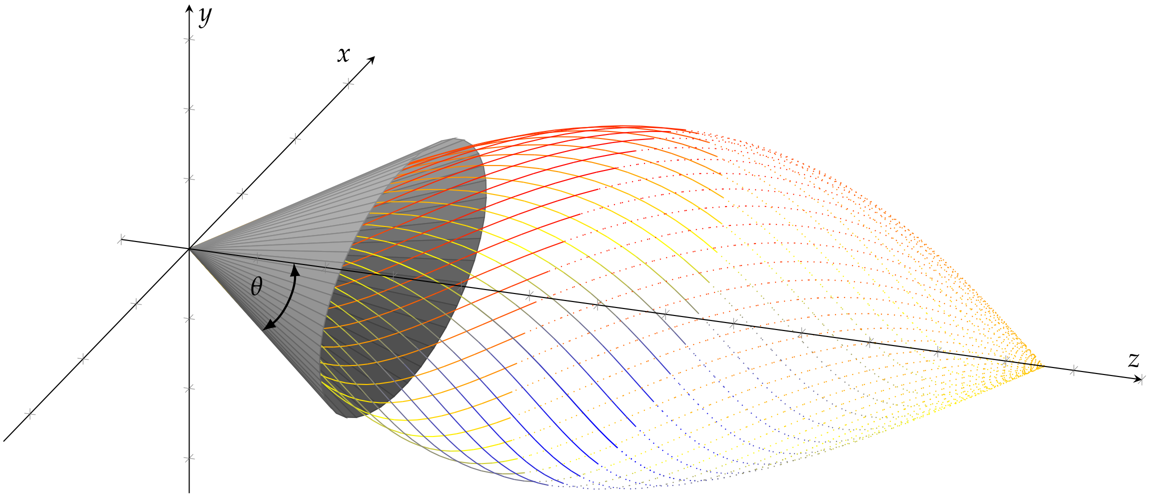


Figure 5.7.: Array of track helices from the IP, with a fixed polar angle θ rotated by ϕ_0 and a volume corresponding to p_T. In the high-p_T limit of straight tracks, the array of tracks can be approximated by the cone with opening angle θ. However, for low-p_T tracks with a large curvature, the array of tracks has to be described using the helix equation (Eq. 3.16). The outwards going part of the tracks are shown as solid curves and the curl-back region as dotted curves.

However, in the case of stereo hits the x', y'-positions have to be expressed as functions of the 3D track parameters. This can be done by searching for the intersection points of the wires with the helical tracks. As an example, the array of helical tracks rotated in the transverse plane by ϕ_0 and with a fixed value of the polar angle θ is illustrated in Fig. 5.7. The tracks have their vertices at the IP, so they are described by the three parameters $p_\mathrm{T}, \phi_0, \theta$. In Fig. 5.7, the helix equation (Eq. 3.16) is used. Therefore, the volume of the array of tracks corresponds to the value of p_T (i.e. the radius of the track circle in the transverse plane).

In the high-p_T limit, all the points that can be reached from the IP with a specific value of θ can be approximated by a cone with half-opening angle θ. The tip of the cone is located at the vertex position and its axis is parallel to the z-axis. For tracks with vertices at the IP, all points $\vec{x}$ on the cone have to fulfill

$$\vec{x} \cdot (0,0,1)^T = |\vec{x}| \cos(\theta) \tag{5.14}$$

All points along the wire are described by the previously introduced wire representation $\vec{w} + \lambda\vec{d}$. In this high-p_{T} approximation, the transverse stereo hit positions can now be found as a function of θ. In order to find the intersection point, the following equation has to be solved for λ

$$(\vec{w} + \lambda\vec{d}) \cdot (0,0,1)^T = |\vec{w} + \lambda\vec{d}| \cos(\theta) \tag{5.15}$$

This equation can be solved analytically to obtain $\lambda(\theta)$ and thus the hit positions for a given value of θ.

However, for general tracks with lower values of p_{T}, the curvature of the tracks has to be considered. The cone approximation deviates from the array of tracks (see Fig. 5.7). Hence, the general intersection point depends on both, θ and ω. For each given value set of 2D track parameters ω and ϕ_0, this can be done analytically. In the transverse plane, the intersection of a circle with a straight line has to be calculated, which can be done by solving a quadratic equation. This is the method pursued by the 3D track reconstruction methods, which are described later for the 3D helix fit method in Sec. 6.1.4.

Without the knowledge of the 2D parameters, the description of the intersection points is more cumbersome. Most importantly, the helical track usually does not have an intersection point with the wire due to the 3D crossing. In order to construct a general method, the point of closest approach of a helix to a straight wire has to be calculated. However, this distance minimization problem cannot be solved analytically. An iterative fitter is therefore required to find the 3D hit positions along the wires.

In general, the explicit methods to find the 3D hit curves strongly depend on the track and hit models with their parametrizations. This complicates a possible modeling of future corrections to the models and poses difficulties for the adjustments of the track finding algorithm. Beneficially, the state of the art corrections are already implemented in the MC simulation. For different particle types, simulations can be used to obtain the average corrections to the idealized helical track model due to multiple scattering. For this reason, a Bayesian parameter estimation combined with a machine learning model to train the hit representations in the track phase space is the method of choice for the 3D track finder presented in this thesis. This approximation of the 3D Hough transformation is presented in the next Sec. 5.2.

5.2. 3D Track Finder Algorithm

The 3D track finder proposed in this thesis is a machine learning approach to obtain the hit representations in a 3D track parameter space. These hit representations can be combined to a Bayesian estimation where the most likely tracks for a given set of hits are found by a clustering algorithm. Both hit types, axial and stereo TSs, are used to find the track candidates in the binned 3D track parameter space: the ω, ϕ, θ-space. In this way, the hit curves become sinusoidal curved planes in the 3D space. The track finding procedure is similar to a binned Hough transformation. After filling the Hough space with the hit representations, tracks are estimated by searching the most probable track for a given set of hits with a peak finding algorithm. However, the hit curves in the parameter space are obtained in a track model independent way. Instead of calculating the Hough plane contributions for each hit from the track model, a training step is used to learn the track-to-hit relations from a training data set. This training data set can be simulated MC tracks as well as real Belle II data from reconstructed tracks. The track model is represented in the training data and enters only indirectly via the track parametrization. This generality of the method simplifies adaptions of the track finding algorithm. For instance, an additional track parameter axis could be introduced to realize 4D track finding. Furthermore, new hit representations of TSs can be incorporated by a retraining of the hit curves in the Hough space. In order to improve the TSF efficiency for low-p_T tracks, such novel TSs with different shapes might be added in future upgrades of the Belle II trigger,

By including the stereo hits, an improvement of the track finding efficiency can be expected. If axial hits are missing due to the limited efficiency of the TSF, these additional stereo hits can recover tracks otherwise lost. In addition, the stereo hits are directly related to the found tracks, which is advantageous for the subsequent hit selection in the neural network preprocessing (see Sec. 6.3.2). Due to the extension of the Hough space to 3D, the polar angle θ can be estimated as a third track parameter during the track finding. As will be discussed below (in Sec. 5.2.5), the stereo hit representations depend on a combination of θ and z_0. Therefore, a 3D background suppression of displaced tracks is already realized prior to the neural network trigger.

Using a machine learning based approach to the track finding has the advantage that all corrections represented in the training data are implicitly used by the track finder. In particular, an error prone remodeling of the corrections which are already present in training data is avoided. This includes the modeled MC corrections in simulated data sets, like detector readout effects, background fuzziness, or field inhomogeneities. Alternatively, the hit representations can also be trained using real data from the Belle II detector. In this way, even corrections missing in the MC can be included in the 3D track finding.

With the presented 3D finder method an implementation in parallel FPGA hardware and an integration into the Belle II first level trigger pipeline (see Sec. 3.3.1) is anticipated. To this end, the pre-trained binned representations of the hit curves in the track parameter space have to be stored on the FPGA in order to construct the full Hough space for the track finding. However, this probabilistic creation of the hit representations in a binned Hough space is an independent new concept. Due to the improved accuracy of the hit patterns, benefits can also be expected if the method is applied to a 2D track finder only.

The concept of the probabilistic track finding is introduced in Sec. 5.2.1. Based on this concept, in Sec. 5.2.2 the binned Hough space of the 3D finder is defined where hits are represented as binned sine curved planes. In order to minimize the memory requirements for the hardware and software implementations, the ϕ-symmetry of the CDC is used in the hit representations as described in Sec. 5.2.3. The details on the training procedure, where the hit representation in the track parameter space are created, are provided in Sec. 5.2.4. After normalization and discretization, the values in the binned hit curves are weights. These weights represent the contributions of each hit to each track in the 3D Hough space. Tracks are found in the joint Hough space of all the hits in the event by the density based clustering algorithm described in Sec. 5.2.5.

5.2.1. Probabilistic Track Finding

The probabilistic Bayesian parameter estimation is the basis for the proposed 3D finder setup. A general concept of this 3D finder was first published in [83] and updated version is accepted for publication in [84]. In a Bayesian manner, the track finding problem can be formulated as follows: given a set of hits H, what is the most probable set of tracks T being responsible for those hits? Treating T and H as random variables, the Bayes' theorem of conditional probabilities allows to rewrite the track finding landscape:

$$\mathrm{P}(T|H) = \frac{\mathrm{P}(T,H)}{\mathrm{P}(H)} = \frac{\mathrm{P}(H|T) \times \mathrm{P}(T)}{\mathrm{P}(H)} \tag{5.16}$$

where $\mathrm{P}(T,H)$ is the joint probability of T and H, $\mathrm{P}(T|H)$ is the probability of a set T given a set H, $\mathrm{P}(H|T)$ is the probability of a set H given a set T, $\mathrm{P}(T)$ is the unconditional probability for a set T and $\mathrm{P}(H)$ is the unconditional probability for a set H.

Track finding is implemented by searching for the maxima in the conditional probability distribution $\mathrm{P}(T|H)$ with some kind of peak finding algorithm. The maxima correspond to the set of tracks T being most likely responsible for the observed set of hits H. This is a very general formulation of the track finding problem. In particular, it is independent of the concrete track- and hit-model. The algorithmic details reside in the concrete parametrization of the set of tracks T, the set of hits H, the modeling of the joint conditional probability $\mathrm{P}(H|T)$, the model for the a priori probabilities $\mathrm{P}(H)$ and $\mathrm{P}(T)$, and in the peak finding algorithm.

Looking at the relation of each single hit h to each single track t, $\mathrm{P}(t|h)$ is the probability distribution of each track t given a hit h:

$$\mathrm{P}(t|h) = \frac{\mathrm{P}(h|t) \times \mathrm{P}(t)}{\mathrm{P}(h)} \tag{5.17}$$

where $\mathrm{P}(h|t)$ is the probability distribution of single hits h for a given single track t.

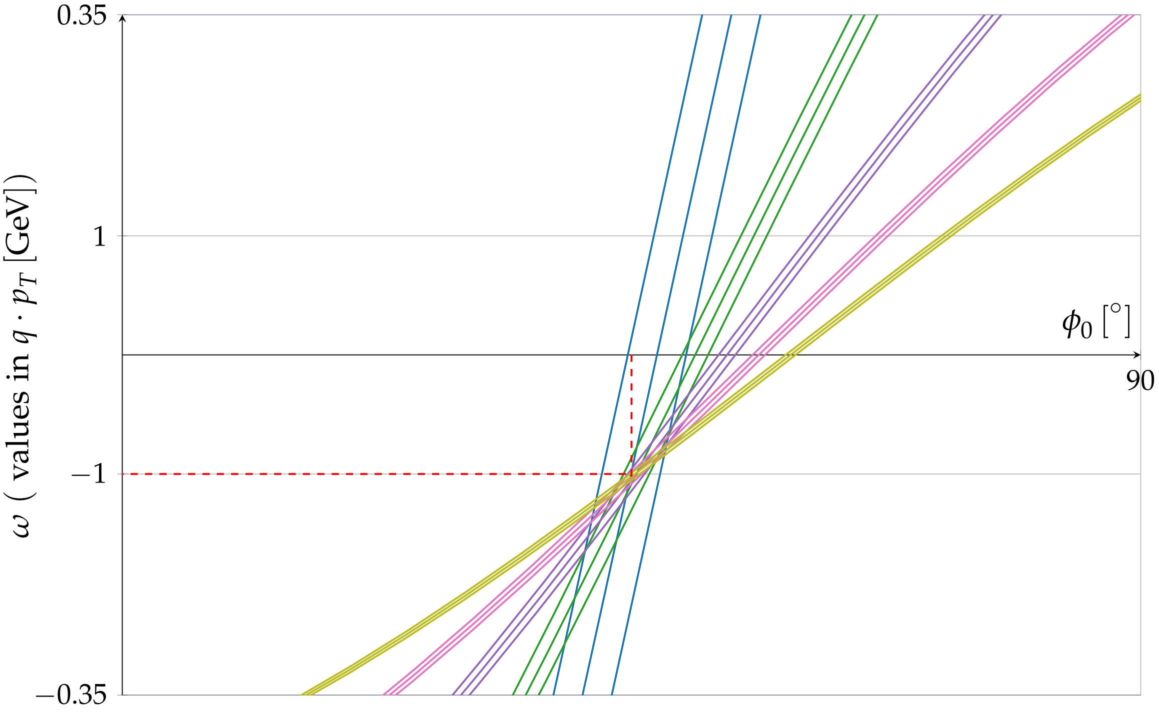


Figure 5.8.: Zoomed into the crossing point of the 2D hit curves in the track parameter space of the same track previously shown in Fig. 5.5. The band around each hit curve illustrates the shift created by the drift time. It is created by shifting the hit position in each SL by a fixed length of 8 mm to the left or to the right, while keeping the radius constant. In order to illustrate the widening of the hit curve width in the Hough space, the same fixed length is used as offset in all SLs. For a correct modeling of the drift time, the probability distribution of the drift times in each SL would be required. The most likely drift times in each SL are shown in Fig. 3.9.

In a binned track parameter space with a discrete number of tracks t, the probability distributions can be approximated as n-dimensional arrays. Furthermore, the arrays approximating $\mathrm{P}(h|t)$, $\mathrm{P}(t)$ and $\mathrm{P}(h)$ can be trained from a training data set. Alternatively, they could be calculated using the explicit analytic model discussed in Sec. 5.1.2 and Sec. 5.1.3. However, in the analytic solution, the width of the hit curves in the track parameter space has to be modeled additionally.

The most important ingredient to this thickness of the hit curves is the drift time. Further effects contribute as smaller corrections. This includes field inhomogeneities, detector inefficiencies and scattering effects of the particles with the detector material. The dependence of the hit curve widths on the drift time is illustrated in Fig. 5.8. Here a fixed distance of 8 mm to the left and to the right of the signal wires is used. This fixed distance is a simplified illustration of the band containing the drift time distribution around the central wire curve. The most important effect seen in this simplified model is the increase of the width of the drift time bands towards the inner SLs. Correctly, the fixed shift of 8 mm should be adjusted according to the average size of the drift cells in each SL. Hence, the effect is slightly overdrawn for illustrative purposes in Fig. 5.8. The largest correction that could be applied to Fig. 5.8 would be found in the innermost SL, where the used small cell chamber results in smaller averaged drift times (see Fig. 3.9). A further improvement to a fixed offset would be the use of a probability distribution for the most likely drift times.

The huge advantage in using a ML approach here, is that explicit modeling is not required. Instead, it is included automatically by training the hit representations in a predefined track parameter space. This is also a distinctive feature to the present 2D finder, where the modeling of the width of the Hough curves due to the drift times is ignored. Instead, all discretized fields crossed by the analytically described central curve are used as a representation of the hit curve.

The a priori probability distributions $\mathrm{P}(t)$ and $\mathrm{P}(h)$ denote the unconditional probabilities for tracks and hits to occur. If required, these distributions can be optimized in favor of specific physics event signatures combined with the geometry of the detector acceptance, i.e. some track parameter ranges can be made more likely to be found. From the topology of the track parameter distribution in the expected physics final states, the track parameter distribution $\mathrm{P}(t)$ can be extracted. For example a peak in p_T around 500 MeV in $\Upsilon(4S)$ events or a boost in the forward and backward region in ISR events with the hadronic final state $\pi^+\pi^-$. Furthermore, in combination with the geometry of the detector the a priori hit probability $\mathrm{P}(h)$ can be optimized, where fore instance the wires in the inner layers are more likely to be hit.

Generally, an event contains n tracks t_i, with $t_i \in T, i \in [1,..,n]$ and m hits h_j with $h_j \in H, j \in [1,...,m]$. Given the set of hits H in an event, the joint probability for each possible track in the binned track parameter space is:

$$\mathrm{P}(t_i|h_1,h_2,...,h_m) = \frac{\mathrm{P}(h_1,h_2,...,h_m,t_i)}{P(h_1,h_2,...,h_m)}. \tag{5.18}$$

This joint probability can be factorized into the single hit contributions by

$$\prod_{j=1}^{m} \mathrm{P}(t_i|h_j) = \prod_{j=1}^{m} \frac{\mathrm{P}(h_j,t_i)}{\mathrm{P}(h_j)}. \tag{5.19}$$

The proposed method of joining the $\mathrm{P}(t_i|h_j)$, namely the individual hit probability distributions in the track parameter spaces, is motivated by the maximum likelihood estimation. The calculation of the product is replaced by the calculation of a sum of $\mathrm{P}(t_i|h_j)$ for all hits h_j in the event. The result is a weight distribution $w(t_i)$ for each possible track t_i:

$$w(t_i) = \sum_{h_j \in H} \mathrm{P}(t_i|h_j) \tag{5.20}$$

where the function $w(t_i)$ cannot be interpreted as a probability distribution. However, it is still suited to extract the most probable tracks by a peak finding in the track weight space. A useful interpretation of the single hit contributions is the logarithm of a probability. The logarithm of a product of probabilities can be disassembled into the sum of the logarithms of the probabilities:

$$\log(\mathrm{P}(t|h)) = \log\left(\prod_{h \in H} \mathrm{P}(t|h)\right) = \sum_{h \in H} \log\left(\mathrm{P}(t|h)\right). \tag{5.21}$$

Therefore, the best analytic interpretation for $w(t_i)$ is the log likelihood of a single track.

Here the idea is, to approximate the weight distributions by an array $D(t|h)$ with the three track parameter dimensions ω, ϕ_0, θ and additionally the hit dimensions required to store such a 3D track representation for each hit. All single hit representations $D(t_i|h_j)$ in the track parameter space are trained

$$D(t_i|h_j) \propto \mathrm{P}(t_i|h_j) = \frac{\mathrm{P}(h_j|t_i) \times \mathrm{P}(t_i)}{\mathrm{P}(h_j)} \tag{5.22}$$

where $D(t_i|h_j)$ is a 5-dimensional histogram (weight matrix) with $\dim(t_i) = 3$ and $\dim(h_j) = 2$. Tracks are parametrized by the three track parameters $\mathrm{P}(t_i) = \mathrm{P}(\omega, \phi, \theta)$ and hits by two parameters uniquely identifying a single reference wire within the TSs: $\mathrm{P}(h_j) = \mathrm{P}(\texttt{TS-id}, \texttt{prio})$.

Without an explicit model of the background and signal distribution in the final experiment, $\mathrm{P}(t_i)$ and $\mathrm{P}(h_j)$ cannot be determined a priori and are therefore assumed to be constant. In the training data set, this is supported by using uniform distributions in each of the track parameters used for the binning of the track parameter space. In order to generate equal probabilities for all tracks, the bins $D(t_i)$ are normalized (see Sec. 5.2.4). Assuming the independence of the single hit and single track probabilities, the histogram for the full Hough space can be constructed by summing up the single hit contributions to each track bin.

$$D(t_i|H) = \sum_{j=1}^{m} D(t_i|h_j) \tag{5.23}$$

After summation, $D(T|H)$ with elements $D(t_i|H)$ for each track $t_i \in T$, is a 3-dimensional equivalent to a track parameter Hough plane. In the software implementation carried out for this thesis, a new 3D binned clustering algorithm was developed, which is based on the density based DBSCAN [85] algorithm. The method of clustering and peak finding in the Hough space $D(T|H)$ is described in Sec. 5.2.5.

parameter	default value	description
`binvars`	ω, ϕ, θ	parameters to be estimated
`phigeo`	32	number of symmetrical ϕ-sectors
`nbins`	$40, 12, 9$	number of bins in each parameter (per ϕ-sector)
`datarange`		trained parameter ranges
`neighbors`	7	allowed extension (in ϕ-sectors) for a single track
`bitwidth`	3	bits to describe single hit values in parameter space
`TS-ids`	$73 = (41 + 32)$	number of TS per ϕ-sector (axial + stereo)
`prio`	3	number of priority wire positions within a TS

Table 5.1.: Parameters defining the data structure of the track-to-hit relations, which is used by the 3D finder to look-up all the single hit contributions to the 3D Hough space. The shown default values of `nbins`, `neighbors` and `bitwidth` correspond to the settings in the 3D finder studies published in [84].

5.2.2. Structure of the Hough Space

In order to construct a Bayesian equivalent of a binned Hough space with relations between hits and tracks, each track parameter is limited to a defined range and discretized into bins. The hits are already discrete and can directly be used as a discrete index of an n-dimensional array holding the hit representations in this binned 3D track parameter space for each hit.

In this section, the data structure for the hit representations is defined. A major achievement is the ϕ-symmetric representation of the hits, which connects the physical symmetry of the tracks in ϕ with the geometrical symmetry of the CDC. The training of the values within the data structure is presented in Sec. 5.2.4. As an overview of the setup, Tab. 5.1 shows the parameters used to define the data structure representing the track-to-hit relations. These parameters will be explained subsequently.

The track parameter space has an axis for each variable in `binvars`. Each of the variables `binvars` are expected to have values in the defined range `datarange` which is binned into `nbins` bins. The `datarange` used in the 3D finder describes tracks with $\left|\omega^{-1}\right| \propto p_\mathrm{T} \geq 0.2\,\mathrm{GeV}$ and with polar angles $19^\circ \leq \theta \leq 140^\circ$. These track parameter limits are discussed in Sec. 5.3.1. The parameter `phigeo` describes the number of symmetrical CDC sectors in ϕ. In Belle II, the wire pattern repeats every 11.25°. Therefore, the full CDC consists of `phigeo` $= 32$ symmetrical sectors ($32 \cdot 11.25^\circ = 360^\circ$). This enables a massive reduction of the memory required to store the hit patterns, which is beneficial for the present software as well as for the future hardware implementation. Practically, a new axis is added to the data structure to describe the ϕ-modulo sectors. This is not an absolute ϕ-value but a distance of the ϕ-value of a track to the ϕ-position of the hit. The size of this additional axis is described by the parameter `neighbors`; it is the number of neighboring ϕ-modulo sectors that are allowed to be related to a single track. This symmetry of the CDC geometry and the detailed meaning of the parameters is described in Sec. 5.2.3. In order to prepare the algorithm for a hardware implementation, the parameter `bitwidth` is introduced. As explained in Sec. 5.2.4, it limits the maximum weight each hit curve can maximally contribute to each possible track in the Hough space.

The hypothetical n-dimensional array D for the full phase space, which can be viewed as an n-dimensional array of weights, is quite large. It holds a representation of each hit in the binned 3D track parameter space. With a total number of 2336 TS-ids and $\texttt{prio} = 3$ possible priority positions within each TS, it has $\dim(D) = 5$ with $O(10^9)$ cells in total

$$\text{cells} = \prod_{v \in \{\omega, \phi, \theta\}} \texttt{nbins}_v. \tag{5.24}$$

However, the required memory to store the track-to-hit relation can be reduced significantly by using the geometrical symmetry of the CDC. As memory resources are limited in hardware as well as in software, the reduction of the required memory is important for an efficient implementation of the new algorithm.

At first, redundant storage of axial hit information in several θ bins is avoided by splitting D into two separate arrays: the array A holds the axial hit representations in the Hough space and S holds the stereo hit representations in the Hough space. With $\dim(\textit{tracks}) = 3$ and $\dim(\textit{hits}) = 2$, S is a 5D histogram in the global picture of the full detector acceptance region. In the case of axial hits in the histogram A, $\dim(\textit{tracks}) = 2$, so a 4D histogram is sufficient. The further reduction of both arrays using the ϕ-symmetry is explained in Sec. 5.2.3.

Digression on the Hough Space Binning

The following criteria are used for the determination of the default values shown in Tab. 5.1: The number of $\texttt{nbins}_\omega = 40$ is chosen slightly larger than the number of bins used by the existing 2D finder [67] of $\texttt{nbins}_\omega = 34$ (see Sec. 3.3.4). With $\texttt{nbins}_\phi = 12$ per ϕ-sector, the number of bins in the full ϕ-region equals the number of TSs in the outermost SL of 384 (see Tab. 5.2). In contrast, the present 2D finder uses the number of TSs in the innermost SL of 160 for the ϕ-binning of the Hough-plane. If drift times are ignored, the useful maximum granularity of the Hough space can be derived from the wire spacing in the CDC. In this context useful means that improvements in the track finding efficiency and in the resolution of the estimated track parameters can be expected. Hence, the value of $\texttt{nbins}_\phi = 384$ used for the 3D finder corresponds to the upper limit in ϕ. The related upper limit for a useful number of bins in ω corresponds to the drift cell spacing in the outermost SL for a maximally curved track that still creates TSs.

This results in a coverage of one quadrant of the CDC: under the assumption of equal side lengths of each drift cell in the TSs, the limiting angle for the TSs acceptance can be calculated. The crossing angles in a TS are illustrated in Fig. 5.9.

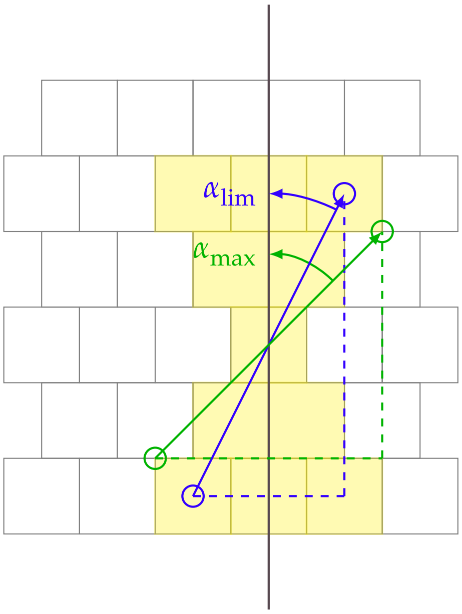


Figure 5.9.: Illustration of the geometrical upper limit for the track angle with the TS. The maximum crossing angle where a TS can be found is $\alpha_{\mathrm{max}} = 45^\circ$, shown in green. At the limiting crossing angle $\alpha_{\mathrm{lim}} = 26.6^\circ$ the TSF starts to lose efficiency as a filled TS pattern becomes less likely. The angles are calculated under the assumption of equal spacing between the drift cells. Note that a track with α_{max} in the outermost SL has $p_\mathrm{T} \approx 350\,\mathrm{MeV}$, a track with α_{lim} has $p_\mathrm{T} \approx 550\,\mathrm{MeV}$ (using Eq. 3.9).

By connecting the center of the lower left drift cell with the upper right drift cell within a TS, the limiting crossing angle for the TSF efficiency is $\alpha_{\mathrm{lim}} = \arctan(\frac{1}{2} \approx 26.6^\circ)$ and the maximum crossing angle is $\alpha_{\mathrm{max}} = \arctan(1) = 45^\circ$. At $\alpha \geq \alpha_{\mathrm{lim}}$ the efficiency of the TSF decreases until no more TSs are found at $\alpha \geq \alpha_{\mathrm{max}}$. The corresponding momenta can be calculated via the radius of the track with Eq. 6.24 and Eq. 3.9 by using the radius of the outermost SL in Tab. 5.2 of $r_\mathrm{L} \approx 1.06\,\mathrm{m}$ (see Fig. 6.6 for an illustration of the geometry):

$$\omega = \frac{2\sin(\alpha)}{r_\mathrm{L}} \qquad\qquad p_\mathrm{T} = \frac{0.3 \cdot 1.5}{\omega}. \tag{5.25}$$

By taking $\alpha_{\mathrm{max}} = 45^\circ$, depending on the charge a maximum angle of 90° has to be covered, which corresponds to one quadrant of the CDC. This means the upper limit where improvements can be expected is $384/4 = 96$. If instead the number of wires in the innermost SL is taken, the number of bins becomes $160/4 = 40$. Note that these are simply upper limits.

Since for each bin in θ a replica of the full 2D Hough space is created, a small number of bins is chosen in θ. In this way, the required memory resources for the 3D finder hardware can be estimated as $\texttt{nbins}_\theta$ multiplied with the 2D finder memory resources. An estimate for the upper limit of the number nbins-θ can be obtained considering the stereo wire offsets in Tab. 3.3. The offsets of 18 to 20 holes at the endplate correspond to an offset of 9 to 10 signal wires with respect to an axial wire configuration parallel to the z-axis. Hence, the value of $\texttt{nbins} - \theta = 9$ is selected for the studies presented in this thesis. Preliminary performance studies of this Hough space setup with nine bins in θ have been published in [84]. Earlier studies have shown [83] that reasonable results can also be achieved in a smaller setup using only six bins in θ.

However, these Hough space sizes are somewhat arbitrary, as they do not affect the underlying logic of the track finding algorithm. In the end, the most important ingredients for their final determination are the available resources on the hardware platform, where the algorithm is going to be implemented. Therefore, the size of the Hough space has to be reconsidered during the future hardware implementation and a tradeoff with the achievable precision has to be made. This precision comprises the resolution of the estimated track parameters, the track finding efficiency and the rate of fake tracks. Since this precision is interdependent with the used clustering algorithm, it can only be measured correctly after the hardware clustering algorithm has been decided. The density based clustering algorithm developed in the software implementation of the 3D finder is presented in Sec. 5.2.5. For example, using twice the number of bins in ϕ means to increase the memory requirements by a factor of two. With the parameters in Tab. 5.1, the size of the Hough space is approximately 18 times larger than the 2D Hough plane of the present 2D finder (two times larger ϕ-size and additional nine θ-bins).

5.2.3. CDC Symmetric Hit Representations

In order to reduce the number of cells required to store the track-to-hit relations, the symmetry of the CDC in ϕ is used: the wire pattern repeats 32 times in ϕ (parameter phigeo in Tab. 5.1). Therefore, it is sufficient to describe a single ϕ-sector of the detector with a width $\Delta\phi = 360^\circ/32 = 11.25^\circ$. In the following, these reduced 11.25° ϕ-sectors will be called ϕ-modulo sectors throughout this Chapter. As shown in Tab. 5.1, with the total number of 2336 TSs in the full detector, a ϕ-modulo sector only contains $\texttt{TS-ids} = 2336/32 = 73$. This usage of the ϕ-symmetry is a major achievement in the 3D finder development. Although it is not required by the underlying theoretical concept, it allows an efficient implementation within the available memory resources of the present software implementation and the future FPGA hardware implementation. In this sense, it can be compared to the memory reduction realized by the neural network trigger where the ϕ-symmetry is used by adopting the preprocessing of the least squares fit (see Sec. 6.1.4). However, the method described here is a discrete approach for the usage of the ϕ-symmetry.

Table 5.2 shows the total number of TSs per SL in the full ϕ-range of the detector and the number of TSs in the reduced ϕ-modulo sectors. The third row in Tab. 5.2

SL	0	1	2	3	4	5	6	7	8
orientation	A	U	A	V	A	U	A	V	A
TS /360°	160	160	192	224	256	288	320	352	384
TS /11.25°	5	5	6	7	8	9	10	11	12
r [cm] / 1^{st} `prio`	19.8	29.3	40.2	51.3	62.0	73.1	83.8	95.0	105.7
r [cm] / 2^{nd} `prio`	20.8	31.2	42.0	53.1	63.8	74.9	85.7	96.8	107.5

Table 5.2.: Number of TSs per SL [21]. The number of sense wires per layer within a SL is constant and equal to the number of TSs per SL. Since the greatest common divisor (gcd) of the number of TSs per SL is 32, the pattern of wires in the CDC repeats every 11.25°. This symmetry is used to reduce the number of cells required for the storage of the hit representations in the 3D Hough space.

`TS-ids-axial`	41
`TS-ids-stereo`	32
`hits-axial`	123
`hits-stereo`	96
`tracks-axial`	480
`tracks-stereo`	2880

Table 5.3.: Summary of the number of hits and the number of tracks in the reduced ϕ-modulo sectors. Two separate arrays are created. Array A stores the axial hit representations in the Hough space. Array S stores the stereo hit representations.

shows that the greatest common divisor (gcd) of the wire numbers in all SLs is 32. Summing up the reduced number of TSs per SL, shown in the third row in Tab. 5.2, for all 9 SLs in a ϕ-modulo sector gives the total number of TSs in the reduced ϕ-modulo sector of 73. The number of TSs in the reduced sector can be decomposed into an axial part of `TS-ids-axial` = 41 TSs represented in the array A and a stereo part of `TS-ids-stereo` = 32 TSs represented in the array S. For the axial part, a partial sum has to be taken over the five axial SLs, corresponding to the columns with orientation A in Tab. 5.2. For the stereos, the partial sum has to be taken over the 4 stereo SLs, corresponding to the columns with orientation U or V in Tab. 5.2.

With the reduced number of TSs in each ϕ-modulo sector, the minimum number of cells in the array of the hit spaces are:

$$\texttt{hits-axial} = \texttt{TS-ids-axial} \cdot \texttt{prio} = 41 \cdot 3 = 123 \tag{5.26}$$

$$\texttt{hits-stereo} = \texttt{TS-ids-stereo} \cdot \texttt{prio} = 32 \cdot 3 = 96 \tag{5.27}$$

where `hits-axial` denotes the total number of axial hits in the ϕ-modulo sector and

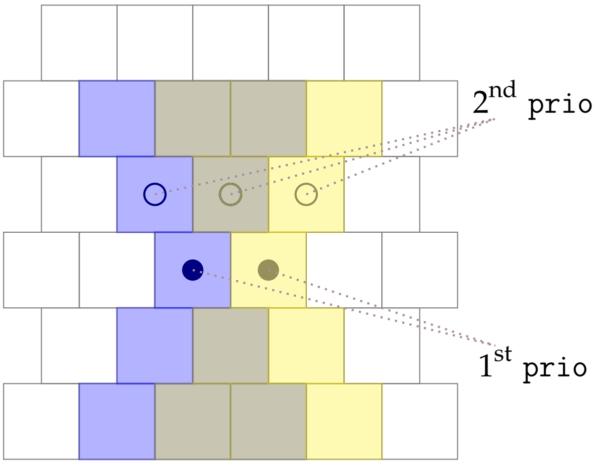


Figure 5.10.: Illustration of the 2nd priority wire position identity in neighboring TSs. The 2nd `prio` positions of the overlapping two TSs are shown as non-filled circles. The 2nd right `prio` position of the left TS shown in blue is identical to the 2nd left `prio` position of its right neighboring TS shown in yellow. Since there is no overlap with the neighboring TSs, the 1st `prio` positions, shown as filled circle in the centers of the TSs, are unique.

`hits-stereo` the total number of stereo hits. The number of hits and the number of tracks are summarized in Tab. 5.3.

A further reduction of the hit space is possible for the 2nd priority hits. The wire positions of 2nd left are identical to 2nd right, shifted by one cell. This overlap with the neighboring TSs in the 2nd `prio` positions is illustrated in Fig. 5.10. Thus, the hit curve of (`TS-id`, 2nd left) can be replaced by (`TS-id` $+ 1$, 2nd right) and only two distinct values are required for the priority.

In the track parameter space, the number of bins in the global ϕ-region is chosen identical to the number of TSs in the outermost SL (see Sec. 5.2.2, $\texttt{nbins-global}_\phi = 384$). Therefore, the number of required bins for the track-ϕ within a ϕ-modulo sector reduces to only $\texttt{nbins}_\phi = 12$. The total numbers of cells in the full 3D track space and in the reduced 2D track space are:

$$\texttt{tracks-stereo} = \prod_{v \in \{p_T^{-1}, \phi, \theta\}} \texttt{nbins}_v = 40 \cdot 12 \cdot 9 = 4320 \tag{5.28}$$

$$\texttt{tracks-axial} = \prod_{v \in \{p_T^{-1}, \phi\}} \texttt{nbins}_v = 40 \cdot 12 = 480 \tag{5.29}$$

Following the above symmetry considerations, both of the reduced arrays A and S have only $O(10^5)$ cells which is well suited for an efficient implementation in software as well as in hardware.

Local ϕ Sectors

Due to the ϕ-reduced data structure, the ϕ-positions of the TS hits and of the tracks need to be transformed from the global Belle II coordinate system with $\phi \in [0, 360]\,^\circ$ to a local one with $\phi_{\mathrm{rel}} \in [0, 11.25]\,^\circ$. For the tracks, this is done by calculating the relative ϕ-position ϕ_{rel} as the modulo of ϕ with 11.25°. Additionally, the parameter ϕ_{mod} identifies which of the 32 ϕ-modulo sectors contains ϕ_{rel}. These parameters are calculated by:

$$\phi_{\mathrm{rel}} = \phi \mod 11.25 \tag{5.30}$$

$$\phi_{\mathrm{mod}} = \left\lfloor \frac{\phi}{11.25} \right\rfloor \tag{5.31}$$

where $\lfloor \rfloor$ denotes the floor function and $\mod$ is the modulo function. The parameter ϕ_{mod} is an integer value with the range $\phi_{\mathrm{mod}} \in [1, .., 32]$ and ϕ_{rel} is a float value with the range $\phi_{\mathrm{rel}} \in [0, 11.25]\,^\circ$. Hence, a binning on the ϕ_{rel} of the tracks needs to be applied to obtain an integer value. In summary, the ϕ-values of the tracks transform like:

$$\phi_{\mathrm{track}} \rightarrow (\phi_{\mathrm{rel}}, \phi_{\mathrm{mod}}) \tag{5.32}$$

$$\phi_{\mathrm{rel}} \rightarrow \phi_{\mathrm{bin}}. \tag{5.33}$$

This method is used to transform global track parameters to ϕ-sectorized local track parameters. Additionally, the other two track parameters, ω and θ need to be binned as well in order to identify a single cell in the 3D Hough space ($\mathrm{track}_{\mathrm{bin}} = (\omega, \phi, \theta)_{\mathrm{bin}}$).

In a similar fashion, the axial and stereo TS hits need to be transformed from their global values to ϕ-modulo sector values. Here it is important to note that TS-ids are arranged circular in the transverse plane. Therefore, the TS-ids within the SLs correspond to ϕ-positions of the TSs. The number of TSs per SL are listed in Tab. 5.2 as "TS /360°".

The calculation starts with a global TS-id $\mathrm{TS} - \mathrm{global} \in [1, 2336]$ and the SL of the TS. Note that 2336 is the sum over the TSs in the nine SLs

$$2336 = \sum_{i=0}^{8} N_i \tag{5.34}$$

where N_i is the number of the TSs per SL i ("TS / 360$''$ in Tab. 5.2). Using the values of the row "TS / 360°" in Tab. 5.2, the TS-id per SL ($\mathrm{TS} - \mathrm{SL} \in [1, N_{\mathrm{SL}}]$) is calculated by subtracting the sum of all TSs in smaller SLs:

$$\mathrm{TS} - \mathrm{SL} = \mathrm{TS} - \mathrm{global} - \sum_{i=0}^{SL-1} N_i. \tag{5.35}$$

The TS-id per SL, $\mathrm{TS} - \mathrm{SL}$ is already proportional to a ϕ-position of the TS, By using the number of TSs per SL per ϕ-modulo-sector, which is listed as "TS / 11.25°"

in Tab. 5.2, the relative TS-id per SL per ϕ-modulo-sector can be calculated. This relative TS-id per SL per ϕ-modulo-sector is called ts_{rel}.

$$ts_{\text{rel}} = ts \mod M_i \tag{5.36}$$

$$ts_{\text{mod}} = \left\lfloor \frac{ts}{M_i} \right\rfloor \tag{5.37}$$

where M_i is the number of TSs in the reduced sector for the modulo calculation shown in the third row in Tab. 5.2 as "TS / 11.25° and ts_{rel} is the relative TS-id per SL per ϕ-modulo-sector. Additionally, $ts_{\text{mod}} \in [1, 32]$ describes which of the 32 ϕ-modulo-sectors the ts_{rel} is contained in.

In order to describe a single hit-cell in one of the arrays A or S, the integrated number of TSs in the smaller SLs with the same orientation is added:

$$ts_{\text{bin}} = ts_{\text{rel}} + \sum_{i \in \text{same orientation}}^{SL-1} M_i \tag{5.38}$$

where the sum over the same orientation means to take either the axial or the stereo oriented columns in the third row of Tab. 5.2. The TS-ID ts_{bin} combined with the priority id of the TS is used as an index in the arrays A or S in order to select a single hit representation in the 3D track parameter Hough space. In summary, the TS-ID transformation can be noted by

$$\text{TS} - \text{global} \rightarrow (ts_{\text{rel}}, ts_{\text{mod}}), \tag{5.39}$$

$$ts_{\text{rel}} \rightarrow ts_{\text{bin}}. \tag{5.40}$$

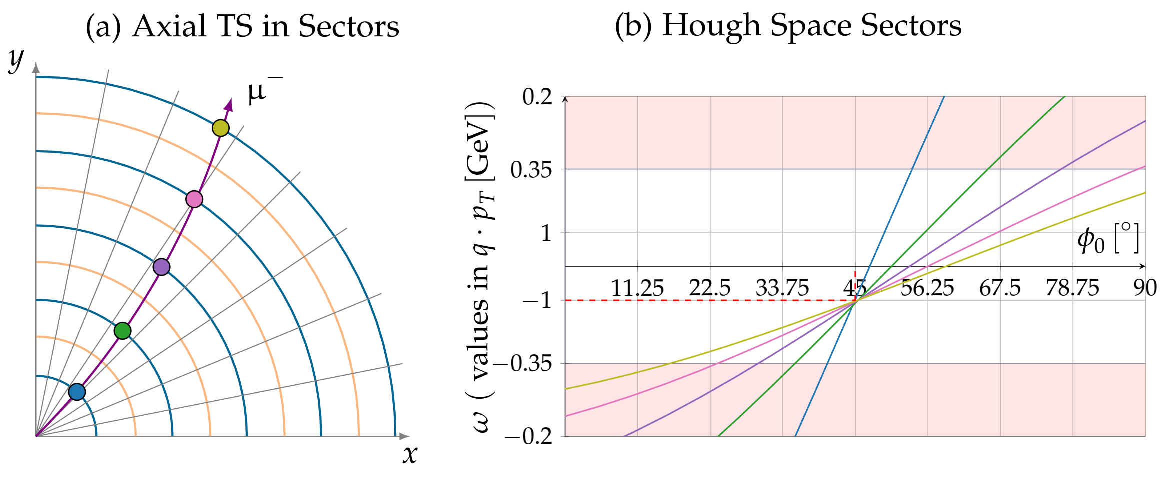
Neighboring ϕ Sectors

So far, the track-to-hit relations have been described within a ϕ-modulo sector with the limited width of 11.25°. However, curved tracks can extend to neighboring ϕ-modulo sectors, as illustrated in Fig. 5.11 (a). A hit in a specific ϕ-modulo sector ts_{mod} does not necessarily belong to a track within the same ϕ-modulo sector ϕ_{mod}. This situation allows for two different, but equally valid interpretations.

The first interpretation takes the perspective of the track as shown in Fig. 5.11 (a). In this interpretation, the hit space needs to be extended to cover several neighboring ϕ-modulo sectors in order to describe tracks within a single ϕ-modulo sector. The second interpretation is the perspective of the hit in the Hough space as shown in Fig. 5.11 (b). Here a single hit can be created by tracks from several neighboring 11.25° ϕ-modulo sectors. The reason for the two different interpretations is that the tracks as well as the hits are described by a position in ϕ. What actually matters is the ϕ-difference between the track and the hit.

Using the above notation for the track sectors ϕ_{mod} and for the hit sectors ts_{mod} this difference can be expressed as

$$d_{\text{mod}} = ts_{\text{mod}} - \phi_{\text{mod}}. \tag{5.41}$$


(a) Axial TS in Sectors

(b) Hough Space Sectors

Figure 5.11.: New sectorized ϕ illustration in the transverse plane. Illustration of curved tracks reaching into neighboring sectors.

The solution is to check the hit-to-track relations in the neighboring ϕ-modulo sectors as well. To this end, several neighboring ϕ-modulo sectors are stored in the histograms A and S. This means an additional axis is added to the data structure which is called `neighbors` in the following. The cell in this `neighbors` axis is the difference d_{mod}. The advantage in using the ϕ-difference here is a significant reduction in the size of the stored hit patterns. Since `neighbors` is an extension of the ϕ-axis, it can be interpreted as either a relative track-ϕ or as a relative hit-ϕ.

This increases the dimensionality of the arrays A and S to $\dim(A) = 5$ and $\dim(S) = 6$. Hence, the optimized data structure used for the track-to-hit relations is 5D for axial hits and 6D for stereo hits. Note that this axis only affects the underlying data structure and no additional axis is added to the resulting Hough space. During the track finding, this additional ϕ-axis will be resolved to create a single absolute ϕ-axis, similar to Fig. 5.11 (b), in the full 360° Hough space (see Sec. 5.2.5).

In the presented setup, seven neighboring ϕ-modulo sectors are used. These correspond to a total ϕ-coverage of $7 \times 11.25^{\circ}$. An additional advantage of the `neighbors` axis is that the ambiguity of the two intersection points of the hit representations in the Hough space is automatically resolved. As shown in Sec. 5.1.2, the sinusoidal hit curves had two intersection points corresponding to an outwards going track and to a track curling back. Using the limitation of the hits to a region of $\Delta\phi = 7 \cdot 11.25 = 78.75$, only outwards going tracks are stored in the hit representations.

Furthermore, the value of d_{mod} corresponds to the curvature of the track. For example, a straight track from the IP will always have $d_{\text{mod}} = 0$, whereas curved tracks will usually cause hits in neighboring ϕ-modulo sectors. Using the value `neighbors` $= 7$, a curved track can reach up to 3 neighboring ϕ-modulo sectors to

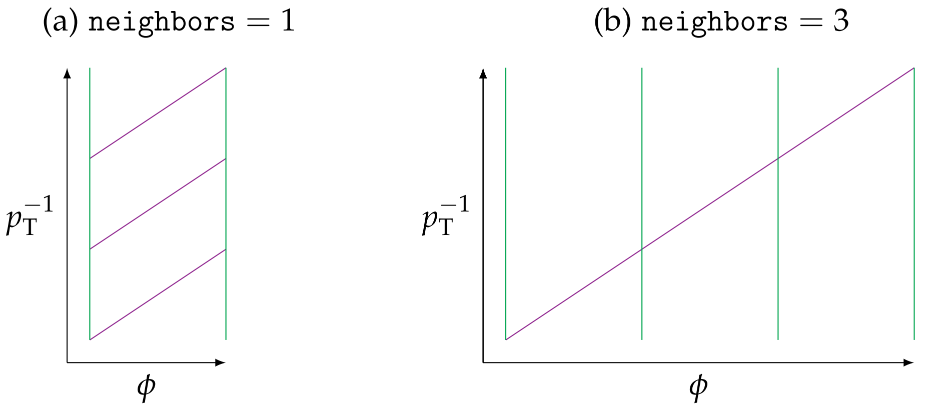


Figure 5.12.: Schematic representation of a hit curve in a 2D track parameter space with different sizes of the `neighbors`-axis. The shown hit is reachable from tracks in three neighboring ϕ-modulo sectors. With the single bin along the `neighbors`-axis in (a), the assignment of track parameters to the hit is no longer unique. By adding a sufficient number of neighboring ϕ-modulo sectors, the track-to-hit relation becomes unique. In the illustration in (b), the ambiguity is resolved for the case `neighbors` $= 3$.

either side which corresponds to a ϕ-coverage of $3 \cdot 11.25 = 33.75^\circ$. Including the ϕ-modulo sector of the track, the maximal coverage becomes $4 \cdot 11.25 = 45^\circ$. Since the efficiency of the TSF goes to 0 for a crossing angle of 45° (see Fig. 5.9), covering three neighboring sectors was chosen as a useful tradeoff. By alternatively setting `neighbors` $= 9$, a full coverage even for the rare case of TSs with large crossing angles can be realized.

The value of `neighbors` $= 7$ may be further optimized in future upgrades of the 3D track finding algorithm. The problem when using too few neighboring ϕ-modulo sectors is illustrated in Fig. 5.12 (a) for `neighbor` $= 1$. Since the single hit curve has a larger extension in ϕ than the ϕ-modulo sector width of 11.25°, the hit pattern in the Hough space element is continued several times in different p_T^{-1}-regions (compare Fig. 5.11 (b)). By constructing the full 360° Hough space, the artifacts of the hit curves would cause multiple ghost lines and pose severe difficulties for the clustering algorithm.

However, introducing an elaborated Hough plane construction algorithm or peak finding algorithm might allow to reduce the number of neighboring ϕ-modulo sectors up to the single ϕ-modulo sector illustrated in Fig. 5.12 (a). For example, the hit pattern in Fig. 5.12 (a) could be cut into p_T^{-1} slices, which are partially selected upon the construction of the full Hough space. Considering the acceptance region of the TSF, the problem of these hit artifacts is sufficiently avoided by setting `neighbors` $= 7$.

track parameter	p_{T} [GeV]	ϕ [°]	θ [°]
range	$[0.2, 1000]$	$[-180, 180]$	$[19, 140]$

Table 5.4.: Track parameter ranges of the $\mu^{\pm}$ tracks in the training data set. Within these ranges, the simulated tracks are generated with a uniform distribution in the inverse transverse momentum p_{T}^{-1}, in ϕ, and in θ.

5.2.4. Training of the Hit Representations

The weight values in the previously described data structure represent the binned hit curves in the 3D Hough space. These weight values, in all the track parameter bins for each hit, are found by supervised training. After a short introduction to the creation of the used MC training data sets, the methods to obtain the hit representations in the track parameter space are presented. Note that in future upgrades, a retraining using reconstructed track data from real Belle II events can be considered. Additionally, the `bitwidth` of the trained values in the data structure is discussed. Since in the hardware only limited computing and memory resources are available and the calculations are carried out using integer arithmetic, this `bitwidth` is an important parameter for the migration of the algorithm to hardware.

Generation of the Training Data

In the basic training data setup with MC training data, single muon track events are simulated for the training. The advantage of muons is that they produce rather "clean" tracks in the CDC. As leptons, they are point like charged elementary particles and they are close to the optimum of a minimum ionizing particle. Due to the about 200× larger mass, they are subject to less multiple scattering than electrons. With the muon lifetime in the order of $\mathcal{O}(\mu\mathrm{s})$, the muons are stable at the time scale seen by the Belle II detector. Future studies should elaborate the usefulness of advanced training data sets with different particle types, track parameter ranges or a training with real Belle II events.

All simulated $\mu^{\pm}$ track vertices are limited to the vicinity of the IP ($|z_0| \leq 1\,\mathrm{cm}$) and without transverse displacement ($d_0 = 0\,\mathrm{cm}$). Since the physics collisions occur at the IP, the track finder is required to have a high efficiency and resolution for tracks from the IP. Ignoring tracks displaced from the IP in z-direction leaves θ as the only 3D track parameter to be estimated. This is analog to ignoring the transverse displacement d_0, which is also done by the present 2D finder. By requiring the track vertices located at the IP, the number of track parameters is reduced from five (helix) to three ($p_{\mathrm{T}}, \phi_0, \theta$), which can be better handled by a 3D track finder. The track parameter ranges in the default training data set are listed in Tab. 5.4. The tracks at the geometrical acceptance region of the CDC in the low-p_{T} and shallow-θ region are discussed later in Sec. 5.3.1.

In principle, higher dimensional track finding can be implemented with the same

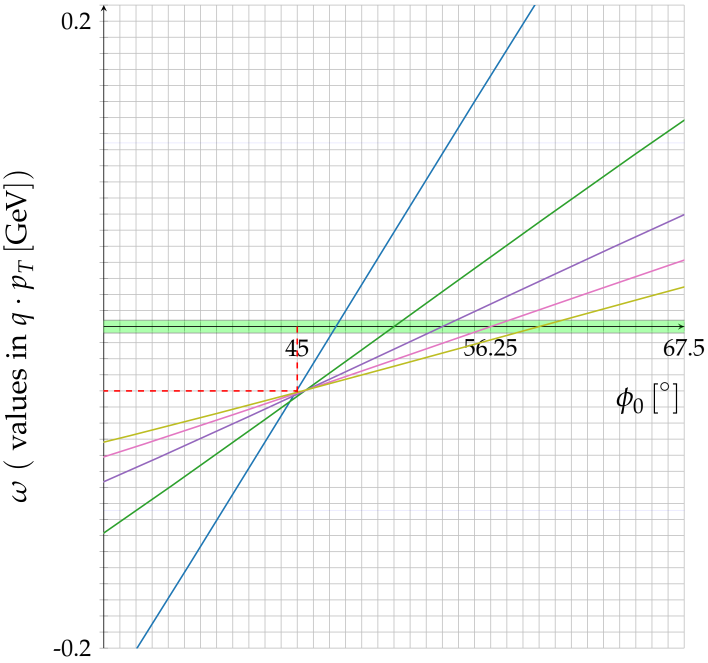


Figure 5.13.: The green band corresponds to the missing high-p_{T} region, if a maximum value of $p_{\mathrm{T}} = 10\,\mathrm{GeV}$ is used.

track finding algorithm as well. However, displaced tracks are background tracks. This means, a precise resolution in the track reconstruction of these displaced tracks is not required. Hence, a 3D Hough finder for tracks constrained to the IP is well suited for the Belle II track trigger.

As will be shown later in Sec. 5.3.1, the θ-range is limited by the geometrical acceptance of the CDC for IP tracks. Positively and negatively charged tracks with a high momentum appear as straight tracks. Thus, their charge cannot be determined from the track curvature. This corresponds to zero curvature tracks $\omega \propto q \cdot p_{\mathrm{T}}^{-1}$. In order to correctly patch the Hough-space in the $\omega \propto p_{\mathrm{T}}^{-1} \approx 0$ region, the upper limit of the p_{T}-range used in the training is set to an unrealistic large value of $p_{\mathrm{T}} = 1000\,\mathrm{GeV}$. As will be discussed subsequently, in this way the central bins along the $q \cdot p_{\mathrm{T}}^{-1}$-axis describing the high-p_{T} tracks are sufficiently represented in the training data set.

Digression on the High-p_{T} Hough Space Gap

With a minimum $p_{\mathrm{T}} = 0.2\,\mathrm{GeV}$ and 40 bins in $q \cdot p_{\mathrm{T}}^{-1}$ the bin-width in $q \cdot p_{\mathrm{T}}^{-1}$ is

$$\text{bin-width} = \frac{\dfrac{1}{0.2\,\mathrm{GeV}} - \dfrac{1}{-0.2\,\mathrm{GeV}}}{40} = 0.25\,\mathrm{GeV}^{-1} \tag{5.42}$$

If a realistic maximum of $p_\mathrm{T} \approx 10\,\mathrm{GeV}$ were used, the gap width would be

$$\text{gap-width}_{10\,\mathrm{GeV}} = \frac{1}{10\,\mathrm{GeV}} - \frac{1}{-10\,\mathrm{GeV}} = 0.2\,\mathrm{GeV}^{-1} \tag{5.43}$$

which corresponds to nearly 80 % of the width of a single bin. This gap in the high-p_T region is illustrated in Fig. 5.13.

Such a gap is expected to introduce problems for high momentum tracks, where in principle the estimation is quite easy. The reason is the infinity introduced by the normal parametrization of the straight line, when the distance ρ is interpreted as the radius of a 2D circle (see Sec. 5.1.2). By using the unrealistic maximum of e.g. $p_\mathrm{T} \approx 1000\,\mathrm{GeV}$, the gap width becomes negligible compared to the bin width (only 0.8 %), with a width of

$$\text{gap-width}_{1000\,\mathrm{GeV}} = \frac{1}{1000\,\mathrm{GeV}} - \frac{1}{-1000\,\mathrm{GeV}} = 0.002\,\mathrm{GeV}^{-1} \tag{5.44}$$

Due to the uniform generation in p_T^{-1}, the high momentum tracks only make up a very small fraction of the total number of events. Thus, no p_T bias towards unrealistic high momentum tracks is introduced. This can be seen by comparing the gap width with the bin width: only 80 % of $\frac{1}{40}$ of the tracks are generated with a $p_\mathrm{T} > 10\,\mathrm{GeV}$, which corresponds to 2 % of the tracks. Overall, the gap is not very big and the motivation to close it comes from continuity considerations of the binned ω, ϕ-space. It is not meant to be actually used for an accurate high-p_T track estimation. In favor of a more accurate physics simulation of the particle tracks created at the IP within the realistic Belle II momentum spectrum, future studies could investigate whether this correction towards a continuous high momentum patch region can be neglected.

However, a future re-introduction of the gap has to be considered carefully: if each hit curve has a gap in the high-p_T region, this might affect the efficiency and resolution of the track finder. Especially the cluster shapes for straight tracks might change. Therefore, it is recommended to close this gap in some way. For example, instead of continuously filling the high-p_T region, it could be cut out of the Hough space resulting in a non-continuous $q \cdot p_\mathrm{T}^{-1}$ axis. Alternatively, the high-p_T patch region could be extrapolated from the bin contents in the neighboring cells of the missing high-p_T cells.

Filling the Data Structures with Training Data

In order to train the hit representations in the track parameter space, the data structures A and S are treated as histograms at first. The histograms A and S are filled in a loop over the training events such that they represent the joint frequency of the track-to-hit relations. These training events have to contain tracks with relations to TS hits. Therefore, simulated MC particles as well as reconstructed tracks can be used as training target.

During this filling process, in each event the valid reference tracks are determined by checking whether the track parameters lie within the defined training range of the finder. This validity check is primarily implemented for the generality of the training method: it allows to use training data sets containing tracks outside of the defined training range and thus enables a reuse of the same training data sets for the training of different range setups of the 3D finder. Even in a MC data sample, where the simulated ranges in the data exactly match the defined range for the finder, some out of range tracks are created as secondary tracks after multiple scattering with the detector material. Although the primary particles are contained within the defined ranges, the secondary tracks actually creating the hits can be out of range. Note that with the simulated muon tracks used in the studies of this thesis, secondaries are rare such that nearly all the simulated tracks can be used as training target. Another source for out of range tracks appears if reconstructed tracks are used as training target. In this case the resolution of the offline reconstruction may distort the track parameters such that they are shifted outside of the defined training range (compare Fig. 5.23). Furthermore, these out of range tracks can be reconstructed tracks from real Belle II events.

In order to use only valid tracks, in the sense that they can be reconstructed based on the hit information in the detector, for MC tracks additionally a relation to a reconstructed track or to a 2D finder track is required. In the case of the reconstruction, the relations are created via the tracking system of Belle II. In the case of 2D finder tracks, a matching is used to obtain the MC track to 2D finder track relations. The used matching method is described in Appendix A.3. As will be discussed in Sec. 5.3.1, the reconstruction efficiency can be read off Fig. 5.22: without background $\approx 99.9\,\%$ of MC particles are reconstructed; with background $\approx 99.7\,\%$. The 2D finder only achieves a track finding efficiency of $\approx 52\,\%$ without background and at $\approx 54\,\%$ with background. A study comparing the different types of reference tracks is presented in Sec. 5.3.3.

The third condition for the tracks used in the training is the number of related TS hits. At this point, the bare limit where track finding is possible is used, corresponding to a minimum of three hits for a 3D finder. This can be seen by considering the hit representations in the Hough space. In the case of a 2D track finder at least two hits are required to find a single intersection point of two sinusoidal 2D curves. In the case of a 3D track finder, the hit representations are planes. Therefore, at least three planes are required to find a single intersection point. If a matching reference track is present, the number of TSs related to the reference track is counted. This means the number of TS related to the 2D finder track or to the reconstructed track. This condition avoids to use underrepresented tracks in the training. With the simulated single muon tracks presently used, it is not necessarily required. However, it will become useful when different particle types are used in the simulation and in the training with reconstructed tracks from Belle II.

For each event and for each valid reference track a loop is carried out over all the related TSs to this reference track. Using the global to local transformations previously introduced in Sec. 5.2.3, each track-to-hit relation becomes an index in one of the data structures A for axial TSs or S for stereo TSs. In summary, this index is composed of $(\text{track}_{\text{bin}}, ts_{\text{local}}, d_{\text{mod}})$, where $\text{track}_{\text{bin}}$ is the 3D cell of the track parameters $(\omega, \phi, \theta)_{\text{bin}}$; $ts_{\text{local}} = (ts_{\text{bin}}, ts_{\text{prio}})$ is the position of the priority wire within the TS in the reduced ϕ-sector; and d_{mod} is the difference between the ϕ-sector of the track and the ϕ-sector of the hit. Hence, a cell in the 6D array S is addressed. For the 5D axial TS array A, a 2D description of the tracks is sufficient using (ω, ϕ). The index ts_{bin} is the successive numbering of the axial TSs in the ϕ-modulo sector for the array A and the successive numbering of the stereo TSs in the ϕ-modulo sector for the array S. As usual when filling a histogram, the indexed cell in the respective data structure is incremented by one:

$$A_{t,h} = A_{t,h} + 1 \qquad\qquad S_{t,h} = S_{t,h} + 1. \tag{5.45}$$

Pseudo code for the full filling process can be found in the Appendix C in Fig. C.1.

Creating the Hit Curves

So far the arrays A and S contain the relative frequencies of all track-to-hit relations in the training data. As an example, an axial hit curve containing these relative frequencies is illustrated in Fig. 5.14 (a). The arrays of weights containing the track-to-hit relations which allow for track finding in the Hough space and for track parameter estimation can be obtained by normalizing the previously filled arrays A and S. The arrays A and S are normalized with the objective to make all tracks equally probable and to flatten out fluctuations in the track-to-hit relations. In general, many different methods of normalization can be used.

In the setup used in this thesis, the normalization is done via a loop over all possible track bins. In the array A a track is identified by the 2D track parameters ω, ϕ, while in S a track is identified by the 3D track parameters ω, ϕ, θ. Note that there is only the dimensional difference to be considered in the arrays A and S. Therefore, the same normalization method has to be applied for the arrays A and S. For each track, the corresponding subspace is selected from the arrays A and S and the distribution of weights in this hit subspace is analyzed. In the following the normalization will be described for the array A only, however, the same procedure has to be applied for the array S.

This normalization loop is only carried out within one ϕ-modulo sector of the track. The additional `neighbors`-axis containing the d_{mod} between the track and the hit is not treated as an extended ϕ-dimension of the track, but as an additional hit dimension. This is possible due to the reciprocal definition of d_{mod} as a ϕ-difference between the track and the hit (see Sec. 5.2.3).

Defining A_t as the array of values in all possible hits h in the subspace of a given track t, A_t has the following hit-axes: $(ts_{\text{rel}}, ts_{\text{prio}}, d_{\text{mod}})$. The elements of the array A_t, i.e. the values in each hit-cell, are denoted by $A_{t,h} \in A_t$. For the normalization, the mean μ_t and standard deviation σ_t of the bin contents in each subspace A_t are

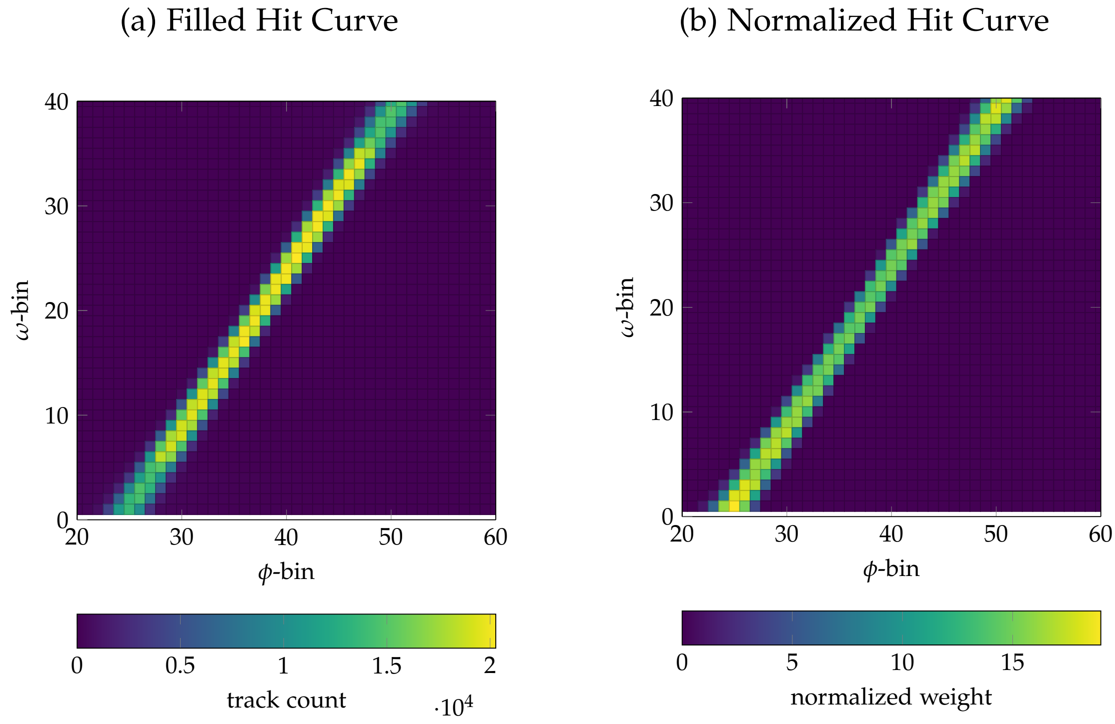


Figure 5.14.: (a) Hit curve of an axial hit directly after the filling corresponding to a 2D histogram of the tracks. (b) Normalized hit curve.

calculated:

$$\mu_t = \frac{1}{N}\sum_h A_{t,h} \qquad \sigma_t = \frac{1}{N}\sqrt{\sum_h (A_{t,h} - \mu_t)^2} \tag{5.46}$$

where N is the number of cells in A_t. This number of cells is the number of TSs in the ϕ-modulo sector (see Sec. 5.2.3) multiplied with the number of bins in the `neighbors` axis:

$$N_{\text{stereo}} = \texttt{TS-ids-stereo} \cdot \texttt{prio} \cdot \texttt{neighbors} = 32 \cdot 3 \cdot 7 = 672 \tag{5.47}$$

$$N_{\text{axial}} = \texttt{TS-ids-axial} \cdot \texttt{prio} \cdot \texttt{neighbors} = 41 \cdot 3 \cdot 7 = 861 \tag{5.48}$$

where the used default values are listed in Tab. 5.1. Using μ_t and σ_t the A_t of each track t are normalized as follows:

$$A'_{t,h} = \frac{A_{t,h} - \mu_t}{\sigma_t} \tag{5.49}$$

After the normalization the $A'_{t,h}$ have $\mu'_t = 0$ and $\sigma'_t = 1$. However, with these weights centered around zero, negative and positive weight contributions are possible. This becomes problematic in the summation of the hit contributions in a multi track event:

all hits related to one track would down-weight the total weight of all other tracks. This has the effect that the total peak weight of each track would be down-weighted proportional to the number of tracks in the event.

This problem rises with the hit representations in a binned track parameter space. Non-zero values should only appear in the close vicinity of the analytic Hough curves for each hit. For example, in Fig. 5.13 this could be achieved by setting all boxes to zero which are not crossed by a hit curve. In principle, the problem would vanish, if the sum is taken over the related hits to a single track. However, this information is only available after the track finding. Therefore, countermeasures have to be installed at this stage, where only the hit information of a full multi-track event is available.

A simple summation of the hit contributions from all tracks in an event into a single multi-track Hough-plane is enabled by using strictly positive weight contributions from each hit. Here this is achieved by shifting the normalized histogram as follows:

$$A'' = A' - \min(A') \tag{5.50}$$

where $\min(A')$ is the minimum value in A'. The resulting hit curves after the shifting to strictly positive values is illustrated in Fig. 5.14 (b).

Bit Width of the Weights

The anticipated representation of each hit is a binned curve of weight contributions of the hit to all possible tracks in the 3D parameter space. Such a discretization of the hit curves is always required for a binned Hough transformation, as it allows to replace the calculation of intersection points by a peak finding in the binned Hough space. In principle, the previously normalized binned hit curves can directly be used for the track finding in a software implementation. However, a hardware implementation of the algorithm is anticipated, which means that the number of bits for each cell in the arrays A and S is restricted. The previously normalized weights are still floating point values and will now be transformed to hardware friendly integer values.

The minimum requirement for this `bitwidth` is a single bit. In Fig. 5.13 this could be achieved by setting all boxes which are crossed by the hit curve to one and all others to zero. This is the method used by the existing Belle II 2D finder (see Sec. 3.3) for the discretization of the hit curves. A disadvantage of this approach is a reduced resolution of the hit curves. Track bins crossed centrally by a hit curve contribute with the same weight as track curves only marginally touched by the hit curve.

Furthermore, a distance information corresponding to the drift time can be embedded in weights with a larger `bitwidth`. As depicted in Fig. 5.8, the hit curves have some thickness corresponding to the drift time distributions of the TSs. Using `bitwidth` $= 3$ such a width of the hit curves can be stored in the hit patterns. In this way, additional correcting effects to the track model, like energy loss, non uniform solenoid field, and multiple scattering, are implicitly embedded.

Increasing the `bitwidth` improves the precision of the following peak finding algorithm, but it also increases the required space and the computing time. A final tradeoff between the achievable precision and the available computing resources has to be made based on the capacities of the concrete hardware platform where the 3D finder is going to be implemented. In the presented setup, a $\texttt{bitwidth} = 3$ is used for each cell in A and S. This means each hit can contribute to each track cell with eight different values and a clear substructure of the hits can be represented. At the same time, the memory requirements on the hardware are only increased by a factor of three compared to a single bit setup. If a future hardware has free memory valences, increased values of the `bitwidth` can be considered.

The `bitwidth` is applied to the weights by binning the weight values. To this end, the maximum values of the histograms $\max(A'')$ and $\max(S'')$ are determined. The maximum value is the largest possible integer number with the defined `bitwidth`. A discretization into the $2^{\texttt{bitwidth}}$ bins is achieved by calculating

$$A''' = \left\lfloor A'' \cdot \frac{2^{b-\epsilon}}{\max(A'')} \right\rfloor \tag{5.51}$$

where a small parameter $\epsilon = 0.0001$ is used to place all the values in one of the $2^{\texttt{bitwidth}}$ cells. Hence, the resulting weight values are integer numbers in the range $[0, 2^{\texttt{bitwidth}} - 1]$. Note that other binning algorithms could be used here as well.

This A''' are the final normalized, shifted and discretized weight values in the array of hit curves in the 3D track parameter space. Later on these hit curves are used for the track finding and track estimation as described in Sec. 5.2.5. For simplicity of the notation, the primes $'''$ are skipped from now on, and the trained weight values in the data structures are simply called A and S. The hit curve of a single axial hit, as it is later used for the track finding, is illustrated in Fig. 5.15.

Discussion of Alternative Methods for the Hit Curve Creation

The weights in the data structure could alternatively be obtained from an analytical calculation. In this case, the nominal position of a hit curve in the track parameter space would be interpreted as mean of a 3D probability distribution. For example, the different distances in the ω-, ϕ-, θ-directions can be modeled by a 3D Gaussian distribution. By defining a variance of this probability distribution, the weights for each track bin could be drawn from the Gaussian. The most important ingredient to this variance is the drift time, as it represents a distance measure between a track and a hit. Since additional physical effects, have to be considered as well, a correct model dependent calculation of the variance is complicated to model. These correcting effects include detector inefficiencies and background contamination as well as model corrections to the idealized assumption of the helical tracks due to field inhomogeneities and energy loss due to multiple scattering at the detector material.

A clear advantage of the presented method is the data driven construction of the hit representations in the Hough space. For example, the energy loss of the particles

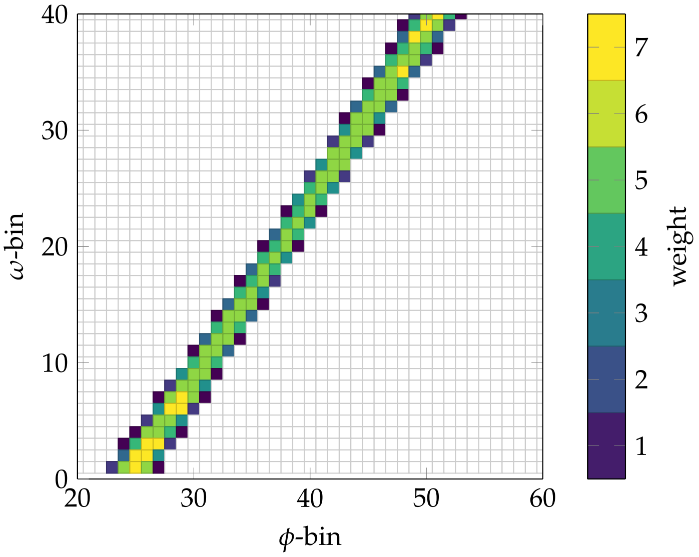


Figure 5.15.: Curve of a single trained axial hit. Due to the value of `bitwidth` $= 3$, a substructure in the hit appears. The large weights close to the central axis of the hit curve decrease towards larger distances in the ω, ϕ-plane. Estimation of the width of a hit curve in the track parameter space. In the presented machine learning approach, the width of this hit curve is automatically determined from the simulation, without the need of additional models.

depends on their type and, therefore, a training with different particle types can be used to achieve more realistic hits in the training data. The method relies on the correctness of the physical effects modeled in the training data set. By using Monte Carlo training data, all simulated error models in the MC simulation are thus implicitly included in the trained hit curves. However, since the physical effects are complicated to model, even the official MC simulation is never fully correct. Improved results can be expected from a retraining of the hit representations using real data recorded from the Belle II detector, which is not affected by the modelling errors in the MC.

5.2.5. Clustering and Track Estimation

The previously trained weight arrays represent the hit curves in the track parameter space and can be used as lookup tables for the track finding. These arrays were trained using a training data set by filling the arrays A and S like histograms to obtain the frequencies of the track-to-hit relations followed by a normalization to make all tracks equally probable (see Sec. 5.2.4). The first step in the track finding is the selection of the correct hit representation from the trained arrays A and S.

As explained below, a proper summation of these ϕ-sectorized single hit contributions provides a complete 3D Hough space covering the full ϕ-acceptance region ($\phi \in [0, 360]^\circ$). Using the parameters in Tab. 5.1 the total number of cells in this Hough space is $(\omega, \phi, \theta) = (40, 384, 9)$. With a value of `bitwdith` = 3, introduced in Sec. 5.2.4, the maximum weight a single hit can contribute to a single track cell is 7.

After summing up all single hit curves to the full Hough space, track finding is carried out with a clustering algorithm in this binned track parameter space. To this end, a modified version of the density based clustering algorithm DBSCAN [85] is developed in this thesis. It is used for the track finding in the 3D track parameter space and implemented in the present 3D finder software version. Once a cluster is found with this clustering algorithm, its peak position is calculated as the weighted average of its member cells. This is similar to a physical center of mass calculation. Since the found peak positions are values in the binned track parameter Hough space, a final back transformation is required. The results provided as output are the estimates of the 3D track parameters $(p_\mathrm{T}^{-1}, \phi, \theta)$ for each found 3D track.

Hough Plane Construction

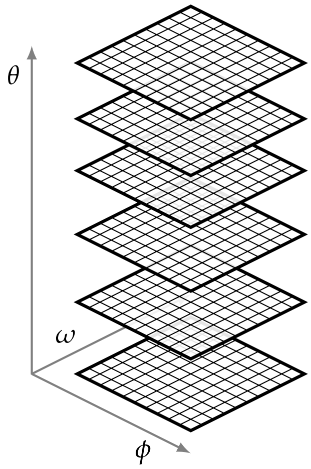


Figure 5.16.: Interpretation of the 3D Hough space as a stack of 2D Hough spaces [83].

In the construction of the Hough space, the trained single hit contributions (see Sec. 5.2.4) have to be added to the full Hough space. As described below, the ϕ-sectorized local hit representations need to be resolved in order to obtain hit representations global in ϕ. Once this is done, the Hough space is created by taking the sum over the single hit representations of the set of axial hits H_axial and the set of stereo hits H_stereo in the event:

$$D = \sum_{h \in H_\mathrm{axial}} \tilde{A}_h + \sum_{h \in H_\mathrm{stereo}} \tilde{S}_h \tag{5.52}$$

where D denotes the 3D Hough space, $\tilde{A}_h \in \tilde{A}$ denotes the array of weights in the global hit subspace relating the axial hit h to all possible tracks and $\tilde{S}_h \in \tilde{S}$ are the corresponding weights of the stereo hit h. Using the elements $\tilde{A}_{t,h} \in \tilde{A}$ and $\tilde{S}_{t,h} \in \tilde{S}$ the weights in each cell of the Hough space are given by

$$D_t = \sum_{h \in H_{\text{axial}}} \tilde{A}_{t,h} + \sum_{h \in H_{\text{stereo}}} \tilde{S}_{t,h} \tag{5.53}$$

where $D_t \in D$ are the elements of the Hough space for each track t. The resulting 3D Hough space can be interpreted as a stack of 2D Hough planes as illustrated in Fig. 5.16.

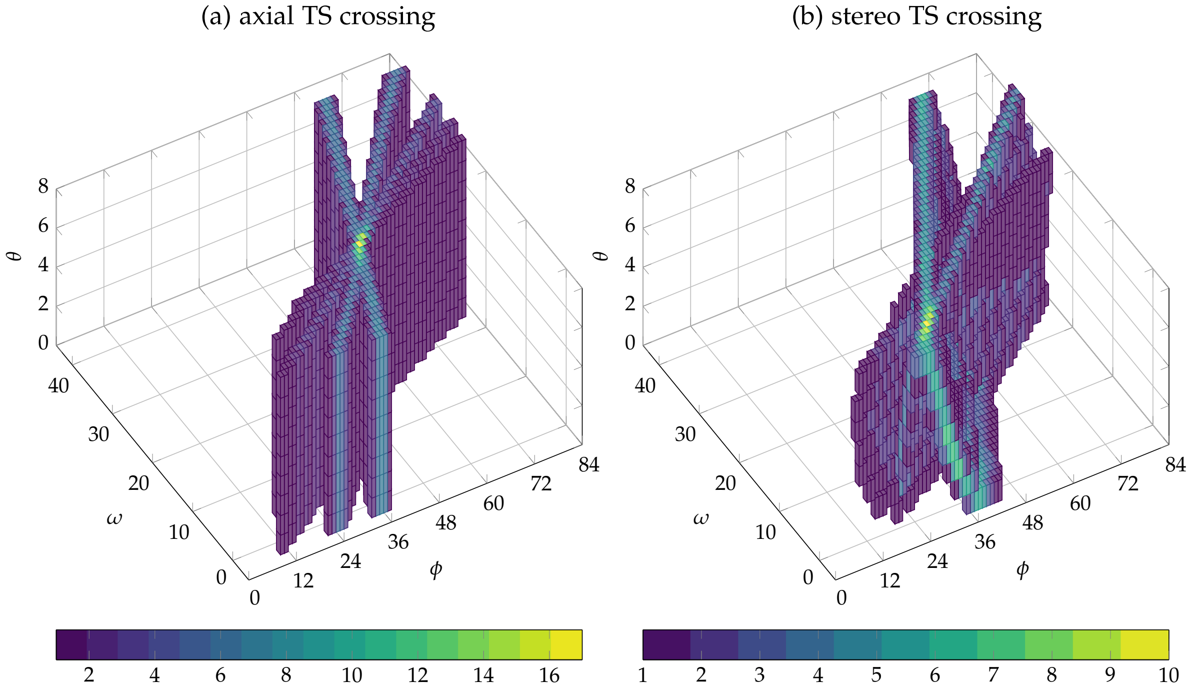


Figure 5.17.: Crossing of the hit representations in the 3D track parameter space. The crossing of five axial TSs is shown in (a) and the crossing of four stereo TSs in (b). While the axial TSs representation in (a) is identical in all θ-bins, the stereo TSs representations in (b) depend on θ. The same example track is used as in the previous figures ($p_{\mathrm{T}} = 1\,\mathrm{GeV}, \phi = 45^\circ, \theta = 90^\circ$).

However, at first the global hit representations need to be calculated. As described in Sec. 5.2.4, the number of cells in the representations A and S were massively reduced by using the ϕ-symmetry of the CDC (see Sec. 5.2.3). These transformations are carried out for each hit h upon filling of the global 3D Hough space.

To this end, the absolute TS-ids need to be transformed into relative TS-ids within the ϕ-modulo sectors using the same method as described in Sec. 5.2.3 for the training ($ts \to (ts_{\text{bin}}, ts_{\text{mod}})$). The resulting sets of ts_{bin} are used to select the local hit-related subspaces (A_h, S_h). Stereo lookups S_h have different contributions along the θ-axis, while for axial lookups A_h the same hit contribution is filled in all θ bins of the resulting Hough space.

The full Hough space is composed out of $32 \cdot 11.25^\circ$ ϕ-modulo-sectors as illustrated in Fig. 5.11. In contrast, for each hit the representation is stored within a ϕ-interval of $\texttt{neighbors} \cdot 11.25^\circ$ only. Now these bins of the hit representation along the `neighbors`-axis, which are created as the ϕ-distance d_{mod} of a track to a hit, have to be properly resolved. To this end, ts_{mod} is used to identify the number of the central ϕ-modulo sector in the Hough space where the hit pattern is added. Corresponding to the distance d_{mod}, the ϕ-modulo sectors in the looked up hit representation are filled to the left and to the right of ts_{mod} into the full Hough space. In this way, the additional axis `neighbors` is resolved into an increased ϕ-region in the track parameter space. For example, with $\texttt{neighbors} = 7$, the seven neighbors in the hit representation correspond to: a central ϕ-modulo sectors, three neighbors to the left, and three neighbors to the right. They are added at the following positions $l_0, ..., l_6$ in the full Hough space:

$$l_0, ..., l_6 = (ts_{\text{mod}} - 3) \mod 32, ..., (ts_{\text{mod}} + 3) \mod 32 \tag{5.54}$$

Since ϕ is a periodic variable, the ϕ-modulo sector within the full Hough space has to be corrected accordingly by calculating the modulo of the positions with 32.

The first ϕ-bin (ϕ_{start}) in the full Hough space for neighbor n of a single hit is given by

$$\phi_{\text{start}} = l_n \cdot \texttt{nbins}_\phi \tag{5.55}$$

where $\texttt{nbins}_\phi = 12$ is the number of ϕ-bins in each ϕ-modulo sector (see Tab. 5.1). The full Hough space is constructed by summing up all the single hit contributions at the correct ϕ-positions.

From a 3D finder with nine bins in θ, two crossings for the axial TSs and of for the stereo TSs are shown separately in Fig. 5.17. These correspond to the first and second summand in Eq. 5.52. With identical values in each bin along the θ-axis, the crossing point of the axial TSs in Fig. 5.17 (a) corresponds to the same 2D Hough picture in each θ-bin. However, for the stereo TSs crossing shown in Fig. 5.17 (b), the single hit representations depend on θ. Due to this slant of the stereo TSs in θ, the peak position is hidden in the center of the intersecting planes. The joint 3D space of the axial and stereo crossings illustrated in Fig. 5.17 is the input to the subsequent track finding algorithm.

For better visibility, Fig. 5.18 illustrates the summed up track parameter weight space shown as separate 2D Hough planes for each of the nine θ-bins. This allows to peak into the central region of the crossing stereo TSs, which was hidden behind the upper θ-bins in Fig. 5.17 (b). In Fig. 5.18, the weight after the summation is indicated by color and the maximum weight in each θ-bin can be read off the color bar legend. In agreement with Fig. 5.18 (a), the axial TS curves have the same single crossing point in each θ-bin. However, the stereo TS curves intersect with the axial

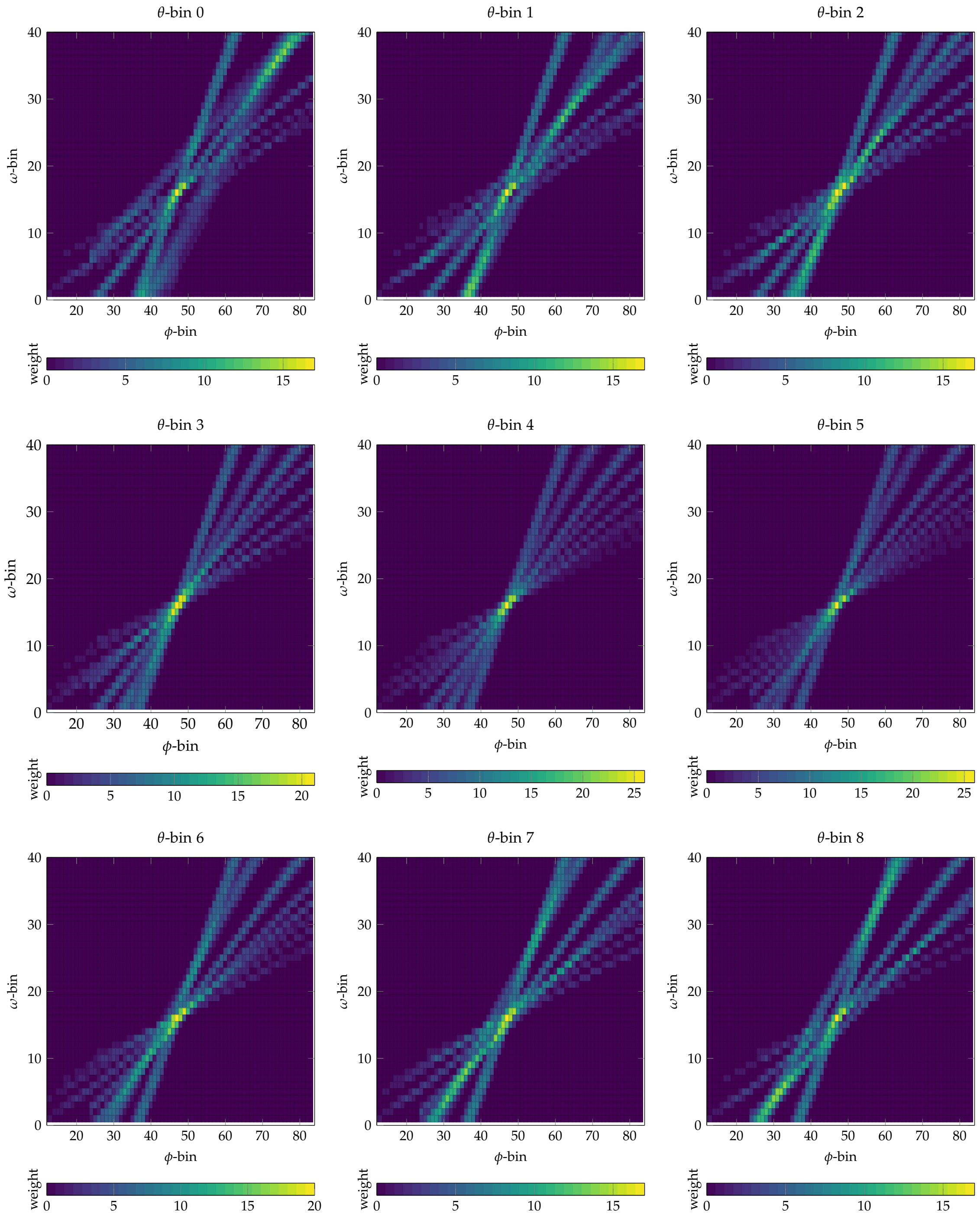


Figure 5.18.: Hough space of a 3D finder setup using nine bins in θ. The 2D Hough planes for the crossing of all axial and stereo TSs are illustrated separately for each θ-bin.

TSs crossing point only close to the correct θ-value. In θ-bins displaced from the correct track θ, the four stereo TS layers have additional separate intersections points displaced from the axial intersection point. Furthermore, the maximum weight in displaced θ-bins is reduced since the stereo TSs do not contribute to the correct peak cell. Since the single muon track is simulated with $\theta = 90^\circ$, the correct θ is the central θ-bin 4.

With four stereo hits, usually up to six separate stereo intersection points can be found in different 2D Hough planes for displaced θ-bins. The number of six corresponds to the pairwise intersections of two out of four stereo TSs. In each θ-bin, such a pairwise intersection appears as crossing point of two stereo curves. However, the total weight in the cells with only two intersecting TSs is much smaller than the weight in the cells close to the correct intersection point. In Fig. 5.18, such displaced stereo crossings appear with less brightness compared to the correct intersection point of all TSs. As will be described below, these false intersections can easily be rejected by the clustering algorithm using a requirement on the minimum weight of cluster the cluster cells.

Discussion of the Stereo Crossing Positions

In general, a displacement in both 3D track parameters, θ and z, changes the positions of the stereo crossing points in a 2D Hough plane for a specific chosen value of the 3D track parameters θ and z. Only if the selected 2D Hough plane is constructed for the correct θ, z-values, the four stereo TS curves and the axial TS curves intersect in a single point. Although the tracks at Belle II have their vertices constrained to the IP, a theoretical discussion of the track parameter space including the z-axis is useful to imagine the effect of background tracks for the presented 3D finder.

The effect of displacements in θ and z can be qualitatively understood in a simplified model by neglecting the curvature of the hit representations. In this way, the 3D stereo TS representations in the ω, ϕ, θ-space can be approximated as slanted planes, where the slanting angle with the θ-axis is proportional to the stereo angle of the wires. In this approximation, the stereo crossing corresponds to the crossing of four planes, each with a different slanting angle. The volume enclosed by these four planes has the shape of a tetrahedron and the position of the correct track parameters corresponds to the limit of vanishing volume of this tetrahedron.

For tracks from $z = 0\,\text{cm}$, the stereo TS planes intersect in a single point. Hence, the pairwise intersections of the four stereo planes in the track parameter space create six lines which meet in the single crossing point for the correct θ value. In this simplified interpretation, θ and z can hypothetically be interchanged to create an analogous 3D Hough space in ω, ϕ, z. This means to fix the correct value of θ and to look at the stereo TS representations for displaced values of z. Since the track parameters θ and z are linearly related for a given set of 2D track parameters, the resulting ω, ϕ, z Hough space has a similar structure to the ω, ϕ, θ Hough space. By neglecting the curvature of the TSs representations, the slanted TS planes in the 3D track parameter space can be interpreted as 3D slices of a more general 4D ω, ϕ, θ, z-

space.

By looking at the usual ω, ϕ, θ-space for a fixed value of $z = 0\,\text{cm}$, a displaced z-position of an observed track corresponds to an increased volume of the tetrahedron. This introduces a huge benefit of the 3D finder: in the presented setup, tracks displaced along the z-axis are less likely to be found since the stereo TSs do not make up a single crossing position. Note that in this description, the hit curves are 4D, for completeness even a 5D description can be considered using all the helix parameters $(\omega, \phi, \theta, z, d)$.

Track Finding with a Clustering Algorithm

The track finding in the 3D Hough space is carried out with a density based clustering algorithm derived from the DBSCAN [85] algorithm. The normal DBSCAN algorithm is specified for a set of points, where the local densities are obtained from a distance measure between the points. Since the Hough space is a binned track parameter space, several adjustments to the algorithm are required. In order to motivate the DBSCAN algorithm, the normal unbinned version is shortly summarized. Afterwards, the clustering algorithm used by the 3D finder in the binned Hough space is presented.

The density based DBSCAN [85] clustering algorithm has two parameters, `eps` and `minpts`. In its usual version, it is applied to a sets of points, where each point is given by its unbinned coordinates. The DBSCAN algorithm checks the neighborhood of each point, which is defined as the distance `eps` with respect to the point. If at least `minpts` other points are contained within this `eps`-neighborhood, the point is considered a "core" point of a cluster. Such a "core" point is in a high density region of the data set. Therefore, it is used as seed to expand a cluster by adding all the points within its `eps`-neighborhood to the cluster.

For all the points newly added to a cluster, the same check as for the initial point is carried out. This means, if at least `minpts` points are contained within a maximal distance of `eps` of a point, the point is considered a "core" point. The search for more points belonging to an existing cluster is repeated for all "core" points found with this method. Points within a distance `eps` of a "core" point which themselves do not have `minpts` points in their own `eps` neighborhood are also associated to a cluster. However, they are not considered as "core" points and therefore not used to search for more cluster members in their neighborhood. In this way, the found clusters are sets of points connected by high local densities.

The algorithm searches for additional clusters until all points in the data set are processed. Since the DBSCAN algorithm finds clusters as regions connected by a high density, the found clusters can have arbitrary shapes.

Binned Density Based Clustering Algorithm

In the binned Hough space D, the total weight in each cell D_t is the sum of the weights of all contributing hits, as indicated in Eq. 5.52 and Eq. 5.53. This means that the weight in each cell corresponds to a measure of the local density of hits. In analogy to the analytic solution of the Hough transformation, these clusters correspond to the intersection points of the hit curves in the Hough space. Therefore, a clustering algorithm can be used to identify the tracks in the events.

Note that the trained hit curves have an extension in the Hough space which mainly approximates the drift times, as illustrated in Fig. 5.8. This is realized by using a number of `bitwidth` bits for the weights in each cell of $A_{t,h}$ and $S_{t,h}$, as described in Sec. 5.2.4. Using the trained hit representations, the sum of the weights for a single track cell contains more information than a simple analytic density of the central hit curves. Only in a simplified setup, where the `bitwidth` for each single hit-to-track relation is set to `bitwidth` $= 1$, the total weight in a single cell D_t in the Hough space would approximate the bare number of hit curves crossing the Hough cell.

parameter	default value	description
`minpts`	1	minimum adjacent neighbor cells
`diagonal`	True	check diagonal neighborhood
`minweight`	24	minimum weight of the cluster cells
`minhits`	4	minimum hits per cluster
`thresh`	0.85	cut off weight depth for the weighted mean

Table 5.5.: Configuration parameters for the binned clustering algorithm and for the track parameter estimation.

In the binned clustering algorithm for the 3D finder developed in this thesis, the weights of the cells in the binned Hough space are inspected, rather than single points. In order to find clusters of Hough space cells, the algorithm processes all the track cells in the Hough space. The set of configuration parameters defined for this binned clustering algorithm and their default values are listed in Tab. 5.5.

A rough estimate leading to the default value of `minweight` $= 24$ listed in Tab. 5.5 can be found by noting its close connection to the parameter `bitwidth`. With the default value of `bitwidth` $= 3$ from Tab. 5.1, the maximum weight a single hit can contribute to a single cell is 7 (see Sec. 5.2.4). Therefore, the default of `minweight` $= 24$ can be interpreted as a minimum of 3.5 hits contributing with a maximum possible weight. In this sense, `minweight` is a weighted cut on the minimum number of hits, with a default value corresponding in the same order as the absolute requirement on the minimum number of hits of `minhits` $= 4$. Note that `minweight` could easily be redefined in a `bitwidth` independent way as `minweight'` $=$ `minweight`$/(2^{\texttt{bitwidth}} - 1)$. Optimization studies with different settings of the parameters `minweight` and `minhits` are presented in Sec. 5.4.2.

In the default setup shown in Tab. 5.5, the parameter `minpts` is set to `minpts` $= 1$. In this way, single cell clusters are avoided by requiring at least a single neighboring cell. However, each candidate cell in a cluster is automatically also a “core” cell. This

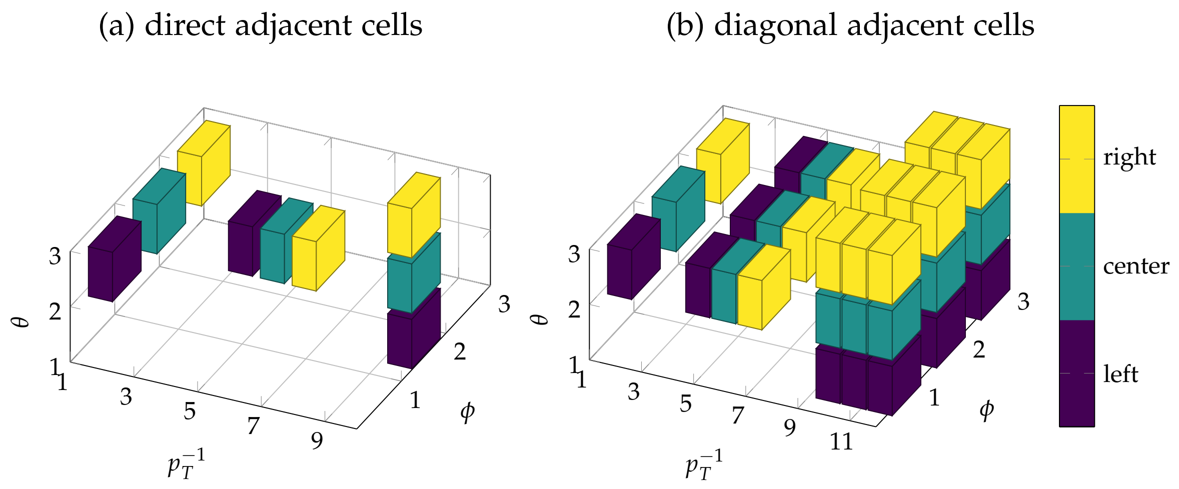


Figure 5.19.: The adjacent cells to a candidate cell with a weight larger than `minweight` are added to the same cluster if they also have a sufficient weight larger than `minweight`. In (a) the six direct neighboring cells are checked, along each of the three track parameter axis ($p_{\mathrm{T}}^{-1}, \phi, \theta$). In (b) additionally all 26 diagonal neighbors are checked as well. The shown units are arbitrary bin indices.

parameter is closely related to the binning of the Hough space and can be changed in future updates of the 3D finder configuration.

Track cells with a weight larger than `minweight` are considered as candidate cells for new clusters. A candidate cell is considered a "core" cell of a cluster if it has itself at least `minpts` neighboring candidate cells. This means at least `minpts` neighboring cells are required to have a weight of at least `minweight`. The neighborhood of these "core" cells are searched for possible expansions of the cluster. As a result, clusters are connected regions of cells, where all cells have a weight larger than `minweight`. Compared to the normal DBSCAN, the counting of the points within the `eps` neighborhood is replaced by directly taking the weight in each cell as a measure of the local density.

Clusters are primarily extended to adjacent cells along the three dimensions of the track parameter space. This means that the left and right neighbor cells along each of the three axes $p_{\mathrm{T}}^{-1}, \phi, \theta$ are checked for their weight. As illustrated in Fig. 5.19 (a), with three track parameter axes, this corresponds to six direct neighbors. If these adjacent cells are also "core"candidate cells, they are added to the cluster. Additionally, the parameter `diagonal` is included in the algorithm to allow for a check of diagonally adjacent cells as well. Direct adjacent cells share one of the six planes of the cuboid with their neighbor. Using the parameter `diagonal` includes adjacent cells sharing one of the twelve edges as well as cells sharing one of the eight vertices in the neighborhood search.

	cluster 1	cluster 2	...	cluster N
hit 1	$w_{1,1}$	$w_{1,2}$	...	$w_{1,N}$
hit 2	$w_{2,1}$	$w_{2,2}$	...	$w_{2,N}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
hit M	$w_{M,1}$	$w_{M,2}$	...	$w_{M,N}$

Figure 5.20.: The confusion matrix for the hits to cluster relations has a column for each cluster and a row for each hit. The weights $w_{i,j}$ are the weight contributions of the M hits i to the peak cells of the N clusters j. Each hit is related to the cluster with the largest weight contribution. Clusters with less than `minhits` hits are removed.

As illustrated in Fig. 5.19 (b), these diagonal neighbors are checked via partially including the direct adjacent cells of the adjacent cells as well. Hence, in total 26 adjacent cells are checked with the option `diagonal`. This is done in a recursive loop over the three dimensions (with the function regionQuery()), as illustrated in the Appendix C in Fig. C.3. Note that the track parameter ϕ has to be treated as a cyclic variable during the clustering. This means that the largest ϕ-cells are adjacent to the smallest ϕ-cells. Similar to the DBSCAN algorithm, the cluster is expanded to the neighborhood of these checked adjacent cells only if they also fulfill the criteria of a "core" cell. Pseudo code for the function expandCluster() is listed in Appendix C in Fig. C.4 and the The main function of the cluster expansion algorithm for this binned density based clustering algorithm is listed in Fig. C.2.

In future studies, the shape, size, and extent directions of the clusters can be taken into account during the clustering. Diagonal neighbors could be checked only along the typical extent directions of the clusters. This can be seen best in a 2D intersection of the hit curves in the ω, ϕ-space shown in Fig. 5.5. By selecting one of the two intersection points to resolve the charge ambiguity, the average crossing angle between the hit curves is limited. Therefore, the clusters tend towards an extension into the upper right and into the lower left direction in the ω, ϕ-space. Alternatively, if the charge convention is globally inverted, i.e. if the second crossing point in Fig. 5.5 is selected, the clusters extend into the lower right and into the upper left ω, ϕ-directions. Furthermore, the extension in θ could be included to limit the search directions. Such limitations or the extent direction are especially useful with the `diagonal` search options, as there are 26 diagonal neighbors in 3D as illustrated in Fig. 5.19 (b).

Relating the Hits to the Clusters

The hit relations to the clusters are determined with a confusion matrix as shown in Fig. 5.20. This is a matrix with all the hits as rows and all the clusters as columns. Each field i, j of the matrix is filled with the weight contribution $w_{i,j}$ of the hit i to the peak cell of the cluster j. For each hit i this allows to extract the cluster j with the maximum weight contribution $w_{i,j}$ to the cluster. For these maximum weight contributions $w_{i,j}$, a relation between the hit i and the cluster j is created.

However, there is a chance that clusters have only few related hits. In order to avoid clusters with too few hits, the configuration parameter `minhits` is introduced. Only clusters with at least `minhits` related hits are kept and interpreted as found tracks, while the smaller clusters are removed.

Due to this removal of clusters, some previously related hits end up unrelated after the cluster removal and could possibly be re-associated to the next-best remaining bigger cluster. Therefore, the calculation of the hit to cluster relations is repeated after the removal of small clusters. After eliminating the columns representing the removed clusters, the reduced confusion matrix is used to find next best track-to-hit relations.

In the present setup, the cluster removal and hit reassignment is carried out for two iterations. In this way, an improvement of the track finding efficiency can be expected, since not the full `minhits` cut is directly applied. For example, a small cluster having only $\texttt{minhits} - 1$ related hits initially, has the chance to pick up additional hits during the second iteration. In general, the number of iterations should be minimized in order to reduce the latency in the hardware implementation. Especially in low multiplicity events, the effect is expected to be small, as the overall occupancy in the CDC is low. If this hit re-iteration feature poses latency problems in the hardware implementation, it could possibly be skipped in favor of a shorter latency.

Estimation of the Track Parameters

Each of the found clusters corresponds to a found track. Using the definition of the track parameter ranges and binnings along the axes of the Hough space, the bin positions in the Hough space can easily be transformed into physical track parameter values. This unbinning is the inverse operation of binning of the track parameters carried out during the training (see Sec. 5.2.3).

The first step in the track parameter estimation is the calculation of the precise cluster centers t_{bin}. To this end, the parameter `thresh` is introduced to select a subset M of cluster member cells $\vec{m}$ with a relatively large weight contribution. Using the maximum weight w_{max} of the cluster, i.e. the maximum weight of a single cell in the cluster, `thresh` defines the minimum required fraction of this maximum weight for the selected cells M. All the cells $\vec{m} \in M$ are required to have a weight $w_m \geq \texttt{thresh} \cdot w_{\text{max}}$. Such a focus on the peak region is desirable, as the cluster can be interpreted as the binned crossing point of the Hough curves. Clusters with very large extensions are not correctly identified tracks. Hence, an improvement in the resolution of the track parameters can be expected by the selection of the local region around the peak. Experiments with different values of the parameter `thresh` are presented in Sec. 5.4.2.

After the subset M is obtained using the parameter `thresh`, the calculation of the weighted mean of the cluster centers can be summarized as follows

$$\vec{t}_{\text{bin}} = \frac{1}{|M|} \sum_{\vec{m} \in M} \vec{m} \cdot w_m \tag{5.56}$$

where $|M|$ denotes the number of selected cell indices $\vec{m}$ in the subset M. The position $\vec{m}$ of each cell within the cluster is multiplied with its bin content, the weight w_m. As weighted sum of all the contributing cluster members, the vector $\vec{t}_{\text{bin}}$ is the peak position of the cluster.

Due to the calculation of the weighted average, resolutions in the order of the binary limit for a parameter represented in a histogram can be achieved for t_{bin}. This intrinsic resolution for a uniformly distributed binned random variable is given by the bin-width divided by $\sqrt{12}$ (see Appendix A.2). Although it is important to note here that this theoretical bin-resolution corresponds to the case of a correctly known value sorted into bins. Hence, it does not fully apply to this clustering case, where the clusters itself have extents due to the extents of the hit curves. Experimental measurements of the track parameter resolutions with the 3D finder are presented in Sec. 5.3.

Since ϕ is a periodic variable, the averaging along the ϕ-axis has to be carried out carefully. This means, to close the circle the largest ϕ-bins are geometrically located next to the smallest ϕ-bins. In the presented setup, the ϕ-bin-values of clusters at the ϕ-border are shifted to a border free region before calculating the weighted average. Afterwards, the found $\vec{t}_{\text{bin}}$ is shifted back to its supposed ϕ-position.

The last step in the estimation of the track parameters for a single track is the back transformation of $\vec{t}_{\text{bin}}$ to physical track parameters. The Hough space position $\vec{t}_{\text{bin}}$ is composed of three components, one for each binned variable v. For the back transformation, the track parameter ranges and the number of bins are required (see Tab. 5.1). With the defined range of a track parameter $v \in [v_{\text{min}}, v_{\text{max}}]$, and $\texttt{nbins}_v$ the number of bins in this track parameter, the bin-width for v can be calculated by:

$$\texttt{binwdith}_v = \frac{v_{\text{max}} - v_{\text{min}}}{\texttt{nbins}_v}. \tag{5.57}$$

Using the $\texttt{binwidth}_v$ of a variable v and the minimum of the range, the back transformation can be calculated by:

$$v = v_{\text{min}} + v_{\text{bin}} \cdot \texttt{binwidth}_v \tag{5.58}$$

which can be calculated separately for the three track parameters $p_{\text{T}}^{-1}, \phi, \theta$. This procedure has to be repeated for each found track in the events.

`minweight`	`thresh`	`minhits`
24	0.85	4

Table 5.6.: Central finder parameters used for the optimization studies of the training and clustering parameters.

5.3. Track Finding Performance

In this section the results of the 3D track finder studies are presented. After introductory studies with different training options and test data ranges, optimization studies of the 3D finder configuration parameters are addressed. Special focus is on tracks with a small number of TSs, where benefits with the 3D finder can be expected in comparison with the existing 2D finder. In particular, tracks at the boundary region of the geometrical CDC acceptance are subject to missing TSs. This boundary region comprises tracks with shallow-θ angles which leave the CDC before passing through all the SLs as well as strongly curved low-p_T tracks which do not create TSs in the outer SLs. Additional contamination with background distorts the events and may lead to misidentified TSs.

In order to define the track parameter ranges for the training and test data sets, an initial study of the geometrical boundaries of the CDC acceptance region is presented in Sec. 5.3.1. As a result of this study tracks with a transverse momentum of $p_\mathrm{T} \in [0.2, 0.35]\,\mathrm{GeV}$ are called low-p_T tracks. Tracks with a polar angle in the forward direction of $\theta \in [19, 35]\,^\circ$ or with a polar angle in the backward direction of $\theta \in [123, 140]\,^\circ$ are called shallow-θ tracks.

Using the track parameter ranges defined in Sec. 5.3.1 for the training and test data, several introductory 3D finder studies are carried out with the central configuration parameters shown in Tab. 5.6. In Sec. 5.3.3 the achievable track parameter resolutions and the track finding efficiencies of 3D finders trained with different training data sets are analyzed. This includes a comparison of reconstructed tracks with MC tracks for the use as training target and the effect of background in the training data. Resolution and efficiency studies at different boundary regions of the CDC acceptance region are presented in Sec. 5.3.4.

The properties of the found track clusters in the Hough space, including their size, weight, and shape, are shown in Sec. 5.4.1 and possibilities for additional configuration parameters in the form of cuts on the cluster properties are discussed. Variations of the present configuration parameters `minweight`, `minhits`, and `thresh` are studied in Sec. 5.4.2. This is done by measuring the effect of the configuration parameters on the track parameter resolution, the track finding efficiency and on the structure of the track clusters in the Hough space as introduced in Sec. 5.4.1. Using the structure of the Hough space defined in Sec. 5.2.2, the central configuration parameters shown Tab. 5.6 are demonstrated to be close to a local optimum of the configuration parameters for the 3D finder. Since the configuration parameters are interdependent with the Hough space structure and with the chosen clustering algorithm, the pursued optimization methods for the configuration parameters are important for future upgrades of the 3D finder algorithm. Such upgrades might be required to meet the limited resources on the future FPGA hardware. Once a hardware friendly setup is chosen, a cluster feature analysis, similar to the one presented in Sec. 5.4.2, needs to be carried out in order to determine the proper hyper-parameters.

5.3.1. Train and Test Data

Due to the use of the additional stereo TSs, an improved track finding efficiency can be expected especially for short tracks, which create TSs only in the inner SLs. The study in this section is carried out to determine the range limits of the CDC acceptance in p_T and θ, with the resulting ranges shown in Tab. 5.8.

In order to test the track finding at the boundary regions, four different test data ranges are defined: a range within the full CDC acceptance region (labelled N for "normal"), where tracks usually create TSs in all 9 SLs; a range extending to the low-p_T region (labelled P for "p_T"); a range containing shallow tracks in the forward and backward region in θ (labelled W for "wide θ"); and a range extending to the low-p_T region as well as the wide-θ region (labelled L for "large ranges"). Each of the four track parameter ranges is simulated with and without background, where the used background is the official mixing background simulated in the Belle II background campaign 15 (see Sec. 4.4.2). This campaign is also used in the Belle II MC production "MC10" (see Chapter 4 for more details on the background). The method of background mixing is used (see Sec. 4.4.1), where background hits are added prior to the creation of the realistic hits by simulating the readout of the detector. The parameter ranges are listed in Tab. 5.7 and the labels for the eight test data set combining a parameter range with a background option are shown in Tab. 5.8.

CDC Acceptance

The acceptance limits in the track parameter ranges p_T and θ are probed using tracks with vertices located at the IP, i.e. $z_0 = 0\,\mathrm{cm}$ and $d_0 = 0\,\mathrm{cm}$. In order to inspect the low-p_T limit of tracks with missing TSs due to a large curvature, a p_T scan is carried out with a minimum $p_\mathrm{T} = 100\,\mathrm{MeV}$. In θ the limiting ranges are the design parameters of the CDC, with $\theta \in [17, 150]^\circ$. At values of $\theta \leq 17^\circ$ as well as $\theta \geq 150^\circ$, tracks from the IP are not contained within the CDC volume and, therefore, no hits are created (see Sec. 3.2.5). Since the CDC covers the full azimuthal angle ϕ, the tracks are simulated with $\phi \in [0, 360]^\circ$. Within the respective ϕ-, and θ-ranges for this acceptance study, 10^5 single $\mu^\pm$ tracks are generated with a uniform distribution. Within the p_T-range, tracks are generated uniformly in p_T^{-1}.

In order to study the limits for low-p_T tracks, tracks with shallow polar angles in the forward region and tracks with shallow polar angles in the backward region,

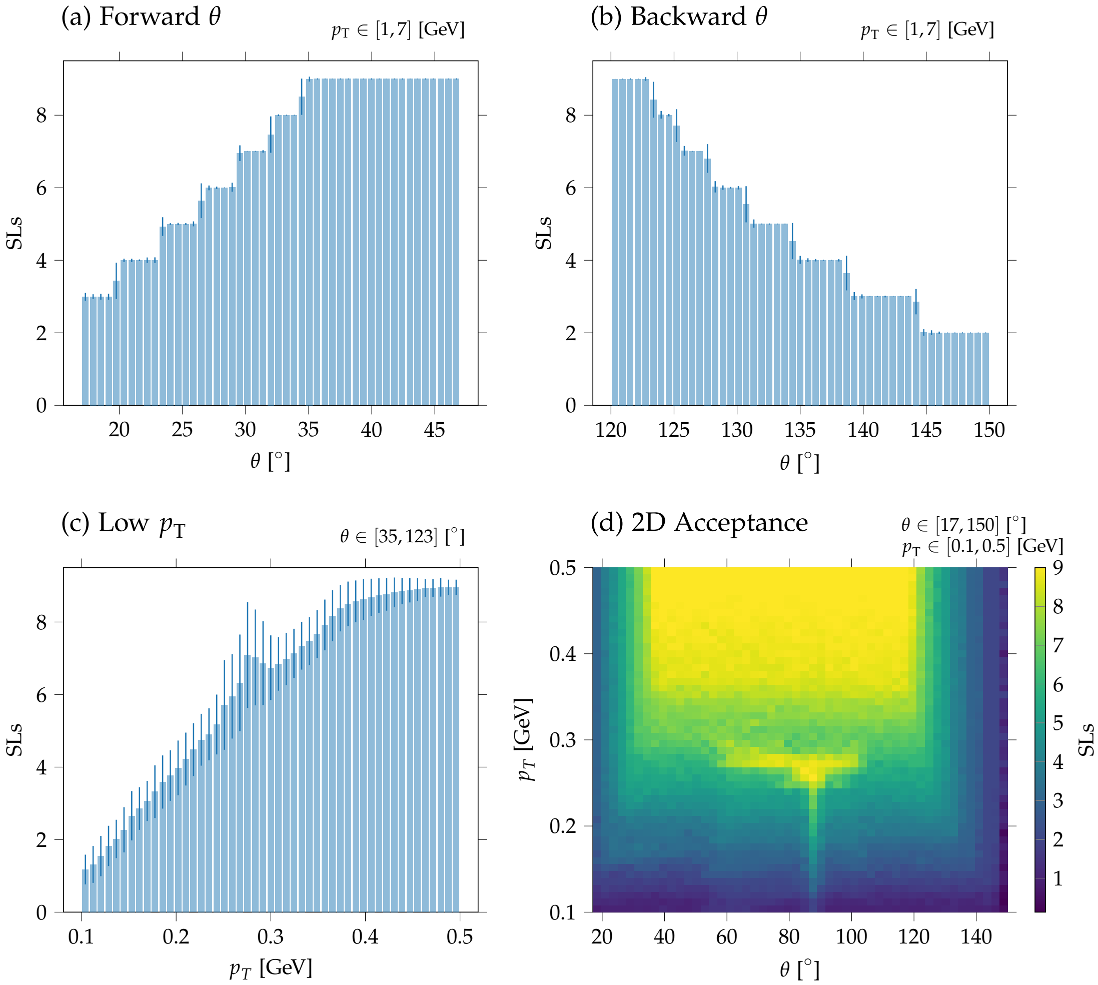


Figure 5.21.: Study of the CDC acceptance in p_T and θ used to define the track parameter ranges in Tab. 5.8. For simulated $\mu^{\pm}$ tracks from the IP, the average number of distinct SLs containing TSs per event is shown separately for the boundary regions in p_T and θ. (a) shows the forward boundary region in θ; (b) shows the backward boundary region in θ; (c) shows the low-p_T region; (d) illustrates the combined p_T and θ boundary regions in a 2D plot.

different subsets of the p_T and θ ranges are used. This means, for the θ-boundaries, only high-p_T tracks with $p_T \in [1,7]$ GeV are used, while for the p_T-boundary test only tracks in a central θ region of $\theta \in [35,123]^{\circ}$ are used. The resulting average number of TSs per simulated single track is shown in Fig. 5.21. The respective p_T- and θ-ranges are indicated in the titles of Fig. 5.21. As discussed below, the limits of the boundary regions are extracted by assuming a minimum of three to four TSs per track for a track to be possibly found by a track finder.

Range	p_T [GeV]	ϕ [°]	θ [°]
N	$[0.35, 10.2]$	$[-180, 180]$	$[35, 123]$
L	$[0.2, 10.2]$	$[-180, 180]$	$[19, 140]$
P	$[0.2, 10.2]$	$[-180, 180]$	$[35, 123]$
W	$[0.35, 10.2]$	$[-180, 180]$	$[19, 140]$

Table 5.7.: Definition of the track parameter ranges covering different boundary regions of the CDC acceptance. The range N – normal – contains only tracks contained in the full CDC acceptance region, while the range L - large - includes low-p_T as well as shallow-θ tracks. The range P - low-p_T - contains low-p_T tracks but no shallow-θ tracks, while the range W - wide-θ - contains shallow-θ tracks but no low-p_T tracks.

Figure 5.21 (a) shows the forward region in the polar angle, with $\theta \in [17, 47]\,^\circ$ and Fig. 5.21 (b) shows the backward region with $\theta \in [120, 150]\,^\circ$. Figure 5.21 (c) shows the low-p_T region, with $p_T \in [0.1, 0.5]$ GeV. The 2D acceptance plot in Fig. 5.21 (d) gives an overview of the full boundary region including correlations of the low-p_T and shallow-θ regions.

Using the distributions in Fig. 5.21 (a) and (b), the θ-range containing both boundary regions is selected as $\theta \in [19, 140]\,^\circ$ and the θ-range of the full acceptance region is selected as $\theta \in [35, 123]\,^\circ$. As will be shown later, the lowest p_T-value where the existing 2D finder is fully efficient is observed as $p_T = 0.35$ GeV. Fig 5.21 (c) and (d) show that tracks with this p_T-value typically have TSs in 8 distinct SLs. This agrees with the setup of the 2D finder, where a minimum number of four axial TSs are required. Using these observations, the four different track parameter range combinations with extensions into different boundary regions are defined and summarized in Tab. 5.7.

The peak in Fig. 5.21 (c) at $p_T \approx 0.28$ GeV is a result of low-p_T tracks curling back inside the CDC volume. Note that matching is not required in this CDC acceptance study. For each simulated single track event, simply the total number of distinct SLs containing TSs is counted. Tracks with $0.25\,\text{GeV} \lesssim p_T \lesssim 0.28\,\text{GeV}$ reach the outermost wire layer of the CDC at $r_{\text{CDC}} \approx 1.11$ m. However, they are not yet stopped by the ECL at the inner radius of $r_{\text{ECL}} = 1.25$ m. By curling back through all CDC layers, TSs are found for the outwards going track and for the curl back track. Since the track finding efficiency depends on the crossing angle of the track with the TSs (see Sec. 5.2.2), these curl-back tracks increase the average number of distinct SLs containing TSs per single track event. However, these curl-back tracks do not result in an improved track finding efficiency of the correct tracks. In the track finding, as well as in the offline reconstruction, only outwards going TSs are combined to single tracks. This means, the curl-back tracks may appear as additional fake tracks. Furthermore, those TSs added during the second outwards going section of a curling trajectory correspond to a track with similar p_T, ϕ, and θ values but with a displaced

z-vertex position. If these TSs mix with the TSs of the initial track, an apparent efficiency increase may be observed which manifests itself in a significantly reduced resolution.

Test Data	p_T [GeV]	ϕ [°]	θ [°]	BKG
N0	$[0.35, 10.2]$	$[-180, 180]$	$[35, 123]$	no bkg
N1	$[0.35, 10.2]$	$[-180, 180]$	$[35, 123]$	bkg 15
L0	$[0.2, 10.2]$	$[-180, 180]$	$[19, 140]$	no bkg
L1	$[0.2, 10.2]$	$[-180, 180]$	$[19, 140]$	bkg 15
P0	$[0.2, 10.2]$	$[-180, 180]$	$[35, 123]$	no bkg
P1	$[0.2, 10.2]$	$[-180, 180]$	$[35, 123]$	bkg 15
W0	$[0.35, 10.2]$	$[-180, 180]$	$[19, 140]$	no bkg
W1	$[0.35, 10.2]$	$[-180, 180]$	$[19, 140]$	bkg 15

Table 5.8.: Definition of the subsequently used test data sets. Each of the four ranges listed in Tab. 5.7 is simulated twice: with and without background (indicated by a 1 or a 0 following the range label). For each range, data sets with and without background (1 and 0) are simulated. The used background are the official background mixing files from background campaign 15 described in Chapter 4. Each test data sets contains 10^5 single $\mu^{\pm}$ track events.

Eight different test data sets are simulated using the labels summarized in Tab. 5.8. Besides the range labels N, L, P, and W from Tab. 5.7, the used background option is indicated by a number; 0 indicates a simulation without background and 1 indicates a simulation with background using background campaign 15.

The two ranges mainly used in the subsequent studies are N and L. In Sec. 5.3.4 additionally the ranges P and W are used to disentangle the effects of the enlarged p_T and θ ranges contained combined in the range L. The data range N contains tracks in the full acceptance region of the CDC for tracks from the IP. Within a θ-range of $\theta \in [35, 123]\,^{\circ}$ and a p_T-range of $p_T \in [0.35, 10.2]\,\text{GeV}$ tracks from the IP typically have sufficient TSs to be found by the 2D finder. Hence, the 2D finder is expected to be fully efficient in the range N.

In the data range L, the track ranges extend into the boundary region in p_T and in θ. Therefore, a loss in the track finding efficiency with the 2D finder is expected. In this data range L a significant efficiency improvement is expected from the 3D finder compared to a 2D finder which uses the axial TSs only. The boundary region limits of the range L are also used for the training of the 3D finders described in Sec. 5.2.4.

Straight tracks at the boundaries of the wider θ-range $[19, 140]$, used in data ranges L and W, reach sense layer 21 of the CDC only which corresponds to TS hits in the innermost three SLs. The extended p_T-range including low-p_T tracks with $p_T \in [0.2, 10.2]\,\text{GeV}$ is used in the data ranges L and P. At the lower limit of $p_T = 200\,\text{MeV}$, shown in Fig. 5.21 (c), TSs are typically found in three to five distinct SLs. Although tracks with 200 MeV reach sense layer 41 of the CDC, the dependence of the TSF efficiency on the crossing angle (see Sec. 5.2.2) reduces the number of TSs in distinct SLs here.

Simulated Track Parameter Distribution

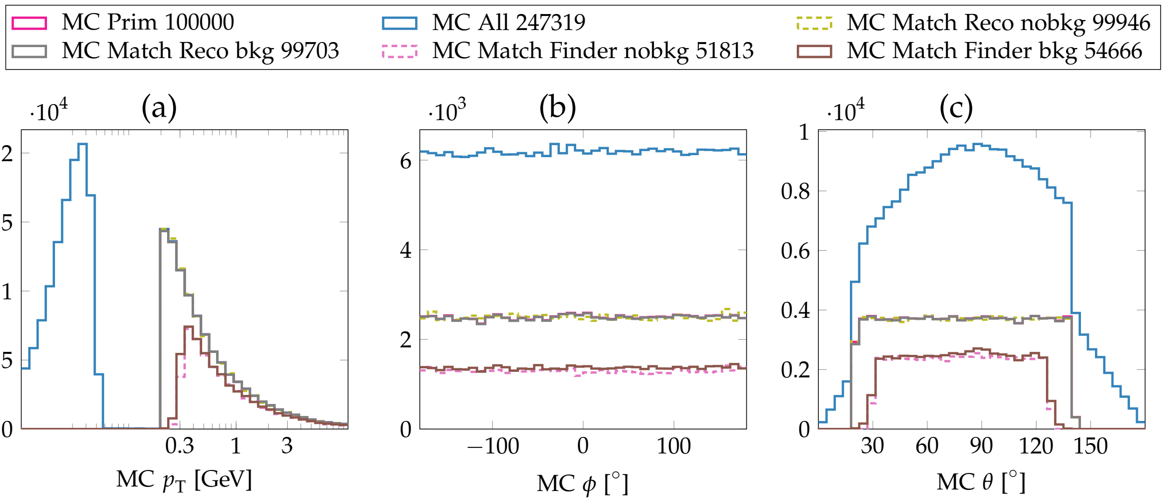


Figure 5.22.: Distributions of the MC track parameters p_T, ϕ, and θ in the test data sets L0 and L1 (with and without background see Tab. 5.8). "MC Prim" are the generated primary MC particles, while "MC All" are primaries as well as secondary particles. The distributions "MC Match Reco/Finder" are the MC track parameters of the primary MC particles with a match to a reconstructed track/2D finder track. The matching via a coincidence of TS hits is described in Appendix A.3.

Each of the test data sets listed in Tab. 5.8 contains 10^5 single $\mu^\pm$ track events, with vertices constrained to the IP. Here the track parameter distribution of the important test data sets L0 and L1 with the largest track parameter range L are presented in detail (see Tab. 5.8). The remaining test data sets contain smaller ranges and can thus be understood as subsets of the track parameter range L. The MC track parameter distributions in p_T, ϕ and θ are shown in Fig. 5.22.

These test data sets have the same boundary limits as the training data set discussed in Sec. 5.2.4. The only difference in the training data are the use of a different random seed, a larger sample size of 10^6 events, and a larger high-p_T limit in order to close the gap in the Hough-space (see Sec. 5.2.4). The existing 2D finder of the trigger and the reconstruction are added to the simulation. In the case of training on MC tracks, the selected training targets are the MC values of those primary MC particles with a match to a reconstructed track or to a 2D finder track. For that matter, the matching mechanism described in Appendix A.3 is used, which is based on

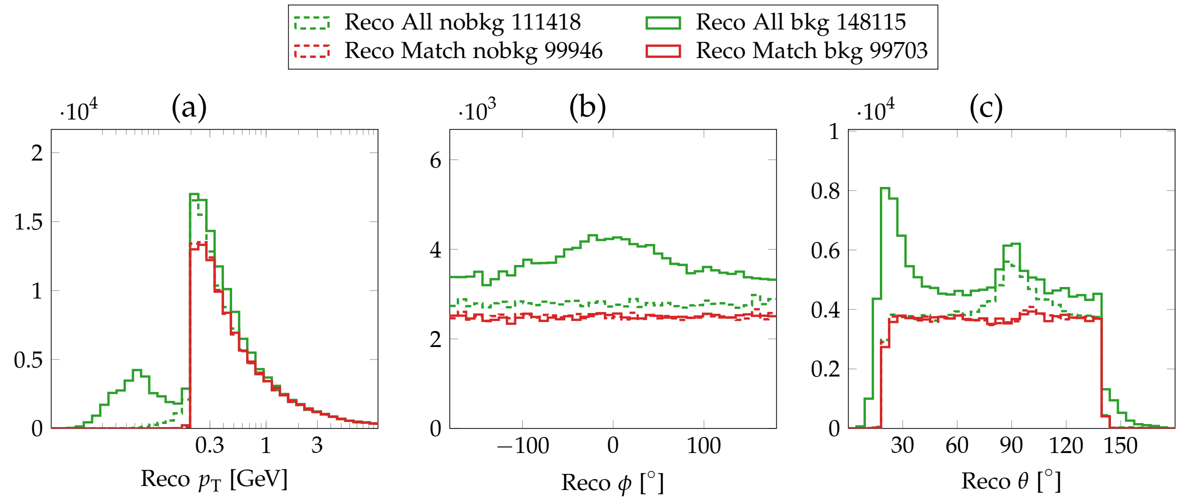


Figure 5.23.: Reconstructed track parameter distributions in the test data sets within the range L. The four curves show all reconstructed tracks as well as reconstructed tracks matched to MC tracks in a simulation with background and a simulation without background.

a confusion matrix of the track-to-hit relations.

Besides the simulated primary and secondary particles, Fig. 5.22 shows the MC track parameter distributions of the four different matched MC tracks used as training target for the 3D finders. The labels of the used training targets used later in Sec. 5.3.3 are (see Tab. 5.9): ("MC Match Reco nobkg", R0), ("MC Match Reco bkg", R1), (" MC Match Finder nobkg", T0), ("MC Match Finder bkg", T1).

After the simulated propagation of 10^6 primary particles through Belle II, a total of $\approx 2.47 \cdot 10^6$ MC particles after the detector simulation (primaries and secondaries). Since a small fraction of the secondaries may end up in the set of reconstructed tracks, the distribution of all the secondaries is included in Fig. 5.22. Since the background is only added after the simulated interactions with the detector material, the distributions of the primary and secondary MC particles is identical for the simulated cases with and without background. However, the reconstruction and the trigger is affected by background.

The distributions of the estimated track parameters from the offline reconstruction are shown in Fig. 5.23 and the distribution of tracks found by the 2D finder of the trigger in Fig. 5.24. Since the reconstruction and the trigger are affected by background these distributions are shown for both cases, with and without background. As shown in Fig. 5.23, in the case with background, the number of reconstructed tracks is $\approx 50\,\%$ larger than the number of primary MC particles, compared to an increase of only $\approx 10\,\%$ in a simulation without background.

However, the matched Reco distributions with and without background are in good agreement. This can be seen by comparing the number of matched recon-

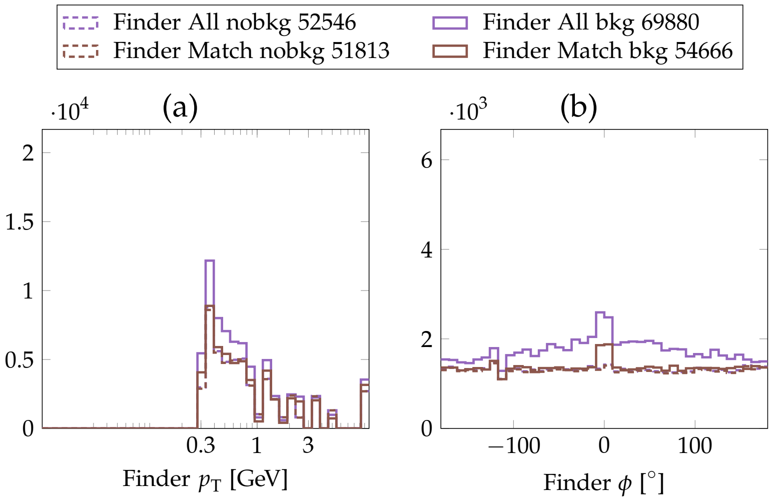


Figure 5.24.: Distributions of the track parameter estimates from the 2D finder in the test data sets with the range L. The four curves show all found 2D tracks as well as 2D tracks matched to MC tracks in a simulation with background and a simulation without background.

structed tracks to the number of primary MC particles in Fig. 5.22. The reconstruction efficiency drops only slightly from 99.9 % to 99.7 % only, which demonstrates the background robustness of the offline reconstruction. Figure 5.22 shows that nearly all primary MC tracks are reconstructed. Therefore, the shown curves for “MC Prim”, “MC Match Reco bkg”, and “MC Match Reco nobkg” are almost identical.

Due to the secondary tracks a large low-p_{T} peak appears for values of $p_{\mathrm{T}} \lessapprox 50\,\mathrm{MeV}$ in the “MC All” distribution in Fig. 5.22 (a). However, such a peak does not appear in the “Reco All” distributions in Fig. 5.23 (a). Since these tracks cannot even be tracked by the offline reconstruction, they are not relevant for the trigger studies. However, with background, an additional peak appears in the “Reco All” distribution in Fig. 5.23 (a) which corresponds to reconstructed low-p_{T} background tracks.

Small differences between the matched MC track distributions in Fig. 5.22 and the corresponding matched reconstructed track distributions in Fig. 5.22 can be seen for p_{T} and θ. The p_{T}-distribution of the matched Reco values is slightly smeared, which shows up as additional reconstructed tracks with $p_{\mathrm{T}} < 0.2\,\mathrm{GeV}$ and a slight abundance of reconstructed tracks wit $p_{\mathrm{T}} > 0.2\,\mathrm{GeV}$. In Fig. 5.23 (c) in the region close to $\theta = 90^\circ$ reconstructed tracks are depleted for $\theta < 90^\circ$ and enriched for $\theta > 90^\circ$. In contrast, the distribution of the matching MC partners is flat in θ. This indicates difficulties of the offline reconstruction with the correct handling of low-p_{T} curl-back tracks affecting the polar angle region of $\theta \approx 90^\circ$. Furthermore, Fig. 5.23 (c) shows a peak in the distribution of all Reco tracks at $\theta \approx 90^\circ$, which can be explained by a double counting of the low-p_{T} curl-back tracks. With background an additional Reco peak appears in the forward region $\theta \gtrapprox 17^\circ$. This peak corresponds to the correct distribution of background tracks in the polar angle θ. The background tracks appear in the ϕ-distribution in Fig. 5.23 (b) as well, where they show up as a broad peak around $\phi \approx 0^\circ$ on top of the simulated flat ϕ-distribution. This $\phi \approx 0^\circ$ is the direction pointing outwards in the accelerator plane.

The distributions of the estimated track parameters from the 2D finder are shown in Fig. 5.24. The binning of the 2D finder Hough plane in p_T^{-1} appears as fluctuations in the high-p_T region in Fig. 5.24 (a). A comparison with Fig. 5.22 (a) and (c) shows that the efficiency of the 2D finder is limited in p_T and in θ. In Fig. 5.24 (b), the 2D finder shows the same broad background peak in ϕ as seen in with the Reco tracks in Fig. 5.24 (b). However, the peak mainly vanishes after the matching with primary MC particles. In "Finder Match bkg", a peak with a small width remains at $\phi \approx 0^\circ$ which disappears in "MC Match Finder bkg". This indicates a possible small systematic error of the 2D finder Hough plane in the region around $\phi = 0^\circ$, where the 2D finder Hough plane is patched to a closed circle.

After all, the distributions of the matched tracks in Fig. 5.22, which are used as training targets, are flat in ϕ and θ. Furthermore, the additional background hits increase the low efficiency of the 2D finder, while the high efficiency of the reconstruction slightly decreases with background.

5.3.2. Analysis Method

The resolution and efficiency studies are based on a matching algorithm, where trigger tracks and reference tracks are matched based on a maximum coincidence of common TS hits. A detailed description of this matching mechanism is provided in the Appendix A.3. This matching algorithm uses a similar matching mechanism as the official Belle II tracking [67]. The resulting matched track pairs are used for the resolution studies, while the ratio of the matched track pairs to a set of all tracks is used for the efficiency studies. Matches between simulated MC particles and tracks from the official Belle II reconstruction are created by the reconstruction modules of basf2 which are provided by the Belle II tracking group. Therefore, the MC to reconstruction matches are assumed to be correctly created within basf2 and not recreated with the trigger matcher from Appendix A.3.

Note that an absolute efficiency of the matching itself cannot be defined, since this would require a non-existing set of correct matches. The finder efficiency introduced below are indeed based on different classes of matching tracks. Hence, one could fear an admixture of the hypothetical matching efficiency with the finder efficiency. Instead of efficiencies, in the matcher two reciprocal fractions are defined which describe how well a track in one class is represented by a track in the other class. A hypothetical matching efficiency would describe the fraction of tracks which are not matched although they should be matched. However, since the matching uses the criterion of common hits, unmatched tracks really should not be matched as they cannot describe the same track. Therefore, the matching mechanism is chosen as a basis for the definition of correct efficiencies. At this point, the question of a matching efficiency can be referred to the question of the correct hit relations. Matching inefficiencies can only occur if both track types are related to a subset of hits corresponding to the same track. See Appendix A.3 for further discussions on the matching.

The following classes of matching tracks are defined for the analysis of the track finders: `Candidate` tracks are the ones that should be found by the finders. In the case of MC reference tracks, the `Candidate` tracks are selected as matches of primary[1] MC particles with reconstructed tracks. By requiring a match with a reconstructed track, the selected primary particles are considered trackable. This means, they can be found by the reconstruction and therefore should be found by the track finder.

The class of `Correct` tracks are estimated trigger tracks from the track finder with a match to both, a primary MC particle and to a reconstructed track. For the matching of trigger tracks to MC tracks and for the matching of trigger tracks to reconstructed tracks, the matching described in Appendix A.3 is used. The third class of `Ghost` tracks describe finder tracks without any match, neither to a MC particle nor to a reconstructed track. The class `All Found` contains all tracks found by the finders regardless of any matching.

The resolutions are calculated from the set of `Correct` tracks. After calculating the differences of the finder estimates with the reference values, the RMS90 is used as a measure of the resolution. This RMS90 is the RMS on a trimmed data set, where the smallest and largest 5 % of the difference values are skipped in order to avoid a bias towards the outliers (see Appendix A).

Since the Hough space uses uniform bin-widths in $p_\mathrm{T}^{-1} \propto \omega$, the unit of the bin-widths in GeV^{-1}. Accordingly the measure for p_T is not calculated in GeV, but in GeV^{-1}. In contrast, a direct calculation of the p_T-differences would depend on the absolute value of p_T. This means the errors would increase with larger values of p_T, where the tracks actually approximate a straight line. The differences of the estimated finder values and of the matched reference values are calculated as follows:

$$\Delta\phi = \mathrm{RMS90}\left[\phi_\mathrm{Est} - \phi_\mathrm{Ref}\right] \qquad [^\circ] \tag{5.59}$$

$$\Delta\theta = \mathrm{RMS90}\left[\theta_\mathrm{Est} - \theta_\mathrm{Ref}\right] \qquad [^\circ] \tag{5.60}$$

$$\Delta p_\mathrm{T}^{-1} = \frac{\Delta p_\mathrm{T}}{p_\mathrm{T,Est}^2} = \mathrm{RMS90}\left[\frac{p_\mathrm{T,Est} - p_\mathrm{T,Ref}}{p_\mathrm{T,Ref}^2}\right] \qquad [\mathrm{GeV}^{-1}] \tag{5.61}$$

where the indices Est and Ref designate the estimated trigger track values and the reference values. Note that MC as well as reconstructed track parameter values can be used as reference values here.

The ϕ- and θ-differences are straight forward. In the inverse-p_T resolution Δp_T^{-1}, defined by Eq. 5.61, the p_T-differences are weighted by the inverse squares of the

[1] Secondary MC particles with a matching reconstructed track are back-scattered particles not pointing to the IP and thus they are skipped in this study.

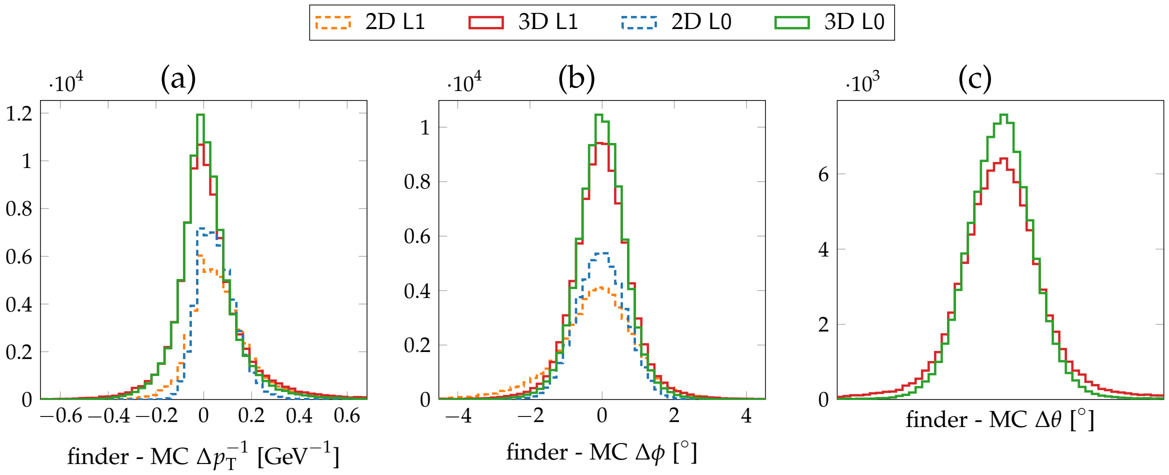


Figure 5.25.: Histograms of the differences between the estimated finder track parameters and the MC reference values using the test data sets L0 and L1 (see Tab. 5.8). In this example plot the estimates of the 2D finder and of the optimal 3D finder selected by the study in Sec. 5.3.3 are shown (label R1 in Tab. 5.9).

reference p_T values. This is identical to the difference of the inverse p_T weighted by $p_{T,Est}/p_{T,Ref}$.

$$\frac{p_{T,Est} - p_{T,Ref}}{p_{T,Ref}^2} = \left(p_{T,Est}^{-1} - p_{T,Ref}^{-1}\right) \cdot \frac{p_{T,Est}}{p_{T,Ref}}. \tag{5.62}$$

Example histograms of these sets of differences which are used for the calculation of the resolution are shown in Fig. 5.25. Note that the momentum resolution of the 2D finder shown in Fig. 5.25 (a) is slightly biased: it deviates from a normal distribution for values ≤ 0. Since the 2D finder Hough plane has a low-p_T limit of 0.3 GeV, those low-p_T tracks with $p_T \leq 0.3$ GeV have by construction systematically increased p_T-estimates.

The uncertainties of the resolutions are calculated as the RMS90 of the RMS90 (see Appendix A):

$$\text{Error[RMS]} = \sqrt{\frac{\mu_4 - \mu_2^2}{N\mu_2}} \tag{5.63}$$

where N is the number of track matches in each bin and μ_2, μ_4 are the second and fourth moment calculated on the trimmed data set[2]. By using the 4th moment, the

[2] The RMS approximates the root of the variance $\text{RMS}[x] \approx \sqrt{\text{Var}[x]}$ of a distribution with expectation value $\langle x \rangle = 0$. In terms of moments, the RMS can be expressed as $\text{RMS} \approx \sqrt{\text{Var}} = \sqrt{\mu_2}$, where the 2nd central moment μ_2 corresponds to the variance $\text{Var} = \langle (x - \langle x \rangle)^2 \rangle = \mu_2$.

used error is a measure of the kurtosis of the distribution and thus describes the deviation from a normal (Gaussian) distribution. Since plausible moments μ_2 and μ_4 can only be calculated if the set is not too small, the binned resolutions are only calculated in bins with $\geq$ 500 entries in the trimmed set. Note that the error is calculated on the trimmed set as well, where outliers are cut off. Hence, the error underestimates the error of the full RMS without trimming.

From the partition of the data sets into two distinct subsets S (signal) and F (fake), efficiencies and fake rates can be calculated. The rates corresponding to the fraction of S or of F in the total set and their errors are calculated by (see Appendix A):

$$\text{rate} = \frac{S}{S+F} \qquad\qquad \text{rateerr} = \sqrt{\frac{S \cdot F}{(S+F)^3}} \tag{5.64}$$

Using the above defined track classes `Candidate`, `Correct`, `Ghost`, and `All Found`, the subsequently used efficiency and fake rate can be defined. The efficiency rate corresponds to the ratio of the number of $S =$ `Correct` tracks with the number of $S + F =$ `Candidate` tracks. This means, the efficiency is the number found tracks out of the tracks that should be found. Additionally, the used fake rate corresponds to the ratio of the number of $S =$ `Ghost` tracks with the number of $S + F =$ `All Found` tracks. This means, the fake rate is the number of unmatched tracks out of all tracks found by the track finder.

$$\text{efficiency} = \frac{\texttt{Correct}}{\texttt{Candidate}} \qquad\qquad \text{fake rate} = \frac{\texttt{Ghost}}{\texttt{All Found}} \tag{5.65}$$

For a detailed derivation of the formulas for the resolution, the efficiency and fake rates and their errors, see Appendix A.

5.3.3. Study of the Training Target Tracks

In this study, the resolutions and efficiencies of differently trained 3D finders are measured and compared with the 2D finder. The goal of this study is to choose a type of reference tracks used as training target tracks and to verify the usefulness of background in the training data. Using the training method described in Sec. 5.2.4, four different 3D finders are trained with the subsequently described training options (see Tab. 5.9). In the training of the 3D finders, the 3D hit patterns in the Hough space are obtained from the track-to-hit relations in a training data set. This comprises the filling a 5D array with the track-to-hit relations and a following normalization to obtain the weight contributions of each single hit to each possible track cell in the 3D Hough space. All four 3D finders are trained on data sets with 10^6 single $\mu^{\pm}$ tracks using the training data set with parameters listed in Tab. 5.4.

A training on MC tracks with matching reconstructed tracks as well as a training on MC tracks with matching 2D finder tracks is carried out. In both cases, the training target values are MC track parameters, however, the subset of tracks is

3D Finder	Reference Tracks	Train Data Background
R0	Reco Tracks	no bkg
R1	Reco Tracks	bkg
T0	2D Tracks	no bkg
T1	2D Tracks	bkg

Table 5.9.: Labels for the four 3D finders with the different training options. A training on MC tracks with matching reconstructed tracks is compared to a training on MC tracks with matching 2D finder tracks. Each training is carried out with and without background. The track parameter range in the training data, listed in Tab. 5.4, has the same low-p_T and shallow-θ ranges as the test data range L in Tab. 5.7.

selected differently. Either a match to a 2D finder tracks or a match to a reconstructed track is required. In this way, the training data set contains only tracks which are already found by another track finder. Using the 2D finder matches, all tracks in the training data set actually can be found by a track finder based on TSs. However, the limited efficiency of the 2D finder is propagated to the newly trained 3D finder. An improved efficiency can be expected from a training on the MC tracks with matching reconstructed tracks. However, some of the reconstructed tracks in the training data set will have a small number of related TSs only. This means that some of the tracks in this training data set cannot be found by a track finder.

For each type of target track selection a training data set with and without background is used. Background introduces noise into the events, with the result of distorted TS information and additionally found pure background TSs (see Chapter 4). These distortions make the training data more realistic and an improved 3D finder performance can be expected from a training with background. Due to the required matching with primary MC particles, pure background tracks are not used for the training.

The test data sets used here have the large track parameter range containing low-p_T and shallow-θ tracks. In Tab. 5.8 these test data sets are listed as L1 for the case with background and as L0 for the case without background. Since the boundary region of the CDC acceptance is reached in p_T and θ, large differences between the 2D finder and the 3D finder can be expected. In the test runs for this study, the trained 3D finders use the configuration parameters listed in Tab. 5.6.

Efficiencies

Averaged efficiencies and fake rates of the finders measured on the test data sets are listed in Tab. 5.10. In the case with background, the best averaged 3D finder efficiency is achieved with R1, followed by R0, T1, and T0. The reason is that a training on tracks found by the reconstruction in a sample with background results in an

Finder	Data	ϵ	g
R0	bkg	92.110 ±0.171	30.641 ±0.241
T0	bkg	90.898 ±0.182	28.550 ±0.241
R1	bkg	92.506 ±0.167	31.929 ±0.240
T1	bkg	91.873 ±0.173	31.391 ±0.241
2D	bkg	54.814 ±0.315	13.220 ±0.256
R0	nobkg	91.433 ±0.177	3.507 ±0.118
T0	nobkg	89.400 ±0.195	3.297 ±0.116
R1	nobkg	91.283 ±0.178	3.549 ±0.119
T1	nobkg	90.353 ±0.187	3.360 ±0.117
2D	nobkg	51.836 ±0.316	1.060 ±0.089

Table 5.10.: Efficiencies ϵ and fake rates g of the 3D finders with different training options and the 2D finder tested on a data set with ranges L (see Tab. 5.8).

improved generalization of the finder compared to a training on a subset of tracks found by the 2D finder. In addition, an improved generalization is expected from a training with background compared to a training without background. A huge difference can be seen between the efficiencies of the 2D finder and of the four 3D finders compared to only small differences among the 3D finders with the different training options. Correlated with this efficiency improvement, the 3D finders show also an increased fake rate g (g for ghost) compared to the 2D finder.

The efficiencies ϵ and fake rates g (g for ghost) as function of the MC track parameters $p_{\mathrm{T}}, \phi, \theta$ are shown in Fig. 5.26.

Due to the requirement of a minimum number of TS hits per track, additional background hits result in an increased efficiency. However, background TSs enter the found tracks. Therefore, this efficiency increase is accompanied by a loss in resolution, as will be discussed subsequently with Tab. 5.11. The fake rate has roughly the inverse background dependent structure of the efficiency. The efficiency ϵ as a function of the true MC p_{T} is shown in Fig. 5.26 (a). In the high-p_{T} region the 3D finders reach $\epsilon \approx 95\,\%$ and the 2D finder $\epsilon \approx 80\,\%$. Due to the wide range in θ, the high-p_{T} efficiencies are still limited by the θ acceptance region. In the low-p_{T} region, track finding is less efficient with all finders. However, the 3D finder efficiency loss starts at a lower p_{T} value of $p_{\mathrm{T}} \approx 0.25\,\mathrm{GeV}$ compared to the 2D finder efficiency loss at $p_{\mathrm{T}} \approx 0.35\,\mathrm{GeV}$. Thus, the 3D finders have a better efficiency for low-p_{T} tracks.

Figure 5.26 (d) shows a low fake rate g in the high-p_{T} region increasing towards the low-p_{T} region. The 2D finder has the lowest (best) g of about 15 % with background and about 1 % without background. However, in the high-p_{T} region the 3D finder fake rates are similar to the 2D finder fake rates. With background, the 3D finder fake rate is lowest with T0, highest with R1 and in between with T1, R0. This is plausible, as training with the strongly skimmed data set for T0 without background creates hit curves with a smaller width in the track parameter phase space. Below the low-p_{T}

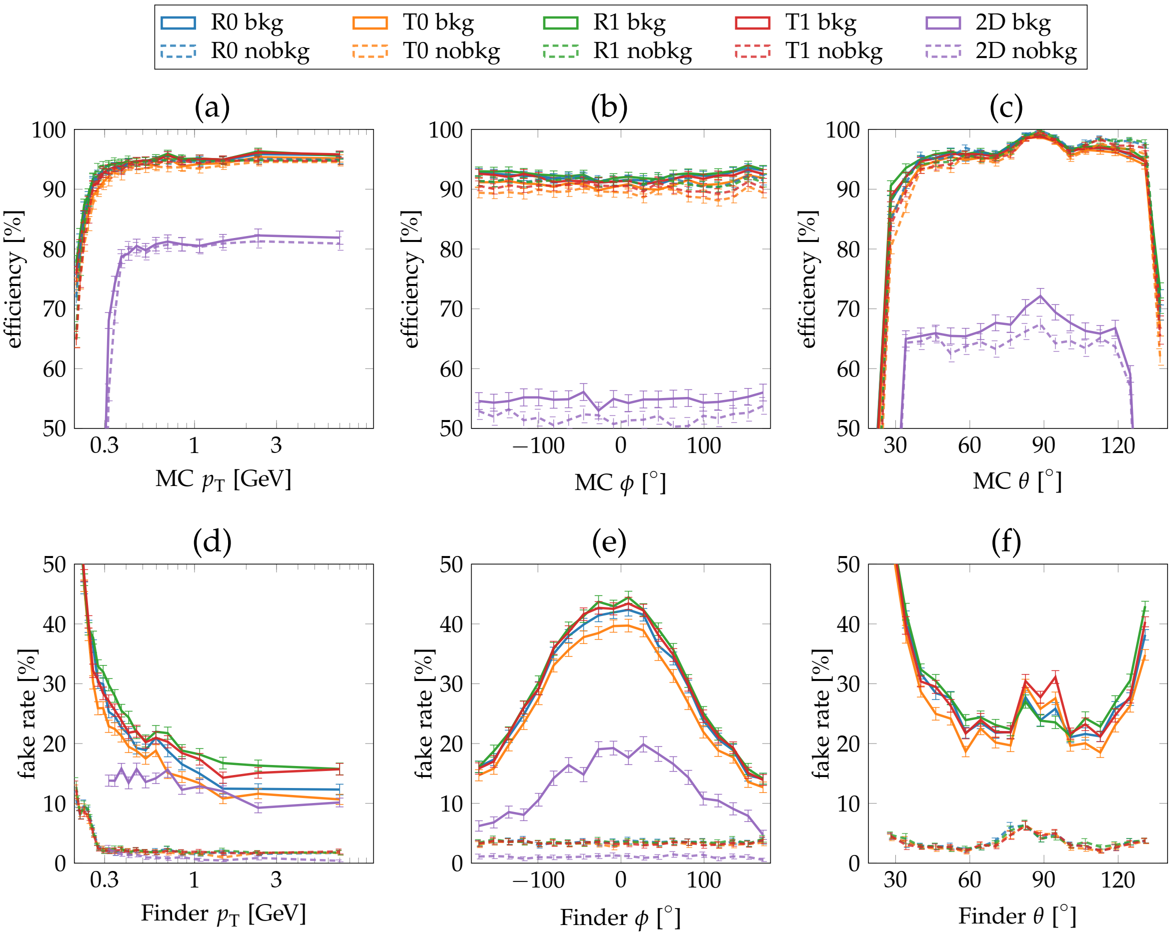


Figure 5.26.: Track finding efficiencies and fake rates of the 3D finders with four different training options and of the 2D finder. Each track finder is tested on the data sets L0 and L1 listed in Tab. 5.8.

limit that can be estimated by the 2D finder of $p_T \lessapprox 0.35\,\text{GeV}$ a large increase in the fake rate g can be seen for the 3D finders in Fig. 5.26 (d). The gain in efficiency of the 3D finders in the low-p_T region, compared to the 2D finders where $\epsilon \approx 0$, comes at the cost of a high fake rate in this low-p_T region. For this reason, the averaged fake rate of the 3D finders in Tab. 5.10 is significantly increased, biased by the low-p_T and shallow-θ boundary regions, where the 2D finder is inefficient (see Tab. 5.10). In contrast, in the high-p_T region in Fig. 5.26 (d) for the case with background, the 3D finders and the 2D finder have similar fake rates.

Fig. 5.26 (b) shows that the efficiencies ϵ as functions of the MC ϕ are almost flat and agree with the values listed in Tab. 5.10. The related fake rates g as functions of the estimated ϕ-values are shown in Fig. 5.26 (e). They are flat without background and show a broad peak at $\phi \approx 0^\circ$ with background. This is in agreement with the broad ϕ peak in the distribution of reconstructed tracks in Fig. 5.23 (b). Therefore, some of the fake tracks correspond to correctly found background tracks. The worst fake rate with background is reached by the finders T1 and R1 with a peak value of $g \approx 45\,\%$. Also the 2D finder shows the background peak around $\phi \approx 0^\circ$ with a fake rate of $g \approx 20\,\%$.

As functions of the MC θ-values, the efficiencies of the 3D finders in Fig. 5.26 (c) show a peak at $\theta \approx 90^\circ$, which is more pronounced in the case with background. In the case with background, also for the 2D finder a clear peak around $\theta \approx 90^\circ$ appears. In the subsequent study with different test data ranges in Sec. 5.3.4, this peak will be identified as low-p_T curl-back tracks. This indicates that the combination of low-p_T curlers and background can affect the 2D finder as well. The larger θ-acceptance region of the 3D finders compared to the 2D finder prominently appears in Fig. 5.26 (c). With background, the 3D finder efficiencies are slightly increased for tracks in the forward region ($\theta < 60^\circ$) and slightly decreased for tracks in the backward region ($\theta > 110^\circ$).

The fake rates g of the 3D finders as functions of the estimated polar angle θ are shown in Fig. 5.26 (f). With background the fake rate g clearly increases at the θ-boundaries and shows a peak at $\theta \approx 90^\circ$. Without background the fake rate g in Fig. 5.26 (f) shows the same structure but it is less pronounced. At shallow-θ angles, R1 and T1 have the largest fake rate g and T0 the lowest. Around the peak at $\theta \approx 90^\circ$ the ordering is changed: R1 and R0 have the lowest g and T1 the largest. Since the peak at $\theta \approx 90^\circ$ is least pronounced with R1, this finder shows the best robustness towards the background enhanced increase of the fake rate close to $\theta \approx 90^\circ$. This indicates that this excess of fake tracks around $\theta \approx 90^\circ$ can be avoided by representing a sufficient amount of the low-p_T and shallow-θ tracks in the training data.

Resolutions

The track finding resolutions calculated as the RMS90 are listed in Tab. 5.11. As described in Sec. 5.3.2, these resolutions are calculated from the differences of the estimated track parameter values with the matched MC track parameter values. As expected, for all finders background results in a worse resolution. Table 5.11 shows no significant differences among the resolutions of the four 3D finders R0, R1, T0, and T1. However, the 3D finders achieve an equal or better resolution than the 2D finder. As shown in Tab. 5.11 in the case without background the 2D finder achieves a similar resolution as the 3D finders.

Considering the improved 3D finder efficiency, this result is notable. This means, the 3D finder resolutions the high-p_T and central-θ region are actually much better compared to the 2D finder resolutions. Only after including all the additional low-p_T and shallow-θ, the 3D finder resolution is decreased to the same level as the 2D finder resolution. The performance on the test data sets with the range N, where both finders are fully efficient, is presented subsequently in Sec. 5.3.4.

The resolutions as functions of the MC track parameters are shown in Fig. 5.27. As expected, the p_T dependent resolutions in Fig. 5.27 (a), (b), and (c) are better in the

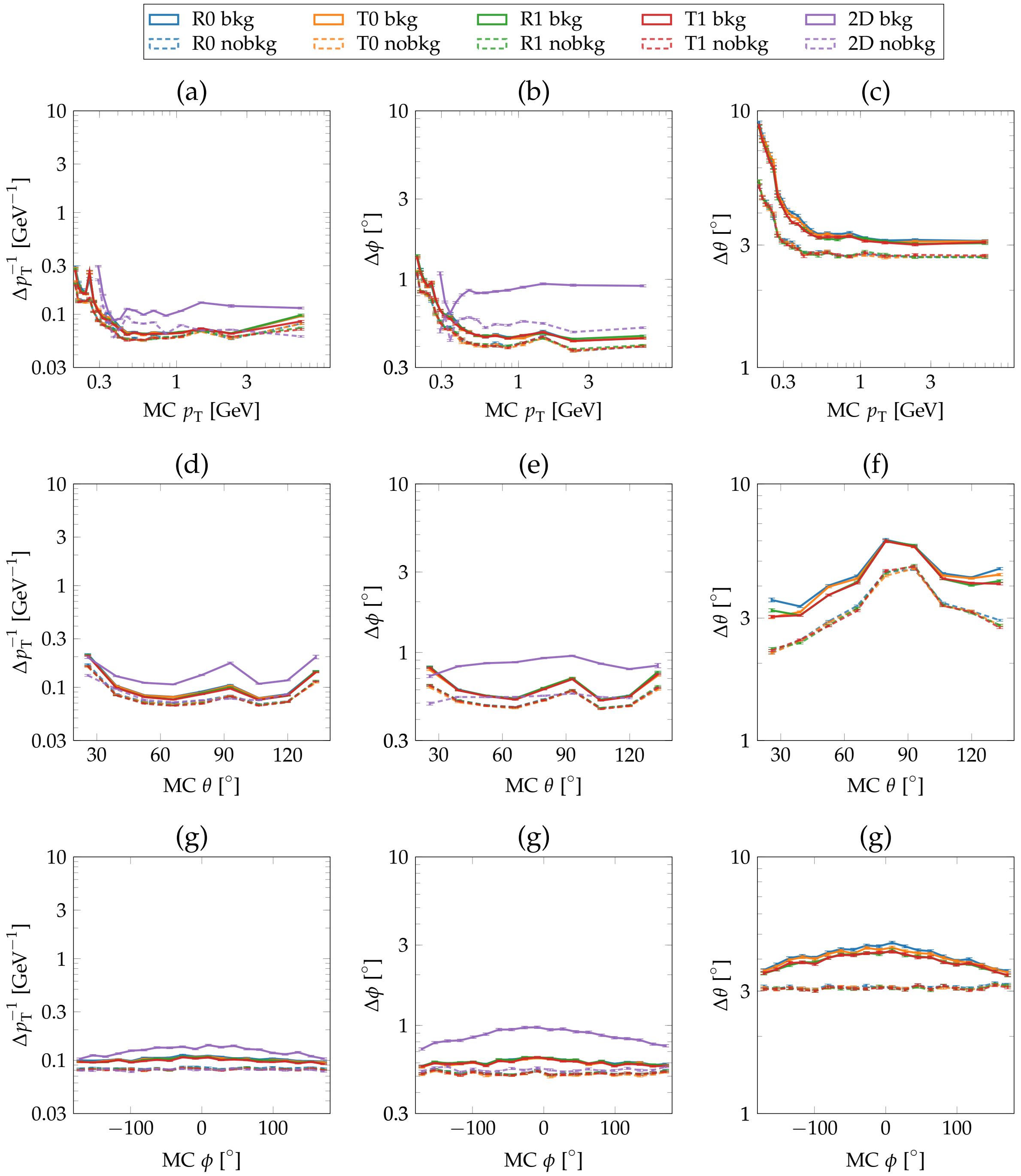


Figure 5.27.: Track parameter resolutions of the track finders as functions of the true MC track parameters. The 3D finders are trained with the different training options listed in Tab. 5.9. All finders are tested on the data sets L0 and L1 listed in Tab. 5.8.

Finder	Data	$p_\mathrm{T}^{-1}[10^{-1}\,\mathrm{GeV}^{-1}]$	$\phi[^\circ]$	$\theta[^\circ]$
R0	bkg	1.035 ±0.003	0.609 ±0.002	4.095 ±0.012
T0	bkg	1.011 ±0.003	0.600 ±0.002	4.018 ±0.011
R1	bkg	1.002 ±0.003	0.609 ±0.002	3.900 ±0.011
T1	bkg	0.987 ±0.003	0.600 ±0.002	3.902 ±0.011
2D	bkg	1.225 ±0.005	0.871 ±0.003	
R0	nobkg	0.839 ±0.003	0.522 ±0.001	3.117 ±0.008
T0	nobkg	0.817 ±0.002	0.513 ±0.001	3.089 ±0.008
R1	nobkg	0.820 ±0.002	0.524 ±0.001	3.076 ±0.008
T1	nobkg	0.812 ±0.002	0.519 ±0.001	3.075 ±0.008
2D	nobkg	0.806 ±0.003	0.547 ±0.002	

Table 5.11.: Resolutions of the four trained 3D finders and of the 2D finder. Each finder is tested on the test data sets L0 without background and L1 with background with the ranges shown in Tab. 5.8.

high-p_T region, where the tracks approximate a straight line, and decrease towards the low-p_T region, where the tracks have a large curvature. In the low-p_T region for values below $p_\mathrm{T} = 0.35\,\mathrm{GeV}$ the 2D resolution breaks down, as can be seen in Fig. 5.27 (a) and (b). In contrast, in this low-p_T region the 3D finders still provide reasonable results. For high-p_T tracks without background, the 3D finders achieve similar p_T- and improved ϕ-resolutions as the 2D finder. However, in the case with background Fig. 5.27 (a) and (b) show a significant resolution loss of the 2D finder in the high-p_T region, while the 3D finder resolutions appear to be unaffected by background. Since the resolution of the 3D finders with the different training data sets is almost indifferent, the reason for this background robustness of the resolution has to be attributed to the training of the hit patterns. In contrast, the 2D finder uses an analytic calculation to obtain the hit patterns in the 2D Hough space.

In bins of θ, the p_T^{-1}-resolution in Fig. 5.27 (d) of all finders decreases for shallow-θ angles, in the forward and in the backward region. Additionally, a peak appears around $\theta \approx 90^\circ$ with the 2D finder as well as the 3D finders. In the case with background, this peak appears to be less pronounced with the 3D finders. This peak corresponds to the low-p_T curl back tracks. Due to the additionally created TSs after the curling, the efficiency for these tracks is increased but the resolution is decreased accordingly (compare Fig. 5.26).

In Fig. 5.27 (e) the 2D ϕ-resolution as a function of θ appears to be almost flat in the case without background, with a slight improvement for shallow-θ angles. With background a peak at $\theta \approx 90^\circ$ appears. The $\Delta\phi$-distributions for the 3D finders have the same structure with and without background, showing a peak around $\theta \approx 90^\circ$ as well. This resolution loss in $\Delta\phi(\theta)$ is again an effect of the low-p_T curl-back track. As a result of the improved efficiency for tracks with shallow-θ angles, this shallow-θ region is accompanied by a resolution loss of the 3D finder. The $\Delta\theta(\theta)$

plot in Fig. 5.27 (f) has the same θ-dependent structure as seen in Fig. 5.27 (d) and (e).

In bins of ϕ, the p_T-, ϕ-, and θ-resolutions in Fig. 5.27 (g), (h), and (i) are all flat in the case without background and have a broad peak around $\phi \approx 0^\circ$ in the case with background. Due to the ϕ-distribution of the background, such a ϕ-dependence of the resolutions is expected. The ϕ-distribution of the background corresponds to the distribution of all reconstructed tracks on top of the reconstructed tracks with a matching MC track in Fig. 5.23 (b).

Selection of the 3D Finder Training Data

The standalone training of the 3D finders on MC track parameter values with matching tracks from the offline reconstruction has turned out to be successful. There is no significant loss in efficiency or resolution compared to a training on MC tracks with matches to 2D finder tracks. Since the training with the reconstructed track matches, used by the finders R0 and R1, is independent of the limited 2D finder efficiency, training with matching reconstructed tracks will be used for the following studies. Furthermore, the overall best efficiency is reached by R1 using a training on reconstructed tracks with background. The improvement by using reconstructed tracks is plausible as more tracks with low-p_T and with shallow-θ angles are represented in the training data. The background makes the training data more realistic as it introduces fuzziness in the form of randomly superimposed noise hits. Since the best generalization can be expected by the training R1 it is suggested to use matches with reconstructed tracks as training targets and to include background in the training data.

5.3.4. Performance at the CDC Acceptance Boundaries

The previous study was carried out on test data with the large data range L listed in Tab. 5.8. This range L contains tracks at both boundaries of the CDC acceptance determined in see Sec. 5.3.1, namely low-p_T tracks as well as shallow-θ tracks in the forward and backward direction. In this study, the three additional data ranges P, W, and N, listed in Tab. 5.8, are used to disentangle the effects of each boundary region separately. As described in Sec. 5.3.1, the labels of the datasets in Tab. 5.8 are combined out of the range label followed by a number indicating the background: 0 for no background, 1 for background. The used background corresponds to the full Phase 3 background using the official background campaign 15 mixing files. Details on the different background campaigns and on the method of background mixing are described in Chapter 4.

Selected by the previous study in Sec. 5.3.3, only a single 3D finder training option is used: training on primary MC tracks with matching reconstructed tracks and with background in the training data (previously labelled R1 in Tab. 5.9). Using this 3D finder and the official 2D finder, at first the test results on the data sets N0 and N1, containing tracks within the full CDC acceptance region, are inspected. A direct comparison with the data sets L0 and L1, containing tracks at both boundary regions, shows which effects in the efficiencies and in the resolutions can be attributed to the boundary regions. Within the data range N, the 2D finder is almost fully efficient. Hence, the results using the data range N are used to verify that there is no performance loss of the 3D finder compared to the 2D finder.

In order to disentangle the effects of the low-p_T and of the shallow-θ region, performance measurements are carried out with the data sets in the ranges P and W in addition. Performance measurements using the test data sets P0 and P1 demonstrate the effect of low-p_T tracks, unaffected by the boundary region in θ. The test data sets W0 and W1 are used to test the effect of tracks with shallow-θ angles, unaffected by the low-p_T region.

Efficiencies in the Full Acceptance and at the Boundaries

Using the test data sets N0 and N1 the efficiencies and fake rates in the full CDC acceptance region, are shown in Fig. 5.28. In order to clearly compare these results to the results in both boundary region, the efficiencies and fake rates on the test data sets L0 and L1 are repeated in Fig. 5.29. Table 5.12 shows the averaged efficiency and fake rate values on those test data sets. Note that the result on the sets L0 and L1 were already included in the previous study in Fig. 5.26 and the values listed in Tab. 5.10.

Due to the limited ranges in p_T and θ in the test data sets N0 and N1, there are no reference tracks beyond the low-p_T and shallow-θ limits in Fig. 5.28. This means, no efficiency data points appear beyond those limits and all tracks correctly found beyond those limits are background tracks. This absence of efficiency points can be seen in Fig. 5.28 (a) in the low-p_T region and in Fig. 5.28 (c) in the shallow-θ

Finder	Data	ϵ	g
2D	N1	98.199 ±0.084	8.228 ±0.164
3D	N1	99.509 ±0.044	29.822 ±0.236
2D	L1	54.814 ±0.315	13.220 ±0.256
3D	L1	92.506 ±0.167	31.929 ±0.240
2D	N0	99.716 ±0.034	0.001 ±0.002
3D	N0	99.957 ±0.013	0.113 ±0.021
2D	L0	51.836 ±0.316	1.060 ±0.089
3D	L0	91.283 ±0.178	3.549 ±0.119

Table 5.12.: Track finding efficiencies and fake rates for the test data sets N0 and N1 in the full acceptance region and in L0 and L1 at the boundaries of the CDC acceptance (see Tab. 5.8).

region. In Fig. 5.28 (d) and Fig. 5.28 (f) appear the correlated significant increases of the fake rates. The reason, why the fake rate beyond this limits is not at 100 % is the resolution of the track finders. Some of the tracks found outside of the test data range set can be matched to an MC track. Hence, the efficiencies and fake rates in Fig. 5.28 should only be considered to be correct within the central region corresponding to the range N.

In contrast, in Fig. 5.29, the ranges shown in the plots correspond to the simulated ranges in the test data set. Therefore, the results in Fig. 5.29 showing test data in the range L complement the results in Fig. 5.28a in the boundary region. Especially in the boundary regions, the correct efficiencies and fake rates can only be read off Fig. 5.29 (a) and (d) for the low-p_T region and Fig. 5.29 (c) and (f) for the shallow-θ region.

In the range N0, the 2D finder and the 3D finder reach $\epsilon \approx 100\,\%$ (see Tab. 5.12). The 3D finder has a better background robustness as achieves an efficiency of $\epsilon_{3\mathrm{D}} \approx 99.5\,\%$ in the range N1. However, the 2D efficiency with $\epsilon_{2\mathrm{D}} \approx 98.2\,\%$ is still very good. This means, the 2D finder is very efficient for tracks fully contained within the CDC acceptance region. Correspondingly, all efficiencies in the central CDC acceptance region shown in Fig. 5.28 are close to 100 %. Note that the scale of the efficiency axis in Fig. 5.28 starts with a lower limit of $\epsilon = 92\,\%$.

Within this shown high-efficiency limit, the dependence on the track curvature still appears in Fig. 5.28 (a) as a drop in efficiency towards the low-p_T region. Figure 5.29 (a) shows the corresponding efficiency drop on a larger scale for tracks with lower-p_T. Since the high-p_T efficiency in Fig. 5.28 (a) reaches almost 100 %, the limited efficiency in the high-p_T region in Fig. 5.29 (a) is a result of the tracks in the boundary regions. The efficiencies in the central-θ region shown in Fig. 5.28 (c) are flat. This means, the structure in Fig. 5.29 (c) for tracks with $\theta \approx 90^\circ$ is created by tracks at the boundary regions as well.

Interestingly, in Fig. 5.28 (b) appears a ϕ-dependency of the efficiencies, which is

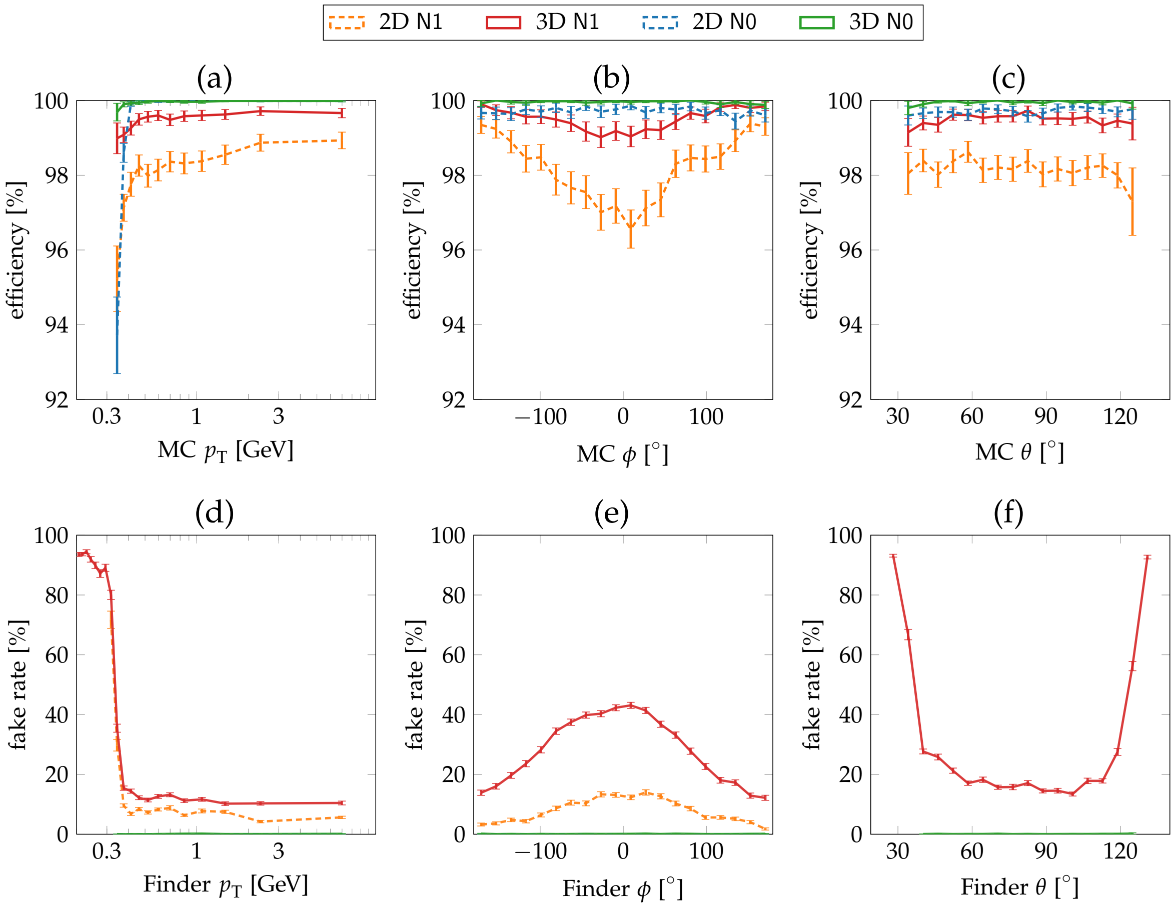


Figure 5.28.: Track finding efficiencies and fake rates of the 3D finder and the 2D finder using the test data sets N0 and N1 where the simulated track parameter ranges are fully contained it the geometrical CDC acceptance region (see Tab. 5.8).

not visible in Fig. 5.29 (b) due to the larger track parameter ranges. Especially the 2D finder is affected by the background tracks: the 2D efficiency drops by almost 4 % around $\phi \approx 0^\circ$ while the 3D finder efficiency loss is only about 1 % in this region.

Fake rates in Fig. 5.28 (e) appear only in the case with background, while small fake rates can also be observed in Fig. 5.29 (e) in the case without background. This means, the tracks in the boundary region are more likely to generate additional fake tracks. Since the same background is used, the fake rate distributions with background in Fig. 5.28 (e) and Fig. 5.29 (e) are very similar.

Efficiencies at the Separate Low-p_T and Shallow-θ Boundaries

The efficiency and fake rate results containing tracks in the low-p_T data sets P0 and P1 are shown in Fig. 5.30 and the results in the shallow-θ data sets W0 and W1 in

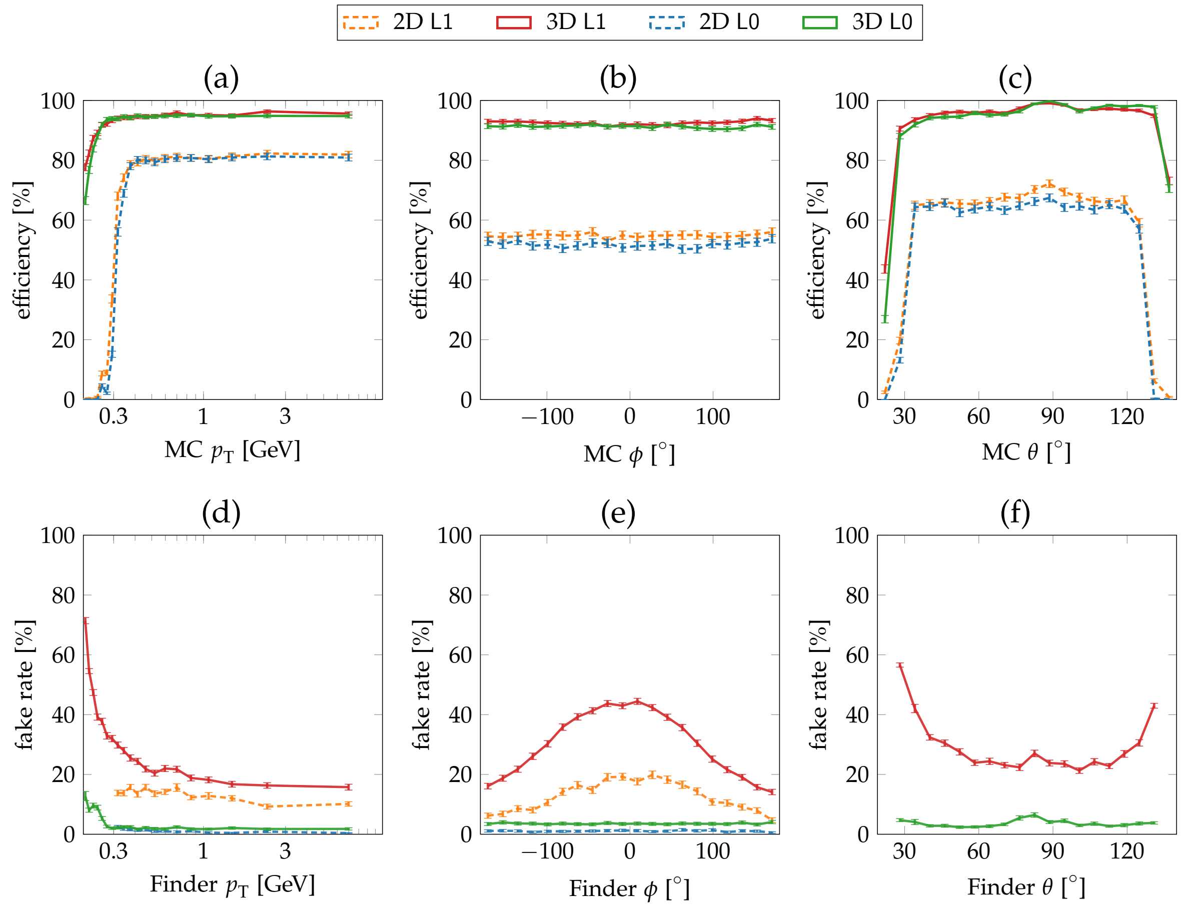


Figure 5.29.: Track finding efficiencies and fake rates of the 3D finder and the 2D finder using the test data sets L0 and L1 with the large track parameter ranges in p_{T} and θ (see Tab. 5.8).

Fig. 5.31. Averaged efficiency and fake rate values in these test data sets are listed in Tab. 5.13. As separate components of the boundary regions combined in the data set L, the results are compared to Fig. 5.29.

The efficiency in the high-p_{T} region is better on the data sets within the θ-acceptance region (N, P) compared to the wider θ-range W. In the high-p_{T} region in Fig. 5.30 (a), all finders reach almost 100 % efficiency. In contrast, the high-p_{T} efficiency in Fig. 5.31 (a) is significantly reduced and similar to the reduced high-p_{T} efficiency in Fig. 5.29 (a). Therefore, the lowered high-p_{T} efficiency in L is identified as tracks with shallow-θ angles. Since the 3D finder has an improved efficiency for shallow-θ tracks, the efficiency loss of the 3D finder in Fig. 5.31 (a) is less pronounced. Using the values in Tab. 5.13, the average efficiency loss due to shallow-θ tracks of the 3D finder is about 5 %, while the loss of the 2D finder is about 20 %.

Due to the limited p_{T} range in W, the efficiencies in the low-p_{T} region in Fig. 5.31 (a) correspond to the low-p_{T} region in the range N shown in Fig. 5.28 (a). In the full

Finder	Data	ϵ	g
2D	P1	67.154 ±0.297	12.517 ±0.229
3D	P1	96.856 ±0.110	32.569 ±0.235
2D	W1	80.647 ±0.250	9.004 ±0.187
3D	W1	95.191 ±0.135	29.933 ±0.241
2D	P0	64.864 ±0.302	1.298 ±0.088
3D	P0	96.599 ±0.115	4.806 ±0.132
2D	W0	80.332 ±0.251	0.000 ±0.000
3D	W0	94.729 ±0.141	0.086 ±0.019

Table 5.13.: Track finding efficiencies and fake rates for different test data sets listed in Tab. 5.8.

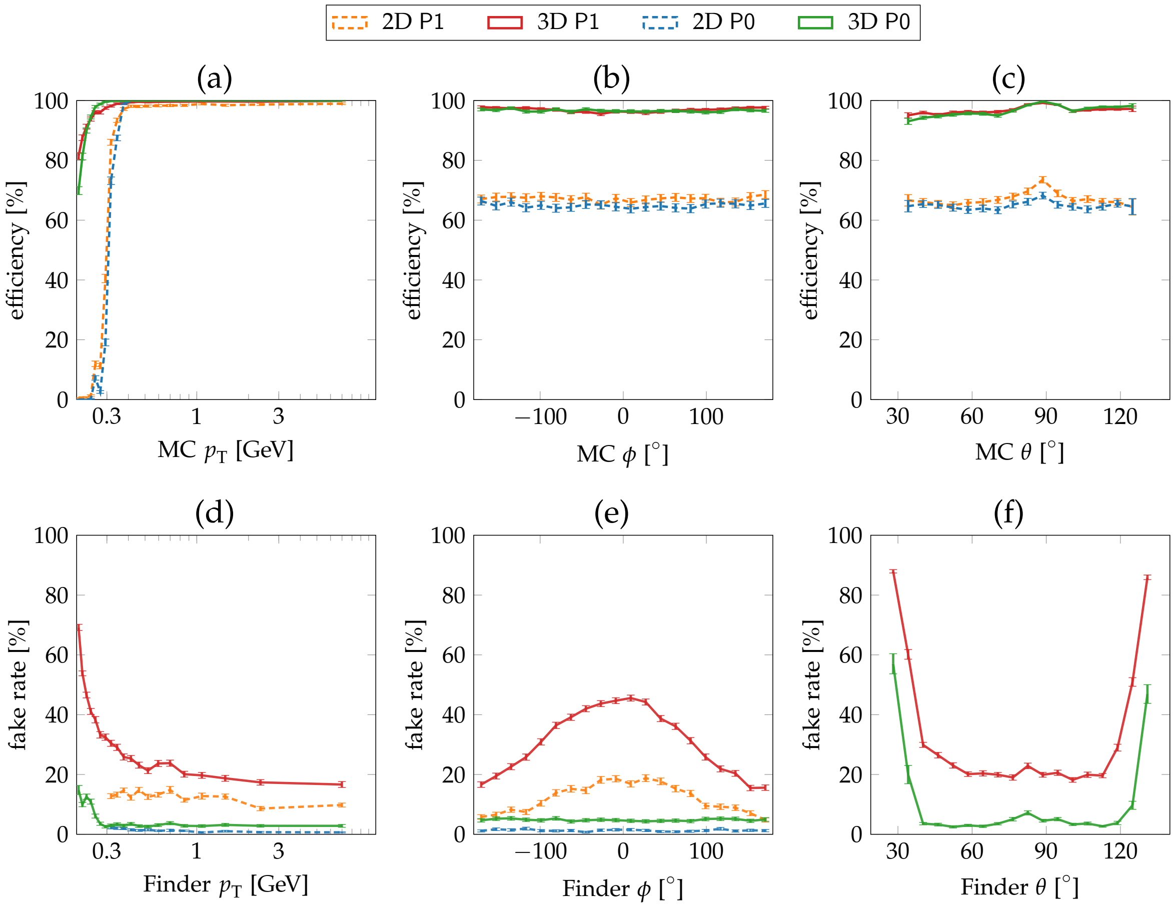


Figure 5.30.: Track finding efficiencies and fake rates in the low-p_T boundary region using the test data sets P0 and P1 listed in Tab. 5.8.

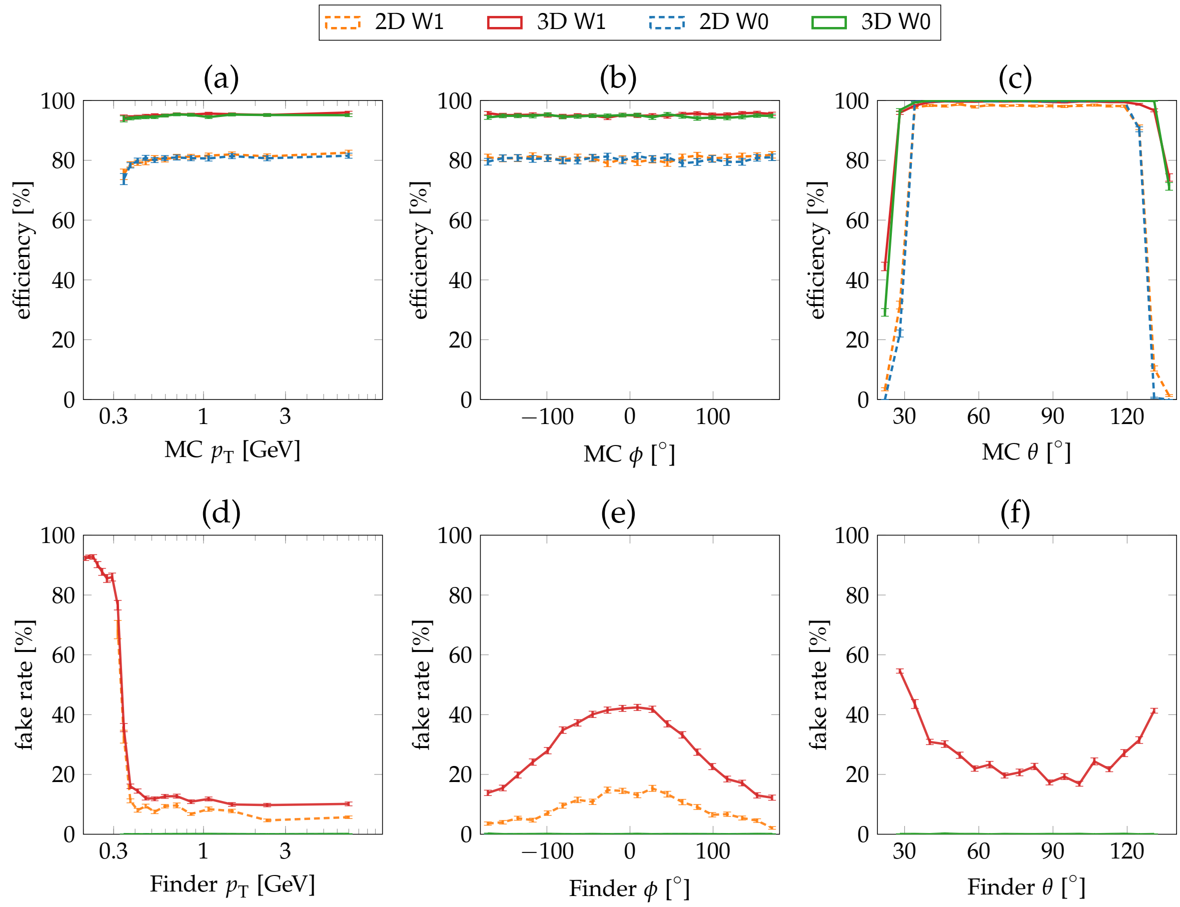


Figure 5.31.: Track finding efficiencies and fake rates of the 3D finder and the 2D finder tested within the test data sets W0 and W1 (see Tab. 5.8).

p_T range, the related fake rates with the range W shown in Fig. 5.31 (d) are similar to the fake rates in range N shown in Fig. 5.28 (d). This means, the additional shallow-θ tracks are not a source for p_T dependent fake tracks.

In analogy, the low-p_T efficiencies and fake rates in the range P shown in Fig. 5.30 (a) and (d) correspond to the range L shown in Fig. 5.30 (a) and (d). Additionally, the fake rates also agree in the high-p_T regions. This is especially seen in the case without background, where fake tracks appear in the high-p_T region. This means, the low-p_T tracks in the test data are a source of high-p_T fake tracks. Since the track finder only searches for outwards going tracks, this is plausible for low-p_T curl-back tracks. The additional TS hits on the curl-back trajectory are found as an apparent second track. Due to resolution of the track finders and enhanced by the fuzziness introduced via the curling, the chance to find high-p_T fake tracks is increased.

By inspecting the central-θ region, similar properties can be identified. The efficiency for the central-θ region, shown for the range W in Fig. 5.31 (c), is flat and reaches almost 100 %, just as in the case of the range N in Fig. 5.28 (c). In contrast, for the range P shown in Fig. 5.30 (c) appears a peak for tracks with $\theta \approx 90^\circ$, similarly to the peak for the range L in Fig. 5.29. Hence, it can be concluded that the peak in the efficiency for $\theta \approx 90^\circ$ is a result of the low-p_T tracks. This is plausible, as the curl-back of low-p_T tracks within the CDC volume requires central polar angles. Towards the shallow-θ region, the low-p_T tracks leave the CDC volume before curling back.

The same property is also seen for the fake rates in Fig. 5.30 (f) and in Fig. 5.29 (f), where the low-p_T curlers are the reason for the peak in the fake rate around $\theta \approx 90^\circ$. Accordingly, such a fake rate peak is absent in the range W shown in Fig. 5.31 (f) and in the central-θ region in the range N shown in Fig. 5.28 (f). Without background, the absence of the peak is obvious. With background, the reason is the θ-distribution of the background tracks, which can be seen from the distribution of all reconstructed tracks in Fig. 5.23 (c). The polar angle distribution of the background tracks corresponds to the difference of "Reco All bkg" with "Reco All nobkg" in Fig. 5.23 (c). Although many reconstructed background tracks (reconstructed tracks without a match) have low-p_T (see Fig. 5.23 (a)), the polar angle distribution of the background shows a peak in the shallow-θ region. Furthermore, the comparison with Fig. 5.23 (c) shows that the offline reconstruction is dealing with the same problem of curling low-p_T tracks with $\theta \approx 90^\circ$.

The flat distribution of the fake rate without background in Fig. 5.30 (c) can be related to the flat distribution of fake tracks in the high-p_T region in Fig. 5.30 (a). Those fake tracks with arbitrary high-p_T values also have arbitrary θ values. This confirms the expectation that these tracks are created by the increased TS hit occupancy in the CDC.

The ϕ-distributions of the efficiencies in Fig. 5.31 (b) and Fig. 5.30 (b) are all flat, with the different average values listed in Tab. 5.13. Due to the already discussed low-p_T effect, in the case without background the ϕ-distributions of the fake rates in the range P shown in Fig. 5.30 (b) correspond to the range L shown in Fig. 5.29 (b). As expected, in the case with background all ϕ-distributions are in agreement.

While the 3D finder has a larger track finding efficiency, the 2D finder achieves lower fake rates. However, considering the track parameter distribution of the background with additional low-p_T and shallow-θ tracks, this is an effect of the limited acceptance region of the 2D finder. Many of the background tracks are simply not found by the 2D finder due to its limitations in p_T and θ.

Resolutions in the Full CDC Acceptance and at the Boundaries

The resolutions of the estimated track parameters in the full CDC acceptance region using the test data sets N0 and N1 are shown in Fig. 5.32. For a direct comparison with the boundary regions, Fig. 5.33 shows the resolutions using the test data sets L0 and L1. The averaged resolution values are summarized in Tab. 5.14.

Overall, the 3D finder achieves better resolutions compared to the 2D finder. The largest improvement of the track parameter resolutions for p_T^{-1} and ϕ, by almost a

Finder	Data	$p_T^{-1}[10^{-1}\,\mathrm{GeV}^{-1}]$	$\phi[^\circ]$	$\theta[^\circ]$
2D	N1	0.962 ±0.003	0.876 ±0.002	
3D	N1	0.554 ±0.001	0.438 ±0.001	3.349 ±0.008
2D	L1	1.225 ±0.005	0.871 ±0.003	
3D	L1	1.002 ±0.003	0.609 ±0.002	3.900 ±0.011
2D	N0	0.687 ±0.002	0.546 ±0.001	
3D	N0	0.501 ±0.001	0.393 ±0.001	2.855 ±0.007
2D	L0	0.806 ±0.003	0.547 ±0.002	
3D	L0	0.820 ±0.002	0.524 ±0.001	3.076 ±0.008

Table 5.14.: Resolutions values (RMS90) of the trained 3D finders and of the 2D finder in test data sets with the different track parameter ranges (see Tab. 5.8).

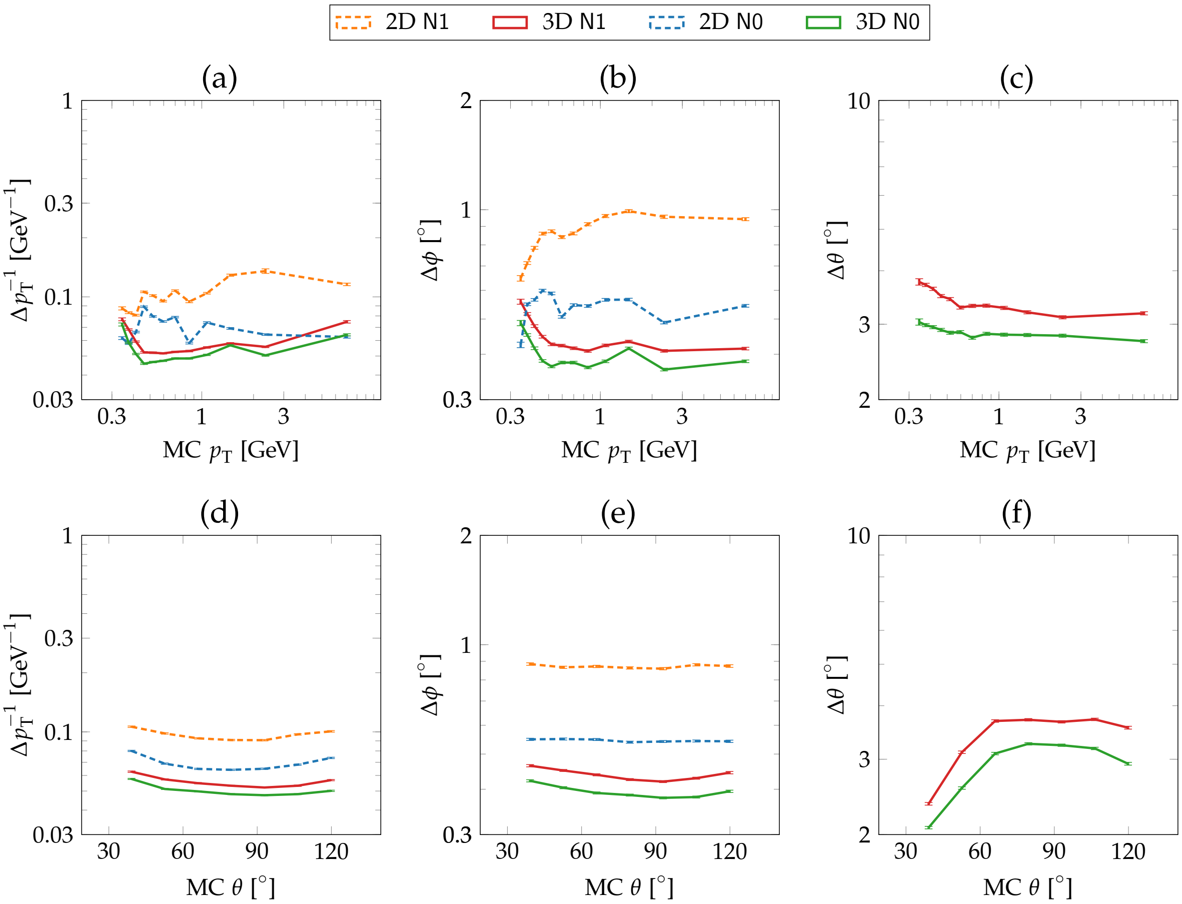


Figure 5.32.: Resolution of the finders on the test data sets N0 and N1 (see Tab. 5.8).

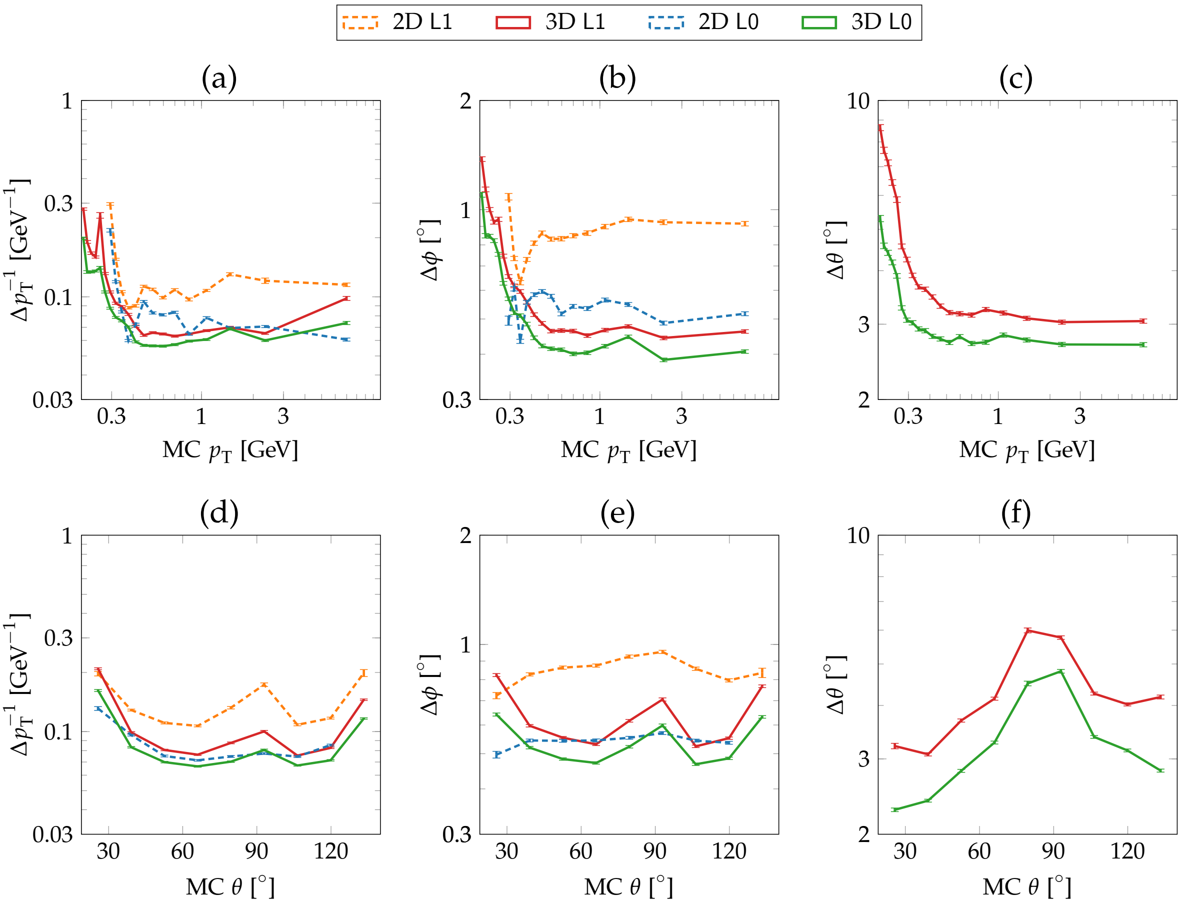


Figure 5.33.: Resolution of the finders on the test data sets L0 and L1 (see Tab. 5.8).

factor of two, can be found in Tab. 5.14 for the test data set N1. In data range L0, the 2D finder and the 3D finder resolutions are comparable.

As expected, additional background results in a worse resolutions. However, the 3D finder resolution is more robust to background (see Tab. 5.14): in the range N the 3D finder resolution is degraded by about 10 % only, while the 2D finder resolution is degraded by 40 % in p_T^{-1} and 60 % in ϕ. In the large range L the 3D finder resolution is degraded by about 20 % in p_T^{-1} and 15 % in ϕ only, while the 2D finder resolution is degraded by 50 % in p_T^{-1} and in ϕ.

More details on the resolutions can be found by comparing the resolutions in the test data sets N0 and N1 shown in Fig. 5.32 with the resolutions in the data sets L0 and L1 shown in Fig. 5.33. The plots show an expected p_T-dependency, with worse resolutions for tracks with a larger curvature. Since the low-p_T region is missing in the range N this can be better seen in the range L shown in Fig. 5.33 (a), (b), and (c). In the low-p_T region in Fig. 5.33 (a) and (b) the limited efficiency of the 2D finder is the reason for the missing low-p_T data points. Hence, the calculation of the average 3D finder resolution using the data sets L0 and L1 is biased by the high 3D finder efficiency in the low-p_T region. This means, a fair comparison of the achieved average resolution values is given by the results in the test data sets N0 and N1 where both finders achieve full efficiency (see Tab. 5.14).

Using the range N the resolutions as functions of θ, shown in Fig. 5.32 (d) and (e), are almost flat. For the 3D finder the θ-estimation is slightly better for tracks in the forward region (small θ) than for tracks in the backward region (large θ). This is expected, as the CDC coverage is asymmetric in θ (see Sec. 3.2.5).

In the corresponding θ-dependent resolution plots using the range L, shown in Fig. 5.33 (d) and (e), a peak appears for tracks with $\theta \approx 90^\circ$. This peak is a consequence of the improved track finding efficiency for the curl-back tracks with low-p_T shown in Fig. 5.30 (c). Since the efficiency for the low-p_T tracks is artificially increased by adding TS hits after the curling, the resolution for these low-p_T tracks with $\theta \approx 90^\circ$ has to decrease. Furthermore, in the shallow-θ region in the range L, additional resolution changes can be observed as a direct effect of the enlarged test data region.

The contributions of the different boundary regions are already described using the efficiency and fake rate study. Hence, the contributions to the resolution changes in L could be directly identified. For completeness, the plots and tables showing the resolutions at the each boundary region separately, using the test ranges P and W, can be found in the Appendix C.2.

5.3.5. Summary of the Track Finding Performance

In studies with single track events, the optimal training target tracks for the 3D finder have been selected: in a simulated data set with background the 3D finder is trained on MC track parameter values with matches to reconstructed tracks. The efficiencies and track parameter resolutions have been measured at the different boundary regions of the geometrical CDC acceptance region. In all track parameter regions, the 3D finder shows an improved robustness against background. Within the full acceptance region of the CDC, where the 2D finder is fully efficient, the 3D finder achieves comparable track finding efficiencies and improved track parameter resolutions.

The efficiency in the low-p_T region and in the shallow-θ region is significantly improved. Due to the improved efficiency, in particular in the case with background, an increased fake rate is observed. Considering the track parameter distributions of the background tracks, the increased fake rate corresponds to correctly found background tracks, which are also seen in the distribution of reconstructed tracks. Especially curl-back low-p_T tracks with polar angles of $\theta \approx 90^\circ$ pose difficulties for the track finders. By adding TS hits along the curl-back trajectory, they increase the fake rates as well as the efficiencies. The efficiency increase created by the false addition of TS hits results in a resolution loss in the region $\theta \approx 90^\circ$.

A new 3D finder algorithm has been developed and implemented within the Belle II simulation framework basf2. Due to the use of machine learning to train the hit representations in the track parameter space, this 3D finder is independent of the concrete track and hit model. Therefore, it can easily be adapted to new requirements or different fields of application. The studies presented in this section were carried out using test data from the IP. An extended study in combination with the neural network, where tracks with a longitudinal displacement along the z-axis are included, is presented later in Sec. 6.5.

5.4. Optimization of the 3D Finder Parameters

The important configuration parameters for the 3D track finder are the number of bins in each of the three track parameters ($p_\mathrm{T}, \phi, \theta$) and the clustering parameters `minhits`, `thresh` and `minweight`. Compared to the binning of a 2D track finder, the binning in θ is completely new. The number of bins in θ is crucial for the resolution of the θ-estimation. Adding more bins improves the resolution, however, it increases the size of the Hough space. The used Hough space binning is defined in Tab. 5.1 and discussed in Sec. 5.2.2. In this section, the optimal configuration parameters for the clustering and the track parameter estimation are studied. These configuration parameters are introduced in Tab. 5.5 and described in Sec. 5.2.5.

The parameter `minweight` is the minimum weight of a single cell in the Hough space to be considered as a possible cluster member. It is important in the first step of the clustering algorithm, where cluster candidates are found. Increasing this parameter has the effect that less cells are considered as cluster candidates. This results in an improved resolution and a smaller fake rate but in a decreased efficiency.

The parameter `minhits` defines the minimum number of hits that have to be associated to a cluster such that it is considered as a found track. Using the confusion matrix illustrated in Fig. 5.20, each hit is associated to the cluster where it has the largest weight contribution to the peak cell of the cluster. This peak cell is designated as the cell in the cluster with the largest total weight, where the total weight per cell is the sum of the single hit contributions to that cell. Clusters with too few hits are removed and, if possible, the hits are reassigned to bigger cluster. Larger values of `minhits` reduce the fake rate but also the efficiency.

For the track parameter estimation, the local region around a peak is selected using the parameter `thresh` (see Sec. 5.2.5). Once cluster candidates are found, `thresh` is used to throw out cells with a relatively small weight contribution to the cluster. It defines the minimum weight of a cluster cell as a fraction of the peak weight in the cluster. By selecting the local area around the peak, the resolution of the track parameters can be improved compared to taking the weighted mean of all cluster cells. Since `thresh` is only applied after the clusters are found, this parameter is uncorrelated to the efficiency and fake rate. Increasing `thresh` reduces the cluster size which improves the resolution. However, if `thresh` is too large the result are clusters only containing the single peak cell. Since the track parameters are calculated as weighted mean of the cluster member cells, in the limit of large `thresh` the resolution drops again. This weighted mean is still calculated in units of Hough space bins with Eq. 5.56. A following back-transformation to physical track parameters is obtained by using Eq. 5.58.

In Sec. 5.4.1, a set of cluster properties is introduced and illustrated using the set of central configuration parameters listed in Tab. 5.6. The histograms of the cluster property distributions illustrate the capabilities to distinguish signal and fake clusters based on cuts on the cluster properties. Special focus is on the cluster properties where the used configuration parameters appear as cuts. In Sec. 5.4.2, the effect of a change in the configuration parameters from Tab. 5.6 is presented. To this end, the mean values of selected cluster property distributions introduced in Sec. 5.4.1 are included in the study.

5.4.1. Properties of the Track Clusters in the 3D Hough Space

An important perspective to study the 3D finder performance is to look at the properties of the found track clusters in the Hough space. Especially signal and fake clusters show characteristic differences in their cluster properties. Based on the observed distributions of the cluster properties, the optimal configuration parameters for the track finding can be selected. In this sense, the configuration parameters `minhits`, `minweight`, and `thresh` can be understood as cuts, which are used to distinguish signal from fake clusters and to improve the resolution of the track estimates. A selection of cluster properties is presented and their relation to the configuration parameters `minhits`, `minweight`, and `thresh` is discussed.

In this study, the central finder parameters listed in Tab. 5.6 are used for the measurement of the cluster properties. Each shown cluster property is a single value per found cluster. This means, the cluster properties are calculated after `minhits` and `minweight` are applied during the clustering and after `thresh` is used to select a local region of Hough space cells around the peak.

The first discussed property is the number of Hough cells contained in the cluster, which is referred to as the "cluster size". Based on the extension of the cluster in the Hough space, additionally the size along each of the three track parameter Hough axes are referred to as "p_T^{-1}-size", "ϕ-size", and "θ-size". For each cluster the p_T^{-1}-size, ϕ-size, and θ-size are the minimum side lengths of a cuboid containing all the cells in the cluster. The sizes of the clusters are closely related to the binning of the Hough space. Furthermore, the sizes are trimmed by all three configuration parameters. There is another important – size like – cluster property: the TS hit multiplicity per cluster, which is referred to as the "number of hits". This distribution is directly trimmed by the parameter `minhits`.

Cluster properties related to the weights comprise the "total weight" of a cluster calculated as the sum of the weights in all the cells and the "average weight" calculated as the mean weight of the cluster. The "peak weight" is the maximum weight of a single cell in the cluster and the "weight width" is the RMS90 of the weight distribution. The trimming effect of the parameter `minweight` can be seen in the "average weight" as well as in the "peak weight" distribution.

Additionally, the following derived cluster properties are defined: Most importantly "avg weight / peak weight" is the "average weight" divided by the "peak weight". Since the local region around the peak is selected by requiring the cells to have a weight divided by the peak weight larger that `thresh`, the lower limit of the "avg weight / peak weight" distribution is given by the parameter `thresh`. Furthermore, the "tot weight / nTS" (the average weight contribution of each hit) and the "tot weight / peak weight" are added to the illustration below.

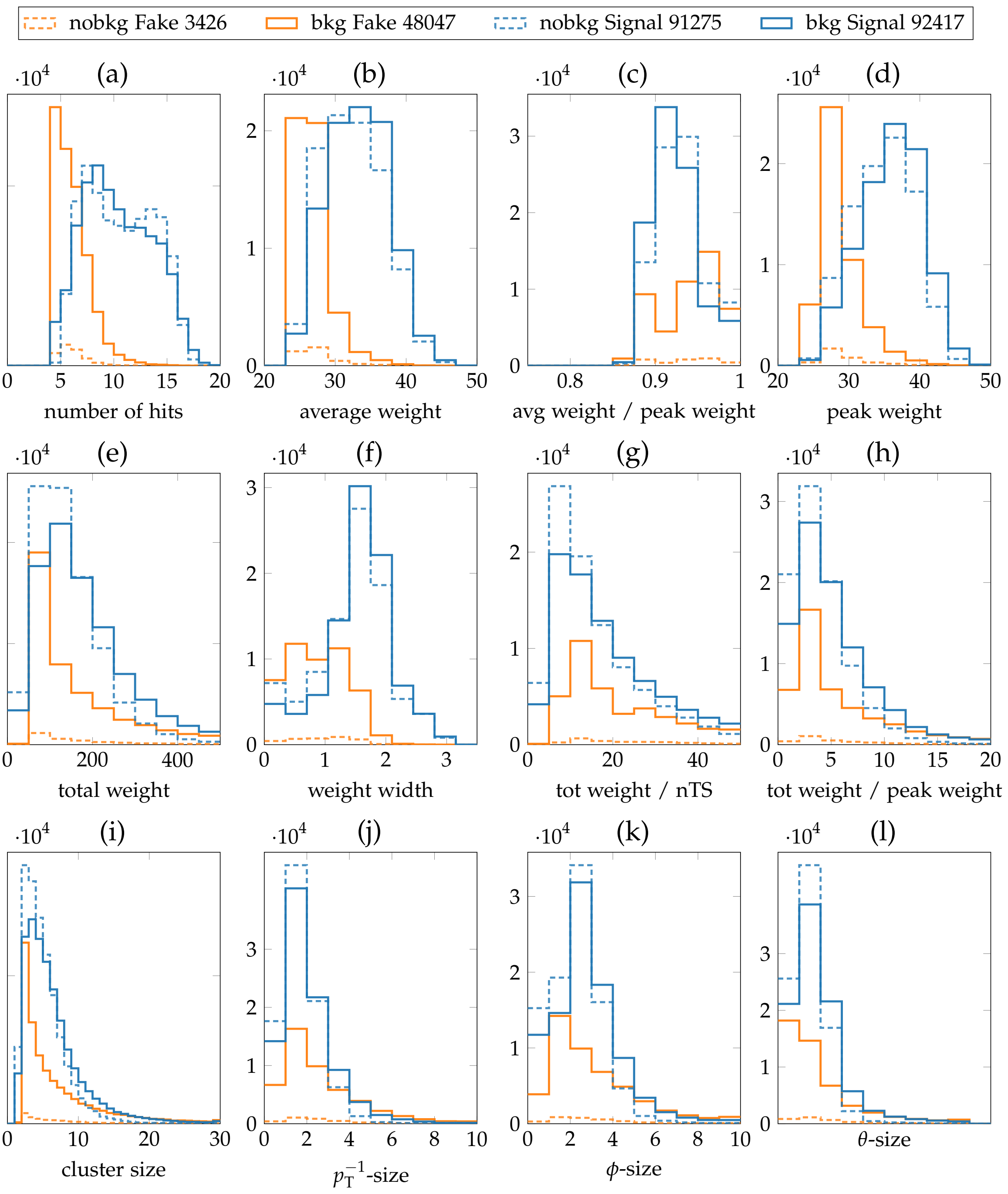


Figure 5.34.: Histograms of the binned cluster properties of signal tracks in blue and fake tracks in orange using the central configuration parameters listed in Tab. 5.6. Dashed lines are measured on the data set L0 without background and solid lines on the data set L1 with background (see Tab. 5.8). The total number of found clusters in each class is listed in the legend.

The “cluster size” in numbers of Hough space cells, is especially important for the realization of the algorithm in hardware. If the same clustering algorithm will be used in the future hardware implementation, the minimum latency required to find clusters is proportional to the logarithm of the cluster size ($\log(n)$ with $n =$ cluster size). The reason for this $\log(n)$ limit on the complexity is that a cluster is locally expanded to the neighbors of each cell (see Sec. 5.2.5). Thus, smaller clusters are desirable for the hardware, as they reduce the latency. On the other hand, a finer binning of the Hough space improves the resolution and the efficiency to separate two tracks in close vicinity.

Figure 5.34 shows the histograms of the cluster properties, tested on the data sets L0 and L1 containing tracks at the p_T- and θ-boundary region of the CDC acceptance (see Tab. 5.8). Each found cluster is either matched to a reference track and counted as signal cluster, or it is unmatched and counted as fake cluster. The four curves in each plot correspond to signal and fake clusters in the cases with and without background. In the legend in Fig. 5.34, the total number of clusters in each class is listed, corresponding to the sum over the bins in each plot. Characteristic differences of the signal and fake distributions can be used to distinguish signal from fake clusters during the track finding. In the present setup, such a distinction is realized with the configuration parameters `minhits`, `minweight`, and `thresh`. The cluster properties in Fig. 5.34 are measured for a 3D finder using the central finder parameters listed in Tab. 5.6. The plots shown at first in Fig. 5.34 (a), (b), (c) are directly related to the configuration parameters `minhits`, `minweight`, and `thresh`. Other cluster properties, which enter the calculations of (a), (b) and (c) follow in the later plots.

The number of related hits per cluster in Fig. 5.34 (a) is limited at the lower end by the parameter `minhits` $= 4$. Since the fake clusters have a peak for smaller values of “number of hits” in Fig. 5.34 (a), a good distinction of signal and fake clusters is possible using the parameter `minhits`. This also means that clusters with a larger number of related hits are more likely signal tracks and less likely fake tracks.

The average weight in the cluster in Fig. 5.34 (b) is the ratio of the total weight Fig. 5.34 (e) divided by the “cluster size” in the cluster Fig. 5.34 (i). This average weight is closely related to the parameter `minweight` $= 24$ (see Tab. 5.6) which is the lower limit of a cell to be considered as a cluster member. In Fig. 5.34 (b), `minweight` appears as the lower limit of the “average weight”. Again, a good distinction of signal and fake tracks can be observed in Fig. 5.34 (b), as fake tracks have typically a lower average weight than signal tracks. Hence, `minweight` is a powerful criterion to distinguish signal and fake tracks.

The “average weight / peak weight” shown in Fig. 5.34 (c) is closely related to the parameter `thresh` $= 0.85$, which is used to select the local area around the cluster peak for the track parameter estimation. The “average weight / peak weight” in Fig. 5.34 (c) is the “average weight”, shown in Fig. 5.34 (b), divided by the “peak weight” shown in Fig. 5.34 (d). Clearly the used cutoff value of `thresh` $= 0.85$ appears in Fig. 5.34 (c) as lower limit of the “avg weight / peak weight”. Fig. 5.34 (c) shows that the “avg weight / peak weight” distribution in signal and fake clusters is very similar. The peak weight in Fig. 5.34 (d) shows a very good discrimination of signal and fake clusters. This indicates that an additional parameter minpeakweight could be introduced with a slightly larger value than minweight.

The plots in the second line, Fig. 5.34 (e) (f) (g) (h) show additional distributions related to the weights in the clusters. Figure 5.34 (e) shows the "total weight" distribution which is calculated as the sum of the weights of all cells in the clusters. Figure 5.34 (f) shows the "weight width" distribution which is calculated as the RMS90 on the distribution of the weights of the cells within the clusters. As can be seen in Fig. 5.34 (f), it is typically lower for fake tracks while clusters with a "weight width" ≥ 2 are all signal tracks. However, this is a direct effect of the cluster size shown in Fig. 5.34 (i), where fake clusters typically have less cells than signal clusters. Figure 5.34 (g) shows the "tot weight / nTS" distribution, which is the average weight contribution of each TS to the cluster. The "tot weight / nTS" is calculated as the "total weight" (Fig. 5.34 (e)) divided by "number of hits" (Fig. 5.34 (a)). Figure 5.34 (h) shows the "tot weight / peak weight" distribution, which is calculated as the "total weight" (Fig. 5.34 (e)) divided by the "peak weight" (Fig. 5.34 (d)). It is in the same order of magnitude as the "cluster size" shown in Fig. 5.34 (i). Since the plots in Fig. 5.34 (e), (f), (g), and (h) do not show additional measures for signal to fake distinction, these quantities are not used to construct cut parameters for the clustering algorithm.

The plots in the third line, Fig. 5.34 (i), (j), (k), and (l) show the size of the found clusters. Fig. 5.34 (i) shows the "cluster size" in the cluster within the local region around the peak. Fig. 5.34 (j), (k), and (l) show the length of the clusters in units of Hough cells along each of the three track parameter Hough dimensions p_{T}^{-1}, ϕ, and θ. Compared to fake clusters, signal clusters have typically a larger "cluster size", ϕ-size, and θ-size while the p_{T}^{-1}-size is very similar. In the case with background, the cuboid enclosing the most likely signal cluster (i.e. the peak values of the histograms) has side lengths of $(2 \times 3 \times 2)$ along $(p_{\mathrm{T}}^{-1}, \phi, \theta)$ and it contains 4 cells. The cuboid enclosing the most likely fake cluster has side lengths of $(2 \times 2 \times 1)$ along $(p_{\mathrm{T}}^{-1}, \phi, \theta)$ and contains 3 cells.

5.4.2. Parameter Variation

This study demonstrates the effect of a change in the 3D finder parameters and gives insight into the approach followed in the construction of the algorithm and the selection of the configuration parameters. With the various Hough space parameters and possible configuration parameters, the search for the optimal solution is a multidimensional optimization problem and testing all combinations is not feasible due to the limited computational and time resources. Hence, heuristic strategies were used during the development of the 3D finder. Here, this strategy is demonstrated using the optimization of the three configuration parameters minhits, minweights and thresh as an example.

The anticipated heuristic optimization strategy to find an optimal parameter set is to vary the parameters one by one. After varying one parameter, its optimum can

`minweight`	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
`minhits`	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
`thresh`	.6	.65	.7	.75	.8	.82	.84	.85	.86	.88	.9	.92	.94	.96	.98

Table 5.15.: Discrete sets of the tested parameter values for the configuration parameters `minweight`, `minhits` and `thresh`.

be selected and used as fixed parameter to optimize the remaining parameters. The procedure can be iterated, until no further optimizations are found. Such a strategy can only find local optima, without a guarantee that the found local optimum is a global optimum as well. However, it can be carried out within a sufficiently short time scale and is therefore preferred over exhaustive scans of the full configuration parameter space. Another problem with exhaustive scans are the multiple optimization objectives. These include the efficiencies, fake rates, and resolutions. Additionally, the number of operations should be limited in order to reduce the latency using the available resources on the hardware. In this sense, a global optimum does not exist.

At the time of the writing of this thesis, the hardware platform where the 3D finder is going to be implemented is not yet decided. Many constraints on the structure and size of the Hough space and on the clustering algorithm will be set by the available resources of this future hardware. Due to the limited number of operations that can be carried out on an FPGA in parallel, especially the timing requirements of the clustering have to be carefully compared to the maximum latency requirements of the first level trigger. Once the resource limitations for a specific hardware platform are known, the parameter optimization has to be repeated considering the new constraints.

In the presented heuristic optimization strategy for the configuration parameters the selected central parameters are: $\texttt{minweight} = 24, \texttt{minhits} = 4, \texttt{thresh} = 0.85$ (see Tab. 5.6). The size of the Hough space is fixed and defined by the parameters in Tab. 5.1 in Sec. 5.2.2. During the optimization step each parameter is varied separately while the other parameters are kept fixed at the central values. This allows to understand the effect of each parameter in the local region of the configuration parameter space around the central parameters.

For each of the three parameters the set of 15 discrete values listed in Tab. 5.15 is used for the study. The tested values for `minweight` and `minhits` are equidistant. In order to show more data points close to the optimum (see Fig. 5.37), the values for `thresh` are non-equidistant (see Tab. 5.15). In the region between 0.80 and 0.98, additional intermediate values are inserted to achieve a finer granularity.

Effect on the Cluster Properties

Figure 5.35 shows the mean values of the cluster properties directly related to the parameters as functions of the varied parameters. As described in Sec. 5.4.1, the

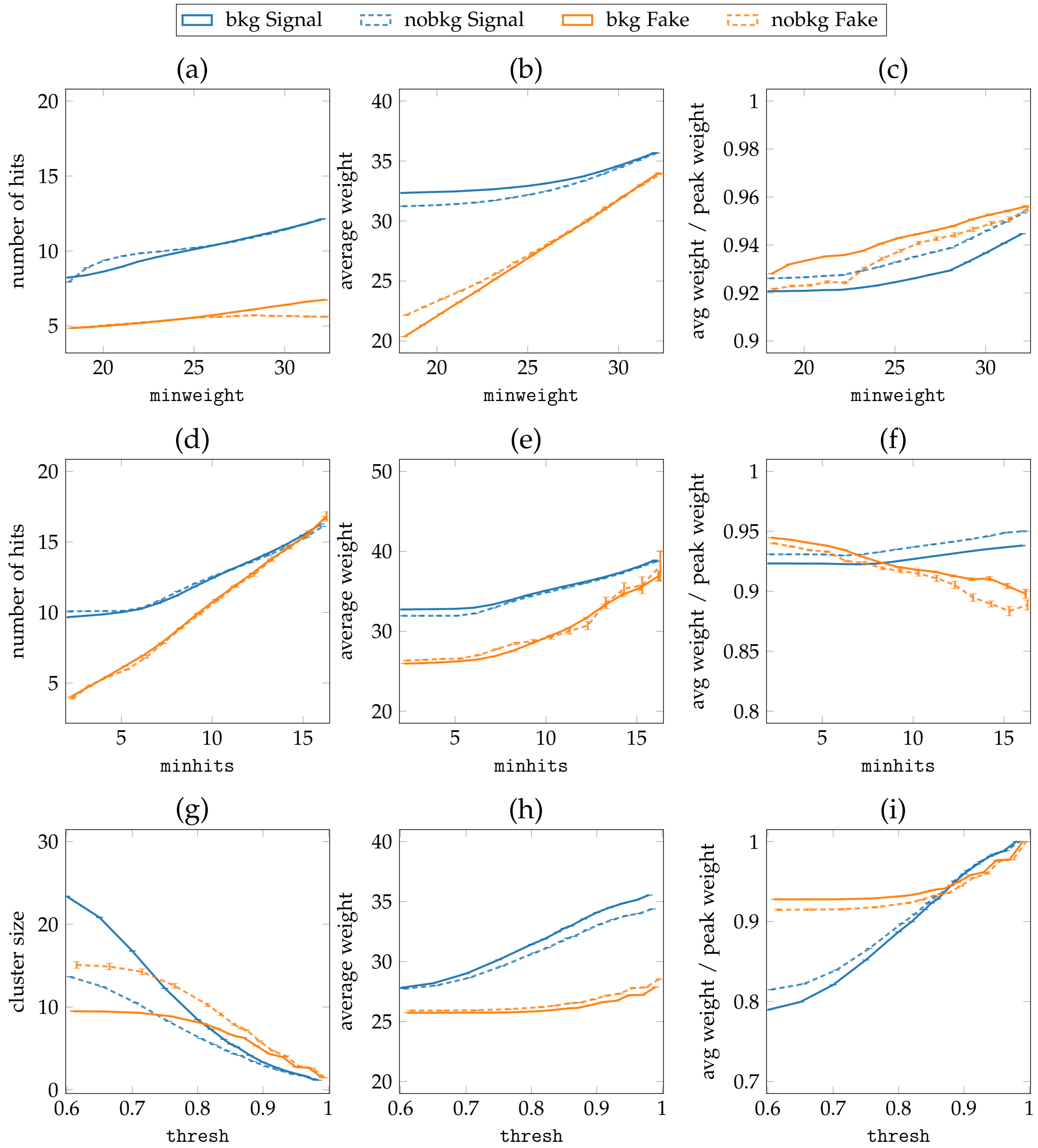


Figure 5.35.: Mean values of the selected cluster properties “number of hits”, “average weight”, and “avg weight / peak weight” as functions of `minweight` and `minhits` in the first two rows. The third row shows the “cluster size”, “average weight”, and “avg weight / peak weight” as functions of `thresh`. The mean values are calculated for signal and fake clusters using the test data sets L0 and L1 (see Tab. 5.8).

parameter minhits is related to the "number of hits" per cluster; the parameter minweight is related to "average weight" of the cluster cells; and thresh is related to the "avg weight / peak weight" (the average weight divided by the peak weight). Since thresh is only applied after the hits are related to the cluster, it is uncorrelated with the "number of hits". However, thresh is related to the "cluster size", as it selects a local region around the peak for the track parameter estimation. Therefore, for the variation of thresh in Fig. 5.35 (g), the "cluster size" is shown instead of the "number of hits".

The curves in Fig. 5.35 show the cluster properties separately for the signal and fake tracks using the test data sets L0 and L1 (see Tab. 5.8). Signal curves in Fig. 5.35 are shown in blue and the fake curves in orange. Additionally, the results of the runs with background (data set L1) are shown as solid curves while the results without background (data set L0) are shown as dashed curves. Each data point in Fig. 5.35 corresponds to the arithmetic mean value of a cluster property distribution on the y-axis, using the respective varied parameter values shown on the x-axis. As an example, the data points at the central parameters correspond to the mean values of the distributions shown in Fig. 5.34. Observed changes in the means of the cluster property distributions correspond to changes in the characteristic differences between signal and fake clusters.

The error bars shown in Fig. 5.35 are the standard errors of the mean. These errors are approximated by the RMS90 as RMS90/$\sqrt{n}$, where n is the number of elements in a specific cluster class (signal or fake, with or without background). This study of the different mean value positions demonstrates which parameters have a potential to separate signal and fake clusters and illustrates their correlations. However, as can be seen in Fig. 5.34, the cluster property distributions also have a width. Therefore, the quality of the cuts is measured in the subsequent efficiency study.

As functions of minweight the cluster properties shown in Fig. 5.35 (a), (b), and (c) continuously increase with increasing values of minweight. With increasing minweight the clusters have typically more related hits (see Fig. 5.35 (a)) and larger average weights (see Fig. 5.35 (b)). This is expected, as a larger number of hits contributing weights to each cell in the cluster increases the average weight per cell. In addition, the ratio of the average weight divided by the peak weight in Fig. 5.35 (c) is also increasing. This is also expected, as the average weights are increasing due to the minweight cut, while the peak weights stay constant.

However, the increasing structure of the signal and fake clusters differ. Over the full region of minweight, the "number of hits" per cluster in Fig. 5.35 (a) is larger for signal clusters than for fake clusters. Furthermore, the difference of the signal curves (blue) and the fake curves (orange) does not change much with minweight. Signal clusters have approximately twice as many hits as fake clusters. This means a classification into signal and fake clusters based on the "number of hits", as it is done with the parameter minhits, is effective and almost independent of the selected value of minweight.

Figure 5.35 (b) shows that especially for low values of minweight the "average weight" is a useful discriminator of signal and fake clusters. With higher minweight, the "average weight" in fake clusters approaches the "average weight" in signal clusters. Thus, `minweight` is a useful quantity to suppress fake tracks. However, the improvement with larger values of `minweight` settles in the region where the signal and fake cluster can no longer be distinguished based on the "average weight". As can be seen in Fig. 5.35 (b), the central value of `minweight` $= 24$ is still in a region where signal clusters can be distinguished from fake clusters based on their "average weight".

Figure 5.35 (c) shows the mean values of the distribution "avg weight / peak weight", which is cut by the parameter `thresh`. Due to the small differences of the signal and fake cluster mean values in Fig. 5.35 (c), within the tested range of `minweight` a cut on "avg weight / peak weight", as realized by the parameter `thresh`, cannot be used to distinguish signal and fake clusters.

As functions of `minhits`, the cluster properties "number of hits" in Fig. 5.35 (d) and the "average weight" Fig. 5.35 (e) are steadily increasing for increasing values of `minhits`. Especially for small values of `minhits` Fig. 5.35 (d) demonstrates the "number of hits" to be a good discriminator for signal and fake clusters. Note that the "number of hits" in Fig. 5.35 (d) is the distribution cut by the parameter `minhits`. Therefore, the differences between signal and fake clusters become smaller for larger values of `minhits`. The central value of `minhits` $= 4$ is still in the region where the different "number of hits" between signal and fake clusters are large. This means that a further reduction of the fake rate can be achieved by increasing `minhits`. However, as will be shown later in Fig. 5.36, the efficiency is affected as well.

The "average weight" in Fig. 5.35 (e) starts out flat for signal and fake clusters for small values of `minhits` ≤ 6 and increases for values of `minhits` > 6. In all regions of `minhits`, the "average weight" of signal clusters is larger than the "average weight" of fake clusters.

In contrast, the dependence of the "avg weight / peak weight" on the parameter `minhits` in Fig. 5.35 (f) appears to be inverted for signal and fake clusters. The "avg weight / peak weight" of signal clusters is continuously increasing with `minhits` while it is continuously decreasing with fake clusters.

The signal and fake track curves with background intersect at `minhits` ≈ 9. This corresponds to tracks having possibly a single TS hit in each SL. Without background the intersection point is at a value of `minhits` ≈ 6. This means fake clusters with many hits tend towards large peak weights compared to the average weight in the cluster. In contrast, in signal clusters with many hits the peak is less pronounced compared to the average cell in the cluster. However, the inverse behavior is not very strong and the signal and fake clusters have "avg weight / peak weight" values in close vicinity.

The "cluster size" in the Hough space is shown in Fig. 5.35 (g). A continuous decrease in the "cluster size" with increasing `thresh` can be observed for signal and fake clusters. In the case with background, the signal and fake curves show an intersection at `thresh` ≈ 0.8, which is absent in the case without background. However, the difference between signal and fake clusters for large values of `thresh` is quite small.

A large difference can be observed for small values of thresh ≈ 0.6 in the case with background, which is not seen in the case without background. At small values of thresh, fake clusters created by the additional background hits have a tendency towards large peak values compared to the average weight in the clusters. In contrast, in the case of signal clusters, background has the effect that the "cluster size" with a weight close to the peak weight is increased.

This is plausible in a simplified model by considering the background as random noise. For signal clusters, a correct intersection of the hit curves exists. By throwing random noise on top of all cells, all cluster member cells are on average increased by the same amount. This has the effect that the weights of the cluster cells relative to the peak weights are increased. At the boundary region of a cluster, where the parameter thresh cuts out a local region, lowering thresh results in more cells to be contained within the local region around the peak. In contrast, clusters created by random noise are uncorrelated to a track such that the peak weight is typically lower. Hence, by lowering thresh, additional cells are not added to the fake cluster as they fall below the minweight limit.

Figure 5.35 (h) shows that the "average weight" as a function of thresh is larger for signal than for fake clusters. With increasing thresh, the distance between the mean of the signal and fake clusters becomes larger. In particular for larger values of thresh the "average weight" of a cluster is useful for signal and fake distinction.

Figure 5.35 (i) shows the distribution "avg weight / peak weight" where the cut thresh is applied. For low values of thresh, signal clusters show a significantly smaller mean values of "avg weight / peak weight". In agreement with the previous observations, this indicates that fake clusters are smaller than signal clusters. For large values of thresh, signal and fake clusters can no longer be distinguished based on "avg weight / peak weight". As shown before in Fig. 5.34 (c), at the central value of thresh $= 0.85$ the means of the "avg weight / peak weight" distributions for signal and fake clusters are almost identical. This indicates that thresh can be used for a separation of signal and fake clusters as well. However, in order to improve the signal efficiency, lower values should be selected than for the selection of the local area around the peak.

Effect on the Efficiency and Fake Rate

The parameter minweight selects cells which are possible cluster members and the parameter minhits is used to skip small clusters. Through this direct use of these parameters in the clustering algorithm, minweight and minhits affect the efficiency as well as the fake rate. Therefore, the efficiencies and fake rates are plotted in Fig. 5.36 as functions of the track parameters minweight and minhits. In order to study the optimal efficiency vs. fake rate tradeoff region, a ROC-like (Receiver Operator Characteristic) curve is included in Fig. 5.36. However, since minweight and minhits are not just threshold values discriminating two classes of output values, the efficiency vs. fake rate plots in Fig. 5.36 (c) and (f) are not real ROC curves. This means that it is not possible to set minweight and minhits to extreme values in order to scan the

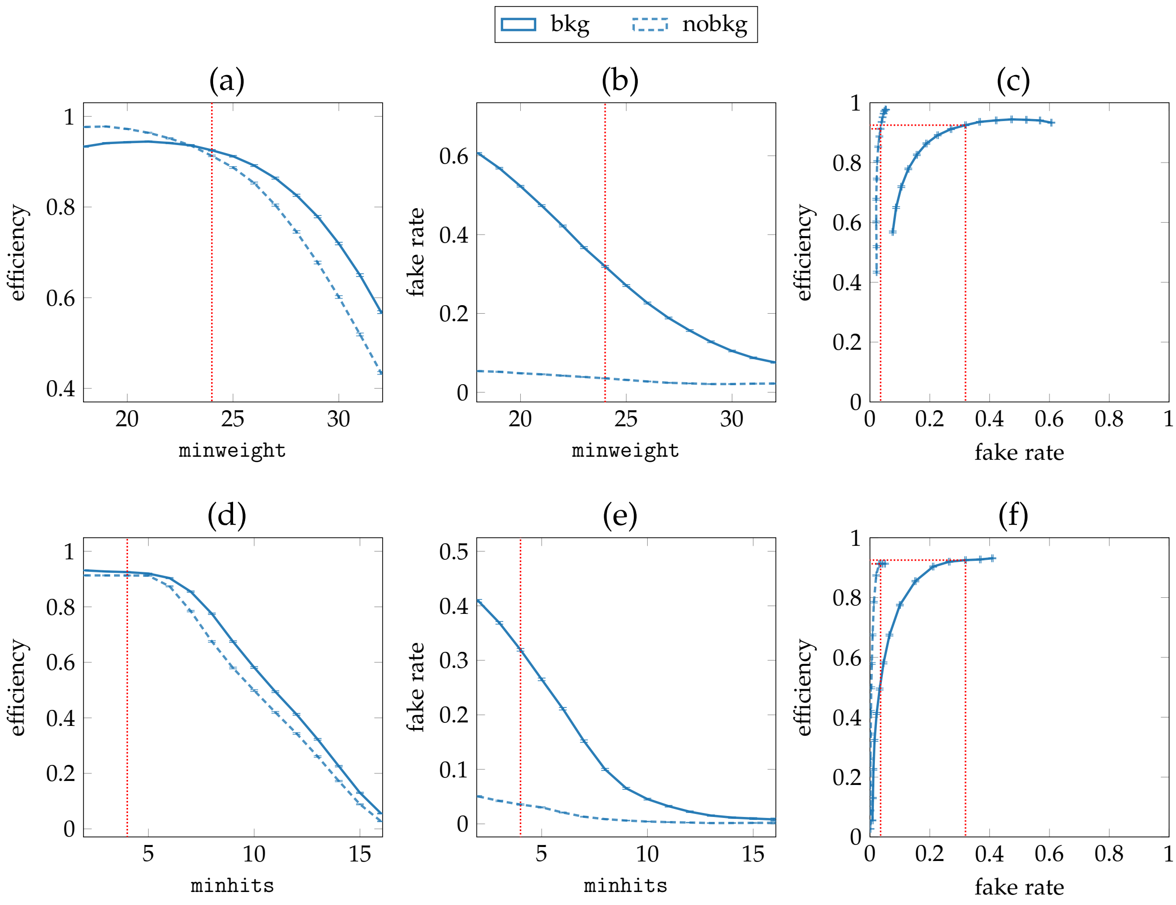


Figure 5.36.: Efficiency, fake rate and a ROC-like curve for the varied clustering parameters `minweight` and `minhits`. The two curves show the results with and without background (test data sets `L0` and `L1` in Tab. 5.8). The dotted red lines indicate central parameter values of $\texttt{minhits} = 24, \texttt{minweight} = 4$.

full efficiency range $\epsilon \in [0, 1]$ and the full fake rate range $g \in [0, 1]$.

Figure 5.36 (a) shows the efficiency ϵ as a function of `minweight`. For small values of `minweight` the efficiency without background (blue dashed line) slowly approximates 100 %. However, in the case with background (blue solid line), a shallow peak around $\texttt{minweight} \approx 22$ can be observed. This shallow peak is a result of the clustering algorithm. With lower `minweight` more and more cells become candidate cluster members such that lots of background cells disturb the signal clusters. If the background contamination becomes too large, the track estimates and hit relations are so inaccurate that the signal tracks can no longer be correctly matched to the target tracks in the test data. Note that the method used for track matching is based on the coincidence of hits of the found tracks with the reference tracks (see Appendix A.3). In the matching with MC reference tracks used here, found tracks are required to have more hits in common with the reference track than with the background. Hence, the peak corresponds to the limit where more than 50 % of the hits related to the found clusters are background hits. With increasing minweight the efficiency in Fig. 5.36 (a) obviously decreases, as small cluster may not be found due to insufficient weights of the contributing cells.

The fake rate, shown in Fig. 5.36 (b), continuously decreases with minweight. This is seen in particular in the case with background, where the fake rate at the smallest value of minweight $= 18$ reaches 60 % and decreases to less than 8 % for the largest shown value of minweight $= 32$. However, for higher values of minweight, the decrease of the fake rate in Fig. 5.36 (b) slows down compared to the decrease in efficiency in Fig. 5.36 (a).

This trade-off is illustrated in the ROC-like curve in Fig. 5.36 (c). It shows the efficiency vs. fake rate for each tested value of minweight. In a ROC curve, the optimally balanced trade-off parameter can be found in the upper left corner and the area under the curve is a quality measure of the classifier. As shown in Fig. 5.36 (c), the selected value of minweight $= 24$ is slightly displaced from this balanced point in favor of an improved efficiency. This means, an small increase in minweight $\gtrapprox 24$ results in a stronger reduction of the fake rate than loss in efficiency. This agrees with the objective to maximize the efficiency. However, if the found track rates or the signal to fake ratio exceeds the limits of the trigger, increasing minweight leaves room for a further reduction of the fake rate. The data point at the value of minweight $= 24$ has an efficiency of $\epsilon = 92.5\,\%$ and a fake rate of $g = 32\,\%$.

The efficiency for varying minhits, shown in Fig. 5.36 (d), does not depend on minhits for small values up to minhits ≈ 6. For larger values of minhits the efficiency decreases linearly with minhits, because tracks with a small number of related hits will be lost. The trends of the fake rate, shown in Fig. 5.36 (e) are different: the fake rate g decreases linearly until minhits ≈ 9 where it becomes negligible. Figure 5.36 (f) shows that the chosen basis value of minhits $= 4$ is also shifted away from the optimally balanced trade-off point in favor of a maximized efficiency. In principle, an increase up to minhits $= 6$ could be considered, in order to achieve a much smaller fake rate while losing only a little efficiency.

Effect on the Resolution

In the following, the effect of the 3D finder parameters on the resolution of the estimated track parameters will be discussed. For that matter, the resolution plots of $\Delta p_{\mathrm{T}}^{-1}$, $\Delta\phi$ and $\Delta\theta$ in bins of the varied finder parameters minhits, minweight and thresh are shown in Fig. 5.37.

Figures 5.37 (a), (b) and (c) show the resolutions as functions of minhits. For values of minhits smaller than 6, these resolutions are almost independent of minhits. By increasing minhits towards larger values, all resolutions can be improved. In principle, for tracks with many hits, the intersection point in the Hough space is determined better and thus the resolution has to improve. However, as previously shown in Fig. 5.36 (d), the efficiency loss sets in for values of minhits > 6. In order

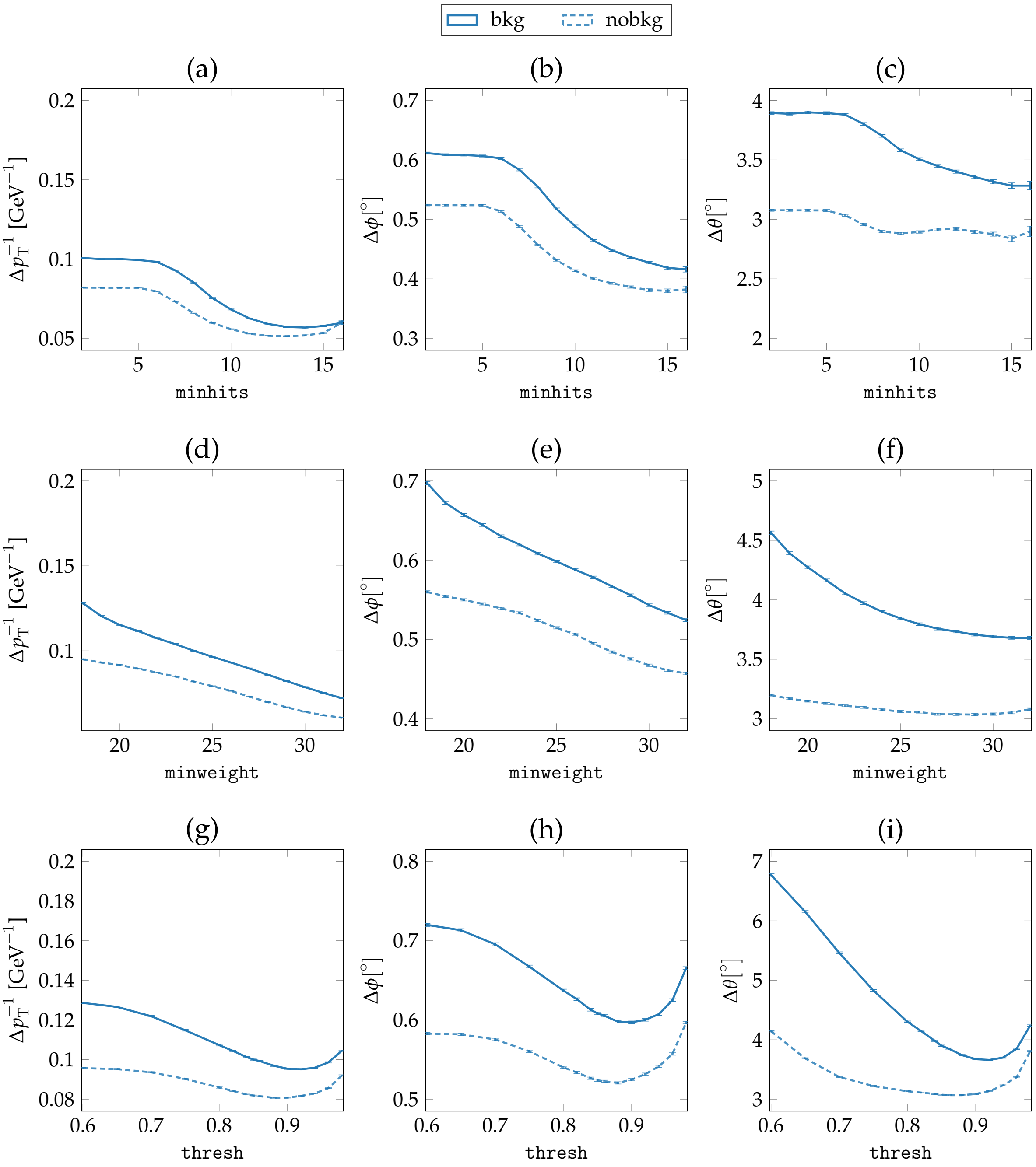


Figure 5.37.: Resolutions of the estimated track parameters as functions of varied clustering parameters.

to avoid an efficiency loss, despite the possible resolution improvements, `minhits` should not have too large values.

Figures 5.37 (d), (e) and (f) show the resolutions as functions of `minweight`. Increasing `minweight` continuously improves the resolutions in all estimated track parameters. Only for $\Delta\theta$, the resolution settles for values of `minweight` > 27. This is an effect of the small number of nine bins in θ used in this study (see Sec. 5.2.2).

In the hypothetical case when only a single peak cell were used for the track parameter estimation, the theoretical upper limit for the resolution is the error introduced via the binning of a histogram (see Appendix A.2). This upper limit is the bin-width divided by $\sqrt{12}$. Given the binning of the Hough space defined in Sec. 5.2.2, this limit corresponds to $(\Delta p_\mathrm{T}^{-1}, \Delta\phi, \Delta\theta) \approx (0.07\,\mathrm{GeV}^{-1}, 0.27^\circ, 3.9^\circ)$. However, these limits only apply to the case when the true value is correctly known. Due to the extent of the clusters in the Hough space, the single peak cell of a cluster does not necessarily correspond to the cell containing the true reference value. One reason for the extent of the clusters is given by the width of the hit representations in the Hough space, which mainly corresponds to the drift time distribution in the TSs (see Fig. 5.8). Additional background hits mixing with the signal hit may result in further displaced peak cell positions within the cluster. Therefore, these theoretical upper limits are not suited to measure the resolutions achieved by the single peak cell clusters.

The third line of plots, Fig. 5.37 (g), (h), and (i), shows the resolutions in bins of the parameter `thresh`. As previously described in this section, `thresh` selects the local region around the cluster peak by selecting only those cells where the ratio of the weight with the peak weight is larger than `thresh`. For larger values of `thresh` this local region becomes smaller. This means, in the limit `thresh` $= 1$, only those cells with a weight equal to the peak weight are left. Due to the used value of `bitwidth` $= 3$ for the weights in the hit representations (see Tab. 5.1), this is usually only a single peak cell.

Limiting the cluster size in the calculation of the weighted mean is useful to focus on the real intersection point of the hit curves in the Hough space. However, if `thresh` is too large, the clusters reach the limit of single cell clusters. In this case possible improvements via the weighted mean calculation are lost. Furthermore, the wrong peak cell might be selected due to the previously discussed extent of the clusters.

All resolutions in Fig. 5.37 (g), (h), and (i) show a minimum for values of `thresh` $\in [0.8, 1.0]$ and the chosen central parameter value of `thresh` $= 0.85$ is close to the minimum of the resolutions. The different positions of the minima along each track parameter axis in Fig. 5.37 (g), (h), and (i) could be investigated in future studies. Due to the different bin-widths along each Hough space dimension, a possible solution for further improvements are different `thresh` values along each axis.

5.4.3. Summary of the Parameter Optimization

An efficient configuration of the 3D finder parameters `minweight`, `minhits` and `thresh` has been suggested and the sensitivity of the track finding algorithm to a change of these parameters has been measured. Besides the efficiencies and resolutions, the structure properties of the clusters in the Hough space have been analyzed. The chosen set of central configuration parameters have been demonstrated to be close to an optimal solution. In favor of an improved efficiency, the cuts are slightly shifted away from the optimally balanced point while accepting an increased fake rate. If lower fake rates are required, especially an increase of `minhits` up to `minhits` $= 6$ turned out to be feasible to achieve a large reduction of the fake rate at a small loss in efficiency.

As part of the first level trigger at Belle II, an implementation on parallel FPGA hardware is foreseen. While the present first level trigger algorithms are implemented on the UT3 boards, a possible candidate that will be evaluated for a possible implementation of the 3D finder is the upcoming UT4 board. Once the available resources in the hardware are known, the results of the described parameter variation studies can be used for a fine tuning of the algorithm to the constraints given by the hardware.

The presented 3D track finding algorithm is very flexible and modifications can easily be implemented. Its independence of the explicit track and hit models allows for a transfer of the same algorithmic model to different track finding applications with other types of hits. Additional improvements for the efficiency to fake rate trade-off can be implemented by introducing further cut parameters to the algorithm. Especially the minimum weight of the peak turned out to be a useful discriminator of signal and fake clusters.

Chapter 6.

Neural Network Trigger

The neural network trigger is the central project developed in the course of this thesis. It is a first level trigger algorithm based on Machine Learning (ML), realizing a stable and robust 3D track reconstruction of single tracks in realtime. Using axial and stereo wire hits from the Central Drift Chamber (CDC), including the drift times as a distance measure of the tracks to the wires, the neural networks estimate the 3D track parameters (θ, z) of single tracks found by the 2D track finder. In face of the expected severe background conditions at Belle II, a precise estimation of the z-vertices of single tracks is crucial, as it allows to suppress background tracks with vertices displaced from the Interaction Point (IP), where the two beams are brought to collision. With the additional 3D track information, the other trigger logics can be relaxed and the trigger efficiency for physics signal events can be significantly improved. Especially the efficiency of signal events with low tracks multiplicities, like initial state radiation events, τ pair production and dark matter searches, will improve with such a z-vertex trigger. A hardware implementation of the neural network trigger is operational at Belle II and presently in a validation phase where the results are compared with the software simulation.

Due to the robustness of ML approaches, under the real conditions of an experimental physics detector with noise (background, electronic, uncertainties), the neural network is expected to provide more accurate results compared to a simple Least Squares (LS) track fit, which is optimal only under ideal conditions. This chapter covers the theoretical background of the neural network trigger, its practical implementation and elementary performance measurements using simulated single track events. Advanced performance measurements using simulated physics signal events and recorded events from early Belle II runs are presented in a subsequent Chapter 7.

Within the neural network trigger system the found single tracks from the 2D finder, parametrized by p_{T} and ϕ, are combined with axial and stereo Track Segment (TS) information and an event time estimate t_0. The 3D track information is extracted especially from the different stereo angles of the CDC wires in the stereo TS, while the axial wire hits allow an improvement of the 2D track information. The precise 3D track parameter estimation is possible by using an event time estimate t_0: relative to t_0, the drift times of each related TS correspond to a distance. After relating the TS to the found tracks, each TS position is represented by two coordinates relative

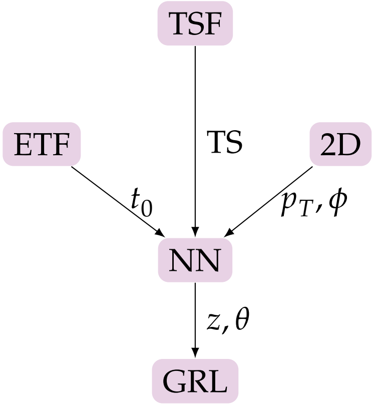


Figure 6.1.: Schematic of the neural network trigger input and output.

to the 2D track parameters as neural network input. Additionally the drift times of each TS are provided as a distance measurement between a TS and a track.

Using parallel FPGA hardware, the neural network trigger is already integrated into the L1 CDC trigger pipeline where new data is processed every 32 ns (see Sec. 3.3.2). The FPGA board of the neural network trigger presently receives input from the Track Segment Finder (TSF) and from the 2D track finder via high-speed optical links. An additional input from the Event Time Finder (ETF) is foreseen, but not yet realized in hardware (see Fig. 6.1 and compare Sec. 3.3.2). Since the development of the separate ETF module is not finished yet, a track based event time is determined in the preprocessing of the neural network, based on the shortest drift time among the TSs related to a track. The output of the neural network is sent to the GRL/GDL, where the final trigger decision is made in combination with the results from the other trigger modules.

The basic operation principle of the neural network trigger is as follows: In a preprocessing step, for each track estimate from the track finder, the related TS are selected. The condition for TS selection is maximally one per SL, hence maximally 9 TSs per track in total. To form the input vector of a single track for the neural network, each of the selected TS are represented by three values: the drift time and additionally two parameters describing the position of the TS. In total, this corresponds to an input vector with 27 values entering a single neural network. As output, the values at two output nodes of the neural network are interpreted as scaled values of the 3D track parameters θ and z. By describing the reference positions of the TS in a 2D coordinate system relative to the found 2D track, the 3D track reconstruction problem is linearized and the task fulfilled by the neural network is analog to a linear least square fit. Here the benefit of the neural network is its noise robustness in the presence of background. Complicated and time intensive error models are required for a correct fitting procedure in the presence of background, like those used for the precise offline track reconstruction at Belle II. The neural network can learn the error model from a training data set, while a short deterministic runtime can be

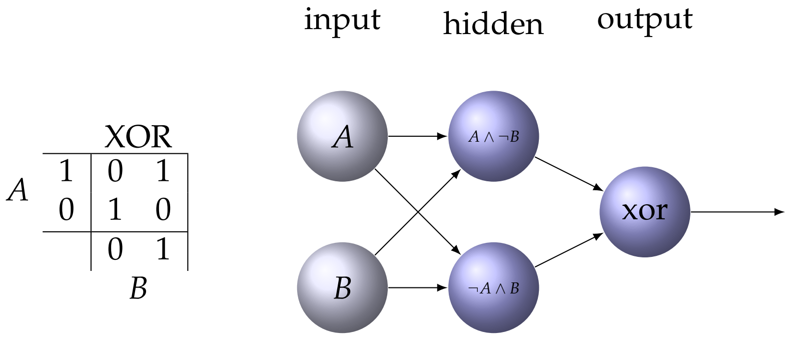


Figure 6.2.: The xor problem for two random variables A and B cannot be linearly decided by a single 1D separating hyperplane, i.e. a single neuron. But it can be solved by a 3 layer network with 2 neurons in the hidden layer [89]. (The hidden neurons solve a linear part of the problem e.g. $A \land \neg B$ and $\neg A \land B$.)

guaranteed. To improve the resolution and efficiency, the neural network trigger is configured as a pool of expert networks, where each network is trained to special conditions that can be known a priori in the preprocessing. In the present network installed in the hardware, the experts cover all the cases of a single missing stereo TS in addition to a network with all stereo TS present. Hence, five expert networks are used in total. Based on the track parameter estimates from the track finder, further extensions are possible where additional expert networks are trained for sectors in the track parameter phase space.

In the early phase of the Belle II neural network trigger project the idea was to use analog neural network hardware [86]. This analog hardware was under development by a physics group in Heidelberg and integrated into the human brain project [87], a flagship project funded by the European Union. These analog hardware neurons calculate a differential equation of the leaky integrate-and-fire type, and achieve a good approximation of the Hodgkin Huxley [88] neuron model. The Hodgkin Huxley model is a complex neuron model with many free parameters that is capable of producing all the firing patterns observed in biological neurons. It is the state of the art of the mathematical models approximating the biological neurons.

But with biological neurons, the problem is that appropriate training algorithms are not developed yet. Although the firing patterns of single neurons are already an accurate representation of the biological model, the complexity of the biological model of a full neural network and the learning process is still a subject of research: biological neural networks are organs in living creatures; they grow up, develop their strengths, can take classes and learn from a teacher. The proposed computational model was liquid computing [90], where a neural network is connected, but not trained in a special way. Only parameters of the whole network are adjusted to obtain different firing patterns. This network is then used as a filter: the input describing the problem to be solved is sent into the network, followed by a readout of the time dependent firing pattern in the neural network. The hope is that the readout of the network can be interpreted more easily than the input sent into to the network. Mathematically this has sound motivation: for many problems a low dimensional input can be projected into a higher dimensional space, via e.g. a neural network, which in turn simplifies a subsequent classification. An illustrative example for this type of projection is the xor problem. A single linear classifier in a 2D space cannot solve the problem. However, by extending the input to 3D space, a separating hyperplane can be drawn that allows to solve the xor problem linearly. The xor problem and its solution with a small neural network is illustrated in Fig. 6.2.

The "tempotron" [91] was studied as a candidate for the readout of the liquid computing network. It is a single time dependent analog neuron which is trained by a supervised machine learning algorithm, similar to a perceptron [92]. In the first studies [86, 93] this liquid computing approach was compared to the direct use of a perceptron neural network. The perceptron neural network already turned out to provide quite accurate results in the 3D track parameter estimation: resolutions of about $\Delta z \approx 1\,\mathrm{cm}$ were achieved for straight tracks [86]. In comparison, the analog neurons were set up as classification networks to discriminate tracks from the IP from tracks displaced by 10 cm along the z-axis. Since the classification correctness of the analog networks only reached 70 %, the precision with the perceptron was found to be significantly better. Therefore, the development of the analog neural networks as a L1 trigger was discontinued. This was the foundation of the present neural network model.

This chapter covers the current development state of the neural network trigger, followed by experimental performance measurements of a basic neural network and studies of possible improvements. In Sec. 6.1 the method of Least Squares (LS) track fitting is introduced in order to demonstrate the general task of a 3D track reconstruction algorithm and to motivate the used representation of neural network input values. An introduction to the theory of the machine learning model used in the setup of the neural network trigger is presented in Sec. 6.2. It addresses the structure of the neural network built as a directed graph of neurons and the supervised training algorithms. The application of the neural network model to the concrete track trigger task and the implementation details are described in Sec. 6.3. This comprises the sectorization of the tracking problem with specialized expert neural networks and the preprocessing algorithm that calculates the input values. Experimental results of a basic neural network configuration with the architecture presently realized in the hardware implementation are presented in Sec. 6.4. Follow up studies of this basic neural network with different training and configuration options of this basic network are shown in the Appendix D.1. Possible improvements with the 3D finder from Chapter 5 as input are be explored in Sec. 6.5. To make full use of the improved 3D finder efficiency, an optimized neural network architecture is developed, which could be used in future hardware in combination with the 3D finder.

6.1. Least Square Track Fit

In a general LS fit, a parametrized function - here the track model - is fitted to a set of measurement points by minimizing the sum of squared distances of the measurement points to that model function. Under ideal conditions for a linear track model an analytic solution exists for this LS fit, which is the optimal solution minimizing the squared distances [94]. But without error corrections, its performance decreases significantly under realistic conditions due to noise from superimposed background and due to required nonlinear corrections to the assumed linear track model. Most prominently, these drawbacks affect tracks with low transverse momenta, where the tracks are strongly curved. This is where the neural network becomes important: it can learn the non-linear corrections to a linear track model from a training data set and provide robust results even in the presence of background.

Since the precise understanding of this LS fitting method is crucial for the optimization of the input features for the neural network, this section covers the theory of LS fits, in a version applicable to the 3D track reconstruction at the Belle II L1 trigger. In the early phase of this thesis, the LS-Fit was implemented in a common development framework with the neural network where it was successfully used to obtain the present input representation for the neural network trigger. The implemented LS-Fit reference model described in this section is oriented on [95]. The important key feature of the LS-Fit is the symmetry of the 3D tracking task in the 2D track parameters. This symmetry was first propagated to the neural network input in [96], which had the advantage that a phase space sectorization was no longer required (see Sec. 6.3.1). By adopting the linear representation of the stereo TS hits used in the LS fit, the neural network can especially learn the nonlinear corrections. Furthermore, the reference model of the LS fit can be used to identify the difficulties that make a track reconstruction fail and thus to optimize the neural network preprocessing.

At the L1 trigger of Belle II, in parallel to the neural network, a 3D fitter is used for the reconstruction of the 3D track parameters of single tracks found by the track finder (see Sec. 3.3). Like in the neural network trigger, the inputs in the Belle II 3D fitter [68] are the CDC TS hits and a set of p_{T}, ϕ-estimates for each track recognized by the 2D finder. As output, a set of precise 3D track parameters $(p_{\mathrm{T}}, \phi, \theta, z)$ for each single track in the event is provided as input to the global decision logic of the first level trigger. This 3D fitter is based on the analytic mathematical model of the LS track fit [68], but it has minor differences in the input representation compared to the reference LS-Fit model presented in this section: while the model presented here (oriented on [95]) uses a quadratic equation to calculate the z-positions of the stereo hits (see Eq. 6.17 in Sec. 6.1.4), in the Belle II 3D fitter [68] trigonometric functions are used for this task. In contrast, in the neural network input representation presented in [96] and refined in [67], these z-positions are approximated by a linear function (see Eq. 6.26 in Sec. 6.2.4).

With both 3D track reconstruction methods installed in parallel in the Belle II trigger, a hybrid method might be considered to achieve the best overall performance;

since for tracks with nearly ideal conditions, the LS fitter results are optimal and can be complemented by the neural network estimates for the case of non-ideal tracks. Such tracks with nearly ideal conditions could be high-p_T tracks, i.e. almost straight tracks, if the background levels are still small. But for curved tracks (low-p_T) with noise, incomplete or nonlinear blurred TS hit information, the track reconstruction with the neural network should be preferred. The benefit of the neural network becomes especially important with the large background rates expected at the design luminosity of SuperKEKB. Since both methods have advantages in their respective domains, they are going to be installed redundantly in Belle II.

6.1.1. Method of Least Squares

In a LS fit, a parametrized function $f_{\vec{w}}$ - the model - with parameters $\vec{w}$, is fitted to a set of data points by minimizing the squared distance of the points to the model function. Under the assumption that a true function f exists, which describes data points $(y_i, \vec{x}_i)$ by $y_i = f(\vec{x}_i)$ the LS fit provides $f_{\vec{w}}$ as an approximation of this true function f. Given a set of N data points $(y_i, \vec{x}_i)$ with $i \in [1, N]$, the model function $f_{\vec{w}}$ is supposed to describe the data points in the same way as the true function f by

$$y_i = f_{\vec{w}}(\vec{x}_i) = f(\vec{x}_i) \tag{6.1}$$

where the parameter vector $\vec{w}$ has to be determined from the fit. The method of least squares minimizes the sum of the squared errors S of the fitted model and the data points.

$$S = \sum_{i=1}^{N} (y_i - f_{\vec{w}}(\vec{x}_i))^2 \tag{6.2}$$

In general, the result can be defined as

$$\vec{w} = \underset{\vec{w}}{\operatorname{argmin}}(S) \tag{6.3}$$

6.1.2. Linear Least Squares

In the case of a linear model function f, the LS fit can be solved completely analytically. Here the question is to find the coefficients $\vec{w}$ of a linear function $f_{\vec{w}}$: $f_{\vec{w}}(x_1, x_2, \ldots, x_M) \to y$ that maps the M input values $x_m, m \in [1, M]$ onto an output value y. This general linear function can be described by:

$$y = f(\vec{x}) = w_0 + w_1 \cdot x_1 + \ldots + w_m \cdot x_m = w_0 + \sum_{j=1}^{M} w_j x_j \tag{6.4}$$

In order to find the analytic solution, at first the data points are rearranged such that the function can be expressed like

$$y = f_{\vec{w}}(x_1, \ldots, x_M) = \begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_M \end{pmatrix} \cdot \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_M \end{pmatrix} = \vec{x} \cdot \vec{w} = x_j w_j \tag{6.5}$$

where $\vec{x}$ is now redefined as an $M+1$ dimensional vector with $x_0 = 1$ and the other elements $x_1, \ldots, x_M$ are the input values from the dataset. So with M input values, the vectors $\vec{x}$ and $\vec{w}$ are $M+1$ dimensional.

With the data set of N data points $(y_i, \vec{x}_i)$, a single matrix equation describing all the data points at once can be formulated

$$Y = X \cdot \vec{w} \tag{6.6}$$

where Y is the $N \times 1$ matrix of all the y_i ($i \in [1, N]$) values in the dataset and X is the $N \times (M+1)$ matrix of all input vectors $\vec{x}_i$:

$$Y = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix} \qquad X = \begin{pmatrix} \vec{x_1}^T \\ \vdots \\ \vec{x_N}^T \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \ldots & x_{1M} \\ \vdots & & \ddots & \\ 1 & x_{N1} & \ldots & x_{NM} \end{pmatrix} \tag{6.7}$$

where x_{ij} are the elements of the matrix X. The rows i are the indices of the data set entries ($i \in [1, N]$) and the columns j are the indices of the arguments ($j \in [0, M]$). Rewritten in this form, the analytic solution for $\vec{w}$ can be easily obtained. Multiplying both sides from the left with X^T gives

$$X^T Y = X^T X \vec{w} \tag{6.8}$$

where the product $X^T X$ on the right side is now a $(M+1) \times (M+1)$ dimensional square matrix which can be inverted analytically. Multiplying both sides with the inverse $(X^T X)^{-1}$ gives the complete analytic solution for $\vec{w}$

$$\vec{w} = (X^T X)^{-1} X^T Y \tag{6.9}$$

Under ideal conditions, the data points are truly distributed according to the fitted linear function and all deviations of the errors are Gaussian distributed. In this case, the analytic solution that minimizes the sum of squared errors is the optimal solution [94]. With the fitted solution $\vec{w}$, the sum of squared errors which is minimized by the LS fit can be expressed as

$$S = (Y - X \cdot \vec{w})^2 = (Y - X \cdot (X^T X)^{-1} X^T Y)^2. \tag{6.10}$$

The LS method is also closely related to the method used in the precise offline track reconstruction. There a Kalman Filter [97] is used, which is an iterative version of the LS fit. Instead of looking at all the data points at once, they are added one by one. In offline tracking, where time is not an issue, it is often useful to add the points along a track iteratively, as it allows to solve the LS fit even for general nonlinear functions. Linearity has only to be assumed in the propagation from one measurement point to the next. Additionally, it allows for the improvement of error models in the offline track reconstruction. A common application in other fields are navigation systems [98], where for example the next gearing moves required to stabilize a flying object on its track can be found by extrapolating from the current state.

6.1.3. Linear 2D Circle Fits

In the case of a 2D circle fit, a linearization of the problem can be used to apply the linear LS fit method. In a 2D plane, a general circle is described by three parameters, the center of the circle $\vec{M} = (m_x, m_y)$ and its radius r. Under the assumption of a circle that passes through the origin, $\vec{M}^2 = r^2$, only two free parameters are left. This is the case for a charged particle starting at the IP. With a magnetic field in z direction, the particle in the transverse plane can be described by only two parameters: p_T and ϕ_0, where p_T is proportional to the radius of the circle and ϕ_0 describes its orientation at the IP. An illustration of a track from the IP and its related hits in a 2D coordinate system is shown in Fig. 6.3. Note that the track parameter ϕ_0 is the direction of the tangent vector of the track at the IP, while the angle ϕ is the polar representation of the circle center in polar coordinates. Hence, the track parameter ϕ_0 is equal to ϕ shifted by $\pi/2$, where the sign of the shift is determined by the sign of the particle charge.

Given a set of Cartesian points (x_i, y_i) on the circle, a quadratic equation for these points on the circle is given by

$$(x_i - m_x)^2 + (y_i - m_y)^2 = r^2 = m_x^2 + m_y^2 \tag{6.11}$$

which can be rearranged to

$$x_i^2 + y_i^2 = 2x_i m_x + 2y_i m_y. \tag{6.12}$$

Beneficially, the $x_i^2 + y_i^2$ can be calculated for each point separately which allows to introduce a conformal mapping on the points:

$$x_i' = \frac{2x_i}{x_i^2 + y_i^2} \qquad y_i' = \frac{2y_i}{x_i^2 + y_i^2} \tag{6.13}$$

This conformal mapping allows to rewrite this quadratic equation as a linear equation

$$1 = x_i' m_x + y_i' m_y \tag{6.14}$$

$$y_i' = \frac{1}{m_y} - x_i' \frac{m_x}{m_y} = m_y^{-1} - x_i' \cot(\phi) \tag{6.15}$$

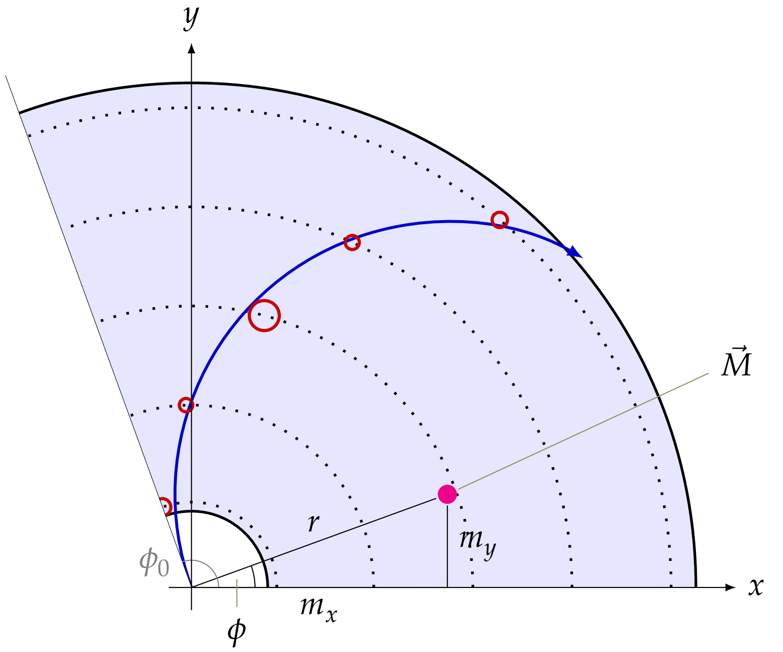


Figure 6.3.: Illustration of a circle fit in the x, y-plane. The hits with the drift times are visualized as red circles. The track (blue) is fitted to the hits, under the constraint that it passes through the origin. From the resulting circle center $\vec{M}$ (magenta) the 2D track parameters p_{T}, ϕ_0 can be calculated. The (ϕ, r) shown here are the polar coordinates of the circle center $\vec{M}$. The track parameter ϕ_0 is the direction of the tangent vector at the IP corresponding to ϕ shifted by $\pi/2$.

This is a linear equation in the coefficients $\frac{m_x}{m_y} = \cot(\phi)$ and m_y^{-1}. These coefficients can be fitted with the basic analytic LS fit equation.

In the 2D fit of the particle from the IP, it is convenient to use only the axial TS as input. As the axial wires are parallel to the z-axis, the x, y-positions of the wires at the endplates can directly be used as hit positions in the transverse plane. But in the case of TS hits, the drift times provide an additional distance measurement between the track and the hit. With the Left/Right (`LR`) information of the TS (see Sec. 3.3.3) in combination with a first estimate of the 2D track parameters from the track finder, the drift times can be translated into directed offsets for the TS hit positions (x_i, y_i). To this end, a model of the x, t-relation, the drift time to distance relation, is required (see Sec. 3.2.5). Since the drift velocity in the CDC gas admixture is almost constant, the x, t-relation can be approximated by a rather simple model: the drift distance is proportional to the drift time multiplied by a constant (i.e. $x = v \cdot t$ with a constant v). Then the x, y-positions of the wire hits simply have to be shifted in the direction given by the 2D track estimate from the track finder combined with the Left/Right information (sign of the direction). In this way the drift times can be used directly to improve the hit positions prior to the 2D circle fit and the 2D fit can be significantly improved.

The required shifting by the drift times can be seen in Fig. 6.3. The TS hits are illustrated as small red circles, where the circle radius is proportional to the drift time. These drift time circles have to touch the track circle and, therefore, the hit can either be placed inside or outside of the circle. With a known 2D track estimate from the 2D finder, i.e. the position of the circle center $\vec{M}$, the shift direction is known. Additionally, the Left/Right information of each circle gives the sign of the shift.

6.1.4. 3D Helix Fit

The idea in the LS fit is to treat the particle trajectory as a perfect helix that passes through the origin. Due to the magnetic field in z-direction, the helix describes the superposition of the transverse motion and the longitudinal motion of the particle. The transverse motion of the track is described by the already discussed circle in the projection onto the transverse plane, where the axial wires appear as points in this projection. This allows to perform a 2D fit of a circle to the axial TS hits in order to obtain the precise values of the 2D track parameters (p_T, ϕ_0). Using the 2D fit result, the 3D positions of the stereo TSs can be calculated and represented in the s, z-plane, where s are the arc lengths along the 2D track to the wire layers containing the stereo TSs and z are the z-positions of the hits along the wires of the stereo TSs. In the s, z-plane, the 3D fit becomes linear [95] and can be solved by the linear LS fit. The schematic trajectory of the 2D fit together with the axial TS hits in the x, y-plane is shown in Fig. 6.3.

In order to describe the longitudinal motion of the track in z-direction with a linear equation, the 3D hit positions on the stereo wires are required. Beneficially, these can be calculated from the already known 2D track parameters combined with the projection of the stereo TSs into the transverse plane. While the axial TSs appear as points in the transverse plane, due to their stereo angle the stereo TSs appear as short line segments (compare Fig. 5.1 in the 3D finder Chapter 5). After the precise 2D values are known, the intersections of the 2D track with the stereo TSs can be calculated and represented in the s, z-plane, where the 3D tracks can be linearly reconstructed.

In order to obtain the intersection points of the stereo sense wires with the 2D track, the intersection point of a straight line, representing the stereo TSs in the transverse plane, with a circle, representing the 2D track, has to be calculated. The line segments of the stereo wires projected into the transverse plane are given by the stereo wire mounting positions at the forward endplate $\vec{F} = (f_x, f_y)^T_\mathrm{FW}$ and at the backward endplate $\vec{W} = (w_x, w_y)^T_\mathrm{BW}$. These stereo TSs line segments can be

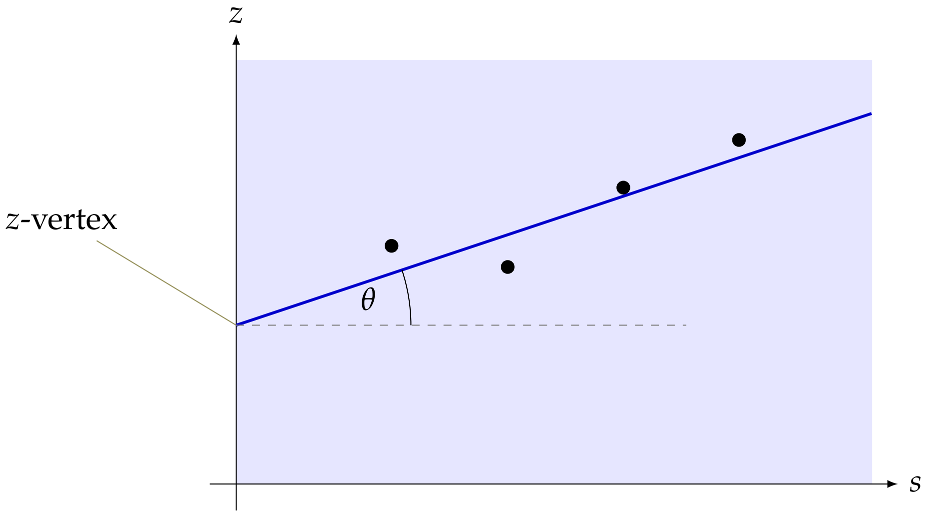


Figure 6.4.: Illustration of the 3D track parameter reconstruction with a line fit, after the stereo TS hits are transformed into the s, z-plane using the 2D track parameters. For the stereo wire hits (represented as black dots), the s-axis shows the arc length along the 2D circle to the hits and the z-axis shows the longitudinal z-positions of the hits.

rewritten as a space point and a direction vector with a single free parameter λ:

$$\vec{L}(\lambda) = \begin{pmatrix} w_x \\ w_y \end{pmatrix}_{\mathrm{BW}} + \lambda \begin{pmatrix} d_x \\ d_y \end{pmatrix} = \vec{W} + \lambda \vec{D}. \tag{6.16}$$

where $\vec{D} = (d_x, d_y)^T$ is the direction vector, which can be calculated as $\vec{D} = \vec{F} - \vec{W}$ and λ is the length along the wire in the transverse plane. Note that each point on the stereo wire given by λ in this transverse plane can simply be used to find the corresponding 3D TS hit position along the z-axis. Now the intersection point of a stereo wire (represented as a straight line) and the 2D track (represented as a circle) can be calculated by inserting the equation for the straight line (Eq. 6.16) into the circle equation (Eq. 6.11) and then solving for λ:

$$(w_x + \lambda d_x - m_x)^2 + (w_y + \lambda d_y - m_y)^2 = r^2. \tag{6.17}$$

This is a quadratic equation in λ and can be solved analytically. Although such a quadratic equation has two solutions $\lambda_{1,2}$, the ambiguity can easily be resolved since the line segment is rather short: only the solution λ that leads to a point on the wire in between the backward endplate position $\vec{W}$ and the forward endplate position $\vec{F}$ is correct. In [95] the solving a quadratic equation for each hit is pointed out as an advantage as no trigonometric functions are required. Although at the trigger level, where the algorithms are implemented in hardware, trigonometric functions can easily be implemented via lookup tables. Therefore, the Belle II 3D fitter [68] uses trigonometric functions instead of a quadratic equation. In the present neural network input [96, 67] this step is linearly approximated (see Sec. 6.2.4).

Once λ is found in the transverse plane and by including the z-components in $\vec{L}(\lambda)$, the 3D hit position $\vec{L}(\lambda) = (l_x, l_y, l_z)$ of the stereo wire hit can be obtained. Besides the z-component of the stereo hit (l_z), the arc length s along the 2D track to the hit is required to linearize the 3D track reconstruction problem. Conveniently, this arc length can be found as the intersection of two circles: the 2D track is followed until it crosses the CDC layer containing the wire (compare Fig. 6.6). For a 2D track from the IP with a radius r_{2D}, this arc length can be described solely by the radii of the two circles. Since the stereo wires are not curved along the CDC layer, a small correction can be applied here: as radius of the CDC layer (r_L), the transverse component of the stereo TS hit position can be used $r_L = \sqrt{l_x^2 + l_y^2}$. The arc length s is then given by

$$s = 2 \cdot r_{2D} \cdot \arcsin\left(\frac{r_L}{2 \cdot r_{2D}}\right) \tag{6.18}$$

which can be seen from the geometry in Fig. 6.6 using a Thales circle.

This calculation has to be repeated for each stereo TS related to the 2D track. The result is a representation of each stereo TS hit in the s, z-plane, illustrated in Fig. 6.4. In this representation, a simple linear fit can be used to extract the 3D track parameters θ and z. The y-intercept corresponds to the z-vertex position and the slope is $\cot(\theta)$. As before in the case of the 2D fit, the drift times are ideally applied prior to the fit. This means, the drift times should be used to shift the stereo line segments, before the intersections with the 2D circle are calculated.

So far this describes the core method of the 3D fit, which is partially used in the preprocessing of the neural network. Of course, to achieve an optimal 3D fit there is room for improvements (especially regarding the exact implementation of the involved approximations). For example, the 2D fit can be iterated to obtain improved directions to apply the drift times. The 3D fit result can also be iterated, in order to include the 3D fit results in the calculation of the drift time offsets. Furthermore, higher order polynomials can be used to describe the x, t-relation. But the optimizations in the use of the drift times have to be compared with the achievable resolution of the t_0. Here the neural network is well suited, as these corrections can be learned from the training data set, without the requirement of an explicit model. Therefore, the simple approximations from the 3D fitter are used in the preprocessing of the neural network as described in Sec. 6.3.2.

6.2. Neural Network Model

Neural networks are a supervised machine learning method to train a universal function approximator. They learn the function to be calculated by example from a training data set and store the information in weights, an internal degrees of freedom. This makes it easy to change the calculated function of a neural network with a fixed graph structure, simply by changing the weights. In general, a sufficiently large training data set is required, with a good representation of the input and output values of the function that should be learned by the neural network.

The neural network used in the trigger is a Multi Layer Perceptron (MLP), which is a supervised ML solution for robust function approximation. It is a feed-forward network with the structure of a directed acyclic graph (DAG). This graph structure of the used three layer network is shown in Fig. 6.5. It has an input, a single hidden and an output layer. The implemented neural networks receive the CDC axial and stereo TS information, combined with the 2D track estimates (ϕ, ω) of single tracks from the 2D finder as input where omega is the curvature of the track and thus proportional to the inverse transverse momentum ($\omega \propto p_\mathrm{T}^{-1}$). Using these input values, an input vector is calculated for each single track such that the neural network can be used to reconstruct the two 3D track parameters (θ, z) as output. To this end, one TS per SL per track is selected in a preprocessing step and each of the 9 selected TS is represented by three values, which results in an input vector of length 27. Furthermore, the network presently implemented in hardware uses a single hidden layer with three times the size of the input layer, i.e. 81 hidden nodes.

Internally, each node (perceptron) computes the weighted sum over its inputs and evaluates this with a nonlinear activation function to calculate its output. The outputs of each layer are fed forward as input to the next layer, hence the name feed-forward network. In general, two main purposes of feed-forward neural networks can be distinguished: classification and regression. In classification networks, a class of an input vector is estimated by applying a threshold on the output value. This has the biological analogy of a neuron that fires if it has sufficient activation from its inputs. In networks used for regression, the floating point output value is directly interpreted as the scaled estimate of a parameter. Given a sufficient size of a single hidden layer (i.e. two layers of weights) and nonlinear activation functions, it is proven mathematically that any continuous function on the $[0, 1]$ hypercube can be approximated with a single hidden layer [99]. Therefore, the use of a shallow neural network with only a single hidden layer is justified. From the hardware perspective this is beneficial, as additional layers increase the latency of the method.

A general introduction to the concept of supervised ML is discussed in Sec. 6.2.1. The model of a single neuron (perceptron) is described in Sec. 6.2.2, followed by the integration of these neurons to a neural network in Sec. 6.2.3. Concrete implementation details on the calculation of the input and output values for the 3D track reconstruction task are described in Sec. 6.2.4. The supervised training algorithm, used to train the weights in the network is presented in Sec. 6.2.5.

6.2.1. Supervised Machine Learning

In supervised Machine Learning (ML), generic estimators learn their concrete task by example with a training algorithm. Inspired by nature, adaptable estimation tools are constructed, which are not designed to calculate some specific function.

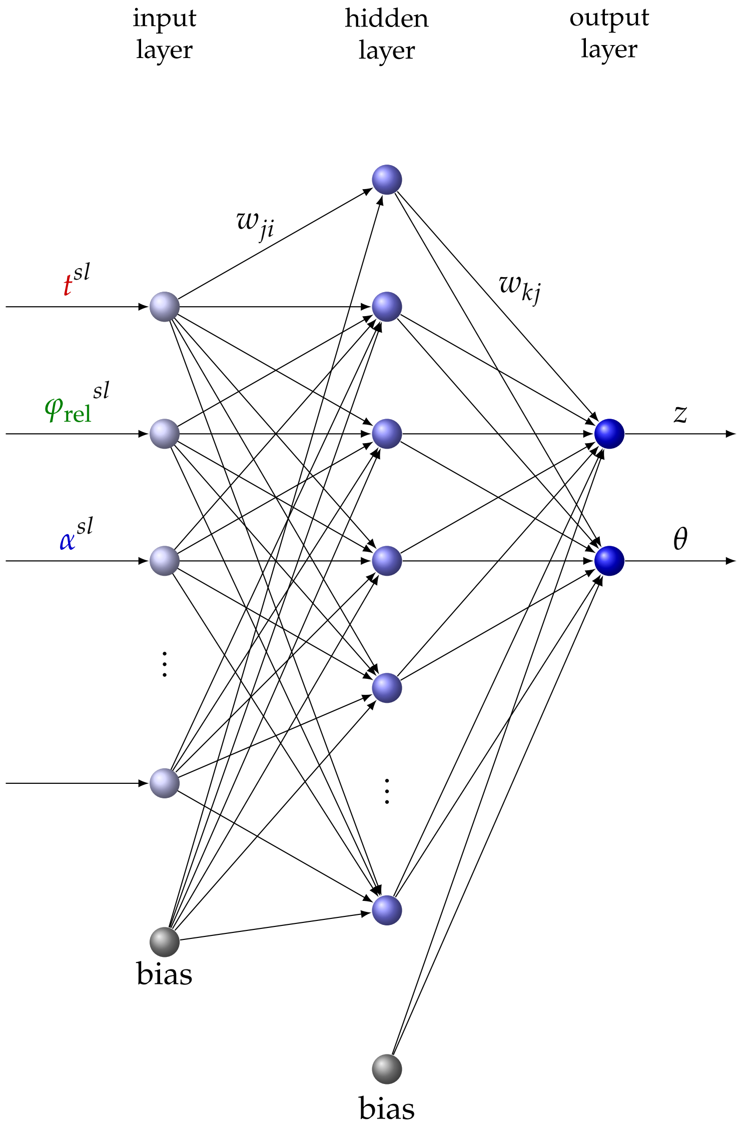


Figure 6.5.: The structure of the three layer neural network used for 3D single track reconstruction. This MLP has 27 inputs input nodes in total to receive the prepared TS hit information for a single track. Each input of the 9 input TS (one per SL), is represented with the three values $\varphi^{\text{rel}}, \alpha, t$. The two values at the two output nodes are interpreted as scaled 3D track parameters θ and z. The size of the hidden layer is normally set to 81 nodes (three times the size of the input layer). The constant bias nodes of the input and of the hidden layer are shown in gray.

Indeed, the calculated function does not even have to be known explicitly, it just has to be represented by a sufficiently large training dataset. The ML model defines a generic estimation procedure $m_{\vec{w}}$ based on internal degrees of freedom (the weights $\vec{w}$) together with a training algorithm. Given an input vector $\vec{x}$ and a weight vector $\vec{w}$ the model $m_{\vec{w}}$ calculates an output vector $\vec{y}_{\text{out}}$

$$\vec{y}^{\,\text{out}} = m_w(\vec{x}) \qquad\qquad \vec{y}^{\,\text{true}} = f(\vec{x}). \tag{6.19}$$

Here $m_{\vec{w}}$ approximates the true function f, which maps an input $\vec{x}$ to its true output $\vec{y}_{\text{true}}$.

The information is learned by adjusting the weights $\vec{w}$ with a training dataset D containing N samples $(\vec{x}_i, \vec{y}_i^{\,\text{true}}) \in D$, where $\vec{y}_i^{\,\text{true}}$ are the training target values. The training algorithm iteratively minimizes a cost function $C(D, m_w)$ defined on the current outputs $\vec{y}_i^{\,\text{out}}$ and on the true target values in the training data $\vec{y}_i^{\,\text{true}}$. The cost function is a measure of the difference of the network output to the training target values. An optimal set of weights $\vec{w}$

$$\vec{w} = \underset{\vec{w}}{\text{argmin}}(C) \tag{6.20}$$

minimizes the cost C on the training data set D.

Since the optimal weight set $\vec{w}$ can not be calculated in general, heuristic algorithms are used with the objective to minimize the cost function. Starting with randomly initialized weights $\vec{w}$, they are iteratively adjusted such that the cost on the training data is minimized. In order to find a close to optimal solution, the training algorithm should avoid to get stuck in the local minima of the cost landscape. As will be discussed later in Sec. 6.2.5, the MLP neural networks for the Belle II trigger are trained with a backpropagation training algorithm using the mean squared error as cost function (Eq. 6.27).

6.2.2. Single Perceptron Neurons

The basic building block of the MLP neural networks are the single neuron nodes of the perceptron type. Each node j has its weight set $\vec{w}_j$ to compute the weighted sum $\vec{w}_j \cdot \vec{x}$ over its inputs $\vec{x}$. This linear equation (scalar multiplication of the weights with inputs $\vec{w}_j \cdot \vec{x}$) is followed by the use of a nonlinear activation function f to calculate the neuron output. This nonlinear activation extends the field of applicability to general nonlinear equations. In order to modulate the activation threshold, an additional constant bias weight w_{j0} is added to the weighted sum. The complete function computed by a single neuron with N inputs is

$$y_j = f\left(\begin{pmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \cdot \begin{pmatrix} w_{j0} \\ w_{j1} \\ w_{j2} \\ \vdots \\ w_{jN} \end{pmatrix}\right) = f\left(\sum_{i=1}^{N} w_{ji} x_i + w_{j0}\right) \tag{6.21}$$

where y_j is the output of neuron j and x_i are its N input values. In order to simplify the notation, the bias weight w_{j0} is included as 0^{th} element in the weight vector. To this end, the input vector $\vec{x}$ is extended to $N+1$ elements, with the fixed 0^{th} element $x_0 = 1$. Furthermore, the $w_{ji} \in \vec{w}_j$ are the $N+1$ weights defining the function calculated by the neuron node. The simplified notation of a single neuron with the $N+1$ dimensional input and weight vectors is:

$$y_j = f\left(\sum_{i=0}^{N} w_{ji}x_i\right) = f(w_{ji}x_i) \tag{6.22}$$

where y_j is again the output of neuron j, the $w_{ji} \in \vec{w}_j$ are the $N+1$ weights and the $x_i \in \vec{x}$ are the $N+1$ inputs. As nonlinear activation function f, the hyperbolic tangent (tanh) is used by the neural networks here. In the last equation, summing over double indices is implicit. Due to the nonlinearity of the activation functions in the single neurons, the full network can approximate general nonlinear functions. The hyperbolic tangent is a frequently used nonlinear activation function with a sigmoid shape symmetric in the output range $[-1, 1]$. Its differentiability is an important feature used in the calculation of the weight updates in backpropagation training algorithm. Depending on the field of application, a threshold could be applied to the output value to obtain a binary result, i.e. whether the neuron fires or not. This is common for classification problems, while the continuous output value is desired in the regression problem at hand.

6.2.3. Multi Layer Perceptron Network

The single neuron is still limited in its power. Once many neurons are combined into a network, the full magic unfolds. The basic network setup which is presently implemented in the hardware, has three layers, an input layer of size 27, a hidden layer of size 81 and an output layer of size two, where the size refers to the number of nodes in each layer. The architecture of such a three-layer network is shown in Fig. 6.5. The w_{ji} are the matrix elements of the first weight matrix W_1 that is used to project the input vector onto the hidden layer. The elements w_{kj} of the second weight matrix W_2 are used to project the hidden layer onto the output layer. In the simplified notation, where each input vector is extended by a 0^{th} element $x_0 = 1$, the full network calculates the following equation:

$$z_k = f(w_{kj}f(w_{ji}x_i)) \tag{6.23}$$

where summing over double indices is again implicit. The expression $w_{ji}x_i$ is the first matrix multiplication $W_1 \cdot \vec{x}$. On each resulting element, the activation function f is applied to obtain the output vector $\vec{y} = f(W_1 \cdot \vec{x})$ of the hidden layer. These intermediate results of the hidden layer are fed forward as input vector to the following output layer. In the output layer, the next matrix multiplication $w_{kj}y_j$ is carried out ($W_2 \cdot \vec{y}$) which again is processed by the activation function f in order to obtain the resulting output vector $\vec{z} = f(W_2 \cdot \vec{y})$. For the present hardware network with 27 input nodes, 81 hidden nodes, two output nodes and two bias nodes (one in the input and one in the hidden layer), the weight matrices have the dimensions $\dim(W_1) = (28 \times 81)$ and $\dim(W_2) = (82 \times 2)$ corresponding to a total number of 2432 weights. An experimental software simulation in basf2 was generated to test the performance of a larger network consisting of 127 hidden nodes, which corresponded to a total number of 3812 weights. Although slightly improved resolutions could be observed with the 127 node network, the 81 hidden node network is already close to optimal [67, 100]. Therefore, the improvement with the larger hidden layer was deemed negligible and the 81 hidden nodes deemed sufficient for future developments.

The parallel structure of the neural network makes it very well suited for an implementation in parallel hardware. With the use of hardware lookup tables, the computational effort for the calculation of the activation functions is small. The main computational complexity resides in the two matrix multiplications, which is a very well known parallelizable task. Therefore, the neural network algorithm allows massively parallel processing while guaranteeing a short deterministic runtime.

6.2.4. Input and Output Features

In order to use the neural network for the 3D track reconstruction, the input and output features have to be properly selected. Starting from the TS hits and the found 2D track parameters as input, a preprocessing step used to select at most one TS per SL per track and to calculate the input vector for the neural network. Using the values at the neural network output nodes, a scaling is used to obtain the 3D track parameters (z, θ).

Input Calculation

The calculation of the input vector exploits the symmetry of the 3D tracking problem in the 2D track parameters p_{T} and ϕ. Similar to the second step in the LS helix fit (see Sec. 6.1.4), the TS positions are represented in a 2D coordinate system relative to the found 2D track ($\phi_i^{\mathrm{rel}}, \alpha_i$), where the estimation of (z, θ) becomes linear. Additionally, the precise drift time information t_i is included as third input value per TS. Thus, the error prone transformation of the drift time into a distance measurement can be avoided (see discussion in Sec. 6.1.4). The resulting input vector has 27 elements: with one hit per SL, there are 9 possible TS hits for each track and each of these TS is represented by the three values $(\phi_i^{\mathrm{rel}}, \alpha_i, t_i)$.

The input calculation starts with the parameters of a found 2D track and its related TS hits. The 2D track parameters are given by (ω, ϕ), where $\omega = q \cdot r_{2D}^{-1} \propto q \cdot p_{\mathrm{T}}^{-1}$ is the signed curvature or signed inverse radius of the 2D track and the sign is given by the charge q of the particle. Additionally, the TS hit information is given (see Sec. 3.3.3) by the tuple $(\text{TS-id}_i, t_i, \texttt{LR}_i, \texttt{prio}_i)$, where TS-id is the id of the TS (TS-id $\in [1, 2336]$), t_i is the drift time at the priority wire in units of trigger timing bins ($\approx$ 2 ns), $\texttt{LR}_i$ is the Left/Right information that describes whether the track passed left or right of the priority wire and $\texttt{prio}_i$ is the position of the priority wire within the TS. This Left/Right information is determined by a lookup table from the pattern of wire hits within the TS. It can have three values: left, right or undetermined. For the priority position, three values are possible: first priority, second left or second right. The second priority hits are one layer outwards of the first priority position. With the TS-id$_i$ and $\texttt{prio}_i$ the priority wire is known and the 2D position of this wire at the CDC end plate can be obtained $(\text{TS-id}_i, \texttt{prio}_i) \rightarrow (\phi_i^{\text{wire}}, r_i^{\text{wire}})$. Here ϕ_i^{wire} is a relative TS-id within the SL and r^{wire} is the radius of the layer containing the priority wire. Since the wires in the second priority layers are shifted by half a wire cell w.r.t. the first priority position, the combined ϕ^{wire} lies in the range $\phi_i^{\text{wire}} \in [-0.5, n_i^{\text{wires}} + 0.5]$, where n_i^{wires} is the number of wires in the SL of hit i.

The geometry of the calculated input values $(\phi_i^{\text{rel}}, \alpha_i, t_i)$ and a description of the parameters, is illustrated in Fig. 6.9. The input parameter α_i is the arc length of the 2D track (s) from the origin to the wire layer with the hit divided by $2 \times r_{2D}$ where r_{2D} is the radius of the 2D track ($r_{2D} \propto p_{\text{T}}$). Hence, it is an approximation of the arc length s_i along the track from the origin to the intersection point of the stereo wire with the 2D track, where the stereo wire is assumed to have a constant wire radius r_i^{L} and is therefore not a straight line segment but a curved line segment on the layer. This approximation is justified since the stereo angles are small and, therefore, the lengths of the stereo line segments in the transverse plane are small compared to the wire radius r_i^{L}.

Equivalently, α_i can be interpreted as the crossing angle of the track with the wire layer containing the priority wire of the TS (see Fig. 6.6). The parameter α_i is calculated by

$$\alpha_i = \arcsin\left(\frac{1}{2} r_i^{\text{L}} \omega\right). \tag{6.24}$$

Thus, the possible range of α values is $\alpha \in [-\frac{\pi}{2}, \frac{\pi}{2}]$.

The parameter ϕ_i^{rel} is the distance in ϕ between ϕ_i^{wire} and the crossing position of the 2D track with the priority wire layer. Therefore, it describes mainly the position z_i of the stereo TS hit in the approximation of small stereo angles, with an additional offset corresponding to the drift time. In the case of axial TS hits, the value of ϕ^{rel} is small compared to the stereo wires and should ideally be proportional to the drift time. Note that an ideal axial TS, which is located perfectly on the 2D track, must have $\phi_i^{\text{rel}} = 0$ and $t = 0$. This parameter ϕ^{rel} is given in units of wires within the SL. With n_i^{wires} the number of wires in SL i, the possible range of its values is $\phi_i^{\text{rel}} \in [0, n_i^{\text{wires}}]$.

In order to get ϕ_i^{rel}, at first the position where the track passes through the layer (ϕ_i^{ref}) is calculated from the 2D track parameters. This ϕ_i^{ref} is the reference position

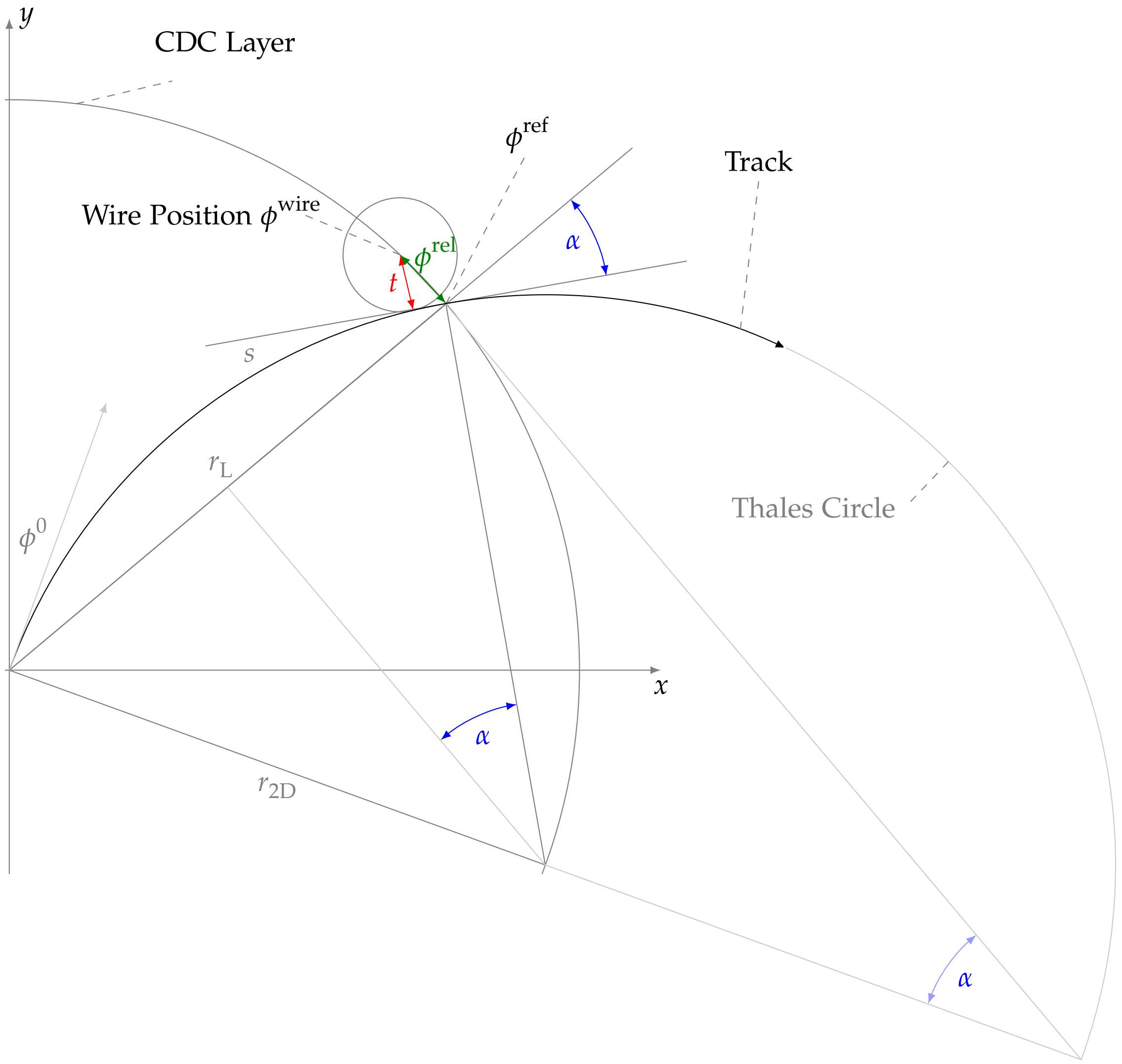


Figure 6.6.: Description of the track segment representation as input for the neural network. Shown is a 2D track with radius $r_{\text{2D}} = \omega^{-1}$ and an axial hit in the CDC layer with wire radius r_{L}. The angle α is the arc length of the track to the layer (s) divided by $2 \times r_{\text{2D}}$, or equivalently the crossing angle of the track at the intersection of the track with the CDC layer (ϕ^{ref}). ϕ^{rel} is the distance between the wire position of the hit and the intersection ϕ^{ref}. The drift time t is proportional to the distance of the track to the hit. For stereo hits, ϕ^{rel} contains the information on the z-position of the hit.

of an ideal axial TS sitting perfectly on the track. Using the previously calculated α_i it is given by

$$\phi_i^{\mathrm{ref}} = n_i^{\mathrm{wires}} \cdot \left(\frac{\phi^0 - \alpha_i}{2\pi} \right) \tag{6.25}$$

where ϕ^0 is the 2D track parameter, i.e. the tangent direction of the track at the IP. Next the relative TS id in the SL (ϕ_i^{wire}) is combined with the priority information. The distance ϕ_i^{rel} is then given by the difference of these two values

$$\phi_i^{\mathrm{rel}} = \phi_i^{\mathrm{wire}} - \phi_i^{\mathrm{ref}} \tag{6.26}$$

The possible range of ϕ_i^{rel} which are used as input for the neural network is determined from the histograms in the preprocessing. An example of such a histogram is shown in Fig. 6.14.

The drift time input value t_i just gets a sign representing the Left/Right information of the TS hit and is thus contained in the range $t_i \in [-\mathtt{tMax}, \mathtt{tMax}]$. In the case of undetermined Left/Right information, the respective drift time input is set to $t_i = 0$. This corresponds to a track passing exactly through the wire position described by $(\phi^{\mathrm{rel}}, \alpha)$.

Note that after the calculation of these three input values $(\phi^{\mathrm{rel}}, \alpha, t)_i$, they are all scaled from there output range to $[-1, 1]$ before feeding the values into the neural network. Therefore, all input values have the same order of magnitude, which is expected to be beneficial for a reduction of the bit-widths in hardware.

Output Calculation

The neural networks are constructed with two output nodes for the 3D track parameters z and θ. The floating point output values are interpreted as scaled track parameter values. Since the output range of the hyperbolic tangent, which is used as activation function, is $[-1, 1]$, an output range has to be defined for each output value of the network. For the polar angle, the natural choice is $\theta \in [0, 180]^\circ$, while different options are possible for z. The setting used in the studies throughout this thesis is $z \in [-50, 50]$ cm. Recently in the hardware a neural network with a larger region of $z \in [-100, 100]$ cm was used. With larger ranges, the resolution at the IP is expected to slightly decrease [100], while structures in the background topology become visible.

6.2.5. Training of the Network

The weights in the neural network are trained with a supervised training algorithm using a training data set that represents the function to be learned. To this end, a training data set is required, which consists of many sample pairs of input and target values. It is presently obtained from a MC simulation, but using real data will also be possible once sufficient data is recorded by Belle II.

Given an input vector from a training data sample, the objective is that the neural network output becomes an optimal estimator of the target value from the training sample. To this end, a cost function on the training data and the network output is defined: it describes the difference between the training targets and the current network outputs. During the training, the weights are iteratively adjusted with the objective to minimize this cost. For the NeuroTrigger an epoch based[1] training algorithm is used: the weight changes of the neural network are only calculated once per epoch. Epoch based algorithms avoid frequent weight changes into wrong direction due to a misleading interpretation of the deviation in single events. To avoid overfitting of the neural network to the training data, cross validation is used as a countermeasure.

Backpropagation

Backpropagation training algorithms calculate the weight changes by propagating the cost backwards through the network. The method relies on the differentiability of the neural network activation function. For the NeuroTrigger, the neural networks are trained by the iRPROP^{-} algorithm [101, 102], within the neural network library FANN [103]. This is a fast and stable gradient based backpropagation algorithm that only uses the sign of the gradient, but not its absolute value.

In each epoch, all the samples in the training data set are processed by the network with the current weight set. The cost C to describe the deviation of the network outputs in the current epoch ($\vec{z}^{\mathrm{NN}}$) and the target values ($\vec{z}^{\mathrm{Target}}$) is the mean squared difference of the outputs and targets with the current weight set:

$$C = \frac{1}{N}\left(\sum_{i=1}^{N} \vec{z}_i^{\mathrm{NN}} - \vec{z}_i^{\mathrm{Target}}\right)^2 \tag{6.27}$$

where N is the number of events in the training data set. To calculate the weight updates Δw_i, the differentiability of the activation function of the neural network is used: it allows to calculate the gradient of the C with respect to each weight w_i.

In the classic backpropagation algorithm [104, 94], the weight updates Δw_i are simply given by the gradient multiplied with a learning rate α_k:

$$\Delta w_i = \alpha_k \cdot \frac{\partial C}{\partial w_i} \tag{6.28}$$

Such a learning rate is typically a decreasing function of the epoch k with the effect that the weight updates become smaller in later epochs. Starting with a random weight set, initially big changes are required to get closer to an optimum. When the weights approach a minimum of the cost function, smaller updates are required for the fine tuning.

[1] One epoch means a single run over all the events in the training data set.

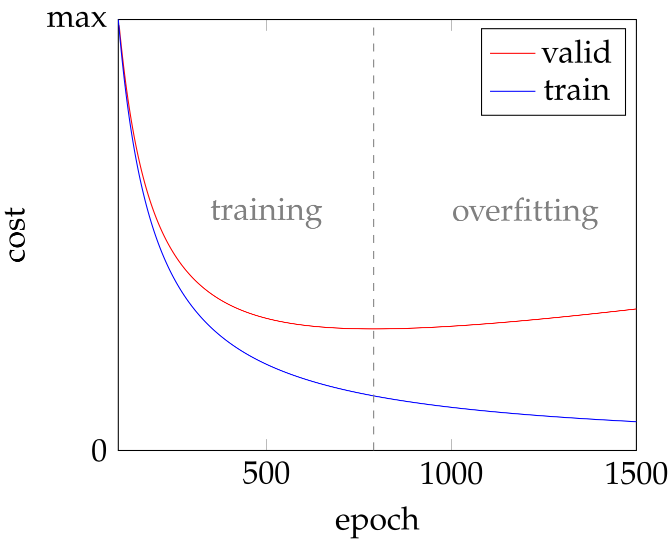


Figure 6.7.: Illustration of the cost on the training set and on the independent validation set as a function of the epoch. The optimal generalization is achieved at the minimum of the validation set. In the overfitting region, the cost on the training set still decreases, while the validation cost increases again.

In the iRPROP$^-$ algorithm, a more advanced method is used to calculate the weight updates from the gradients: it uses only the sign of the gradient, not its magnitude, but it remembers the sign of the changes of the previous epoch. A weight update Δw_j is determined as follows: if the gradient in epoch k has the same sign as the gradient in the previous epoch $k - 1$ (i.e. the change is in the same direction), the step width is increased. If the change is in the opposite direction, there was a jump over a minimum and the step width is decreased. The details of the iRPROP$^-$ algorithm can be found in [101, 102].

Cross Validation

A stopping criterion for the training is required to decide when the network is sufficiently trained. If the training is stopped too early, the estimation quality could be further improved. But if the training is stopped too late, the network will overfit to the fluctuations in the training data and the generalization is getting worse. Cross validation [94, p. 372] is a countermeasure to avoid overfitting: an independent validation data set is used during the training to measure the generalization of the network in each epoch with the current weight set. Once the optimum is reached, the cost on the validation set reaches a minimum and will increase again in the following epochs.

The validation data should be an independent data set, representing the same

parameter	default	description
repeatTrain	10	Training is repeated with different initial weights.
nTest	5000	Number of test events used to select the weight set of the best training run.
nTrain	10	Number of training events = number of weights * nTrain.
nValid	5000	Number of events used for the cross validation.
checkInterval	500	Previous epochs used for cross validation.
maxEpochs	10000	Maximum number of training epochs.

Table 6.1.: Parameter description and default values to configure the neural network trainer.

problem as the training data. This can be achieved by partitioning the data into the train data set and the validation data set. With MC data, the validation set can also be generated with the same method as the training data, but with a distinct random seed.

During the training the cost on the validation data set is used as a stopping criterion for the training. This is realized by storing the weight sets and costs on the validation data after each epoch. Once a minimum in the validation cost is recognized, the training is stopped. The weight set at the minimum of the validation cost is selected as the final trained network.

An illustration of the cost on the train data and the validation data is shown in Fig. 6.7. In the early epochs, the cost decreases in both, the training set and on the validation set. But in later epochs, the cost only continues to decrease on the train data, while it increases again on the validation data. At this point the neural network is overfitting the training data.

Trainer Parameters

In the implemented neural network trainer, several parameters can be configured to optimize the training, which are shown in Tab. 6.1. The number of events used for the cross validation is defined by the parameter nValid. With checkInterval, the number of previous epochs are defined, where a minimum in the validation error is searched after each epoch. Once a minimum is found, the training is stopped and the weight set with the minimum validation error within the last checkInterval epochs is selected. Alternatively, if the number of epochs reaches maxEpochs the training is stopped. But the typical number of epochs is 1000 to 3000 due to the cross validation and the maxEpoch $= 10000$ is almost never reached.

The parameter nTrain defines the size of the training data sample. It is expressed as a factor to be multiplied with the number of weights in the network. Hence, for the basic network with 2432 weights, nTrain corresponds to 24320 training samples. With repeatTrain, the same training is carried out several times with independent initial random weights. Since each network typically finds a local minimum, rolling the dice several times allows to improve the result. Out of a set of `repeatTrain` the optimal weight set is selected using a further independent test data set containing `nTest` samples.

6.3. Neuro Trigger Setup

The neural network trigger is composed of a preprocessing module, followed by the actual execution of the network. For each found track, the preprocessing starts by selecting the related TS hits. In addition to the calculation of the input vector for the neural networks, the preprocessing uses information on the tracks available before the execution of the network to categorize the tracks into different classes (a.k.a. sectors). For each of these classes, a trained expert neural network is installed. Those classes can be defined by a geometrical sectorization of the track parameter phase space or refer to a specialization to missing input values. The sectorization concepts are described in Sec. 6.3.1. After determining the sector of the track, the preprocessing loads the correct weight set and proceeds with the input calculation of the neural network, which is described in Sec. 6.3.2. Then the neural network selected from the pool of experts is used to estimate the 3D track parameters of the single charged tracks, found by the track finder of the L1 trigger. An overview of the implementation within the simulation framework of the Belle II detector (basf2) and its connection to the hardware implementation via the tools for Data Quality Monitoring (DQM), are addressed in Sec. 6.3.3.

6.3.1. Expert Neural Networks

All information available a priori, should be included in the preprocessing of the neural network, to let the network focus on its actual purpose. If a sectorization of the input data into classes of distinct subtasks is possible, this can be used to train expert networks. This is possible, because neural networks tend to specialize to the specific conditions provided in the training data set.

Such a divide and conquer approach is an established method in machine learning solutions. In [105] the input was presented to a pool of expert networks and a gateway network. Out of the outputs of all the experts, the best result is selected as estimate of the whole ensemble. While the generalization of a single network to all input data conditions needs to be guaranteed in the case without sectorization, the local expert networks can achieve a higher accuracy with their specific expertise. Given a preprocessing algorithm that sectorizes the input data before the execution of the neural networks, only the single appropriate expert network can be selected from the ensemble for execution. This approach reduces the computational complexity, which is desirable for the L1 trigger system.

During the development of the NeuroTrigger project, different sectorization methods were studied: a sectorization of the track phase space was considered and expert

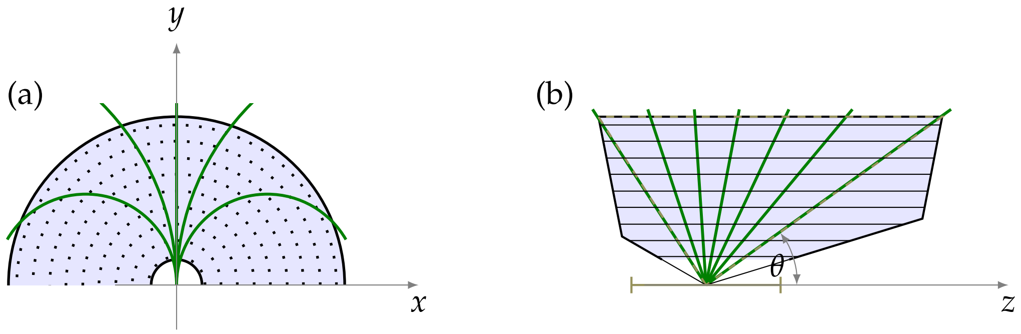


Figure 6.8.: Sectorization of the phase space for tracks coming from the IP. (a) Sectors in the signed transverse momentum $q \cdot p_\mathrm{T}$, where q is the charge and p_T is proportional to the 2D track radius. (b) Longitudinal sectors in the 3D track parameter θ.

networks were trained for missing input values. In the future, further specialization of the neural networks can be considered. For example, neural networks can be trained to recognize specific physics decay channels with a unique signature. Once the signal and background model observed in Belle II is sufficiently understood, a classification network could be trained that incorporates information for specific physics processes.

2D Sectorization

Sectorization in the 2D track parameters p_T and ϕ is implemented in the neural network simulation, but not used in the current networks. Since the 3D track reconstruction problem is linearized w.r.t. the 2D track, the track reconstruction is symmetrical in the 2D track parameters. However, the errors for low-p_T tracks might differ from the errors for high-p_T tracks such that studies with sectorization in p_T might still be considered for a fine-tuning of the neural network results.

The 2D sectorization dates back to the early development phase of the neural network [86], where each input of the neural network represented the scaled drift time at a specific priority wire in a TS. With 2336 track segments, each with Left/Right information, priority hit information and a drift time, the dimensionality of the inputs to a neural network covering the full (p_T, ϕ)-region is very large. With the intention to reduce the dimensionality of the input, where each wire was associated with one input to the network, the track parameter phase space was sectorized in the 2D track parameters p_T, ϕ [86, 93]. In each (p_T, ϕ)-sector (see Fig. 6.8 a)), only a small number of wires could actually get hit which massively reduced the number of inputs per expert network. Though this method used an extremely large number of sectors ($O(10^6)$), each with a network of the same architecture, but trained with a specific weight set. A hardware solution for this heavily sectorized network was proposed in [106]. Using an external memory for the FPGA board, a huge number of trained weight sets can be stored and selected according to the 2D track parameters. It was a proof of principle for the MLP neural networks applied to the 3D track reconstruction at the first level track trigger of Belle II.

But further studies were carried out and a different solution to use the 2D track parameters was found: The number of sectors could indeed be reduced efficiently by using the symmetry of the tracking problem in the 2D track parameters. This heavily reduces the number of required sectors while losing only a little accuracy and the external memory was no longer required. The breakthrough was the adaption of the preprocessing from the LS fit to the NeuroTrigger [96]. Instead of associating each input with a specific wire, the coordinates of the wires are presented as input. The procedure is analog to the preprocessing in the 3D LS fit (see Sec. 6.1): there at first the 2D tracks are reconstructed in the transverse plane using the axial hits. After this step, the positions of the stereo hits, which include the 3D track information, are arranged in a coordinate system relative to the 2D tracks. The 3D track parameters are reconstructed by fitting a straight line to the transformed stereo hit positions. Here, the fit of a straight line to the transformed stereo information, is now replaced by the neural network. Beneficially, the neural networks can learn to handle nonlinearities and disruptions in noisy input data, while the LS fit is optimal only under ideal conditions.

3D θ Sectorization

While the problem is symmetrical in the 2D track parameters, a sectorization in the 3D track parameters might still be favorable: In θ, the track finding problem is not symmetric and the phase space could be sectorized as shown in Fig. 6.8 b). The problem is that 3D information is not available after the 2D track finder. Presently, the 3D track parameters (θ, z) are only available as output values of the 3D trackers, i.e. after the neural network. But if the advanced 3D track finder will be realized that estimates θ already at the track finding level (see Chap. 5), improvements via a θ sectorization can be considered. Besides the axial TS used by the 2D finder, the 3D finder uses the stereo TS as well for an early 3D track parameter estimation. Simulation studies with θ-sectorization in combination with the 3D finder as input are presented in Sec. 6.5.3. Alternatively, the 3D tracking could be iterated [106], if the speedup achieved with newer hardware allows to meet the latency requirements.

Missing Inputs

If possible, the preprocessing selects one TS per SL per track as input to the neural network. But sometimes TS information is missing, either because there is no TS hit in a SL, or because the cuts used in the TS selection are too tight. These cases with missing TS hits are covered by specialized expert networks. In the present setup of the neural network trigger, expert networks are only trained for single missing stereo TS hits. This corresponds to five experts in total, one for the case with all four stereo TS present, plus four networks, each with a single stereo layer missing.

parameter	default	description
`SLpatternMask`	[010101010]	SL TS hit expert pattern mask.
`SLpattern`	[111111111] [101111111] [111011111] [111110111] [111111101]	SL TS hit expert patterns.
`invptRange`	$[[-5.0, 5.0]]$	charge/p_{T} sectors [GeV^{-1}].
`phiRange`	$[[0.0, 360.0]]$	ϕ sectors [°].
`thetaRange`	$[[0.0, 180.0]]$	θ sectors [°].

Table 6.2.: Default sectorization parameters of the neural network trainer.

In combination with the 3D finder, the efficiency for low-p_{T} tracks is significantly improved. Experiments with additional low-p_{T} expert neural networks, where two of the outer stereo TS may be missing, are presented in Sec. 6.5.2.

The configuration parameters for the missing input are the binary bit masks called `SLpatternMask` and `SLpattern` in Tab. 6.2. Each list of nine boolean values in the `SLpattern` defines a single expert neural network. Within each list, each of the nine entries corresponds to a SL of the CDC (to be read from right to left). In the `SLpattern` 1 means the SL is required to have a hit and 0 means the SL is required to have no hit. Hence, the five `SLpatterns` describe the five possible cases with maximally a single missing stereo hit.

The `SLpatternMask` defines whether the value of the `SLpattern` at the respective SL position is used. A 1 in `SLpatternMask` means the `SLpattern` is used, while a 0 means the `SLpattern` is not used. In the default configuration, shown in Tab. 6.2, this is used to skip the patterns for the axial SL. Therefore, if axial TSs are missing default values are used as input and the expert networks are only selected by the stereo hit pattern. In principle, this allows to carry out studies with missing axial TSs as well.

A sectorization in θ can be configured with the parameter `thetaRange` shown in Tab. 6.2. It is a list of the ranges for each θ sector. In the same way, also the 2D sectorization in p_{T}^{-1} and ϕ can be configured with the parameters `phiRange` and `invptRange`. The total number of experts is the product of the number of experts of each type.

6.3.2. Preprocessing

The preprocessing is important to reduce the dimensionality of the input and to extract the essential features required by the network. It calculates the input vector for the neural network and selects the appropriate expert weight for execution. The known event information prior to the preprocessing, is the set of all active TS in the event, the found 2D track parameters (p_{T}, ϕ_0) and the axial TS related to the found

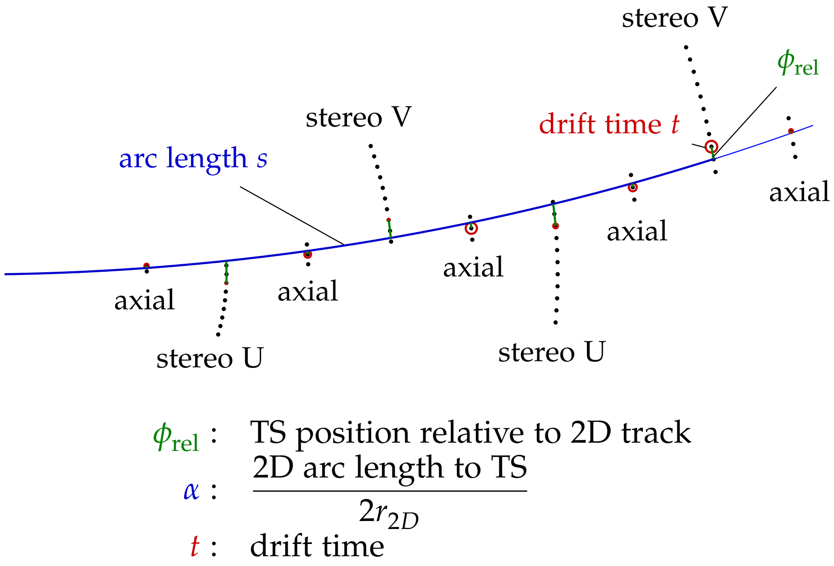


Figure 6.9.: Calculation of the input values used by the neural network. Each hit is represented by the three values ($\alpha \propto s, \phi_{\text{rel}}, t$). These are 2D coordinates of the TS relative to a given 2D track ($\alpha, \phi_{\text{rel}}$) and the drift time t. Only TSs in a SL specific small ϕ_{rel} region close to the 2D track are considered. These regions, obtained from a training data set, are asymmetric due to the stereo angle in the stereo SL.

2D tracks. If available, an additional event time estimate t_0 from the ETF can be used in the calculation.

The three main steps carried out in the preprocessing of the neural network are: the TS hit selection, the input vector calculation and weight selection. In the hit selection, axial and stereo hits are related to the 2D track. The goal is to select one TS hit per SL per track, which defaults to 9 TS as input for a single neural network. But if input TS are missing, dedicated expert networks with specialized trained weight sets are selected. The calculated input vector for the neural network is a representation of the related TS hits in combination with the p_{T}, ϕ-information from the 2D finder.

A preliminary version of this preprocessing method using (ϕ_{rel}, s, t) as TS representation was published in [96]. The presently used refined setup, where the 2D arc length s is replaced by the angle α, was published in [107]. In the following, the preprocessing concept and the involved configuration parameters are summarized. A thorough documentation of the preprocessing steps can be found in [67].

parameter	default	description
`nTrainPrepare`	5000	number of tracks used to fill the ID-range histograms
`relevantCut`	0.02	cut to determine the ID-ranges
`tMax`	256	maximum drift time (in trigger time bins $\approx 2\,\text{ns}$)

Table 6.3.: Parameters configuring the id-map ranges for the hit selection of the neural network.

ID-ranges for the TS Hit Selection

For each track found by the 2D finder, the (p_T, ϕ_0)-estimates are used to select the related axial and stereo TS hits. Therefore, a small range of TS around the estimated 2D track is inspected to find the hit candidates. These TS ID-ranges are determined in a pre-training step using the configuration parameters shown in Tab. 6.3. Using a sample of `nTrainPrepare` 2D finder tracks, for each SL a histogram is filled with the ϕ-distances between the 2D track and the related TS. An example of such an ID-range histogram is shown in Fig. 6.14.

The ϕ-ranges are obtained by applying a cut on this histogram: at least a number of `nTrainPrepare` × `relevantCut` TS are required for each ϕ-distance to be contained within the ID-range. In the preparation of the ID-ranges, the histogram is searched from the peak position in both directions until the first distance falls below that cut. With the default value of `relevantCut` $= 0.02$, this means that $\approx 96\%$ of the TS are contained within the obtained ID-ranges. An illustration of the ID-ranges in each SL is shown in Fig. 6.15. Due to the alternating orientation of the stereo angle in the different stereo SL, their allowed ϕ differences extend to different sides of the 2D track, as illustrated in Fig. 6.9.

In principle, each expert network has its own ID-ranges. Although with the missing input experts, the ID-ranges are almost identical (except for the single missing SL). Therefore, in the basic setup only the ID-ranges from the experts with stereo TS hits in all SLs are used for the hit selection. Larger differences can be observed with θ-sectorization as will be discussed in Sec. 6.5.3.

Event Time Calculation

Without input from the ETF, the event time is determined as the shortest drift time among the selected TS hits of a track. After the geometrical sector of the expert network is selected (i.e. θ-sectorization based on the input from the track finder), the ID-ranges of the expert network where all stereo SL have a hit is used to select the related axial and stereo hits. Among all TS hits contained within the ID-range, the shortest drift time is used as t_0 and its value is subtracted from the other drift time values.

Hit Pattern Calculation

With the known event time, at most one TS per SL is selected. Like for the event time calculation, after selecting the geometrical sector, the ID-ranges of the network with a hit in all stereo SLs is used here. All TS candidates for a track are required to lie within these ID-ranges. Only TS with drift times larger than t_0 and smaller than the configuration parameter `tMax` are selected (see Tab. 6.3). With the known hit pattern, the expert neural network can be selected.

TS Hit Selection

In general, a track can cause several hits within a single SL such that an additional hit selection is required. Since the expert network has already been selected based on the hit pattern, the exact ID-ranges of this expert network are used. Like before in the pattern calculation, each TS hit is required to be contained within the ID-ranges and additionally have a drift time $t = t_{\text{TDC}} - t_0$ with $0 < t < \texttt{tMax}$. If several TS fulfill the same condition within the same SL, TS with known Left/Right information are preferred. Furthermore, the TS with the shortest drift time is selected. Since a first priority hit is more likely to have the correct Left/Right information, in future studies a preference of first priority hits over second priority TS hits can be considered. After the related TS are selected, the input vector for the neural network is calculated.

6.3.3. Implementation

The simulation of the neural network trigger and its preprocessing is implemented within basf2, the simulation framework of the Belle II detector. It is used to train the neural network weights to the L1 trigger task with a special trainer and to evaluate its performance on test data sets. The integration in basf2 allows to simulate realistic MC physics cases for the performance studies. Since basf2 is used for the physics analysis in Belle II, it can also be used to study the network response on real data from the detector. To be used in the real experiment, the trained and tested weight sets are installed in the Belle II trigger hardware on dedicated FPGA trigger boards. The hardware results can be compared to the software results to implement measures for Data Quality Monitoring (DQM).

Training with Real Data

In the early development phase, the training of the basic neural network is carried out only on simulated MC data. But in later stages, when sufficiently background data is recorded from the Belle II detector, an alternative training with real data as training targets is anticipated.

Using MC data for the neural network training requires good trust in the MC data - the neural network can only learn what is represented in the training data. If the

parameter	default	description
wMax	63.0	After each training epoch, weights are limited to [-wMax, wMax].
fixedPoint	False	Use fixed point arithmetic in net execution (for FPGA simulation).
precision	[12, 8, 8, 12, 10, 10]	fixed point precision in bit after radix point (for track phi, scaling factor, reference id, MLP nodes, MLP weights, MLP activation function)

Table 6.4.: Hardware related configuration parameters of the neural network simulation.

training data does not generalize, the neural network will fail in generalization as well. In particular if the MC data contains some additional falsely modeled features that are not present in the real detector data, the neural network possibly learns these features instead of the function it is supposed to learn. Vice versa, the performance of the neural network may improve significantly by learning characteristic hidden features in the data which are missing in the MC model. Detecting additional "fake" features introduced by the MC is difficult by looking at the MC data alone. Therefore, the MC data needs to be carefully compared to the real detector data to detect differences and install countermeasures. A neural network that was trained on MC data can be tested with real data to measure its generalization capabilities.

Neural Network Hardware

The neural network trigger is a newly integrated component in the CDC subtrigger system of the Belle II experiment. The hardware platform for the neural network trigger are four FPGA boards. These run in parallel with the 3D fitter boards and receive the same information as input via high speed optical links (GTH, GTX). Details on the FPGA implementation of the basic neural network setup are presented in [108]. It shows the Universal Trigger board 3 (UT3) setup of the 3D neural network trigger, as used in the cosmic runs and in Phase 2. Many Belle II trigger components are implemented on the UT3, which is a specially designed FPGA board equipped with a Virtex 6 FPGA.

With future upgrades of the algorithm, the requirements on computational resources could exceed the UT3 such that more powerful hardware is required. Preliminary studies have shown that a complex setup using many different neural network weight sets, and therefore high memory requirements, can be realized with the Virtex 7 [106] FPGA. Presently the upcoming candidate for more complex trigger algorithms is the UT4 board, equipped with an UltraScale FPGA. This new custom designed FPGA board will serve as the future upgrade of the UT3 board for all the trigger components at Belle II.

In order to transfer the trained neural networks to the hardware and to compare the hardware results with the software simulation, the additional parameters listed in Tab. 6.4 are used. The parameter `wMax` is used during the training in order to put a limit on the weight range. This is important since each weight in the hardware is only represented as a fixed-point integer with a limited number of bits. In order to obtain the same results in the simulation as in the hardware, the neural network simulation can be switched to `fixedPoint` arithmetic. Then the simulation is bit-precise, using the bit-widths configured with the parameter `precision`. This is especially important for the comparison of the simulation results with the hardware results.

Data Quality Monitoring

The neural network hardware runs in the clocked L1 trigger pipeline, where new input data is processed every 32 ns. Output values are recorded via Belle II Link (B2Link) a dedicated protocol used in the Data AcQuisition (DAQ) of Belle II. Upon a trigger signal, when an event is written out from the detector, presently 48 consecutive clock cycles of the trigger pipeline outputs are stored for offline analysis as raw trigger information.

Via its B2Link format, the neural network hardware stores information about its input values, the intermediate results of the preprocessing and the calculated 3D estimates. In basf2, an unpacker module has been implemented to read this raw trigger data and to create the corresponding basf2 data objects. This allows to test the validity of the hardware preprocessing and the network results by a comparison with bit-precise values from the software simulation. Additionally, a Data Quality Monitoring (DQM) module has been implemented in basf2. It allows to perform consistency tests of the data transferred between the boards of the trigger pipeline. For example, the output of the track finder can be compared to the input of the neural network. Furthermore, the DQM module is used for online monitoring of the hardware. For this purpose, the DQM module will be activated within the online High Level Trigger (see Sec. 3.3).

6.4. Performance of the Neural Network Trigger

The goal in this section is to study the settings and conditions of the basic neural network trigger which is presently implemented in the running Belle II experiment. To this end, important neural network configuration options of a network architecture that fits into the existing hardware are tested on simulated test data with different properties. Extended network architectures in combination with the 3D finder, which may require new hardware implementations, will be presented in the subsequent Sec. 6.5. Studies with simulated physics signal events as well as studies with real data from Belle II are subject of the next Chap. 7. Depending on the specific properties of the test data sets, different neural network settings are expected to be optimal. With the knowledge of how the neural network settings affect the performance on specific test data types, an elaborated choice of the settings for the hardware implementation can be made in conclusion.

Since it is not feasible to present all tested single network types on the single data sets separately, only an excerpt of representative studies is presented. These studies are selected using an integrated analysis method, where a set of features on the network results is defined such that many test runs can be easily compared. Thus, the optimal neural networks for a specific data set can be selected. These features can for example be the network with the best z-resolution or the network with the highest efficiency ϵ. Using this strategy, a selection of relevant examples, out of the $\approx 2 \cdot 10^5$ neural network test runs, is made and presented in the following sections.

6.4.1. Tested Event Types

The test data sets can roughly be distinguished in three different groups: simulated single track events, simulated physics events and recorded events from early Belle II runs. With the single track events, different track parameter ranges and different track parameter distributions are used. In particular, two different classes defining the range in p_T and θ are used: tracks contained within the full acceptance region of the CDC and tracks with missing TS hits at the boundary region of the CDC accptance (compare definitions of the 3D finder test data in Sec. 5.3.1). For the polar angle θ a uniform generation in θ as well as uniform generation in $\cos(\theta)$ is simulated for testing. Additionally, for the z-vertex position, data sets with a uniform distribution with $z \in [-50, 50]\,\mathrm{cm}$ as well as tracks from the IP with $z \in [-1, 1]\,\mathrm{cm}$ are used. The used labels for these main data ranges are listed in Tab. 6.6. Besides the main data sets with single muon ($\mu^\pm$) tracks, a data set containing single pion ($\pi^\pm$) tracks is generated in addition. All single track data sets are simulated with the four different background types presented in Chap. 4.

As candidate for physics signal events, initial state radiation (ISR) events are simulated (See Sec. 7.1). These are events where one ore more photons are radiated from the initial e^+ or e^- particles before the collision. In this way the center of mass energy is effectively reduced and hadronic cross sections can be measured in this

Feature	Description
`NNpar`	Label of the training parameters
`Train`	Particle type in the training data set ($\mu^{\pm}$ or $\pi^{\pm}$)
`TS-LUT`	TSF lookup table used in training and testing
`Finder`	The used track finder (2D or 3D)
`Target`	Target tracks used in the training (MC or Reco)
$\texttt{Bkg}_{\text{train}}$	Background type used in the training
$\texttt{Bkg}_{\text{test}}$	Background type in the test data set

Table 6.5.: Selected features of the training data, the test data, the neural network and of the trigger simulation.

reduced center of mass system. Since these events typically have a low track multiplicity, they are difficult to trigger. Here the neural network trigger can provide a substantial efficiency improvement: by recognizing the characteristic ISR signature based on the 3D track estimates at the L1 trigger level, the other trigger logics can be relaxed. The basic tests in this section are carried out with the muon data sets, while the pion data sets will be used in connection with the ISR events in Sec. 7.1.

The different settings of the training and test data sets and of the tested networks are listed in Tab. 6.5. Throughout the following studies, the labels introduced in Tab. 6.5 will be used in the result summary tables. The parameter `NNpar` is a label for a setting of neural network training options. In the selected networks, `NNpar` describes especially the different sizes of the training data set, which are listed in Tab. D.1 in the Appendix D.1. The label `Train` describes the particle types used in the training data in Sec. 7.1 ($\mu^{\pm}$ or $\pi^{\pm}$). Out of the CDC trigger simulation options, `TS-LUT` identifies different options of the Left/Right lookup table of the TSF (studied in Sec. D.1.4) and `Finder` is the choice of track finder used (studied in Sec. 6.5). In the studies throughout this chapter two track finders are used: the 2D finder in the normal setup, as it is presently installed in hardware in Belle II, and the 3D finder which was presented in this thesis in Chapter 5. The simulated background in the test data sets is labelled $\texttt{Bkg}_{\text{test}}$. The `Target` label provides information on whether reconstructed or MC track parameter values were used as target tracks during the neural network training. Similar to the description of the background in the test data, $\texttt{Bkg}_{\text{train}}$ identifies the type of background used in the training data. Extended studies of the neural network training with different values for the parameters listed in Tab. 6.5 can be found in the Appendix D.1.

6.4.2. Single Track Simulation

Simulated single track data sets with the track parameter ranges listed in Tab. 6.6 are used for the training and the basic tests of the neural networks. The training data sets for the neural network studies contain single $\mu^{\pm}$ or $\pi^{\pm}$ track events with

Data	p_T [GeV]	ϕ [°]	θ [°]	z [cm]	d [cm]
NIP	$[0.35, 10.2]$	$[-180, 180]$	$[35, 123]$	0	0
N50	$[0.35, 10.2]$	$[-180, 180]$	$[35, 123]$	$[-50, 50]$	0
ZIP	$[0.2, 10.2]$	$[-180, 180]$	$[19, 140]$	0	0
Z50	$[0.2, 10.2]$	$[-180, 180]$	$[19, 140]$	$[-50, 50]$	0

Table 6.6.: Track parameter ranges of the basic training & testing datasets for the neural network trigger. Similar to the ranges used for the 3D finder in Chapter 5, the data sets NIP contains IP tracks in the full CDC acceptance region and ZIP contains tracks at the boundary region of the CDC that do not create TS hits in all SLs. In order to cover a uniform distribution of track vertices along the z-axis as well, the data sets N50 and Z50 are introduced.

vertices distributed uniformly along the z-axis ($z \in [-50, 50]$ cm) and without a transverse displacement (i.e. $d = 0$ cm). In the test data sets with single track events, data sets containing only tracks from the IP are simulated in addition. For the transverse momentum p_T and the polar angle θ, the same data ranges are used as in the previous Chapter 5 on the 3D finder in Sec. 5.3.1: in a range NIP, tracks from the IP ($z = 0$ cm) are fully contained in the CDC volume such that TSs are usually found in all 9 SLs. A range ZIP contains tracks with lower p_T and shallower polar angle θ, where typically several TSs inthe outel SLs are missing (see Fig. 5.21). The maximum value of $p_\mathrm{T} = 10.2$ GeV is selected because it is the maximum value of p_T that can be estimated by the existing 2D finder. It is a result of the 2D finder Hough plane binning in the straight track limit, which is chosen larger than the expected high-p_T peak at the beam energy of the HER (7 GeV). The ranges of the five helix track parameters are listed in Tab. 6.6. Accordingly, the data sets with displaced tracks are labelled N50 and Z50. For all single track data sets, the track parameters are distributed uniformly in the inverse transverse momentum (p_T^{-1}), the azimuthal angle (ϕ_0) and the z-vertex position (z_0).

In the polar angle θ two distributions are simulated: uniform in θ and uniform in $\cos(\theta)$. In the training data sets, the polar angle θ was generated with a uniform distribution. This has the advantage that all possible θ values are represented in the training data. Note that the output value of the neural network is θ rather than $\cos(\theta)$. Therefore, a uniform distribution in the output value of the neural network helps to avoid a false feature learning by the neural network: if certain θ output values are a priori more probable than others, the neural network might learn this. Hence, the default training data set has the range Z50 with a uniform distribution in θ. The data sets with the uniform distribution in $\cos(\theta)$ correspond to a spherical distribution of the tracks in the detector and are thus used for the testing.

The data simulation is carried out as a two step process: at first, for each data range and distribution type, four data sets are simulated with different background types and stored in separate data files. At this level, the track reconstruction is carried out in addition. Only the data objects relevant for the following trigger simulation (especially the CDC hits) and the analysis of the neural network results (MC and reconstructed track parameters) are stored to reduce the size of the data sets. In order to accelerate the processing, for each file type several runs with different random seeds are carried out and merged only upon the training and analysis runs.

In the second simulation step, all the trigger components relevant as input for the neural network are simulated. Besides the simulated trigger components already known from the 3D finder studies several additional simulation steps and configuration options are required here to provide the input for the 3D track reconstruction with the neural network. In general, the input modules to be simulated are the TSF, the track finder and the ETF (schematically shown in Fig. 6.1). In this second simulation run, for each component alternative options are simulated and stored as alternative inputs for the later neural network training or testing runs.

Regarding the simulated alternative options in the trigger components, for the track segment lookup table (`TS-LUT`), the LUT of the TSF which is used to determine the Left/Right information in the track segment hits, two different options simulated: `LUTbkg` and `LUTv22`. The `TS-LUT` = `LUTv22` is presently loaded in the hardware, while `LUTbkg` is a background optimized `TS-LUT` which is presently the default in the software simulation. Since the Left/Right information is not used by the track-finders, the choice of `TS-LUT` was not important in the previous studies on the 3D finder in Chapter 5. For the neural network this information is used in the input calculation as sign of the drift time inputs and, thus, the choice becomes important here. The default setting selected in the presentation of the subsequent studies is the `TS-LUT` `LUTbkg`. Studies comparing both LUTs are presented in the Appendix D.1.4.

A simulation of the event time finder (ETF) has been carried out with four different threshold options $(0, 1, 2, 3)$ and with the true event time. However, to date the ETF is not yet available in the hardware implementation. Furthermore, the resolution of the ETF t_0 appears to be strongly affected by background in the simulation [100]. Hence, the studies presented here avoid the use of the ETF input at all. Instead, the `et_option` = fastestpriority is used, where a track based t_0 is calculated in the preprocessing of the neural network as shortest priority time of the selected TSs (see Sec. 6.3.2). The default neural network settings are summarized in Tab. 6.7.

For the track finder, two alternative options are simulated: the official 2D finder and the newly developed 3D finder which was presented in the previous Chapter 5. The parameter settings of the selected 3D finder can be found in Chapter 5 in Table 5.6. Further details are discussed in connection with the related studies in Sec. 6.5.

Polar Track Angle in the Training Data

The neural network is trained to estimate the two 3D track parameters θ and z, where the floating point output values at the two output nodes are directly inter-

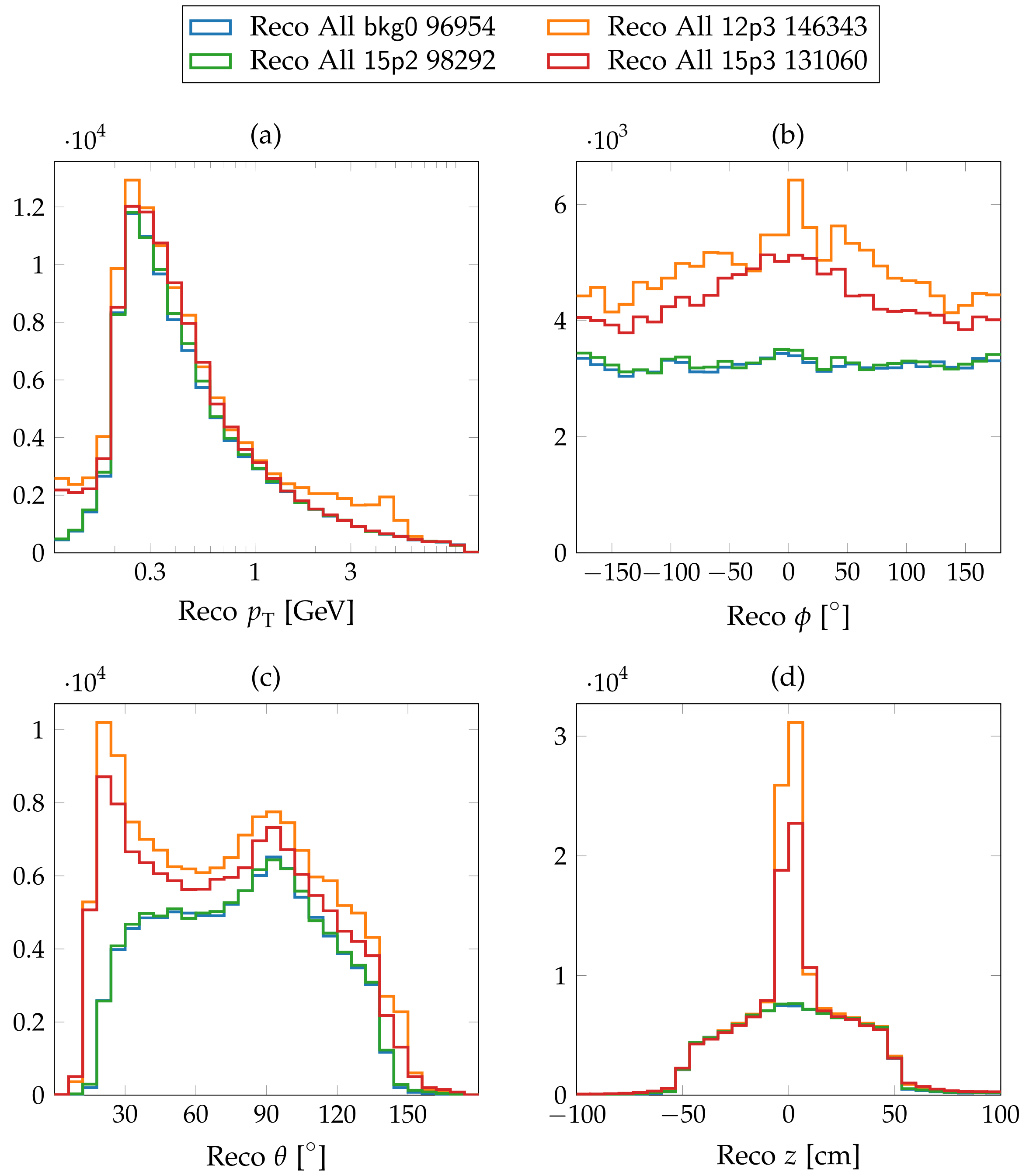


Figure 6.10.: Histograms of all reconstructed track parameters in the data sets used for the neural network training. The polar angle θ is generated with a uniform distribution and four different background types are added in the simulation. The track parameter range is listed as Z50 in Tab. 6.6.

preted as (linearly) scaled track parameters. In order to have an equal amount of all possible outcome values in the training data set, the polar angle θ is generated with a flat distribution here. All training data sets are simulated four times with the data range Z50 (see Tab. 6.6), with four different background options. The background types, in increasing order of background level, are: bkg0, 15p2, 12p3, 15p3 (for details of the simulated background types see Chap. 4). The backgrounds 12p3 and 15p3 are both simulated Belle II Phase 3 background types, i.e. the full luminosity of $\mathcal{L} = 8 \times 10^{35}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$ is assumed. They are generated in different background simulation campaigns by the Belle II background group and were used in the official large scale MC production of physics signal events on the grid. In the newer background campaign 15, besides the Phase 3 background 15p3 additionally a Phase 2 background sample 15p2 was simulated, where a lower luminosity of $\mathcal{L} = 2 \times 10^{34}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$ was assumed. Because the luminosity in the Belle II experiment is presently still lower than the design luminosity, the neural network presently loaded into the hardware was trained with the Phase 2 background 15p2. Therefore, the background 15p2 is included in the following studies. Note that the single background type used in Sec. 5.3.1 corresponds to the full Phase 3 luminosity background sample 15p3 with the highest background level used in this section.

The following histograms show the track parameter distributions of 10^5 simulated single $\mu^{\pm}$ track events from the training data sets. Figure 6.10 shows all reconstructed tracks in the training data. A typical effect from the different background types can be observed: with higher background loads more tracks are reconstructed. These tracks are additional tracks contained only in the mixed background samples (see Fig. 4.6) and additionally fake tracks created by a pileup of signal and background hits. A comparison of the distributions with and without background indicates the positions of the background tracks in the track parameter phase space. In agreement with the observations in the previous chapter on the 3D finder (see Sec. 5.3.1, there is typically a ϕ-dependence, with an increased number of background tracks pointing outwards in the accelerator plane ($\phi \approx 0^\circ$).

As an effect of the different background models for Belle II Phase 3, differences can be observed in the distributions with 12p3 and 15p3: the increase with 12p3 in the high-p_T region indicates that this background sample contains more high momentum tracks (up to $\approx 4\,\mathrm{GeV}$). Both Phase 3 background types result in additionally reconstructed low-p_T tracks; they show a strong peak at the IP ($z \approx 0\,\mathrm{cm}$) and they have a peak in the forward direction for small values of the polar angle ($\theta \approx 20^\circ$). This background peak for small values of θ already appeared before if Fig. 5.23, in the IP constrained data sets used for the 3D finder studies. A peak in the polar angle around $\theta \approx 90^\circ$ also appears without background and indicates problems of the offline reconstruction with low-p_T curl-back tracks. This reconstruction error can be explained by low-p_T tracks from the IP, which curl inside the CDC and thus cannot be reconstructed accurately (compare Sec. 5.3.4).

After the reconstructed tracks are matched to primary MC particles, no additional background tracks can be seen in the distributions. Only the efficiency can be affected, which can be seen in Fig. 6.11. It shows the track parameter distributions of

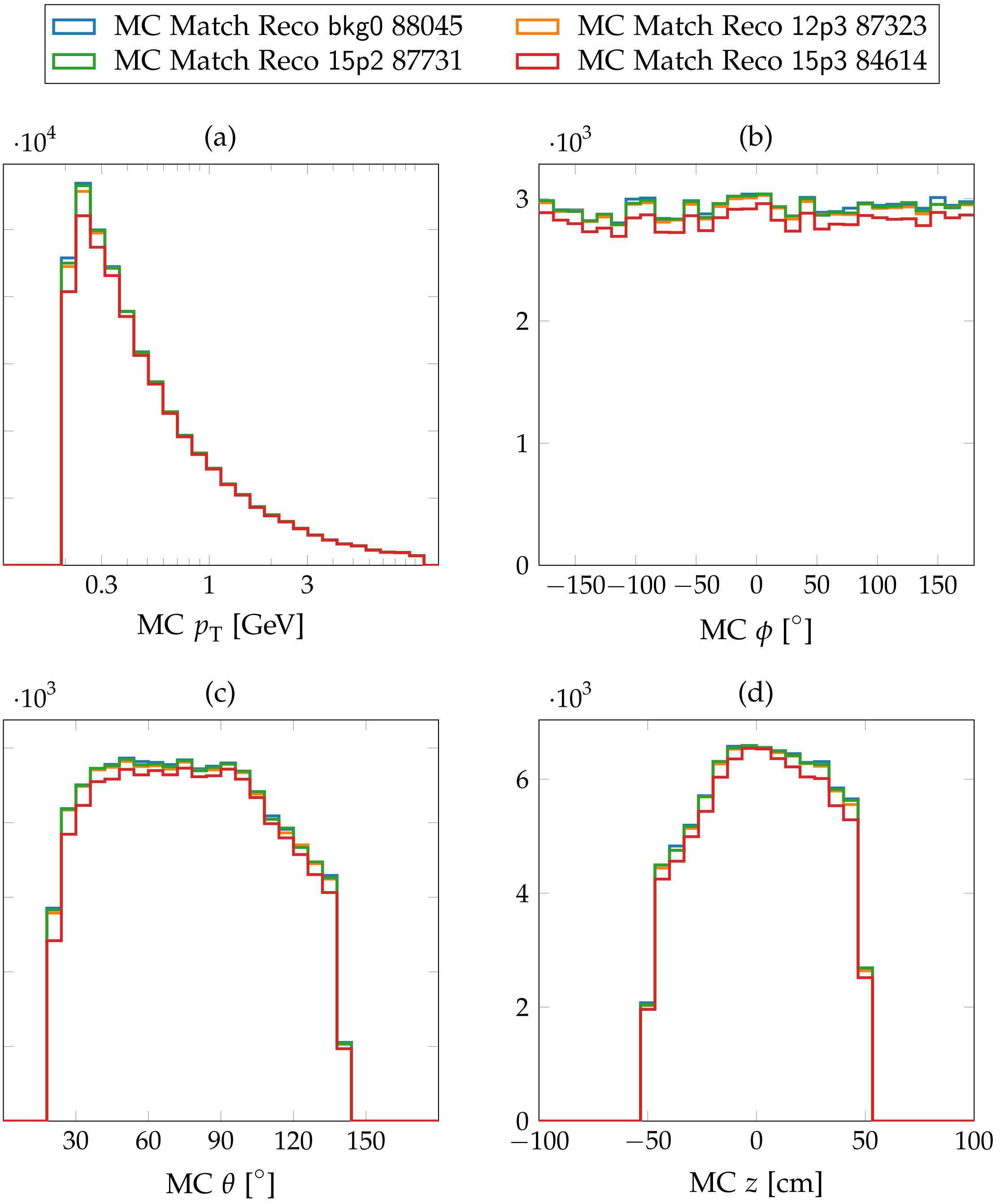


Figure 6.11.: Track parameter histograms of the primary MC particle with a match to a reconstructed track in the data sets used for the neural network training. After the matching, the differences between the different background types almost vanish. The track parameter range is listed as Z50 in Tab. 6.6.

those primary MC tracks with a match to a reconstructed track. As expected, the background with the largest occupancy (15p3) results in the lowest reconstruction efficiency. With the older Phase 3 background model 12p3, the efficiency is close to the efficiency without background. The reason for the efficiency loss of the offline reconstruction with the newer Phase 3 background 15p3 are the increased background hit occupancies mixing with the signal hits (see Fig. 4.6). In Fig. 6.11 (c) and (d), the geometrical structure of the CDC becomes visible as deviations from the simulated uniform distributions in z and θ.

Figure 6.12 shows the reconstructed track parameter values of the same matching track pairs (i.e. match of a primary MC particle with a reconstructed track). It shows the general tendency of the reconstruction: a small peak in the polar angle at $\theta \approx 90^\circ$ and a small increase of low-p_T tracks. These systematic structures are not introduced by the used background, as the same structure appears even without background. Overall the reconstructed values are quite close to their MC partner values such that both values can be used as a useful training target. The z- and θ-distributions of the matched track pairs are mostly flat, with small deviations as a direct effect of the CDC geometry. As a result of the track parameter distribution, which contains some combinations of z and θ where the detector does not receive a hit at all, the reconstruction efficiency for the data range Z50 is in the range $[84\,\%, 88\,\%]$. The geometrical limitation in the $z\theta$-region becomes most visible in Fig. 6.11 (c) and Fig. 6.11 (d). While z and θ were generated flat, the matching distributions are not flat anymore. Due to the asymmetric polar angle coverage of Belle II (designed for $\theta \in [17, 150]^\circ$), in the θ-distribution especially tracks in the backward region (large values of θ) are depleted. Similarly, in the z-distribution, more tracks are missing on the backward side (negative z) than on the forward side.

Polar Track Angle in the Test Data

In the test data sets, the polar angle is generated with a uniform distribution in $\cos(\theta)$. Using this polar angle distribution, the main two simulated single track data sets used in the following studies either contain tracks from the IP only (ZIP), or tracks with a uniform distribution along the z-axis in the range $z \in [-50, 50]\,\mathrm{cm}$ (Z50). The different ranges of the test data sets and their labels are summarized in Tab. 6.6. The distribution of all reconstructed tracks in the test data sets with the range Z50 is shown in Fig. 6.13. It shows the same structural dependence on the background types as seen in the training data with a uniform distribution in θ (see Fig. 6.10). For brevity, the other distributions are not repeated here for the test data set.

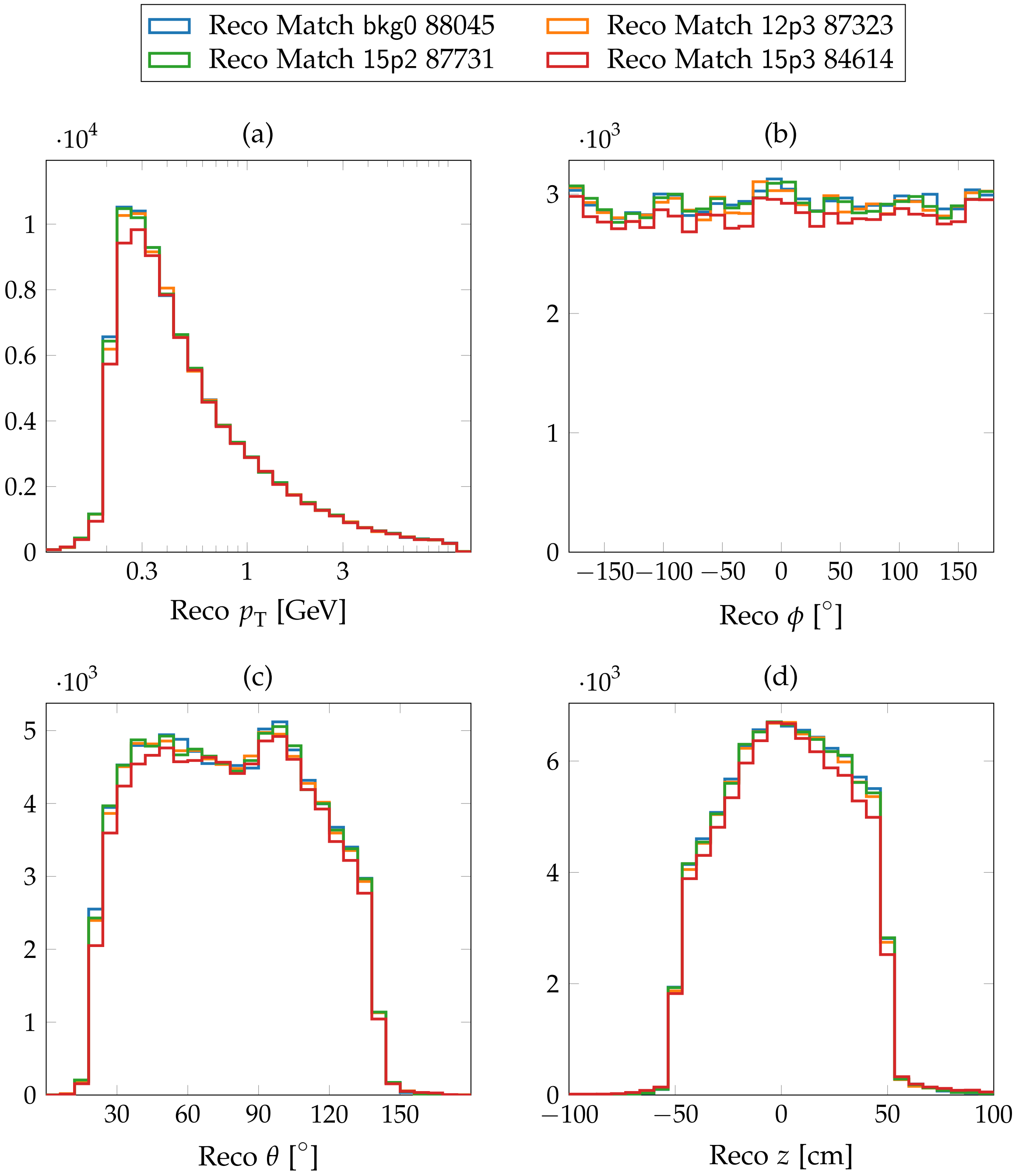


Figure 6.12.: Track parameter histograms of the reconstructed tracks with a match to a primary MC particle in the data sets used for the neural network training. The similarity to the matching MC track parameters in Fig. 6.11 illustrates the quality of the offline reconstruction. The track parameter range is listed as Z50 in Tab. 6.6.

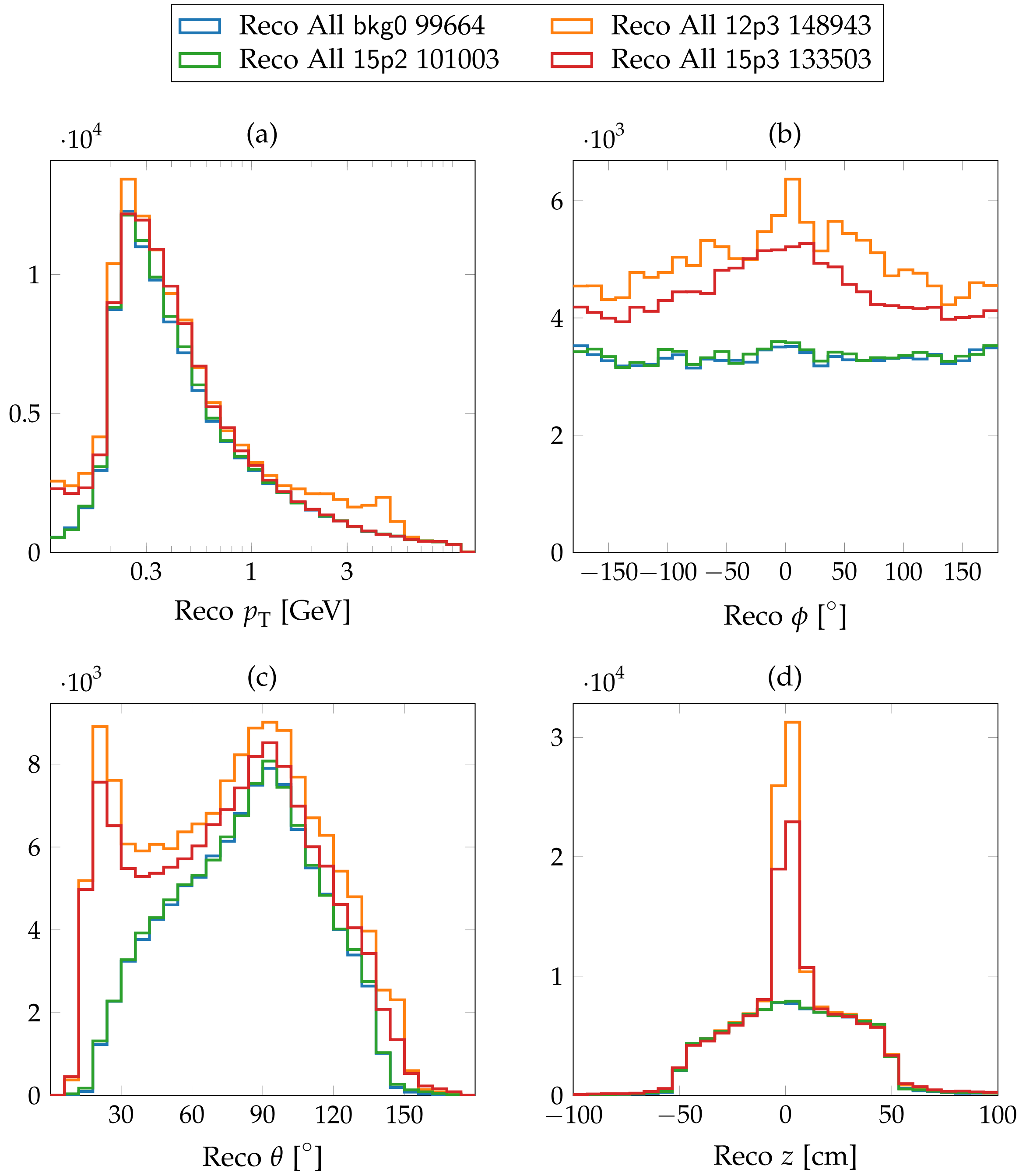


Figure 6.13.: Histograms of all reconstructed track parameters in the data sets Z50 used for the neural network testing (see Tab. 6.6). The polar angle θ is generated with a uniform distribution in $\cos(\theta)$ and four different background types are added in the simulation.

6.4.3. Basic Network Training

parameter	value
`et_option`	fastestpriority
`nTrainPrepare`	5000
`relevantCut`	0.02
`nMLP`	5 experts
`outputScale`	$z \in [-50, 50]\,\mathrm{cm}, \theta \in [0, 180]^\circ$
`nTrain`	1000×
`nValid`	50000
`nTest`	50000
`repeatTrain`	10

Table 6.7.: Overview of the training parameters for a typical network configuration with a large training data set.

The first performance measurements are carried out with a basic network configuration with an architecture that can be loaded into the present hardware. The values of the parameters used to train this network are summarized in Tab. 6.7 and are shortly discussed here. A detailed description of the parameters and of the training algorithm can be found in the previous sections on the neural network model Sec. 6.2 and on the neural network setup Sec. 6.3.

A rather big training data set is used for this introductory study: the size of the training sample is chosen as 1000× number of weights in the network, the number of training samples is about $2.5 \cdot 10^6$ events. In the Appendix D.1 in Tab. D.1, this large training sample is labelled NNE. Note that with 27 input nodes, 81 hidden nodes and 2 output nodes, the number of weights in a network is $(27+1) \cdot 81 + (81+1) \cdot 2 = 2432$. This network receives input from the 2D finder and from the TSF only. With the `et_option` = fastestpriority, the event time t_0 is determined per track as the shortest priority time among the selected hits (see Sec. 6.3.2). In this sense, the used t_0 is a track t_0 but not an event t_0. In this way an additional input from the ETF is not required. This setup uses `nMLP` = 5 expert networks, specialized to single missing stereo TSs in the input. The exact patterns of missing stereo TSs were previously discussed in Sec. 6.3.1 and shown in Tab. 6.2.

The parameter `outputScale` defines the range in the 3D track parameter s z and θ that are supposed to be estimated by the neural network. In this basic setup, the neural network can estimate the z-vertices of tracks within the limited range of $z \in [-50, 50]\,\mathrm{cm}$, while the full polar angle θ is covered. Since the Belle II detector has a polar angle coverage of only $\theta \in [17^\circ, 150^\circ]$, there are no tracks with $\theta < 17^\circ$ or with $\theta > 150^\circ$. However, by using an enlarged `outputScale`, the values represented in the data are shifted towards the linear region of the activation function of the neural network (see Sec. 6.2). Since the output float value is simply interpreted as scaled floating point value, improved resolutions are expected from the use of enlarged ranges in the `outputScale`. Furthermore, in preliminary studies a full polar angle coverage in the `outputScale` turned out to provide significantly improved resolutions compared to a limited setup. Since no improvements are expected from a small reduction of the output scale in θ, all the neural networks presented in the subsequent studies are trained for a full polar angle coverage in the `outputScale`. In order to avoid numeric instabilities, the `outputScale` should not be set to extremely large values.

As described in Sec. 6.2.5, the training of the neural networks is carried out using the iRPROP$^-$ algorithm using the neural network training library FANN [103]. This training is repeated 10 times with different initial random values for the network weights. The training is stopped, once the training error on an independent validation set, containing `nValid` $= 50000$ samples, begins to increase again. For each of the `repeatTrain` $= 10$ training runs the optimal weight set is selected at the minimum of the training error in the validation set. After the training, the best of the `repeatTrain` $= 10$ trained weight sets is selected by using the independent test data set with `nTest` $= 50000$ independent samples.

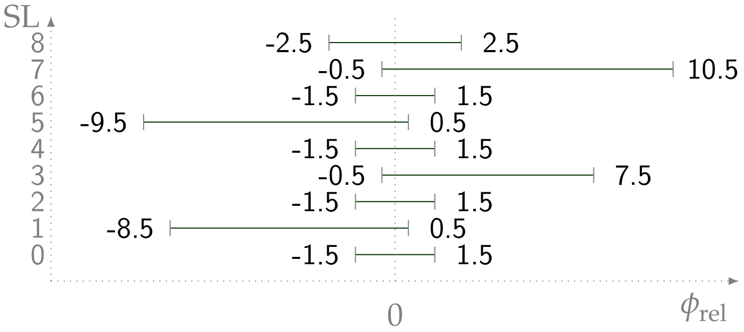


Figure 6.14.: ID-ranges for the TS selection obtained after applying the `relevantCut` on the ID-range histograms shown in Fig. 6.15.

The relevant ID-ranges for the hit selection are determined by the two training parameters `nTrainPrepare` and `relevantCut`. A data set of `nTrainPrepare` $= 5000$ track samples is used prior to the training, in order to fill a histogram for each of the 9 CDC SLs with the ϕ-distances of the 2D tracks to the TS positions. Figure 6.15 shows such histograms, where the x-axis shows ϕ_{rel}: the ϕ-distance between the 2D tracks and the TSs in each SL in units of TS-ids. The y-axis shows the TS rates in bins of this ID-range distance (i.e. the TS counts are divided by `nTrainPrepare`).

Especially in the inner axial SLs (SL0, SL2) in Fig. 6.15, rates larger than 100 % appear. This means that typically more than one TS is hit at the same distance, which corresponds to first and second priority hits within the same TS. The distributions in the axial SLs (SL0, SL2, SL4, SL6, SL8) are narrow and close to the position of the 2D tracks ($\phi_{\text{rel}} = 0$). In the stereo SLs (SL1, SL3, SL5, SL7) broader distributions with alternating shifts are observed. The alternating shift is a result of the alternating stereo angles in these SLs. In order to obtain the hit selection for the neural network, a cut is applied on these histograms. Starting from the peak bin of each histogram, bins from the right and from the left are added to the ID-range as long as their bin content is larger than 2 %. This can be configured with the parameter `relevantCut` $=$ 0.02 as described in Sec 6.3.2.

In principle, each of the expert networks determines its own ID-ranges. But with the setup used here (without geometrical sectorization), the differences are small and it is sufficient to study a single ID-range histogram of the hit selection. This will be different when stereo sectorization is introduced, as will be shown later in Sec. 6.5.3. After the cut is applied, only the ID-ranges are used for the hit selection, not the full histograms. The resulting ID-ranges, as they are used in the preprocessing, are shown in Fig. 6.14. Further studies with different training runs of the neural network, including different sample sizes of the training data, different training targets and different track segment types are shown in Appendix D.1.

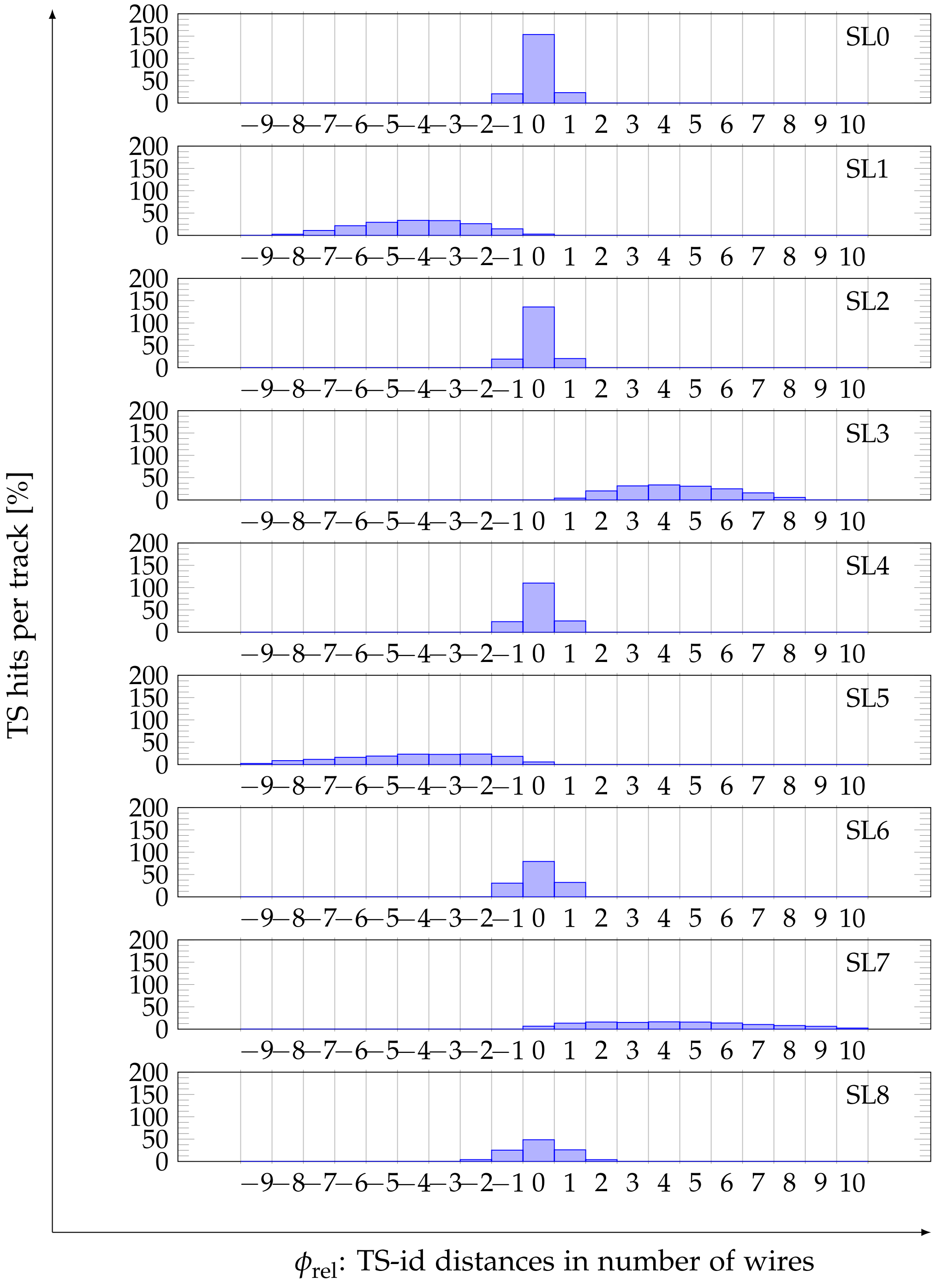


Figure 6.15.: Trained ID-range histograms of the basic neural network. From these histograms, the ID-ranges are selected, which are used by the neural network preprocessing for the selection of the input TSs.

6.4.4. Single Track Analysis

The most important features studied with single tracks are the efficiencies and the resolutions. At first a set of reference tracks needs to be chosen to be compared with the estimated neural network tracks. Similar to the 3D finder studies in Sec. 5.3.1, these can either be MC tracks or reconstructed tracks (Reco). In the case of MC tracks, multiple scattering might result in tracks not visible in the CDC. Additionally, in the data sets Z50, there is a region of (z, θ) combinations where the tracks geometrically do not pass through the CDC. In the case with background, the reconstruction might find additional tracks contained in the background sample, or wrongly identified fake tracks which strongly deviate from the simulated ones. Hence, some of the tracks in the reference sample (MC or Reco) might not be adequate. In the MC studies, the suitable reference tracks are selected by matching the simulated MC particle values with the Reco track values. Within basf2, this matching is carried out by the tracking modules via the relations to the hits in the detector. For the CDC trigger simulation, the method from the tracking system was adopted to build a matcher for the trigger tracks. This matcher allows to match estimated trigger tracks, from the track finder as well as from the neural network, to the MC or Reco reference tracks. The method of matching is described in Appendix A.3. Further details on the matching in the trigger simulation can be found in [67].

The efficiency of the full CDC trigger to estimate 3D track parameters with the neural network ϵ_{Trg} is a product of the track finder efficiency ϵ_{Finder} (introduced in the previous chapter, Sec. 5.3.1) with the neural network efficiency ϵ_{NN}. At the neural network level, only the tracks that are already found by the track finder can be processed. From this perspective, valid tracks that should be processed are only those found tracks received as input which have a match to a reference track. In this way, inefficiencies due to particle loss via multiple scattering or due to the inefficiency of the offline reconstruction are trimmed away.

As in the previous chapter on the 3D finder, the finder efficiency is defined here as the ratio of the number of correctly found tracks $N_{\texttt{Correct}}$ with the number $N_{\texttt{Candidate}}$, the tracks that should be found by the track finder (see "Analysis Method" in Sec. 5.3.1).

$$\epsilon_{\text{Finder}} = \frac{N_{\texttt{Correct}}}{N_{\texttt{Candidate}}} \tag{6.29}$$

In the MC data sets, the number of candidate tracks is selected as primary MC tracks which are reconstructed by the offline reconstruction. In the data sets recorded in the early runs of Belle II, no MC information is available and all reconstructed tracks are considered as reference tracks. Hence, the finder efficiency with MC test values $\epsilon_{\text{Finder}}^{\text{MC}}$ use a different main unit $N_{\text{Candidate}}$ than the finder efficiency with Reco test values $\epsilon_{\text{Finder}}^{\text{Reco}}$. The difference between the efficiencies will be shown in Sec. D.1.2. With real data, the only method to select "good" tracks, is by applying cuts to select only tracks from the IP (as will be discussed in Sec. 7.2).

The neural network efficiency ϵ_{NN} is the ratio of the tracks correctly processed by the neural network $N_{\text{NN out}}$ with the number of correct tracks received from the

Feature	Description
$\Delta\theta$	RMS90 of θ
Δz	RMS90 of z
ϵ_{NN}	Neuro efficiency (Found tracks processed by NN)
$\epsilon_{\mathrm{Finder}}$	Finder efficiency
ϵ_{Trg}	Trigger efficiency ($\epsilon_{\mathrm{Finder}} \times \epsilon_{\mathrm{NN}}$)

Table 6.8.: Basic resolution and efficiency features on a single track basis.

finder $N_{\mathrm{NN\ in}}$. Here correct tracks means, the finder tracks in $N_{\mathrm{NN\ in}}$ and the neural network tracks in $N_{\mathrm{NN\ out}}$ are required to have a match with a reference track.

$$\epsilon_{\mathrm{NN}} = \frac{N_{\mathrm{NN\ out}}}{N_{\mathrm{NN\ in}}} \tag{6.30}$$

In the simulation $N_{\mathrm{NN\ in}} = N_{\texttt{Correct}}$, which means the neural network input is the same as the finder output. However, in hardware transmission errors may occur such that not all tracks send by the finder are necessarily received by the network.

With the efficiency of the finder and the efficiency of the neural network, the full single track efficiency of the trigger can be defined as the product:

$$\epsilon_{\mathrm{Trg}} = \epsilon_{\mathrm{Finder}} \times \epsilon_{\mathrm{NN}} \tag{6.31}$$

where ϵ_{Trg} denotes the combined finder and neural network tracking efficiency.

After the neural network tracks are matched to valid reference tracks (see Appendix A.3), the resolution of the estimates can be calculated. Since the neural networks estimate the 3D track parameters z and θ, the interesting resolutions here are Δz and $\Delta\theta$. As before in the studies on the 3D finder, after trimming the histograms of the differences between the estimates and the matched reference values, the RMS (Root Mean Square) is used as estimator for the resolution. Trimming removes outliers and allows to give insight into the central distribution. Usually the RMS90 is used to calculate the resolutions in the following studies, where on each tail of the distributions 5 % of the entries are trimmed away. Since no additional degrees of freedom are introduced, as it would be the case in a fit of the resolution with a model including the tails, this RMS90 is a quite stable estimator for the resolution. The details of the error calculation for the efficiencies and the resolutions are described in Sec. 5.3.2 in application to the 3D finder studies and on a general basis in the Appendix A. Table 6.8 is a summary of the basic resolution and efficiency features.

6.4.5. Basic Neural Network Performance

In the following, this basic network is tested on simulated single muon track events originating from the z-axis. The data sets for the training and the first test are generated with the same track parameter range Z50 in Tab. 6.6, however, with independent random seeds and a different distribution of the polar track angle θ. While the training data was generated with a uniform distribution in θ, the test data was generated with a uniform distribution in $\cos(\theta)$. For details on the track distribution in the train and test data sets see Sec. 6.4.2.

Parameter	value
`Background`	[bkg0, 12p3, 15p2, 15p3]
`Finder`	2D
`Target`	MC
`TS-LUT`	LUTbkg

Table 6.9.: Parameters of the test data for the basic network.

The data settings used in this study are listed in Tab. 6.9. Four different background types are tested: bkg0 is a simulation without background; 12p3 are the official background files from the older background campaign 12; 15p2 and 15p3 are the official Phase 2 and Phase 3 background mixing files from the campaign 15. More details on the background simulation can be found in Chapter 4. The same data settings are used for the test as were used in the training. Most importantly, the same background type is used. The networks were trained with the MC true values as training target and with the 2D finder as input (see Sec. 6.4.3).

The performance of this network is measured by comparing its output values to the MC true values (primary MC particles with a match to a reconstructed track). Table 6.10 (a) shows an overview of the single track results with the four different background types. In order to discuss the relative size of the errors in this first presentation of the results, additionally the errors for each result value are shown here in Tab. 6.10 (b). The good z-resolution without background of $\Delta z = (1.991 \pm 0.008)$ cm drops to $\Delta z = (6.314 \pm 0.026)$ cm with the Phase 3 background 15p3. The same tendency is observed in the θ-resolution, with $\Delta\theta = (2.471 \pm 0.010)\,^\circ$ without background and $\Delta\theta = (5.735 \pm 0.025)\,^\circ$ with 15p3. With the 2D finder as input, the tracks processed by the network are pre-selected by the limited efficiency of the 2D finder. Therefore, the efficiency of the trigger of $\epsilon_{\mathrm{Trg}} \approx 50\,\%$ ($\epsilon_{\mathrm{Trg}} = (56.427 \pm 0.169)\,\%$ in the case with 15p3 and $\epsilon_{\mathrm{Trg}} = (51.700 \pm 0.167)\,\%$ in the case without background) is dominated by the finder efficiency $\epsilon_{\mathrm{Finder}} \approx 50\,\%$ ($\epsilon_{\mathrm{Finder}} = (57.551 \pm 0.169)\,\%$ with 15p3 and $\epsilon_{\mathrm{Finder}} = (51.962 \pm 0.167)\,\%$ without background). In turn, all neural network efficiencies ϵ_{NN} are close to 100 % ($\epsilon_{\mathrm{NN}} = (98.048 \pm 0.062)\,\%$ with 15p3 and $\epsilon_{\mathrm{NN}} = (99.495 \pm 0.033)\,\%$ without background). All discussed errors discussed so far are smaller than 1 % of their respective value. For reasons of clarity,

Results

	$\mathrm{Bkg}_{\mathrm{test}}$	$\Delta\theta_{\mathrm{MC}}$	Δz_{MC}	$\epsilon_{\mathrm{NN}}^{\mathrm{MC}}$	$\epsilon_{\mathrm{Finder}}^{\mathrm{MC}}$	$\epsilon_{\mathrm{Trg}}^{\mathrm{MC}}$
		[°]	[cm]	[%]	[%]	[%]
(a)	bkg0	2.471	1.991	99.495	51.962	51.700
	12p3	3.573	3.353	99.355	53.489	53.144
	15p2	2.733	2.451	99.494	52.540	52.274
	15p3	5.735	6.314	98.048	57.551	56.427

Errors

	$\mathrm{Bkg}_{\mathrm{test}}$	$\Delta(\Delta\theta_{\mathrm{MC}})$	$\Delta(\Delta z_{\mathrm{MC}})$	$\Delta(\epsilon_{\mathrm{NN}}^{\mathrm{MC}})$	$\Delta(\epsilon_{\mathrm{Finder}}^{\mathrm{MC}})$	$\Delta(\epsilon_{\mathrm{Trg}}^{\mathrm{MC}})$
		[°]	[cm]	[%]	[%]	[%]
(b)	bkg0	0.010	0.008	0.033	0.167	0.167
	12p3	0.015	0.013	0.037	0.168	0.168
	15p2	0.012	0.010	0.033	0.168	0.168
	15p3	0.025	0.026	0.062	0.169	0.169

Table 6.10.: Results of the basic neural network test with four different background types. The networks are tested on single muon tracks with vertices uniformly distributed along the z axis.

the error values will not be shown explicitly in the result tables of the subsequent studies. Instead, these errors will only be shown as error bars in the selected plots.

Track Parameter Resolutions

In the following plots, the resolution of the estimated 3D track parameters (θ, z) are shown as functions of the binned true *MC* track parameters $(p_{\mathrm{T}}, \phi, \theta, z)$. This allows for studying the performance in the different track parameter regions. All resolution plots show the measured resolutions $\Delta\theta$ and Δz in log-scale on the y axis and using 20 bins for the respective variable on the x-axis. In order to avoid underrepresented bins with large errors, each bin in the plots is required to contain at least 500 track pairs. With 10^5 simulated single track events uniformly distributed in the track parameter shown on the x-axis and with the 20 bins on the x-axis, this means the minimum trigger efficiency for the shown resolution values is 10 %. In this study, each plot shows four different networks, each trained and tested with a different background type.

Figure 6.16 shows the neural network resolutions for different values of the 2D track parameters (p_{T}, ϕ). Most prominently appears the accuracy loss with the different background types. The best resolutions are achieved without background nobkg, followed closely by the Phase 2 background 15p2. As expected, the Phase 3 background simulations 12p3, 15p3 result in the worst resolutions. In particular, with the latest background campaign 15 15p3 a significant resolution loss can be observed. Figure 6.16 (a) and (c) show a decrease in resolution for tracks with low-p_{T}. Figure 6.16 (b) and (d) show Δz and $\Delta\theta$ as functions of ϕ. Since tracks are uniformly distributed in ϕ and the detector is symmetric in ϕ, the ϕ-dependent changes in the resolutions are mainly caused by the background. With the Phase 3 backgrounds, a broad peak appears in the Δz and $\Delta\theta$-distributions around $\phi \approx 0^\circ$. However, there are also multiple scattering effects of the tracks with the detector material, causing the bumpy structure for the case bkg0 in Fig. 6.16 (d). Without background, $\Delta z(\phi)$ in Fig. 6.16 (b) is flat, followed closely by the Phase 2 background conditions 15p2. Since this ϕ-dependence is not seen in the Δz-distribution without background, this indicates a deflection of the tracks in close vicinity to the vertex positions. The structure in $\Delta\theta(\phi)$ can be explained by tracks displaced from the IP, where the z-axis is not contained within the beampipe, but lies in the metal of the crotch part between the two separate beam lines of the high energy ring and the low energy ring.

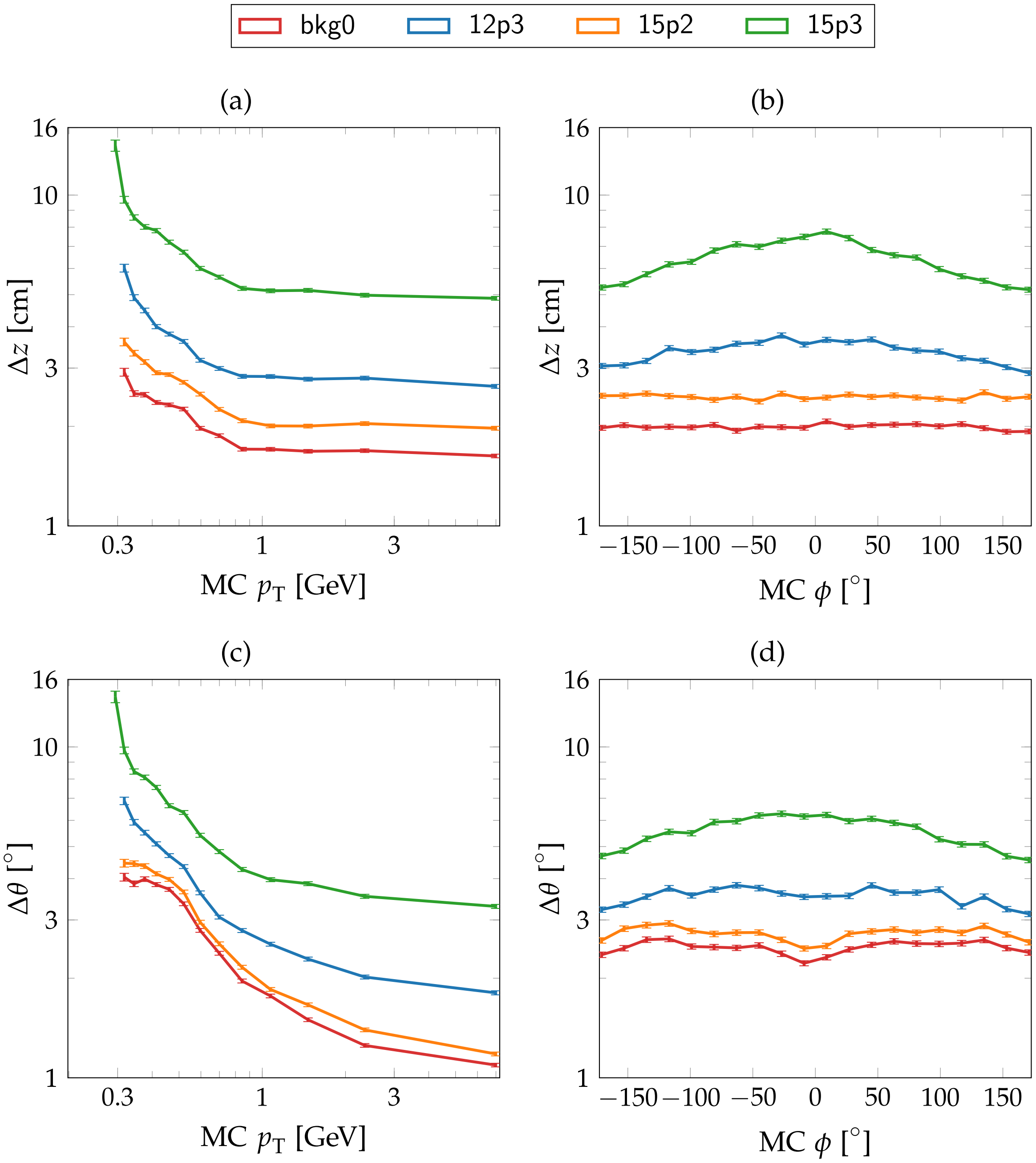


Figure 6.16.: Resolutions of the 3D track parameters as functions of the MC 2D track parameters p_T and ϕ.

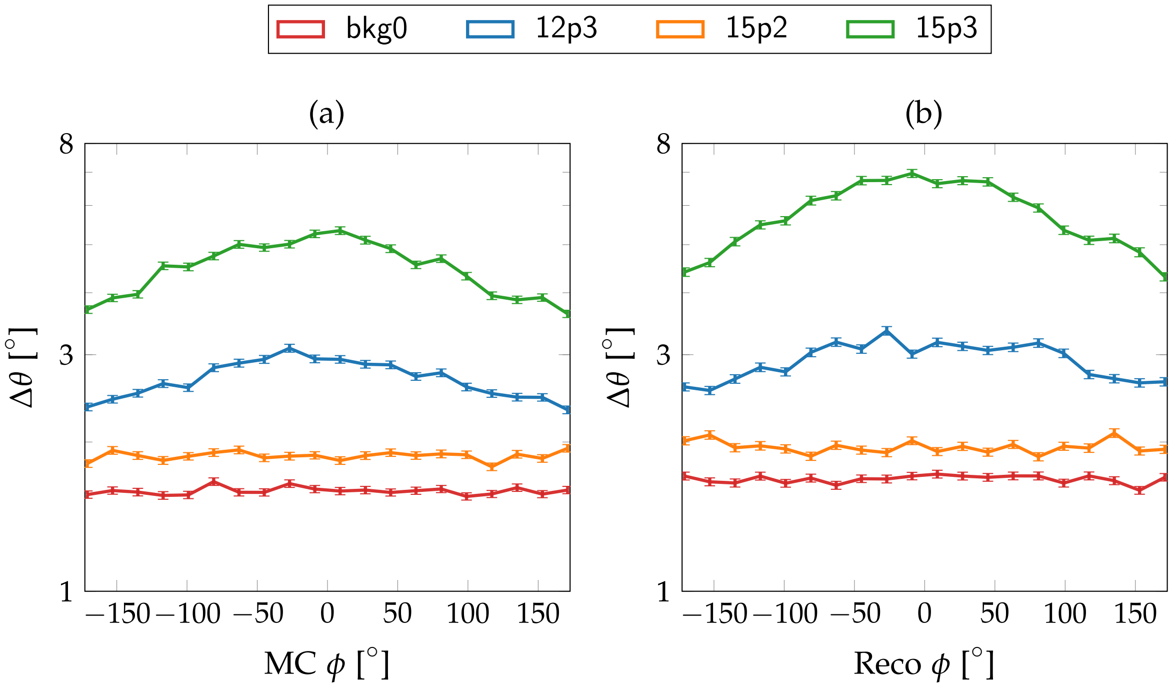


Figure 6.17.: Clarification of the ϕ-dependence of $\Delta\theta$. (a) shows tracks only coming from the IP. In (b), the $\Delta\theta$ is calculated with tracks along the z-axis, but with the reconstructed track parameter values as target (instead MC track parameter values).

In order to verify the scattering at the detector material as the source of the ϕ-dependent disturbance in $\Delta\theta$, additional plots are shown in Fig. 6.17. Fig. 6.17 (a) shows $\Delta\theta(\phi)$ for a test data set with tracks only coming from the IP. Since there is no heavy material in the way for tracks coming from the IP, this distribution is flat as expected. Additionally, Fig. 6.17 (b) shows $\Delta\theta(\phi)$ on the tracks again distributed along the z-axis in the range $z \in [-50, 50]$ cm. However, $\Delta\theta$ is now calculated with the reconstructed tracks instead of the MC tracks. Since the reconstruction is run after the scattering effects, the reconstructed value is expected to be more adequate to describe the actual track seen in the detector. As expected, the $\Delta\theta(\phi)$ in Fig. 6.17 (b) is also flat.

The neural network resolutions as functions of the true MC 3D track parameters θ and z are shown in Fig. 6.18. Since the detector is built around the IP, the resolutions along the z-axis in Fig. 6.18 (a) and Fig. 6.18 (c) are best around $z = 0$ cm. The resolution $\Delta z(\theta)$ in Fig. 6.18 (b) is nearly flat in the region of $\theta \in [40, 120]^\circ$, which corresponds to the region where tracks from the IP can reach all CDC SLs. The decrease at shallow-θ angles is owed to the geometrical acceptance of the CDC. Fig. 6.18 (d) has the same decrease at shallow-θ angles, but a linear dependence of $\Delta\theta(\theta)$: the geometrical asymmetry of the CDC enables an improved θ-resolution towards the forward region (small values of θ).

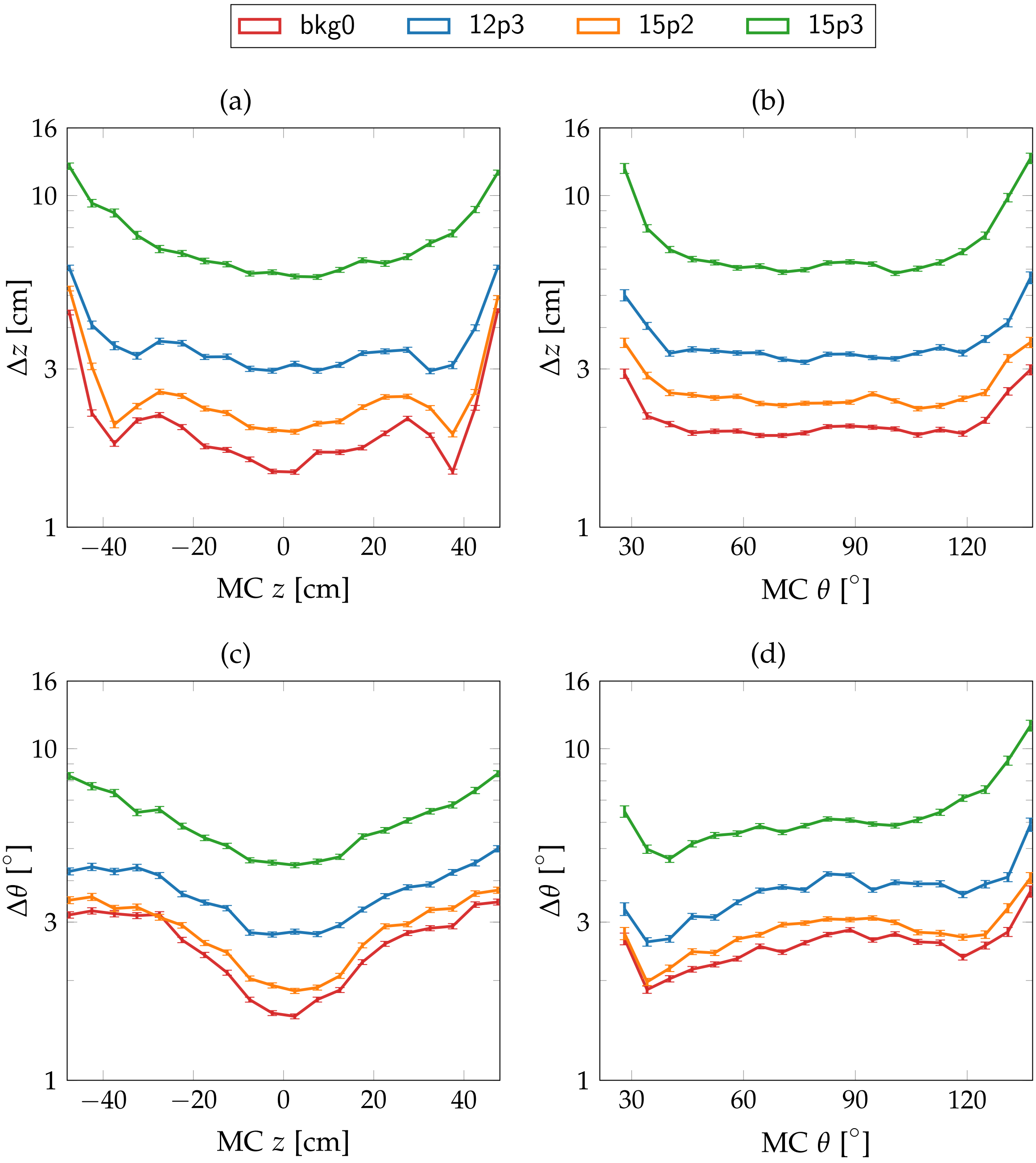


Figure 6.18.: Resolutions of the neural network 3D track parameter estimates, as functions of the MC 3D track parameters θ and z.

Track Processing Efficiencies

The distributions of the efficiencies $\epsilon_{\mathrm{Finder}}$, ϵ_{NN} and ϵ_{Trg} in the three track parameters $p_{\mathrm{T}}, \theta, z$ are shown in Fig. 6.19. Similar to the previous resolution plots, the distributions of the efficiencies are shown with 20 bins in each shown track parameter. In order to avoid underrepresented bins with large fluctuations, in each bin, at least 500 tracks are required as main unit in the efficiency calculations. For the track parameters simulated with a uniform distribution this corresponds to 10 % of the simulated tracks per bin. Because the efficiencies as functions of ϕ are flat, they are not shown in Fig. 6.19.

With the 2D finder as input, the efficiency of the neural network ϵ_{NN} is quite good, which can be seen in the left column of Fig. 6.19 (a), (d), (g). Tracks which are difficult to reconstruct are already sorted out, as they are not found by the 2D finder. For this reason, the range of the efficiency axis is zoomed into the interesting region close to 100 %. The neural network can only drop found tracks if they have too few stereo TSs (≤ 2). Since the 2D finder requires at least four axial TSs, a found track typically has at least three stereo TSs such that the observed good neural network efficiencies match the expectation here. Only at the geometrical acceptance region of the CDC, sufficient stereo TSs might be missing such that the track is lost. This can be seen in Fig. 6.19 (a) and (d), where the neural network efficiencies are plotted against the track parameters p_{T} and θ. Only in the low-p_{T} region in Fig. 6.19 (a), and at shallow-θ angles in Fig. 6.19 (d), there is a substantial efficiency loss. Interestingly, Fig. 6.19 (g) shows that the neural network efficiency does not depend on the z-position, as could be expected from the CDC geometry. This is a result of the good neural network efficiency combined with the low 2D finder efficiency and will be studied in combination with the 3D finder as input in Sec. 6.5.1.

The second column (Fig. 6.19 (b), (e), (h)) shows the finder efficiency $\epsilon_{\mathrm{Finder}}$. In order to visualize the differences between the four different background types, the efficiency axis is limited to the appropriate region for the 2D finder efficiencies. The third column (Fig. 6.19 (c), (f), (i)) shows the full trigger efficiency ϵ_{Trg}, which is the product of the neural network efficiency and the finder efficiency. In order to show the full picture, here the absolute efficiency range $[0, 100]$ % is chosen for ϵ_{Trg}. Obviously, the trigger efficiency is dominated by the finder efficiency. Fig. 6.19 (h) shows that the 2D finder efficiency is almost flat in z, with a little improvement for small positive values of z that can be attributed to the asymmetric CDC geometry. The 2D finder efficiencies in p_{T} and θ agree with the efficiencies for single tracks from the IP. For a general discussion of the 2D finder efficiency in comparison with the 3D finder efficiency, see Sec. 5.3.3. Here it is interesting to note that the 2D finder efficiency increases with background, while the neural network efficiency slightly decreases (compare Sec. 5.3.4). The overall trigger efficiency, dominated by the 2D finder efficiency, still increases with background. The reason for this inverted tendency is that the tracks additionally found with larger background levels contain background TSs which are more likely to be cut off by the neural network preprocessing.

Summary of the Basic Network Studies

The basic network was tested on a test data set containing tracks uniformly along the z-axis and shallow-θ as well as low-p_{T} tracks. Although these deviations from the full CDC acceptance region for tracks from the IP limit the achieved neural network resolutions and efficiencies, the achieved averaged results are still quite good. In the azimuthal angle ϕ, the neural network efficiency depends on the background level only. However, the resolution shows to have a combined ϕ, z-dependence, which can be explained by the asymmetric material distribution of the supporting heavy material in Belle II. No z-dependence of the neural network efficiency has been observed so far, but this will be discussed again with the more efficient 3D finder in Sec. 6.5.1. All tested networks have the same basic architecture which can be loaded into the present hardware. Using simulated single track events, the studies in Appendix D.1 demonstrate the effect of a change in important configuration options during the training and the performance in different single track parameter test data ranges.

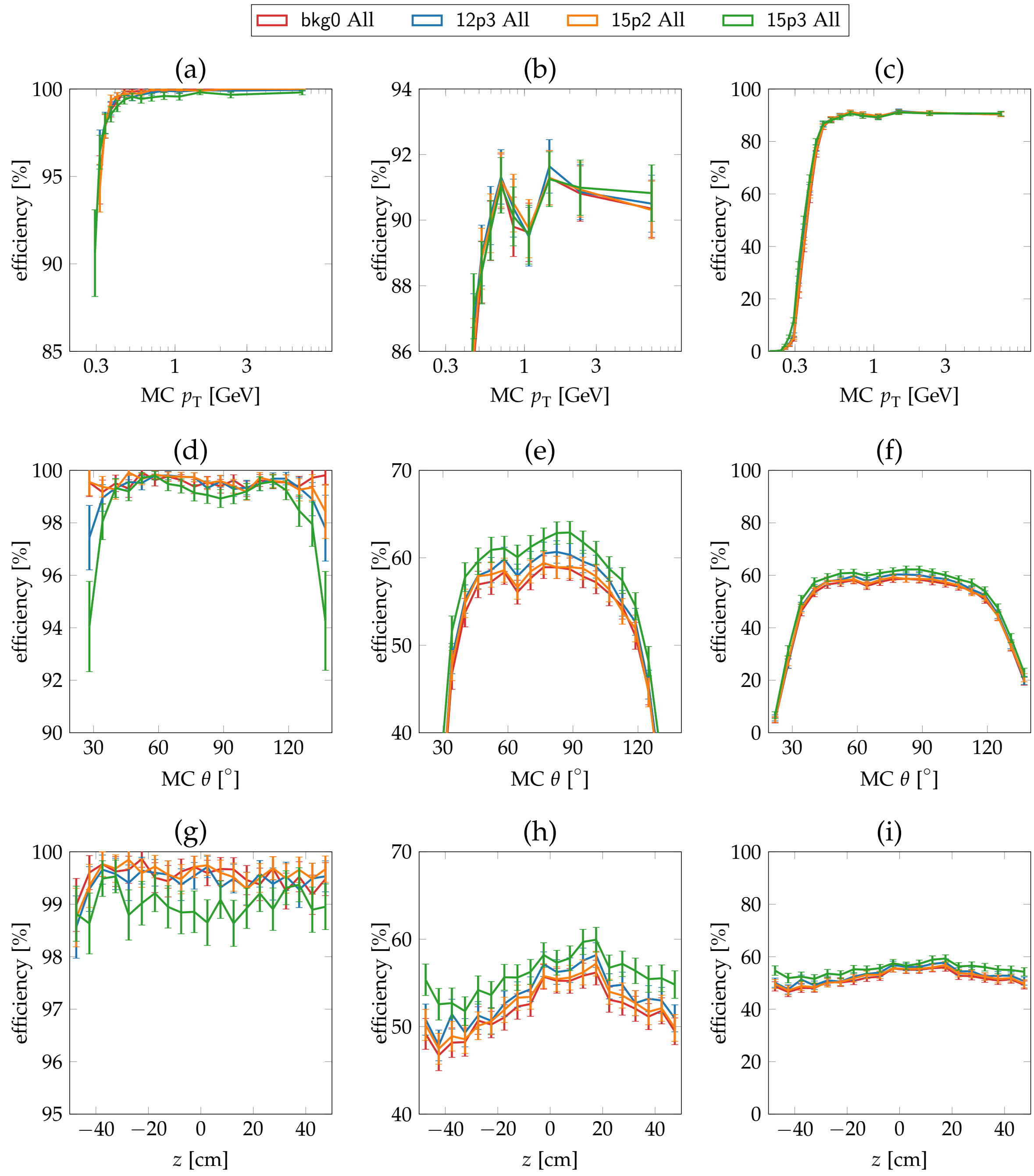


Figure 6.19.: Single track efficiencies of the neural network (ϵ_{NN}), the 2D finder (ϵ_{Finder}) and the full CDC track trigger pipeline (ϵ_{Trg}) in a test run with the data ranges Z50 and with four different background types. In the column ϵ_{Trg} the full range of the efficiency is shown, while for ϵ_{NN} and ϵ_{Finder} the range of the efficiency axis is zoomed into the interesting region.

6.4.6. Advanced Single Track Features

As an extension of the basic single track resolutions and efficiencies, in this section more complex observables are introduced for the performance studies of the neural network. At first, the expert pattern-rates are counted, which allows to check how many stereo SLs typically contain a TS related to a track. This means, the pattern-rate for an expert network *i* describes and how often expert *i* is called per total calls of the network. In addition, track cuts are introduced, as they could be implemented in the GDL of the trigger to make use of the neural network estimates. These single track cuts can be combined to a full trigger logic for multi track events. Studies with these event based cuts on multi track event are presented in the initial state radiation studies in Sec. 7.1.

Pattern Rates

The basic neural network setup uses five expert networks: four networks are trained for a single missing stereo TS and one is trained for the case where all four stereo SLs have a TS. In order to inspect the loss rate of the TSs in the SLs, the columns `SL1`, `SL3`, `SL5`, and `SL7` are added to the summary tables. They show the rate of tracks with a related TS hit in the respective SLs. This means, for a SL *i*, this rate is the number of TS hits in SL *i* related to an estimated neural network track per neural network tracks estimated in total.

binary	decimal	column name	description
1111	15	`p15`	all SLs present
1110	14	`p14`	SL7 missing
1101	13	`p13`	SL5 missing
1011	11	`p11`	SL3 missing
0111	7	`p7`	SL1 missing

Table 6.11.: Description of the pattern columns introduced in the result tables. The binary patterns describe only the stereo TSs while the intermediate axial TSs, which appeared in the full pattern definition in Tab. 6.2, are left out.

For the evaluation of the individual expert neural networks, the pattern-rates are introduced as new columns of the result tables. In the five-expert network setup, the labels of these columns are: `p7`, `p11`, `p13`, `p14`, and `p15`. The shown integer number following the "`p`" corresponds to the decimal value of the corresponding binary patterns listed in Tab. 6.11. The `SL`-rates are obviously correlated with the pattern-rates.

(a) `Gain`

	partner	cut
base	MC/Reco	$-1\,\mathrm{cm} < z_{\mathrm{base}} < 1\,\mathrm{cm} \wedge d_{\mathrm{base}} < 1\,\mathrm{cm}$
target	NN	$-20\,\mathrm{cm} < z_{\mathrm{target}} < 20\,\mathrm{cm}$
multi track		All tracks in base. ≥ 1 tracks in target.

(b) `Loss`

	partner	cut
base	NN	$-20\,\mathrm{cm} < z_{\mathrm{base}} < 20\,\mathrm{cm}$
target	MC/Reco	$-40\,\mathrm{cm} < z_{\mathrm{target}} < 40\,\mathrm{cm}$
multi track		All tracks in base. No track in target.

(c) `Fake`

	partner	cut
base	MC/Reco	$\lvert z_{\mathrm{base}} \rvert > 40\,\mathrm{cm}$
target	NN	$-20\,\mathrm{cm} < z_{\mathrm{target}} < 20\,\mathrm{cm}$
multi track		All tracks in base. ≥ 1 tracks in target.

Table 6.12.: Cut definitions for the event efficiency features.

z-Cut Based Trigger Rates

With the z-cut based features, an implementation of new event trigger logics based on the neural network estimates is anticipated. Most obviously, this is a z-cut where the z-estimate of a single track is used to decide whether this track originates from the IP. For the study of different z-cut values, as before in the calculation of the resolutions, at first matching pairs of the estimated neural network tracks $(\theta_{\mathrm{NN}}, z_{\mathrm{NN}})$ with some reference tracks $(\theta_{\mathrm{ref}}, z_{\mathrm{ref}})$ are selected, where these reference tracks can either be MC tracks or Reco tracks. Then cuts are applied on these track pairs in order to calculate the track cut rates (see Tab. 6.12). Abstractly, by using different cuts on the reference tracks and on the estimated tracks, a class of "target" tracks fulfilling the target-cut and a class of "base" tracks fulfilling the base-cut is defined. Each calculated rate is the relative frequency of "target" per "base". This enables the transition from single track cuts to event based cuts, as the cut results of all single tracks can be combined.

In detail, three z-cut based rates are selected for the following studies: `Gain`, `Fake` and `Loss`. These cut types are summarized in Tab. 6.12. A single track cut based efficiency, `Gain`, can be obtained by selecting the tracks truly coming from the IP and counting how often these tracks are estimated within a defined cut interval by the neural network. A typical definition of the number of tracks correctly estimated within a z-cut at 20 cm (i.e. the “target” tracks) is

$$N_{\text{zcut}} = |z_{\text{ref}}| \leq 1\,\text{cm} \wedge |z_{\text{NN}}| \leq 20\,\text{cm} \tag{6.32}$$

where $\wedge$ denotes the logical and operator, the IP reference interval is $z \in [-1, 1]$ cm and the estimation interval is $z_{\text{NN}} \in [-20, 20]$ cm. Combined with the number of tracks truly coming from the IP (the “base” tracks)

$$N_{\text{IP}} = |z_{\text{ref}}| \leq 1\,\text{cm} \tag{6.33}$$

the Gain-rate can be defined as the ratio of these two track counts

$$\texttt{Gain} = \epsilon_{\text{zcut}} = \frac{N_{\text{zcut}}}{N_{\text{IP}}} = \frac{|z_{\text{ref}}| \leq 1\,\text{cm} \wedge |z_{\text{NN}}| \leq 20\,\text{cm}}{|z_{\text{ref}}| \leq 1\,\text{cm}.} \tag{6.34}$$

Hence, the Gain-rate is a definition of a signal efficiency. For single track events, it can also be interpreted as the true positive rate. In extension to multi track events, the requirement is that any of the track pairs (NN,ref) fulfills this condition. This means, at least one of the tracks truly coming from the IP is required to be estimated at the IP. The cut Gain is summarized in Tab. 6.12 (a).

Closely connected is the definition of a Fake-rate, which describes how often tracks that are truly displaced from the IP are estimated within the given cut region (“target” tracks):

$$N_{\text{wrong}} = |z_{\text{ref}}| \geq 40\,\text{cm} \wedge |z_{\text{NN}}| \leq 20\,\text{cm}. \tag{6.35}$$

Combined with the number of truly displaced tracks (“base” tracks)

$$N_{\text{outside}} = |z_{\text{ref}}| \geq 40\,\text{cm} \tag{6.36}$$

the Fake-rate can be defined as

$$\texttt{Fake} = \frac{N_{\text{wrong}}}{N_{\text{outside}}} = \frac{|z_{\text{ref}}| \geq 40\,\text{cm} \wedge |z_{\text{NN}}| \leq 20\,\text{cm}}{|z_{\text{ref}}| \geq 40\,\text{cm}}. \tag{6.37}$$

The Fake-rate can also be interpreted as the rate of false positives out of the true negatives. Hence, a background suppression efficiency is given by the inverse of the Fake-rate. In this sense, a background track is a track displaced from the IP and the background suppression efficiency is the number of tracks correctly classified as background per background tracks in total.

$$\text{background suppression} = 1 - \texttt{Fake}. \tag{6.38}$$

In extension of the Fake-rate to multi track events, any track is required to be estimated within the z-cut region (see Tab. 6.12 (c)). For the inverse background suppression rate, this means that for multi track events containing only displaced tracks, none of the displaced tracks are allowed to be estimated as IP track.

The Loss-rate, is defined here as the number of tracks which are estimated to be within the cut interval, but which have their true vertices far away. Here, the "base" tracks are the tracks estimated within the z-cut of 20 cm:

$$N_{\text{base}} = |z_{\text{NN}}| \leq 20\,\text{cm}. \tag{6.39}$$

Out of the these tracks, the number of tracks where the reference partner value is outside of ±40 cm is counted ("target" tracks):

$$N_{\text{target}} = |z_{\text{ref}}| \geq 40\,\text{cm}. \tag{6.40}$$

The Loss-rate used here is then defined by the ratio:

$$\texttt{Loss} = \frac{|z_{\text{NN}}| \leq 20\,\text{cm} \wedge |z_{\text{ref}}| \geq 40\,\text{cm}}{|z_{\text{NN}}| \leq 20\,\text{cm}}. \tag{6.41}$$

The Loss-rate can also be interpreted as the rate of false positives out of the total estimated positives. Hence, the Loss-rate can be interpreted as the inverse of a signal purity of the estimated tracks.

$$\text{signal purity} = 1 - \texttt{Loss}. \tag{6.42}$$

In the extension of the Loss-rate to multi track events, all of the reference tracks are required to be outside of the reference z-cut region $|z_{\text{target}}| \geq 40\,\text{cm}$ (see Tab. 6.12 (b)).

Example Summary Table

Table 6.13 is an example for the summary tables used in the subsequent studies. It shows all the newly described rate types, using a test data set with tracks uniformly distributed in the range $z \in [-50, 50]$ cm (ranges Z50 in Tab. 6.6) and a neural network with train data size NND (see Tab. D.1). The table has been sliced into three parts: the first part, Tab. 6.13 (a), shows the already known resolution and efficiency values; the second part, Tab. 6.13 (b), shows hit rates in each SL and the pattern rates; the third part, Tab. 6.13 (c), shows the z-cut rates.

In the SLi columns in Tab. 6.13 (b), the integer i in the column label describes the number of the SL in the CDC. Only the stereo SLs are shown, as expert neural networks are trained for missing stereo TSs in the neural network input. The SLi columns in Tab. 6.13 are all close to 100 % except for SL7. This is expected, as SL7 is the outermost stereo SL and it is thus most likely to not receive a TS hit. This may either be due to low-p_{T} tracks curling back before SL7 is reached, or because of shallow polar angles θ letting the track leave the CDC volume before SL7 is reached. Note that for a track from the IP, if SL7 is missing the outwards following axial SL8 is usually missing as well. Due to the additional background occupancy, the finding of TSs (TSF efficiency) is more likely and, thus, the rate SL7 increases with higher background levels.

	Bkg$_{\text{test}}$	$\Delta\theta_{\text{MC}}$	Δz_{MC}	$\epsilon^{\text{MC}}_{\text{NN}}$	$\epsilon^{\text{MC}}_{\text{Finder}}$	$\epsilon^{\text{MC}}_{\text{Trg}}$
		[°]	[cm]	[%]	[%]	[%]
(a)	bkg0	2.4	1.9	99.5	52.0	51.7
	15p2	2.7	2.4	99.5	52.5	52.3
	12p3	3.6	3.3	99.4	53.5	53.1
	15p3	5.7	6.3	98.0	57.6	56.4

	Bkg$_{\text{test}}$	SL1	SL3	SL5	SL7	p7	p11	p13	p14	p15
		[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]
(b)	bkg0	99.3	100.0	99.9	87.9	0.7	0.0	0.1	12.1	87.1
	15p2	99.3	100.0	99.8	87.9	0.7	0.0	0.2	12.1	87.0
	12p3	99.5	99.9	99.8	88.8	0.5	0.1	0.2	11.2	88.0
	15p3	99.6	99.9	99.7	90.8	0.4	0.1	0.3	9.2	90.0

	Bkg$_{\text{test}}$	Gain1	Loss1	Fake1
		[%]	[%]	[%]
(c)	bkg0	99.7	0.0	0.1
	15p2	99.0	0.2	0.5
	12p3	98.4	0.8	2.2
	15p3	95.5	2.4	7.0

Table 6.13.: Newly introduced columns for the rates of TS per SL per track, the pattern rates of the expert networks and the z-cut based rates Gain, Loss and Fake. The results are shown for a network trained with training data size NND and tested on tracks with the test data ranges Z50.

In the pattern columns in Tab. 6.13 (b), the integer number in the column label describes the binary pattern of stereo TSs related to the track (see Tab. 6.11). Hence, the percentages in the pattern columns correspond to the frequency with which each trained expert network is used. The pattern rates p7, p11, p13 are all close to zero, which means that these experts are almost never in use. About 90 % of the tracks are processed by p15, with TSs in all four stereo SLs. Around 10 % of the tracks are processed by p14, where the stereo SL7 does not have a TS hit. Thus, the pattern rates are in agreement with the SL-rates.

Each z-cut rate label in the first row in Tab. 6.13 (c) is followed by an integer number describing the track multiplicity (i.e. Gain1, Fake1, Loss1). In multi track events, this integer describes the number of tracks estimated by the neural network per event, which have a matching reference track. Since single track events are studied here, the z-cut based rates in Tab. 6.13 (c) are only shown for a track multiplicity of one. A description of the z-cut rate labels is summarized in Tab. 6.12. Table 6.13 (c) shows that Gain1 reaches nearly 100 % in the case without background and decreases with larger amounts of background. The rates Fake1 and Loss1 are close to zero without background, but they increase with larger background levels.

Introduction to the Event Efficiency Features

In the studies with simulated initial state radiation events (presented in Sec. 7.1) and the studies with real data from the Belle II experiment (presented in Sec. 7.2), the data samples contain multi track events. In order to study the full event efficiency, the single track statistics need to be combined to full event efficiencies. For completeness, the extension of the just defined z-cut rates to event based cuts is shortly introduced here. Besides the z-cut rates `Gain`, `Fake`, `Loss`, additional rates based on the θ-difference between two tracks will be introduced in Sec. 7.1.2.

The z-cuts for multi track events are defined in the same way as for the single track statistics. The difference is that in each event all the track pairs are tested whether they fulfill the cuts. Thereby, the following logical conditions have to be distinguished: A cut could be satisfied by “all” the tracks, by “no” track, by “any” track and by “some” tracks. The distinction between “any” and “some” is introduced, because “all” is a subset of “any”. Therefore, “some” means there is “any” track within the cut, but not “all”.

The last row in Tab. 6.12 showed the chosen recipe for the track combination in the multi track events. In the event extension, the `Gain`-rate describes IP events, where all tracks are coming from the IP. Out of these, at least one track is required to be contained within the cut interval. Therefore, the `Gain`-rate is expected to increase with a larger track multiplicity.

The `Loss`-rate describes events estimated by the neural network within the z-cut region. Out of these, no track must be found within an enlarged z-cut region. This rate is expected to decrease with a larger track multiplicity.

The `Fake`-rate describes events, where all tracks are truly displaced from the IP. Out of these, the neural network is required to make at least one false estimate. Therefore, this rate is expected to increase with a larger track multiplicity.

As noted earlier, the integer numbers in the rate labels describe the track multiplicity. These labels appear as column descriptions in the first row of the result tables. However, in events with many tracks, the event multiplicity might exceed the number of columns shown in the result tables. Therefore, in the subsequent multi track studies, the column with the largest number corresponds to $\geq n$ matching track pairs. All labels with smaller numbers correspond to exactly n matching track pairs.

6.5. Benefits of the 3D Finder

Following the successful studies with a standalone 3D finder presented in the previous Chapter 5, here the integration with the neural network is tested. The main achievements of the 3D finder were an improved track finding efficiency and a good track parameter resolution in the three track parameters $(p_\mathrm{T}, \phi, \theta)$. The improvement in the efficiency became apparent especially for tracks with low-p_T and tracks with shallow-θ angles. For IP tracks contained within the full CDC acceptance region, where the 2D finder reaches almost full efficiency, the 3D finder achieves the same levels of high efficiencies (see Sec. 5.3.4). This means that no IP tracks are lost with the 3D finder in comparison with the 2D finder. Besides the combination with the neural network, in the subsequent studies the 3D finder will be tested on single tracks with displacements along the z-axis. Since all signal tracks at Belle II have their vertices at the IP, studies on simulated displaced tracks demonstrate the performance on background tracks. Following the theoretical discussions in Sec. 5.2.5, with the 3D finder a suppression of these displaced background tracks is expected. The reason are the hit representations of the stereo TSs as slanted planes in the 3D track parameter Hough space, where a single crossing point of all four stereo TSs only exists for tracks from the IP. Such an active background suppression of the 3D finder is a huge advantage, as it reduces the work load for the neural network trigger. The 3D finder used for all the following neural network studies is configured with the optimized parameters presented in Chapter 5. This 3D finder configuration parameter are summarized in Tab. 6.14.

parameter	value
`minhits`	4
`minweight`	24
`thresh`	0.85
`nbins` p_T	40
`nbins` ϕ	12
`nbins` θ	9

Table 6.14.: Summary of the 3D finder parameters used for the studies with the neural network. These settings were found with the studies presented in the previous Chapter 5.

The first study with the neural network in Sec. 6.5.1 will use the basic neural network setup, where only the 2D track parameters (p_T, ϕ) are used in the preprocessing of the neural network (i.e. the estimated θ from the 3D finder is not used). This allows a simple comparison of the 2D finder and the 3D finder as input for the neural network. In Sec. 6.5.2, neural networks optimized for low-p_T tracks are introduced, which are more efficient in processing the additional low-p_T and shallow-θ tracks found with the 3D finder. As a possible approach to use the θ-estimate of the

3D finder, a study of expert networks for sectors in the polar angle θ is presented in Sec. 6.5.3.

6.5.1. Comparison of the 2D Finder with the 3D Finder

The first studies of the neural network in combination with the 3D finder are carried out on simulated single track events with vertices at the IP, using the test data ranges NIP and ZIP from Tab. 6.6. This allows a comparison with the results in the previous Chapter 5 where IP tracks were used as well. In the second study simulated tracks displaced along the z-axis are tested. To this end, the test data ranges Z50 listed in Tab. 6.6 are used. Besides the effect on the neural network, this will, for the first time, show the important background suppression capabilities of the 3D finder for displaced tracks.

IP Tracks Fully Contained in the CDC

For tracks from the IP within the full geometrical acceptance region of the CDC, the 3D finder as well as the 2D finder are very efficient (see Sec. 5.3.4). Hence, the first study is supposed to demonstrate the effect of the different finders on the performance of the neural network trigger. This is done by using the range NIP, which does not contain short tracks with low-p_{T} or shallow-θ Tab. 6.6.

Note that this study directly connects to the setup of studies in Chapter 5: the range ZIP used here corresponds to the range L in the 3D finder studies; the range NIP used here corresponds to the range N in the 3D finder studies (see Tab. 5.7). The only difference in the test data sets is the θ-distribution, where a uniform distribution in $\cos(\theta)$ is simulated here. The single background type previously used in Chapter 5 corresponds to the Phase 3 background type labelled 15p3 here. In the test runs, the 2D finder as well as the 3D finder are used as input for the neural network on the same test data sets and background types.

Table 6.15 shows the summarized results for this comparison on the test data set NIP. The additional column Finder indicates whether the 2D finder or the 3D finder was used in the respective test runs. At the first glance, all efficiencies are close to 100 %. Only with the background type 15p3, a marginal efficiency loss of the total trigger efficiency up to 99.6 % can be observed. Without background, the neural network can achieve resolutions of $\Delta z \approx 1.3\,\mathrm{cm}$ and $\Delta\theta \approx 1.3^{\circ}$. With larger background levels, a resolution loss is observed with both finders as input to the neural network. This resolution loss appears to be slightly better handled using the 3D finder as input.

In order to show more details, on this data set where the finders are fully efficient, the resolutions achieved with the neural networks are shown in Fig. 6.20. In the resolutions as functions of p_{T}, in Fig. 6.20 (a) and (b), the neural networks with the 3D finder as input clearly achieve improved resolutions. Towards the low-p_{T} region, the resolutions with both finders become very similar. This is related to the limited efficiency of the 2D finder, which already applies at the lower bound of this test data

Bkg$_{\text{test}}$	Finder	$\Delta\theta_{\text{MC}}$	Δz_{MC}	$\epsilon^{\text{MC}}_{\text{NN}}$	$\epsilon^{\text{MC}}_{\text{Finder}}$	$\epsilon^{\text{MC}}_{\text{Trg}}$
		[°]	[cm]	[%]	[%]	[%]
bkg0	3D	1.3	1.3	100.0	100.0	100.0
bkg0	2D	1.4	1.3	100.0	99.7	99.7
15p2	3D	1.6	1.7	100.0	100.0	100.0
15p2	2D	1.7	1.7	100.0	99.7	99.7
12p3	3D	2.2	2.4	100.0	100.0	99.9
12p3	2D	2.4	2.5	100.0	99.8	99.8
15p3	3D	3.8	4.6	99.9	99.7	99.6
15p3	2D	3.9	5.1	100.0	99.6	99.6

Table 6.15.: Neural network with the 2D finder as input compared to the 3D finder as input. The test data set contains only tracks originating from the IP. Low-p_{T} and shallow-θ tracks are excluded (data set NIP in Tab. 6.6). The results are sorted by Δz_{MC}.

set. As previously shown in Fig. 5.28, the efficiency of the 2D finder towards the low-p_{T} limit of 0.35 GeV already drops to 95 %, while the 3D finder still achieves an efficiency of 99 %.

The resolutions as functions of θ in Fig. 6.20 (c) and (d) show only small differences between the 3D finder and the 2D finder input. The values are consistent with the slight improvements seen in the averaged values in Tab. 6.15. With the background 15p3 the resolution $\Delta z(\theta)$ of the neural network achieved with the 3D finder is slightly improved by 0.5 cm. In summary, the efficiencies and resolutions achieved with both finders as input to the neural network are very similar for tracks fully contained within the CDC acceptance. Small improvements can be observed with the 3D finder as input in the data set NIP: the neural network track parameter resolutions are slightly improved in the high-p_{T} region and the resolutions show an improved background robustness.

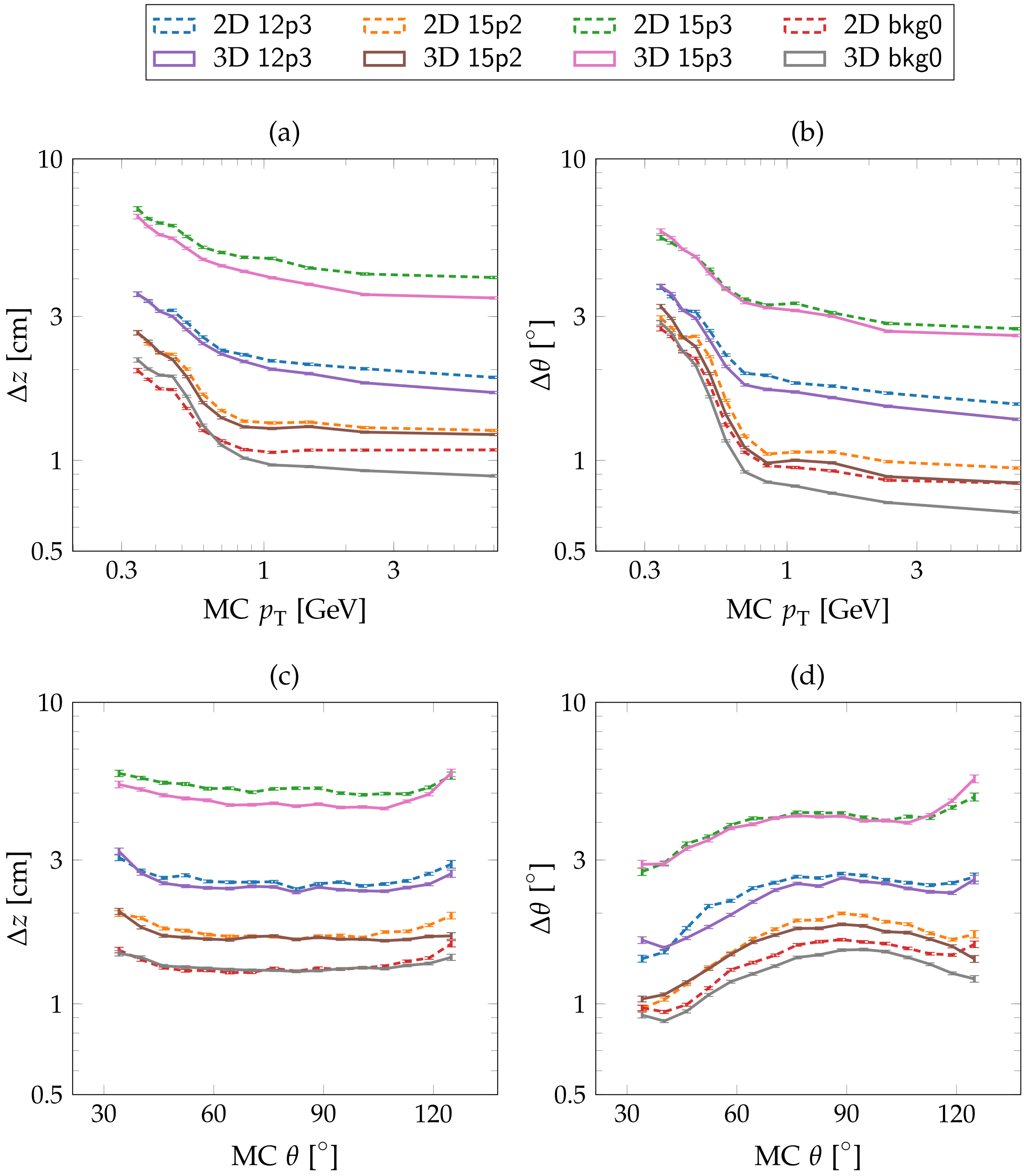


Figure 6.20.: Resolution of the neural network on the test data set NIP from Tab. 6.6 with the 2D finder compared to the 3D finder as input. The test data sets contain only tracks in the full acceptance region of the CDC. Four different background types are tested.

IP Tracks at the Boundary Region of the CDC

$\texttt{Bkg}_{\text{test}}$	$\texttt{Finder}$	$\Delta\theta_{\text{MC}}$	Δz_{MC}	$\epsilon_{\text{NN}}^{\text{MC}}$	$\epsilon_{\text{Finder}}^{\text{MC}}$	$\epsilon_{\text{Trg}}^{\text{MC}}$
		[°]	[cm]	[%]	[%]	[%]
bkg0	2D	1.6	1.5	99.5	55.2	55.0
bkg0	3D	2.0	1.8	72.4	92.8	67.2
15p2	2D	1.9	1.9	99.5	55.5	55.2
15p2	3D	2.4	2.3	73.3	93.1	68.3
12p3	2D	2.7	2.9	99.4	56.4	56.1
12p3	3D	3.6	3.6	77.1	93.7	72.2
15p3	2D	4.6	5.8	98.1	59.3	58.2
15p3	3D	6.2	6.4	84.3	93.6	78.9

Table 6.16.: Neural network with the 2D finder as input compared to the 3D finder as input. The test data set contains low-p_{T} and shallow-θ tracks originating from the IP (data set ZIP in Tab. 6.6). The results are sorted by Δz_{MC}.

The second study is carried out with the test data set ZIP (see Tab. 6.6) which contains low-p_{T} and shallow-θ tracks at the geometrical boundary regions of the CDC acceptance. In this range the 3D finder can achieves nearly twice the efficiency of the 2D finder (see Sec. 5.3.4). Due to these additionally found tracks in the low-p_{T} and shallow-θ region, deteriorated averaged resolutions are expected with the 3D finder as input. In contrast, the 2D finder has a significantly lower efficiency for these tracks (compare Fig. 5.29).

Table 6.16 shows the averaged efficiencies and resolutions achieved with both finders as input to the neural network in a test run on the enlarged test data range ZIP. As expected from the increased efficiency of the 3D finder, a comparison of the neural network resolutions Δz and $\Delta\theta$ shows apparently better resolutions with the 2D finder than with the 3D finder. However, at the same time, the total trigger efficiency $\epsilon_{\text{Trg}}^{\text{MC}}$ is significantly improved with the 3D finder. The achieved neural network resolutions in the full CDC acceptance region have already been verified to be comparable with both finders (see Tab. 6.15). Therefore, the reason for the reduced average resolution with the 3D finder as input can definitely be attributed to the additional low-p_{T} and shallow-θ tracks that are only found by the 3D finder. Since these tracks typically have smaller numbers of related TSs, the average resolution on the test data sets with the enlarged range ZIP (see Tab. 6.6) matches the expectation.

For all background types, the 3D finder efficiency is about $\epsilon_{\text{3D Finder}}^{\text{MC}} \approx 93\,\%$ while the 2D finder has only $\epsilon_{\text{2D Finder}}^{\text{MC}} \approx [55 - 59]\,\%$. Although the total trigger efficiency ϵ_{Trg} in Tab. 6.16 is still increased with the 3D finder, the huge efficiency improvement of the 3D finder is now limited by the efficiency of the neural network. The neural network efficiency without background is $\epsilon_{\text{NN}}^{\text{MC}} \approx 72\,\%$ and increases up to

$\epsilon_{\mathrm{NN}}^{\mathrm{MC}} \approx 84\,\%$ with the Phase 3 background 15p3. In contrast, the neural network efficiency with the 2D finder as input is close to 100 % with all background types. This increase of the neural network efficiency with background is a result of the increased occupancy in the CDC and the related increased number of TSs (see Chapter 4). A comparison with Tab. 6.15 shows that this reduced efficiency of the neural network is only observed in the test data containing additional low-p_{T} and shallow-θ tracks. In the used five-expert neural network, only those tracks with less than three stereo TSs are discarded. Since the 2D finder requires at least four axial TSs, tracks found by the 2D finder usually have at least three stereo TSs from the inclined stereo SLs. Hence, this limited ϵ_{NN} only appears with the 3D finder as input.

This highlights an area that can be improved upon: if the neural network were able to handle all the tracks additionally found by the 3D finder, an improvement of the full trigger efficiency ϵ_{Trg} by almost a factor of 2 can be expected. Therefore, it is important to optimize the neural network for tracks with low-p_{T} and shallow-θ angles. The corresponding study is presented later in Sec. 6.5.2.

More details on the achievable resolutions $\Delta\theta$ and Δz within the large range ZIP are shown in Fig. 6.21. The resolution loss with increasing background levels can be seen with both finders. As functions of p_{T}, shown in Fig. 6.21 (a) and (b), the resolutions are better in the high-p_{T} region than in the low-p_{T} region. In agreement with the limited data range seen before in Fig. 6.20, in the high-p_{T} region improved neural network resolutions are observed with the 3D finder as input. Since the resolutions are only calculated in bins containing more than 500 matched track pairs, the reduced 2D finder efficiency appears as shorter resolution curves in the low-p_{T} and shallow-θ region. This reaffirms that the average resolutions shown in Tab. 6.16 are dominated by the low-p_{T} and shallow-θ tracks. Note that the test data was simulated with a uniform distribution in p_{T}^{-1}. Since the 2D finder does not find most of these low-p_{T} tracks, they do not enter the averaged resolution values.

The same holds for the resolution plots as functions of the polar angle θ, where the deteriorated resolution with the 3D finder is corresponds to the low-p_{T} tracks in the test data. An interesting finding in Fig. 6.21 (c) and (d) is the small peak for $\theta \approx 90^{\circ}$ propagated from the finder resolutions to the neural network resolutions. According to the previous 3D finder studies in Sec. 5.3.4, this peak corresponds to low-p_{T} curl-back tracks in the CDC.

In order to confirm the effect of the low-p_{T} and shallow-θ tracks, Fig. 6.22 shows the efficiencies of the track finding, the neural network and the full trigger as functions of p_{T} and θ. The central column shows the finder efficiencies $\epsilon_{\mathrm{Finder}}$. Huge improvements can be observed with the 3D finder compared to the 2D finder, while the different background levels have only a minor impact on the finder efficiencies (compare Sec. 5.3.4). In the low-p_{T} region in Fig. 6.22 (b) and at shallow polar angles θ in Fig. 6.22 (e) the 3D finder efficiency is significantly improved.

The left column in Fig. 6.22 shows the efficiency of the neural network (ϵ_{NN}). While the efficiency is close to 100 % with the 2D finder as input for all background types, it varies from $\epsilon_{\mathrm{NN}} \approx 70\,\%$ to 90 % with the 3D finder for the different back-

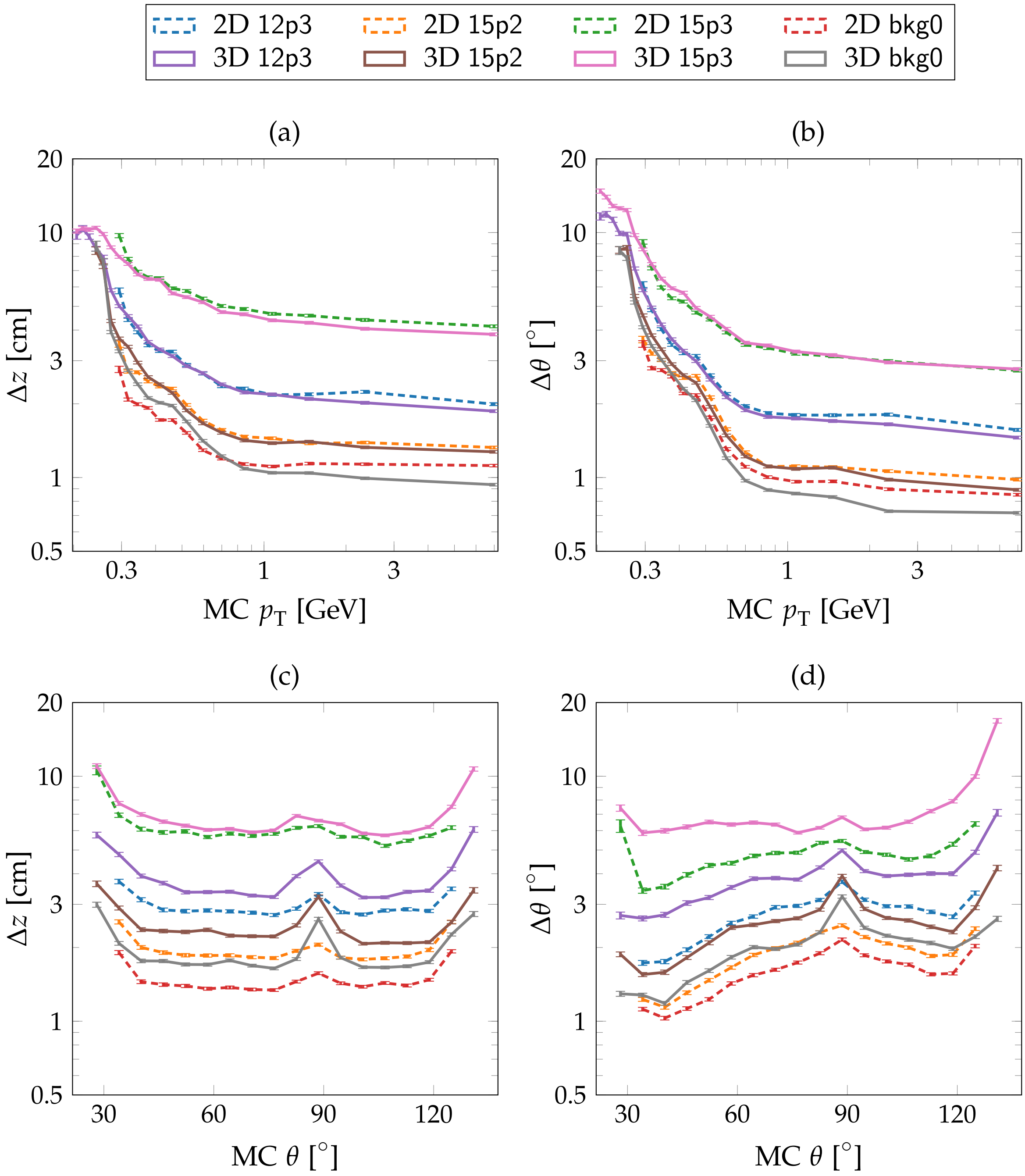


Figure 6.21.: Resolution of the neural network with the 2D finder compared to the 3D finder as input using the test data ZIP from Tab. 6.6.

ground levels. This is plausible, as the larger background levels create additional hits and thus increase the chance for the tracks to be processed. Note that at least three stereo SLs containing TSs are required with the used five-expert neural network.

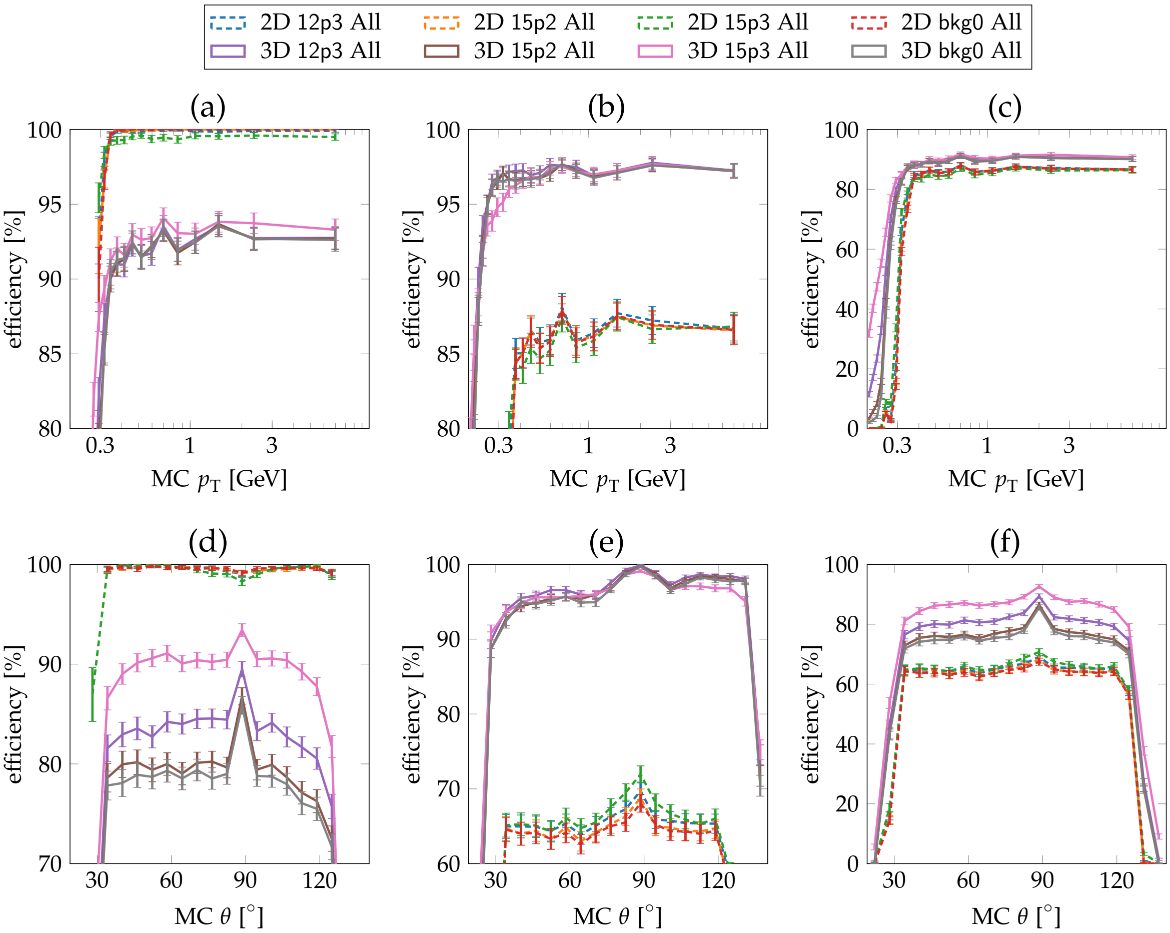


Figure 6.22.: Comparison of the trigger track efficiencies with the 2D finder compared to the 3D finder for single tracks from the IP. The track finder efficiencies ϵ_{Finder} are shown in the central column, the neural network efficiencies ϵ_{NN} in the left column, and the full trigger efficiencies ϵ_{Trg} in the right column.

Like the finder efficiencies, the neural network efficiencies with the 3D finder input have a peak close to $\theta \approx 90^\circ$. This apparently improved efficiency is created by curl-back tracks with low-p_T and is thus related to the resolution loss at $\theta \approx 90^\circ$ in Fig. 6.21. All efficiencies are almost flat in ϕ and therefore not included in the plot. The flatness in ϕ indicates that the asymmetric background distribution in ϕ does affect neither the finder nor the neural network efficiency.

Beamline Tracks

This study is carried out on a test data set with range Z50 that contains single tracks uniformly distributed along the z-axis. Furthermore, low-p_T tracks and shallow-θ tracks are included. Hence, this study demonstrates the structure of the 3D finder efficiency for displaced tracks. The same five-expert neural network is used as in the previous IP track study.

$\mathtt{Bkg}_{\mathrm{test}}$	`Finder`	$\Delta\theta_{\mathrm{MC}}$	Δz_{MC}	$\epsilon_{\mathrm{NN}}^{\mathrm{MC}}$	$\epsilon_{\mathrm{Finder}}^{\mathrm{MC}}$	$\epsilon_{\mathrm{Trg}}^{\mathrm{MC}}$
		[°]	[cm]	[%]	[%]	[%]
bkg0	2D	2.5	2.0	99.5	52.0	51.7
bkg0	3D	2.9	2.1	77.0	75.8	58.3
15p2	2D	2.7	2.5	99.5	52.5	52.3
15p2	3D	3.3	2.7	77.7	76.6	59.5
12p3	2D	3.6	3.4	99.4	53.5	53.1
12p3	3D	4.4	4.0	80.3	78.5	63.0
15p3	2D	5.7	6.3	98.0	57.6	56.4
15p3	3D	7.4	7.7	84.4	84.6	71.4

Table 6.17.: 2D finder input tracks are compared to 3D finder input tracks. The data set contains tracks uniformly distributed along the z-axis (dataset Z50 in Tab. 6.6). The results are sorted by Δz_{MC}.

Table 6.17 summarizes the resolutions and efficiencies observed in this study. As before with the IP tracks in Tab. 6.16, the resolutions with the 2D finder are slightly better than the resolutions with the 3D finder. As expected, the average resolutions are getting worse for the tracks displaced from the IP shown here. The average finder efficiencies are still improved with the 3D finder compared to the 2D finder. However, the improvement became smaller than in the previous studies with IP tracks. This indicates that the 3D finder is indeed more efficient for tracks from the IP than for displaced tracks, which is beneficial as these displaced tracks should not be triggered. The neural network efficiencies ϵ_{NN} are similar to the efficiencies for IP tracks in Tab. 6.16: with the 2D finder they are also close to 100 %, while the efficiency varies from 77 % to 84 % for the 3D finder with the different background levels. Due to the decreased 3D finder efficiency for the tracks displaced from the IP, the improvement in the total trigger efficiency seems to be much smaller here (compare Tab. 6.16).

Figure 6.23 shows all the relevant efficiencies with this data set and can be compared to the IP efficiencies in Fig. 6.22. The last row shows the new efficiencies as functions of z. Of special interest is the distribution of the 3D finder efficiency in Fig. 6.23 (h). It shows a very good efficiency of $\epsilon_{\mathrm{Finder}} \geq 90\,\%$ for small values of $|z|$, i.e. for tracks from the IP. With larger distances from the IP, the 3D finder efficiency drops significantly by almost a factor of 2 for a z-distance of $z = 50\,\mathrm{cm}$ from the

IP. In comparison, the 2D finder efficiencies are almost flat. This means that tracks displaced from the IP are found by the 2D finder just as well as tracks from the IP.

Since this data set contains tracks uniformly distributed along the z-axis, the 3D finder efficiencies as functions of the other track parameters $(p_\mathrm{T}, \phi, \theta)$ are down-weighted by these displaced tracks. In the high-p_T region in Fig. 6.23 (b), the 2D finder and the 3D finder apparently achieve the same efficiency.

For the neural network efficiency as a function of z, an efficiency drop can be observed for tracks from the IP in Fig. 6.23 (g), which becomes smaller with larger amounts of background. This agrees with the background dependence of the neural network efficiencies observed for IP tracks.

With the displaced vertices in this test data sets, the chance for curling low-p_T tracks with $\theta \approx 90^\circ$ is reduced. For this reason, the peak previously observed in Fig. 6.22 (d), (e), and (f) for tracks with $\theta \approx 90^\circ$ vanishes here in the plots Fig. 6.23 (d), (e), and (f).

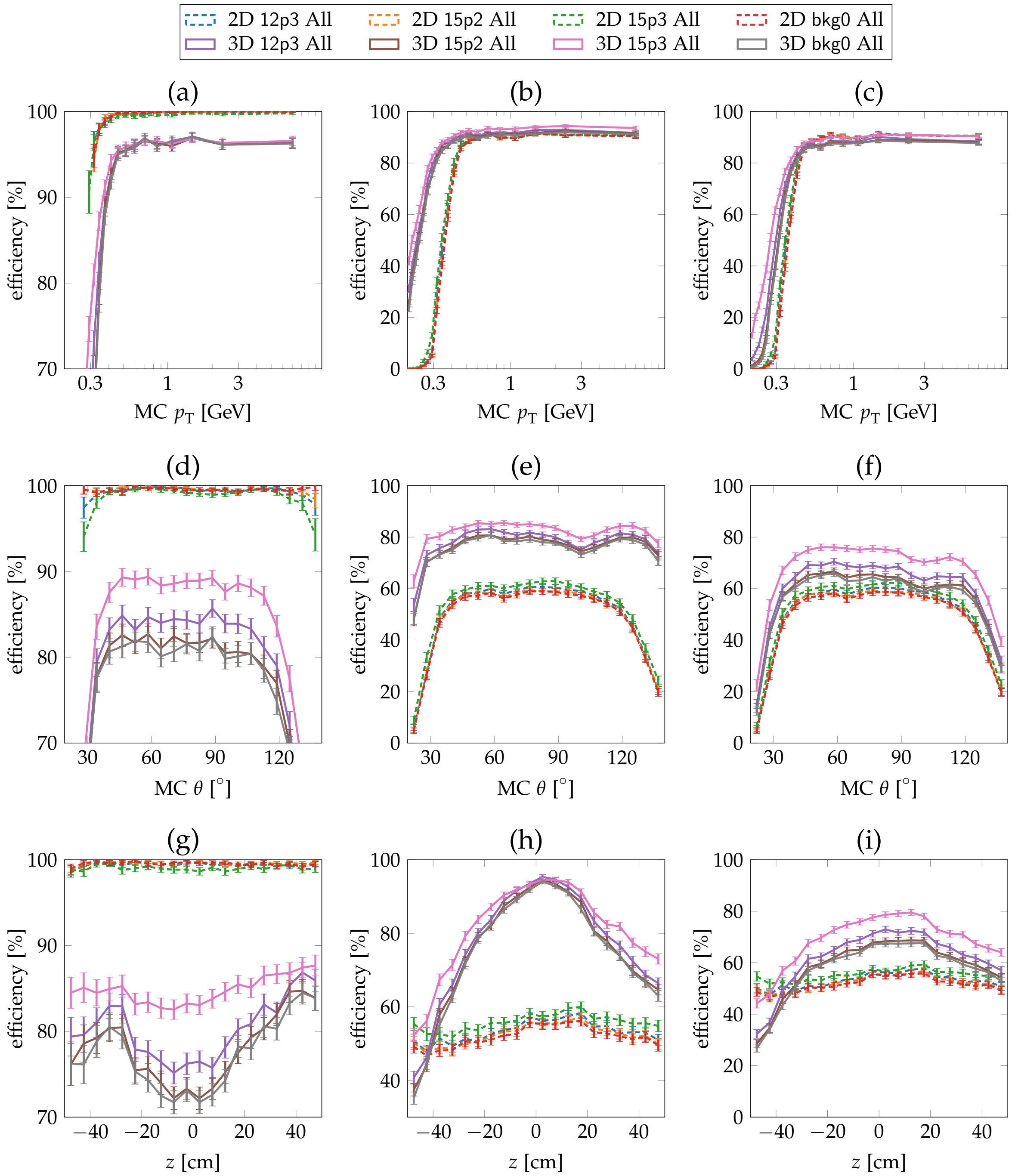


Figure 6.23.: Neural network, track finder and the total trigger efficiencies with the 2D and 3D finders tested on single tracks displaced along the z-axis with four different background types (test data ranges Z50). Preliminary versions of (h) and (i) have been presented at [109].

Track Finding Rate at Large Displacements

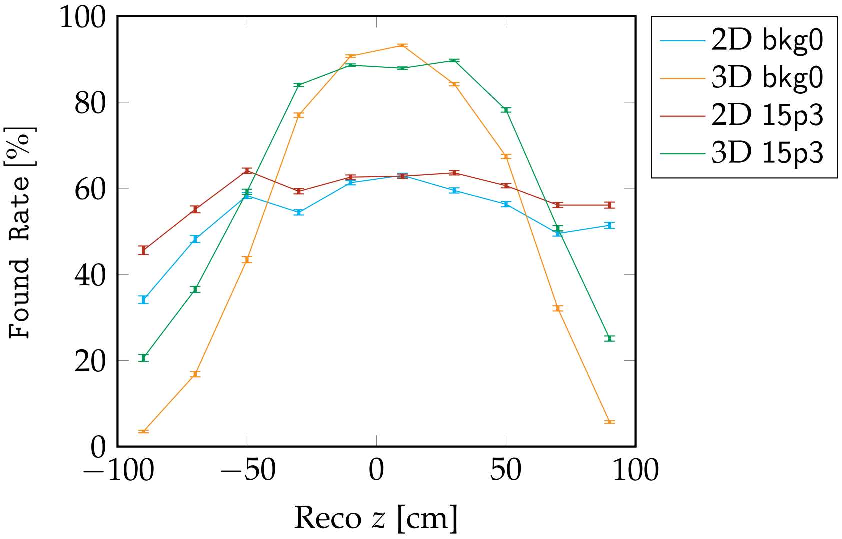


Figure 6.24.: Track finding efficiencies of the 2D finder and the 3D finder in a different test data set with where tracks are simulated uniformly in $z \in [-100, 100]$ cm. As reference values, all reconstructed tracks are selected which have a $p_T \geq 0.25$ GeV. This test data set contains low-p_T and shallow-θ track, using the track parameter ranges $p_T \in [0.2, 10.2]$ GeV and $\theta \in [19, 140]^\circ$. Note that no matching with the simulated MC tracks is required in this study.

In order to provide a thorough demonstration of the 3D finder background suppression of displaced tracks, a new test data set is added and the achievable track finding rates are illustrated in Fig. 6.24. This test data set is simulated with tracks uniformly distributed in the range $z \in [-100, 100]$ cm. Low-p_T and shallow-θ tracks are included as indicated in the caption text of Fig. 6.24. For the calculation of the `Found Rate`, reconstructed tracks are used rather than simulated MC tracks. This means, the `Found Rate` is the number of tracks found by the track finder per reconstructed track. In general, for tracks with larger displacements along the z-axis, reconstructed tracks should be preferred of MC tracks. Especially the heavy material in the crotch part between the two beamlines results in deterioration of the simulated track parameters, which can be compensated by using reconstructed tracks.

Fig. 6.24 demonstrates that the 3D track finding rate for tracks with displacements larger than 50 cm indeed falls below the track fining rate of the 2D finder. In contrast, the `Found Rate` of the 2D finder is almost flat along z. Only for values of $z \leq -60$ cm, the `Found Rate` of the 2D finder is reduced due to the geometrical acceptance region of the CDC. Note that the inverse of the `Found Rate` for displaced values of z corresponds to the background suppression rate. In contrast, the `Found`

Rate at the IP corresponds to the track finding efficiency for signal tracks. As noted before in the discussion of Fig. 6.23 (h), the track finding efficiency of the 3D finder at the IP is significantly larger than the track finding efficiency of the 2D finder. At a distance of $|z| \approx 50\,\text{cm}$ the Found Rate of the 3D finder is already reduced by a factor of 2 compared to the 3D finder efficiency at the IP. For larger displacements of $|z| \gtrapprox 80\,\text{cm}$ the Found Rate of the 3D finder, in the case without background, almost goes to zero. This means, the 3D finder realizes an ideal background suppression for tracks with large displacements in z. As will be shown in Sec 7.2.2, a large fraction of the background tracks is observed at such large distances.

Resolution for Displaced Track

Using again the test data sets Z50, the resolutions for the displaced tracks as functions of the MC 2D track parameters p_T and ϕ are shown in Fig. 6.25. The corresponding resolutions and function of the 3D track parameters θ and z are shown in Fig. 6.26. As functions of p_T and θ, the resolutions show a similar structure as observed in test data set with IP tracks in Fig. 6.21. The resolutions as functions of ϕ show the typical background dependent resolution loss with a broad peak pointing in positive x-direction (around $\phi \approx 0^\circ$.

New are the resolutions as functions of z in Fig. 6.26 (a) and Fig. 6.26 (c). As expected, the resolutions are best in the vicinity of the IP ($z \approx 0\,\text{cm}$). However, the structure of $\Delta\theta(z)$ in Fig. 6.26 (c) and $\Delta z(z)$ Fig. 6.26 (a) differs. With lower background levels, the resolution $\Delta z(z)$ is almost flat until $|z| \approx 30\,\text{cm}$ and decreases strongly for values of $|z| \geq 40\,\text{cm}$. In $\Delta\theta(z)$ the situation with low background levels differs: the resolution drops almost linearly with the distance from the IP and flattens out for larger distances. With larger background levels (15p3) the z-dependence of both resolutions decreases continuously with an increasing distance from the IP.

This means that with larger background levels, tracks with a larger displacement z from the IP are more difficult for the neural network to estimate. Since the goal of the trigger is to improve the efficiency of signal tracks from the IP, this does not affect the signal efficiency. However, it could affect the purity of the triggered events, as displaced background tracks become more likely to be estimated falsely as signal tracks.

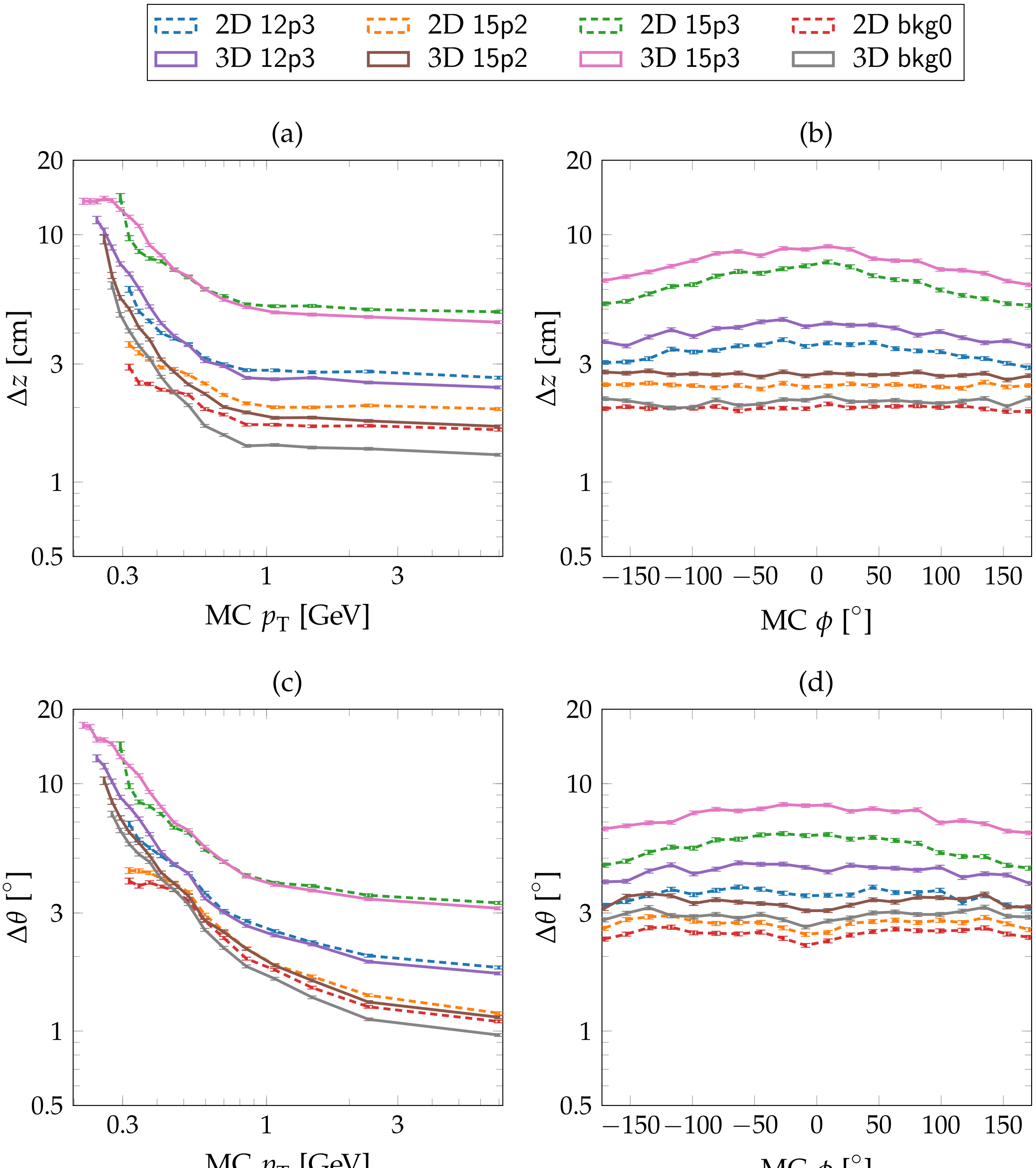


Figure 6.25.: Resolutions of the neural network with the 2D finder compared to the 3D finder as input as functions of the MC 2D track parameters p_T and ϕ. The test data sets Z50 is used with four different background type.

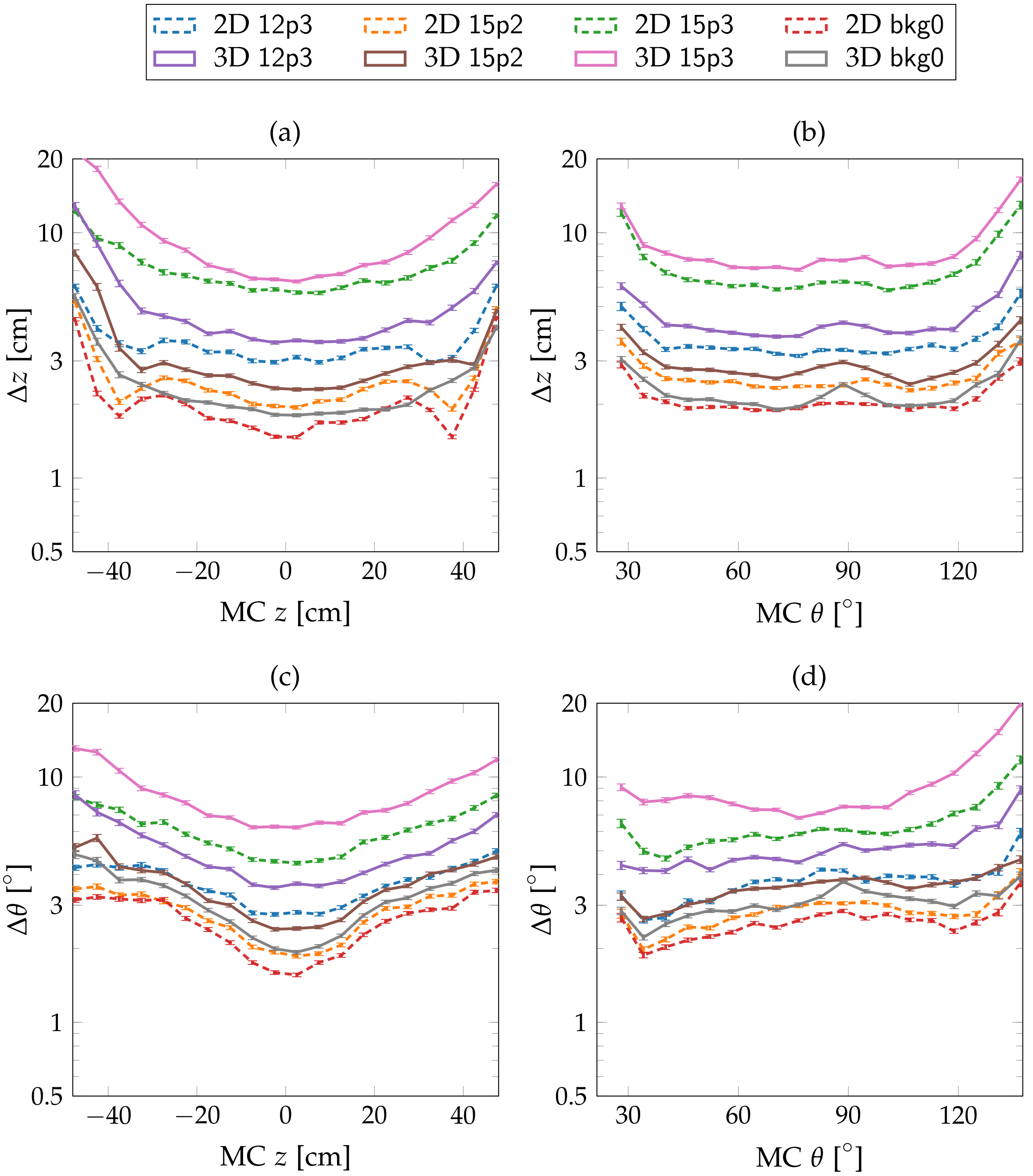


Figure 6.26.: Resolutions of the neural network with the 2D finder compared to the 3D finder as input as functions of the MC 3D track parameters θ and z. The test data sets Z50 is used with four different background type.

Pattern Rates

In analogy to the pattern rate and z-cut studies with the 2D finder in Sec. 6.4.6, here the corresponding 3D finder results are presented. The used neural network in this study is trained with a training data set of size NND (see Tab. D.1) and the test data set contains single tracks uniformly distributed along the z-axis (test ranges Z50).

	$\mathrm{Bkg}_{\mathrm{test}}$	$\Delta\theta_{\mathrm{MC}}$	Δz_{MC}	$\epsilon^{\mathrm{MC}}_{\mathrm{NN}}$	$\epsilon^{\mathrm{MC}}_{\mathrm{Finder}}$	$\epsilon^{\mathrm{MC}}_{\mathrm{Trg}}$
		[°]	[cm]	[%]	[%]	[%]
(a)	bkg0	2.9	2.1	77.0	75.8	58.3
	15p2	3.3	2.7	77.7	76.6	59.5
	12p3	4.5	4.1	80.3	78.5	63.0
	15p3	7.4	7.7	84.4	84.6	71.4

	$\mathrm{Bkg}_{\mathrm{test}}$	SL1	SL3	SL5	SL7	p7	p11	p13	p14	p15
		[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]
(b)	bkg0	99.9	100.0	99.7	74.6	0.1	0.0	0.3	25.4	74.2
	15p2	99.9	100.0	99.5	73.5	0.1	0.0	0.5	26.5	72.9
	12p3	99.9	100.0	98.8	73.7	0.1	0.0	1.2	26.3	72.3
	15p3	99.8	99.9	97.2	76.1	0.2	0.1	2.8	23.9	73.1

	$\mathrm{Bkg}_{\mathrm{test}}$	Gain1	Loss1	Fake1
		[%]	[%]	[%]
(c)	bkg0	99.2	0.3	1.2
	15p2	98.4	0.6	2.1
	12p3	97.9	1.3	4.9
	15p3	94.1	4.1	16.4

Table 6.18.: Results of the neural network with the 3D finder as input, tested on four different background types on tracks in the range Z50 (see Tab. 6.6. The pattern rates of the expert networks and the z-cut based rates Gain, Loss and Fake are included. The results are shown for a network trained with training data size NND (see Tab. D.1).

The summarized results for the pattern rates and single track z-cuts with the 3D finder are shown in Tab. 6.18. Except for the used finder, all conditions are identical to the study in Sec. 6.4.6. Hence, the results can directly be compared to Tab. 6.13. The first table, Tab. 6.18 (a), shows the resolutions and the efficiencies. A comparison to the 2D finder for the resolutions and efficiencies has already been presented in Sec. 6.5.1 with a very similar network and is not repeated here for network with a slightly smaller training data size (nTrain $=$ 500 here instead of nTrain $=$ 1000).

Table 6.18 (b) shows the rates of TSs per track per SL for the neural network tracks with the 3D finder as input (SL-rates) and the rates pattern-rates corresponding to the runtime share of each expert network. For the SL hit rates, the main difference to the 2D finder is in SL7. The rate with the 3D finder of SL7 $\approx$ 74 % is lower compared to SL7 $\approx$ 89 %) with the 2D finder. This larger fraction of processed tracks without a

TS hit in the outermost stereo SL, is related to the improved 3D finder efficiency for low-p_T tracks. This agrees with the observed pattern-rates: the pattern p15, where all stereo TSs are present, is used less often with the 3D finder. Consistently with both finders, the experts with the patters p7, p11, p12 are almost never in use.

The third table in Tab. 6.18 (c) shows the z-cut rates Gain1, Loss1, Fake1 with the 3D finder (for a description see Sec. 6.4.6). The Gain1-rate is close to 100 %, but decreasing with an increased background levels. Compared to the case with the 2D finder in Tab. 6.13, there is no difference in Gain1, while Loss1 and Fake1 are both twice as high with the 3D finder. This is again a result of the increased efficiency of the 3D finder. With almost double the amount of tracks found by the 3D finder, the neural network has to process more difficult low-p_T and shallow-θ tracks where the accuracy of the estimation is expected to be reduced. This is consistent with the observed pattern-rates: in Tab. 6.18 (b) the pattern-rate p14 is about twice as large as with the 2D finder in Tab. 6.13 (b). This means that the amount of processed tracks with only three instead of four stereo SLs is doubled. Since more short tracks are processed by the network, this increase in Loss1 and Fake1 is reasonable.

6.5.2. Neural Network Short Track Experts

In the previous studies with the 3D finder as input to the neural network, the neural network efficiency was reduced to 72 % without background and 84 % with the background 15p3. This had the effect that the full power of the improved 3D finder efficiency was additionally reduced by the neural network. The main source of this problem was identified as so-called short tracks, which have not enough stereo TSs to be processed by the neural network. These short tracks may either be tracks with low-p_T or tracks with shallow polar angles θ.

In order to overcome this deficit, new short track expert networks, specialized to tracks with low-p_T and shallow-θ, are introduced in this section. In the previously used five-expert networks, only a single stereo TS was permitted to be missing. The obvious extension is to include expert networks where two stereo TSs are permitted to be missing. A full coverage of all possible combinations with two missing stereo TSs would introduce six new expert networks. However, the previous pattern-rate study showed that those combinations which are not accessible for normal tracks from the beamline are indeed almost never used (patterns `p7, p11, p13`). In fact, only the two outer SLs can be missing for ideal low-p_T or shallow-θ tracks if detector inefficiencies are neglected. This means that already a single additional expert is expected to provide a substantial efficiency improvement with the 3D finder input: an expert for the case when two outer stereo SLs are missing.

From the mathematical track reconstruction perspective, two stereo TSs combined with the 2D track parameters are sufficient to allow for a 3D track reconstruction. With a known 2D track from the track finder, each stereo TS hit can be transformed into a single point in the s, z-plane, where s is the arc length along the 2D track (see Sec. 6.1.4). . The 3D track reconstruction carried out by the neural network corresponds to the line fit through the stereo TS hit points in this s, z-plane. Since two points are the minimum requirement to define a line in a 2D space, two stereo TSs, combined with the 2D track parameters, are sufficient for the 3D track reconstruction. Of course, additional stereo TSs improve the accuracy and a loss in resolution is expected from a two stereo TS 3D track reconstruction. Furthermore, additional axial TSs are required in order to determine the 2D track parameters. For more details on the fitting mechanism, see Sec. 6.1.

Short Track Network Settings

The main goal of this study is to achieve an improved efficiency for low-p_T and shallow-θ tracks. However, possible detector inefficiencies can also be compensated by additional expert networks as well. In order to keep the overhead of the additional experts small, only three out of the six possible two stereo TSs experts are added: in the new experts the outermost SL is always missing, while the three combinations with missing inner stereo TSs are added as new experts. The list of the full eight-expert patterns, including the inclined axial TSs, is shown in Tab. 6.19. It is an extension of the five-expert pattern definition in Tab. 6.2. Using this ex-

parameter	default	description
`SLpatternMask`	[010101010]	SL TS hit expert pattern mask.
`SLpattern`	[111111111] [101111111] [111011111] [111110111] [111111101] [101011111] [101110111] [101111101]	SL TS hit expert patterns.

Table 6.19.: Sectorization parameters of the neural network trainer for short track expert networks. The TS hit patterns are to be read from right to left to get the order of the SL from 0 to 8 (i.e. SL8, SL7, . . . , SL0). The corresponding five-expert pattern parameters are listed in Tab. 6.2

pert configuration, the eight-expert network is trained using the train data size NND (see Tab. D.1). In order to deploy an updated configuration of expert networks to the hardware, the selection of the experts should be properly revised to meet the available resources in the FPGA hardware. An eight-expert network setup is even expected to fit into present UT3 hardware.

binary	decimal	column name	description
1111	15	`p15`	all SLs present
1110	14	`p14`	SL7 missing
1101	13	`p13`	SL5 missing
1011	11	`p11`	SL3 missing
0111	7	`p7`	SL1 missing
0110	6	`p6`	SL1 and SL7 missing
1010	10	`p10`	SL3 and SL7 missing
1100	12	`p12`	SL5 and SL7 missing

Table 6.20.: Extended list of all eight-expert pattern rates. The three new experts `p6`, `p10`, and `p12` require only two stereo TSs. The binary patterns describe only the stereo TSs and have to be read from left to right (i.e. SL1, SL3, SL5, SL7). For completeness, the full list of patterns is shown here, including the patterns already listed in Tab. 6.11.

To study the different properties of the low-p_T expert networks, three additional columns are introduced for the three new experts patterns. An overview of all the eight patterns is shown in Tab. 6.20. The three new patterns for the cases with two missing stereo TSs are `p6`, `p10` and `p12`.

Improved Short Track Results

Similar to the previous pattern study results in Tab. 6.18, the summary tables with the 3D finder as input to the short track expert networks is listed Tab. 6.21. All test runs are carried out with the already known four background types (see Chapter 4). However, this time a different test data set containing only tracks from the IP is used (ZIP). Since tracks displaced from the IP are required to calculate `Loss1` and `Fake1`, out of the z-cut rates, only `Gain1` can be shown here. In addition, a table with the 3D track parameter resolutions is shown for the frequently used expert networks in Tab. 6.21 (c).

A consistency test with a 2D finder as input to the short track expert networks in not shown here. Since the 2D finder requires a minimum of four axial TSs per track, almost no short tracks with less than three stereo TSs are found. As expected with the 2D finder, only the already known experts for three and four stereo TSs are used leading to the same results as with a five-expert setup.

	$\mathrm{Bkg}_{\mathrm{test}}$	$\Delta\theta_{\mathrm{MC}}$	Δz_{MC}	$\epsilon^{\mathrm{MC}}_{\mathrm{NN}}$	$\epsilon^{\mathrm{MC}}_{\mathrm{Finder}}$	$\epsilon^{\mathrm{MC}}_{\mathrm{Trg}}$	Gain1
		[°]	[cm]	[%]	[%]	[%]	[%]
(a)	bkg0	3.4	2.8	98.5	92.8	91.5	98.2
	15p2	3.9	3.6	98.6	93.1	91.7	97.3
	12p3	5.1	4.7	98.8	93.7	92.5	96.4
	15p3	7.7	6.9	99.0	93.6	92.6	93.9

	$\mathrm{Bkg}_{\mathrm{test}}$	SL1	SL3	SL5	SL7	p7	p11	p13	p14	p15	p6	p10	p12
		[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]
(b)	bkg0	99.9	99.9	72.8	51.9	0.0	0.0	0.2	21.0	51.7	0.0	0.1	27.0
	15p2	99.9	99.9	73.5	51.7	0.0	0.0	0.3	22.0	51.3	0.0	0.1	26.2
	12p3	99.9	99.8	76.2	54.5	0.0	0.0	1.1	22.7	53.3	0.0	0.2	22.7
	15p3	99.9	99.7	81.4	61.8	0.0	0.1	2.6	22.0	59.1	0.0	0.2	16.0

	$\mathrm{Bkg}_{\mathrm{test}}$	Δz15	$\Delta\theta$15	Δz14	$\Delta\theta$14	Δz12	$\Delta\theta$12
		[cm]	[°]	[cm]	[°]	[cm]	[°]
(c)	bkg0	1.3	1.3	3.0	3.8	6.5	8.0
	15p2	1.7	1.7	4.3	4.4	7.1	8.6
	12p3	2.7	2.5	5.8	6.1	7.8	10.0
	15p3	5.4	4.6	8.7	10.0	8.8	14.6

Table 6.21.: Pattern-rates and TS hit rates per SL per track with the low-p_{T} expert networks, when using the 3D finder as input. The test data contains single muon tracks from the IP. The new short track patterns are shown in the columns `p6`, `p10`, and `p12`. For the three frequently used expert networks (patterns `p15`, `p14`, `p12`), the θ and z-resolutions are shown in (c). All resolution errors are in the order of 1 %.

The advantages of using the short track experts with the 3D finder as input can be seen in Tab. 6.21. The first table Tab. 6.21 (a) shows the resolutions, efficiencies, and the z-cut based signal efficiency Gain1. A significant increase of the total trigger efficiencies ϵ_{Trg} can be observed. All trigger efficiencies ϵ_{Trg} are now strictly larger than 91 %. This corresponds to an efficiency increase of 20 % to 30 % compared to the five-expert setup. As previously shown in Tab. 6.16 in a five-expert setup with the 3D finder as input ϵ_{Trg} is limited to 67 % without background and up to 79 % with 15p3.

This huge increase in efficiency comes along with only a little loss in the average resolution over all the experts (compare Tab. 6.16). Therefore, the z-cut rate Gain1 is quite similar to the previous case shown in Tab. 6.18. Note that the Gain1-rate in Tab. 6.18 calculated on the data set Z50 can be used for a comparison because a cut on the IP is carried out for the calculation of Gain1 (see Tab. 6.12).

The reason for the observed efficiency improvement can be identified in the hit rates and in the pattern-rates shown in Tab. 6.21 (b). Only the innermost two stereo SLs (SL1, SL3) are hit by almost every track. The trained experts for the case of only two stereo TSs have the effect that more tracks with missing stereo TSs in the outer SLs are processed. The SL5 TS hit rate in the estimated tracks is SL5 $\approx$ 71 % without background and up to SL5 $\approx$ 81 % with the background 15p3. Similarly, in the outermost stereo SL, the hit rate without background is further reduced to SL7 $\approx$ 52 % and raises only to SL7 $\approx$ 62 % with 15p3.

This agrees with the observed pattern-rates, where in particular the new expert pattern p12 processes a substantial amount of the tracks. Without background the short track pattern p12 is used for 27 % of the processed tracks. With the background type 15p3, this rate is decreasing to p12 $\approx$ 16 %. The rate p12 corresponds to the expert network where both outer SLs are missing (SL5, SL7) and only the two inner stereo SLs are present (SL1, SL3) (see Tab. 6.20). The rates of the short track expert networks with missing intermediate stereo TSs (p6, p10) are very low.

Interestingly, the pattern rate p15 shows a small increase with larger background levels, p14 seems to be constant for all background levels while p12 is decreasing. This is plausible since a larger background level introduces additional TS hits. Thus, a track with only two TSs plus background might appear as a three TS track. A track with three TSs plus background might appear as a four TS track. Therefore, p15 can only increase and p12 can only decrease. By simultaneously losing tracks to p15 and receiving tracks from p12 the rate of the three TSs tracks processed by p14 stays rather constant with changing background levels.

The comparison of these results with those in Tab. 6.16 in Sec. 6.5.1 is notable: without the additional short track experts, the trigger cannot make full use of the improved 3D finder efficiency. The bad efficiencies ϵ_{NN} of the five-expert neural network in Tab. 6.16 are now improved to almost 100 % with the eight-expert networks. Note that the observed average efficiency improvements applies to the simulated single tracks with a uniform distribution in p_{T}^{-1}. Even larger efficiency improvements can be expected for low-p_{T} events only or for tracks with a majority of shallow polar angles θ.

For those experts which are frequently used (i.e. p12, p14, p15), a closer look on the resolutions achieved by the separate experts is shown in Tab. 6.21 (c). Using the same numbering as for the pattern-rate labels, the experts corresponding to the resolutions $\Delta\theta$ and Δz in Tab. 6.21 (c) are identified by the integer in the label. As expected, with smaller number of TSs, the resolution decreases. However, the resolution achieved with the two stereo TS expert (12) is still comparable to the three stereo TS expert (14) in the case with background. Without background, the resolution is deteriorated by factor of 2. This important result demonstrates that cuts on the track parameters are still feasible with the resolutions achieved by the short track experts.

Figure 6.27 gives an overview on the average resolutions achieved with the eight-expert networks. It shows the short track expert networks, which are trained and tested with four different background types using the 3D finder as input. For reference, the resolutions can be compared to the five-expert resolution averages in Fig. 6.21.

In the high-p_T region in Fig. 6.27 (a), the resolutions are similar to the ones shown in Fig. 6.21 (a). The reason is that high-p_T tracks from the IP usually create TSs in all SLs. Therefore, the high-p_T tracks are processed by the same expert network (p15) as before in the five-expert setup. However, in the low-p_T region, the majority of the tracks in Fig. 6.27 (a) is now processed by the new two track expert (p12). In the low-p_T limit, a resolution of $\Delta z \approx 8\,\mathrm{cm}$ without background and up to $\Delta z \approx 10\,\mathrm{cm}$ in the case with the background 15p3 can still be achieved. The structure of $\Delta\theta(p_\mathrm{T})$ in Fig. 6.27 (b) is similar to $\Delta z(p_\mathrm{T})$.

In the resolution as functions of θ, in particular the tracks at shallow polar angles are processed by the new expert p12. Figure 6.27 (c) shows $\Delta z(\theta)$ and Fig. 6.27 (d) shows $\Delta\theta(\theta)$. The resolution loss for $\theta \approx 90^\circ$ is now stronger pronounced than before with the five-expert network in Fig. 6.21 (c) and (d). The reason is that a larger number of low-p_T curl back tracks can be processed by the additional expert network. Relative to the average resolution, this loss in resolution is largest without background and with the lightweight Phase 2 background 15p2.

Summary of the Short Track Experts

The additional short track experts turned out to provide a significant improvement of the track finding efficiency. In particular, the expert for two stereo TSs in the inner two stereo layers processes up to 30 % of the tracks in the tested data sets. Since less stereo information is used, the resolution of the two track expert is necessarily reduced. Compared to the three TSs expert without background, a resolution loss by a factor of two has been observed. This limited resolution with the two stereo TSs expert is still usable for the construction of trigger cuts. Since quite good detector efficiencies are assumed in the MC simulation, inner TSs are rarely missing. Therefore, all experts for missing inner stereo TSs are almost never used. The studies indicated that only three experts are really required for the cases: all stereo SLs present, SL7 missing, SL7 and SL5 missing.

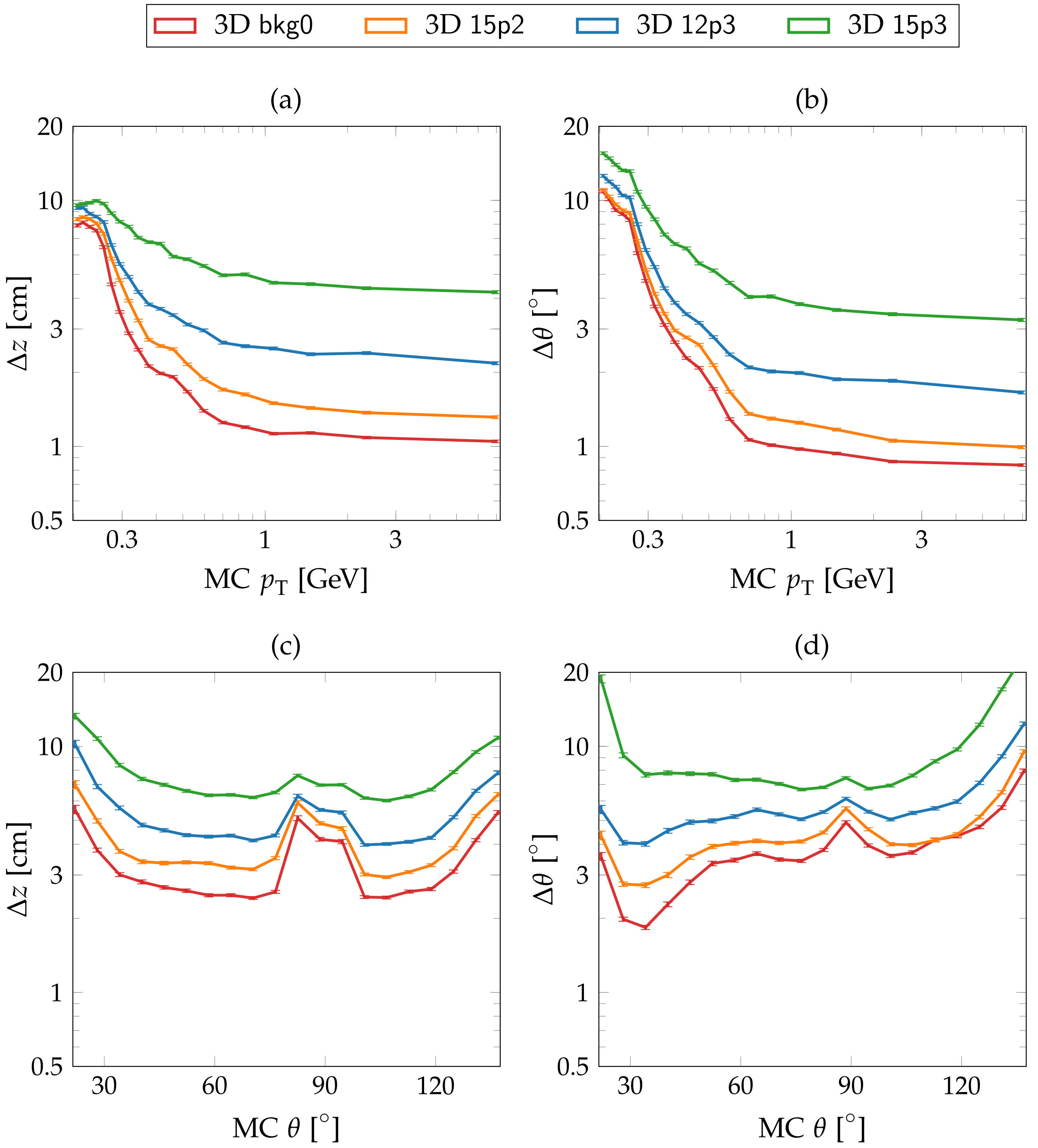


Figure 6.27.: Average resolution of the low-p_T and shallow-θ expert neural networks tested on tracks in the range ZIP with four different background types.

6.5.3. Neural Networks Theta Sector Experts

In this section the method of θ-sectorization is introduced, where expert networks are trained for a number of θ-sectors. The studies described so far only used the 2D track parameter estimates (p_T, ϕ) of the 3D finder to calculate the input vector for the neural network, while the additional θ-estimate provided by the 3D finder was unused. In this way, the neural network could only benefit from the improved resolution in the 2D track parameters and from the improved track finding efficiency of the 3D finder. By loading θ-sector specific expert neural networks, the θ-estimates of the 3D finder can be used in the preprocessing of the neural network.

Each θ-expert neural network is assigned to a limited sector in θ. By binning the θ-estimates of the 3D finder, a corresponding θ-expert network can be selected in the preprocessing. Within each θ-sector, the basic five-expert setup is used for missing stereo TSs. Note that the approach to use θ conceptually differs from the use of the 2D track parameters p_T and ϕ. Only p_T and ϕ are explicitly used in the calculation of the neural network input vector, in order to linearize the 3D reconstruction problem (see Sec. 6.2.4).

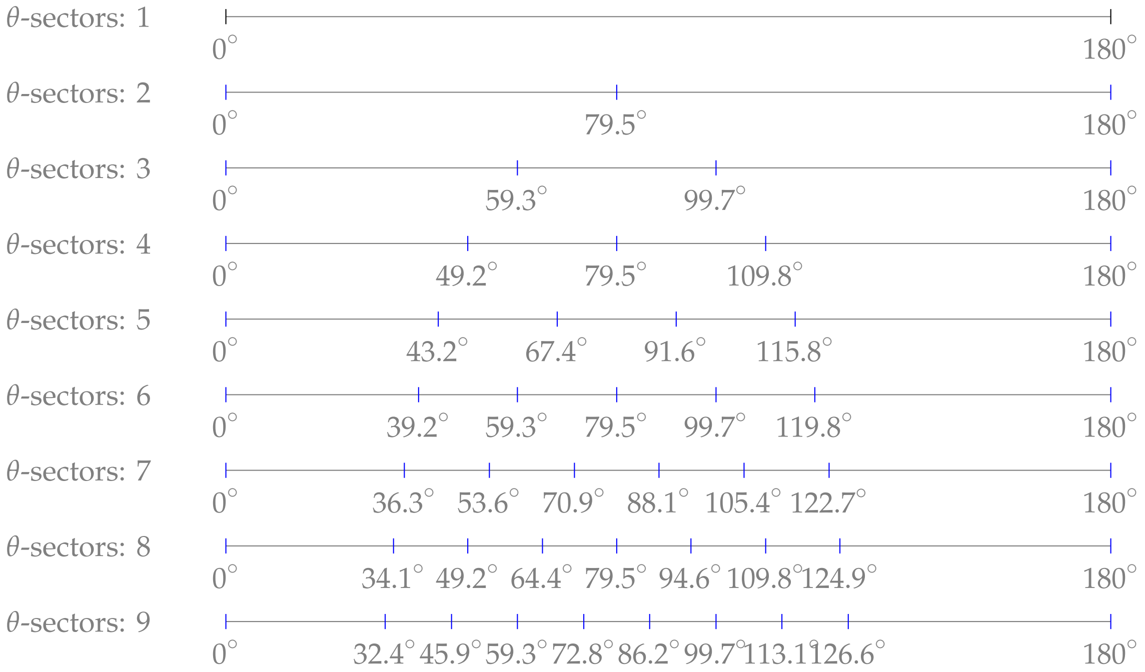


Figure 6.28.: Expert ranges with different numbers of sectors in the polar angle θ. For each of the 1 to 9 θ-sectors the usual five-expert networks are trained for missing stereo TSs.

Several neural network setups with a different number of θ-sector experts (from 1 to 9) are tested in this study. The defined ranges for each θ-sector in the tested neural network setups are listed in Fig. 6.28. Note that the case of a single θ-sector (first line in Fig. 6.28) is identical to the previously presented networks without θ-sectorization.

The sectorization parameters for the θ-expert training are obtained by setting the parameter `thetaRange` in Tab. 6.2 to the ranges listed in Fig. 6.28. Since a five-expert setup is used within each θ-sector, the total number of trained experts is the number of θ-sectors times five. Therefore, the training time increases significantly with larger numbers of θ-sectors. In order to avoid extensively long training times, only the two smallest train data sample sizes were used in this study (NNB in Tab. D.1). As an angular preserving example, the correct angles of the θ-sectors in a setup with nine θ-sectors are illustrated in Fig. 6.29. This setup corresponds to the last entry in the linearized illustration in Fig. 6.28.

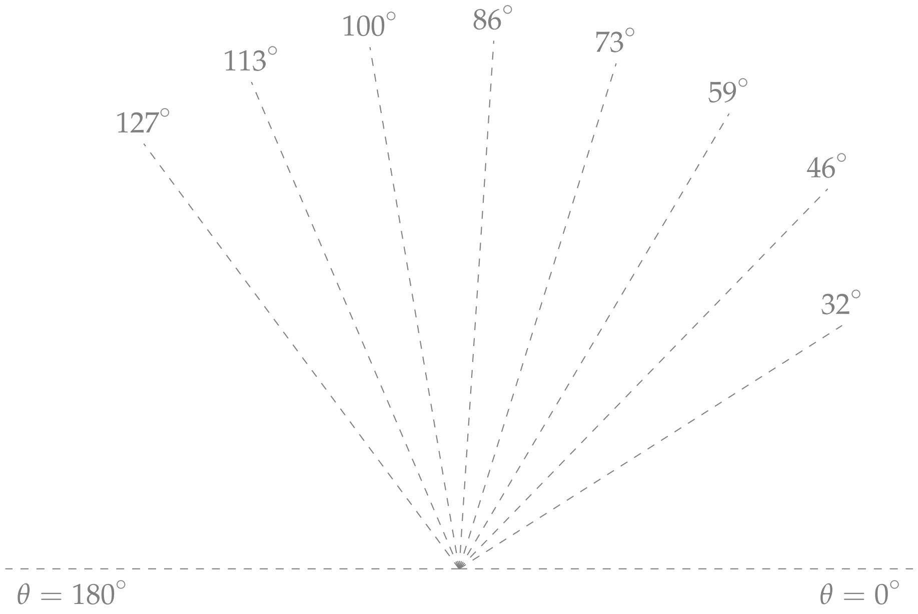


Figure 6.29.: Illustration of the sector borders for a neural network setup with nine expert sectors in θ.

The method of sectorization is implemented by dividing a θ-range into a number of bins. In the end, the full polar angle ($\theta \in [0°, 180°]$) is covered, in the same way as in the previous cases without θ-sectorization. This simplifies the comparison of the θ-sectorized networks and avoids an efficiency loss for tracks with shallow-θ angles. According to the trained 3D finder polar angle coverage of $[\theta_{\text{min}}, \theta_{\text{max}}] = [19°, 140°]$, at first the θ-sector ranges for the neural network are calculated uniformly within the same range. Once they are calculated, the outermost bins are extended to obtain the full coverage $\theta \in [0°, 180°]$.

In general, a full coverage of the polar angle, larger than the coverage of the CDC with $\theta \in [17°, 150°]$, is not required. On the contrary, no improvements are expected from a limited θ-coverage. By using the full θ-range in the θ-sectorization setup, errors in the θ-input due to a limited resolution of the 3D finder can be compensated. Furthermore, different 3D finders with different θ-ranges can easily be interchanged.

If the received θ-estimates are strictly contained within the range $\theta \in [19^\circ, 140^\circ]$, the exact same results are obtained as without enlarged outermost θ-expert sectors.

Note that the parameter `thetaRange` (see Tab. 6.2) configures the range of validity of each θ-expert, while the parameter `outputScale` (see Tab. 6.7 and Sec. 6.4.3) defines the range of output values that can be estimated by each neural network. In all studies presented in this thesis, the `outputScale` for θ is set to the full range of $\theta \in [0^\circ, 180^\circ]$. For the θ-experts presented in this section, this means that each θ-expert can estimate tracks with arbitrary polar angles $\theta \in [0^\circ, 180^\circ]$. In this way, wrong θ-estimates from the 3D finder still have a chance to be correctly estimated by the neural network. If the 3D finder θ-estimates are correct, the enlarged `thetaRange`

Performance of θ-Expert Networks

Bkg$_{\text{test}}$	θ-sectors	$\epsilon^{\text{MC}}_{\text{Finder}}$	$\epsilon^{\text{MC}}_{\text{Trg}}$	Δz_{MC}	$\Delta\theta_{\text{MC}}$	$\epsilon^{\text{MC}}_{\text{NN}}$
		[%]	[%]	[cm]	[°]	[%]
bkg0	X9	92.82	67.36	1.62	1.81	72.57
bkg0	X1	92.82	67.21	1.76	1.94	72.41
15p2	X9	93.08	68.59	2.20	2.27	73.69
15p2	X1	93.08	68.28	2.39	2.45	73.36
12p3	X9	93.71	72.16	3.26	3.29	77.00
12p3	X1	93.71	72.23	3.78	3.63	77.08
15p3	X9	93.60	77.84	5.76	5.48	83.17
15p3	X1	93.60	78.93	6.50	6.24	84.33

Table 6.22.: Results of the expert neural networks with one and with nine θ-sectors.

For this study, the trained networks are tested on tracks from the IP (range ZIP in Tab. 6.6) and the networks are tested with the same background type as used in the training. The resolutions and efficiencies as functions of the different numbers of θ-sectors are shown in Fig. 6.30. Each curve corresponds to a different background type and each data along the curve to a network with a different θ-sectorization. Additionally, the values for the extreme θ-sectorization cases are listed in Tab. 6.22: the case with a single θ-sector, labelled X1 in Tab. 6.22 and the case with nine θ-sectors, labelled X9 in Tab. 6.22.

The resolutions Δz and $\Delta\theta$, shown in Fig. 6.30 (a) and (b), clearly improve with increasing numbers of θ-sectors. However, large improvements can only be observed for small number of θ-sectors, while for a large number of θ-sectors the further improvements become marginal. In particular, the largest improvement step is achieved by using more than one θ-sector. Furthermore, with higher background levels the improvement in the resolutions due to the θ-sectorization becomes larger.

The efficiencies of the neural network ϵ_{NN} are shown in Fig. 6.30 (c). Since the same 3D finder is used for all different θ-expert test runs, the 3D finder efficiencies

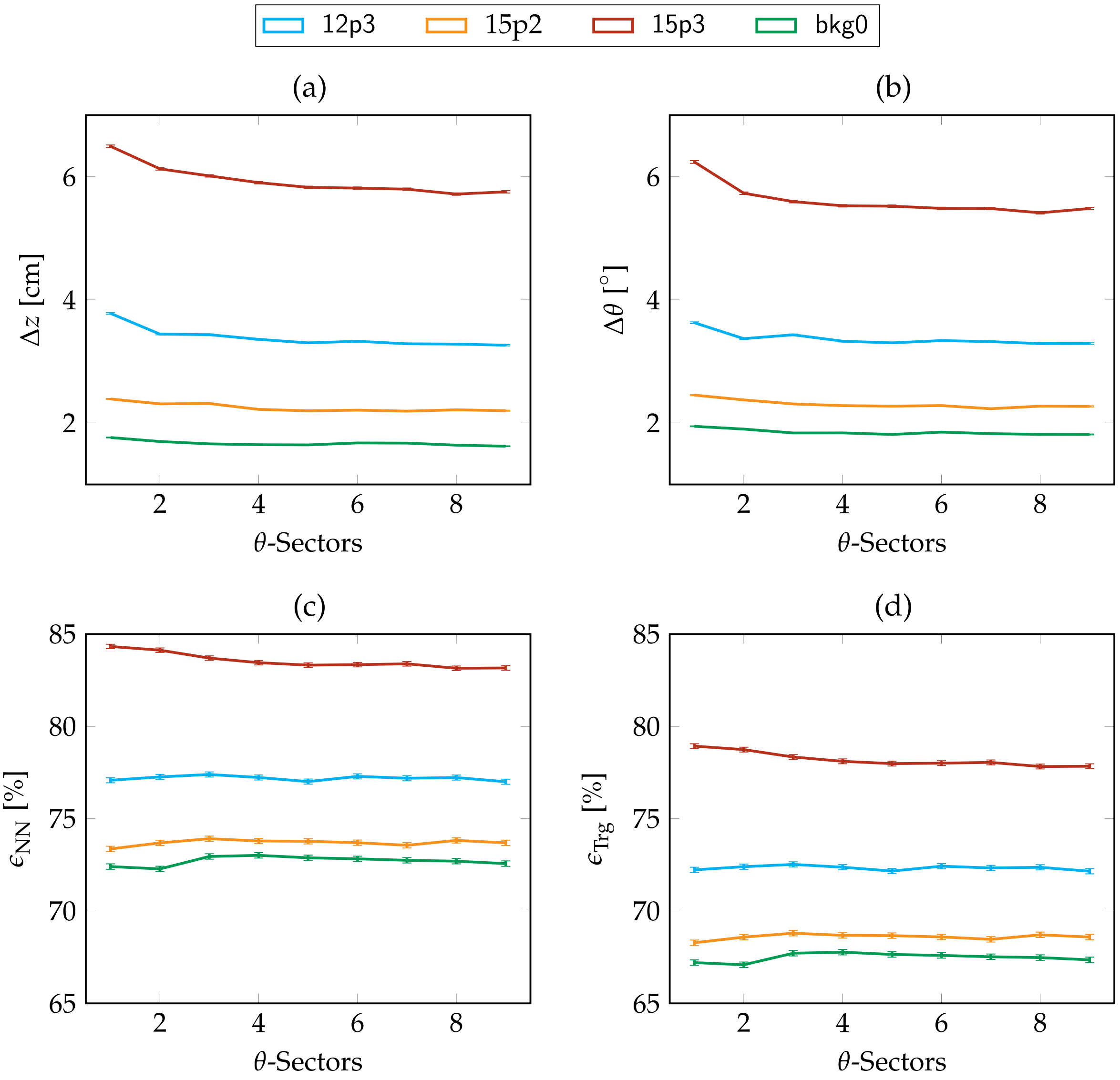


Figure 6.30.: Resolutions, neural network efficiencies and total trigger efficiencies as functions of the number of θ-sectors with trained expert neural networks. The test data set contains only tracks from the IP (range ZIP in Tab. 6.6).

$\epsilon_{\mathrm{Finder}}$ only change with different background types. However, $\epsilon_{\mathrm{Finder}}$ is constant for the different θ-sectors with the same background type (see Tab. 6.22). In order to illustrate the absolute scale of the efficiency, the total trigger efficiencies ϵ_{Trg} are shown in Fig. 6.30 (d).

The efficiencies in Fig. 6.30 (c) and (d) are almost flat. Only for the background type 15p3 a small efficiency loss of about 1 % is observed. Since an efficiency loss is only observed in the high occupancy background 15p3, it only affects those tracks additionally found due to the mixing of background hits with signal hits. Note that the efficiency of the neural network is determined by the hit selection. Only if less than three stereo TSs are selected in the preprocessing, a found track is not processed by the neural network. This hit selection is based on the ID-ranges, where each of the θ-expert networks uses its own, θ-sector specific ID-ranges. Background TSs which increase the track finder efficiency via mixing with signal clusters in the Hough space may not be selected as input for the neural network due to the ID-range cut. This means, the ϕ-distance of the background TSs position to the ϕ-reference position where the 2D track intersects the radial shell of the CDC wire is larger than allowed by the stored ID-ranges (see Sec. 6.3.2). Since the efficiency loss of about 1 % is quite small and only appears with high background levels, it is considered to be tolerable in favor of an improved resolution.

For the case with background 15p3, where the largest average resolution improvement was observed, the resolutions Δz and $\Delta\theta$ as functions of the MC track parameters p_T and θ are shown in Fig. 6.31. Since the improvement in the resolutions is settling for larger number of θ-sectors, only three representative cases are shown in Fig. 6.31: $\theta-$sectors $\in [1, 2, 9]$. The shown labels X1, X2, and X9 correspond to the cases of one, two, and nine θ-sectors. As expected, the single θ-sector network shows the worst resolution in all regions of the track parameters p_T and θ in Fig. 6.31 and improved resolutions are achieved for larger numbers of θ-experts. Figure 6.31 shows that by using two θ-sectors instead of one, more than 50 % of the possible resolution improvement can already achieved.

In the resolutions as functions of p_T, shown in Fig. 6.31 (a) and (b), larger improvements can be observed in the high-p_T region compared to the low-p_T region. As functions of θ, the structure of $\Delta z(\theta)$ and $\Delta\theta(\theta)$ slightly differ. The improvement in $\Delta z(\theta)$ shown in Fig. 6.31 (c), is almost constant over the full shown θ-region. In contrast, the improvement of the resolution $\Delta\theta(\theta)$ (see Fig. 6.31 (d)) is small in the central θ-region of $\theta \in [60, 110]^\circ$ and increasing for shallow-θ tracks. In particular in the forward region, for tracks with a polar angle of $\theta \lessapprox 30^\circ$, the θ-resolution improves by almost a factor of 2.

This indicates that a unique spacing of the θ-sectors might not be required and additional θ-experts should be installed in the forward region. Over the full tested p_T and θ-range a small number of θ-sectors results in small but significant improvement in the resolution. However, for larger numbers of θ-sectors, no significant further improvements are observed. This means that at some point the maximal possible improvement of additional θ-sectors is reached. In order to reduce the overhead of additional sectors in hardware, it is therefore suggested to focus on a small number of θ-sectors. Since the observed improvements in the resolutions are small, the θ-sectorization can be considered as an option for the fine tuning of the neural network trigger. Note that the neural networks were only trained with a smaller number of events in this study (NNB in Tab. D.1). In future fine tuning studies, possible further improvements with larger training data samples could be tested as well. If θ-sectorization is employed, a possible efficiency loss due to the limited ID-ranges in the θ-experts has to be monitored. Possible countermeasures are enlarged ID-ranges in each θ-sector or the use of the same fixed ID-ranges in all θ-sectors.

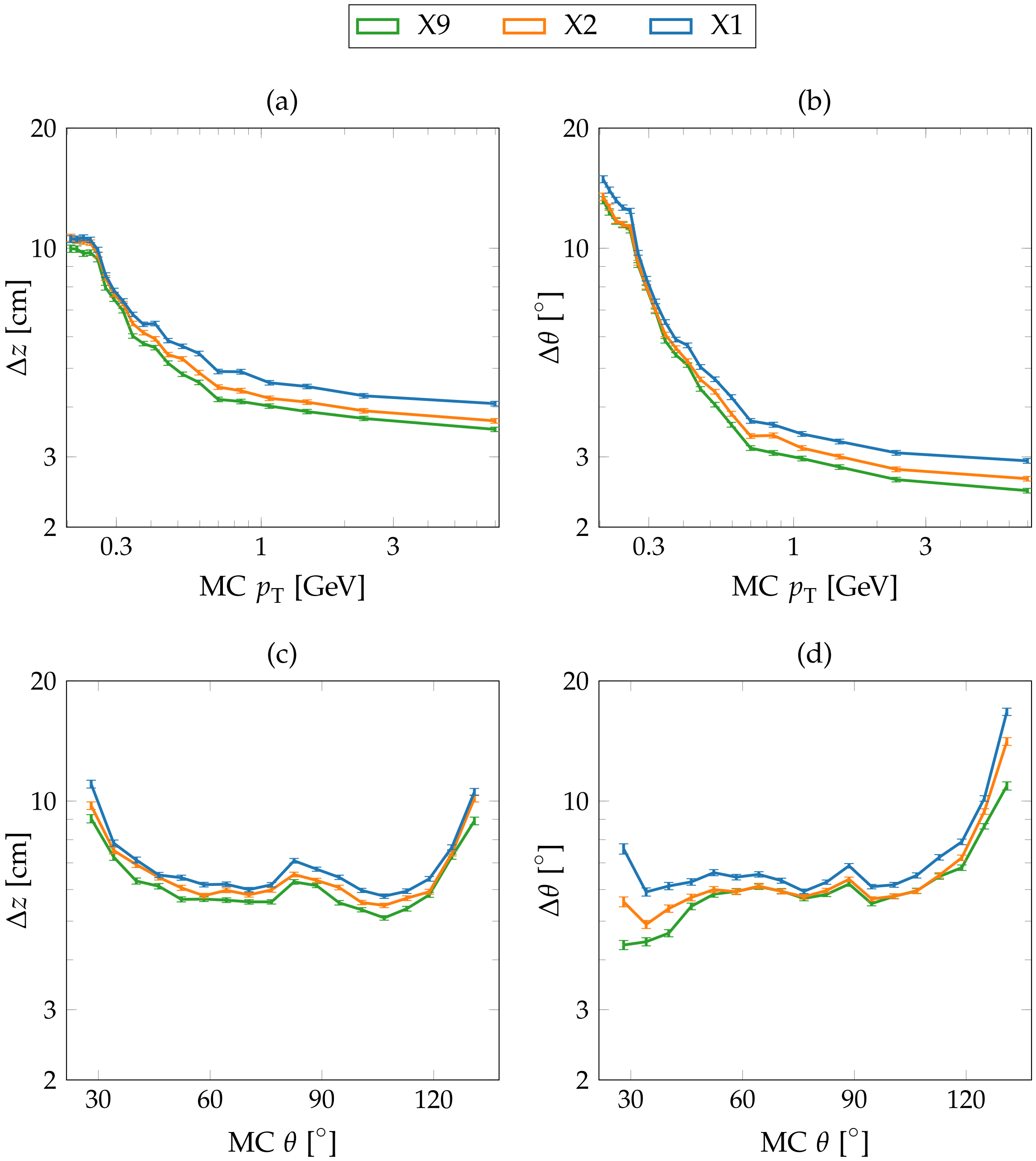


Figure 6.31.: Resolution of the θ-sectorized neural network using the test data ZIP with the background 15p3. For clarity, only three cases are shown: nine θ-sectors, two θ-sectors and a single θ-sector.

6.5.4. Summary of the 3D Finder Benefits

The studies with the 3D finder as input for the neural network demonstrated that the 3D finder is a viable alternative to the 2D finder. With its high efficiency for short tracks, the total trigger efficiency (ϵ_{Trg}) can be significantly improved. Even with the basic five-expert neural networks, an efficiency increase of 12 % without background and up to 20 % with background has been observed (see Tab. 6.16). This efficiency improvement applies to those short tracks which have at least three stereo TSs such that they can be processed by the five-expert network, although they have insufficient axial TSs to be found by the 2D finder.

However, taking full advantage of the efficiency improvement with the 3D finder poses new difficulties for the neural network. Many tracks which are not found by the 2D finder at all now must be processed by the neural network. In particular low-p_{T} or shallow-θ tracks found by the 3D finder, frequently appear as short tracks with less than three stereo TSs selected in the neural network preprocessing. With the basic network setup, a limited efficiency of the neural network (ϵ_{NN}) was observed with the 3D finder as input (77 % to 84 % in Tab. 6.16).

A solution to achieve a significantly improved neural network efficiency is the introduction of short track expert networks. Consequently, the acceptance of the first level trigger is be extended into the low-p_{T} and shallow-θ region. Since the cross sections of the physics interactions typically increases towards shallow-θ angles in the forward and in the backward direction, an enlarged polar angle coverage is an important measure to increase the efficiency for signal events in the detector. Especially for events with a low track multiplicity, significant improvements of the signal efficiency can be expected.

In the simulated single track test events with a uniform distribution in p_{T}^{-1} and lower limit of $p_{\mathrm{T}} \geq 0.2\,\mathrm{GeV}$, a significant improvement of the total trigger efficiency has been observed: while with the 2D finder values of only $\epsilon_{\mathrm{Trg}} \approx 55\,\%$ are achieved (see Tab. 6.16), with the 3D finder in combination with the short track experts efficiencies of $\epsilon_{\mathrm{Trg}} \approx 92\,\%$ are achieved (see Tab. 6.21). Furthermore, only three expert networks turned out to process most of the tracks. In order to reduce the resources on the hardware, a proposal could be made to use experts only for these three cases: all four stereo TSs are present (pattern `p15`), SL7 is missing (pattern `p14`), SL7 and SL5 is missing (pattern `p12`). Since the achieved resolutions per expert scale with the number of stereo TSs used as input, the selected expert can be used as a quality indicator of the neural network estimated. For example, this can be applied in the trigger logic by using different cut values for the different expert networks.

A θ-sectorization study demonstrated a method for the explicit use of the θ-estimate provided by the 3D finder in the neural network preprocessing. Especially, small numbers of θ-sectors provide a significant improvement in the resolution. More than 50 % of the average improvement is already achieved with two θ-sectors. Under the condition that sufficient hardware resources are available, it can be considered as an option for the fine tuning of the neural network precision. The largest observed improvement with a fine θ-sectorization was a factor of two in the θ-resolution for shallow-θ tracks in the forward region. Therefore, an additional shallow-θ expert in the forward region is suggested. In future studies, alternative methods of using the θ-estimates should be investigated as well. Most importantly, the estimate from the 3D finder should be entered directly as input value to the neural network. This can be realized as an additional input node, or as an advanced input calculation method similar to the present use of the 2D track parameters (p_{T}, ϕ).

Chapter 7.

Physics Measurements

A trigger for physics signal events has to process multi track events with characteristic track topologies determined by the involved physics processes. For the measurement of physics signal trigger rates and background rejection rates, in this chapter studies with full events are carried out. Benefits from the developed 3D finder in Chapter 5 and the neural network in Chapter 6 are expected especially for events with a low track multiplicity. While single track and two track events have to be rejected without a 3D trigger, the 3D finder and the neural network allow to open up the trigger for the interesting low multiplicity physics signatures. In the trigger analysis, signal rates and background rejection rates are constructed based on the neural network estimates and new trigger logics are suggested.

As typical candidate for low multiplicity events, simulation studies with Initial State Radiation (ISR) events are carried out in Sec. 7.1. Especially the ISR events with hadronic final states are of importance to reduce the theoretical error on the muon $g-2$ measurements (see Sec. 2.5.1). The muon $g-2$ has a tension with the SM of three standard deviations [110] and is therefore a viable candidate for the search of physics beyond the SM. With an effective trigger for low multiplicity events, precision measurements of the hadronic final states in ISR events at Belle II can contribute to unravel the significance of the tension. Experiments with recorded events from early Belle II runs are presented in Sec. 7.2. The signal efficiencies are measured by selecting events with tracks from the IP, while background rejection rates are measured on selected displaced tracks. Preliminary results of the studies in Sec. 7.1.4 and Sec. 7.2.2 have been presented at [109].

7.1. Initial State Radiation Events

In initial state radiation (ISR) events a photon γ is emitted from an initial state particle (e^+ or e^-) before the collision. This effectively reduces the center of mass energy ($\sqrt{s}$) of the initial state by the amount of energy the photon carries away and thus gives access to a variety of different physics processes at lower collision energies. Directly, these processes would only be accessible by running the e^+e^- collider at a lower center of mass energy $\sqrt{s}$. In the initial state radiation processes

$$e^+e^- \to \gamma e^+e^- \to \gamma q\bar{q} \tag{7.1}$$

with a quark anti-quark pair ($q\bar{q}$) in the final state the cross section for hadron production in e^+e^- collisions can be measured. For the calculation of the hadronic component of the anomalous magnetic moment of the muon (a_μ^{HAD} in Eq. 2.56), this is typically expressed as $R(s)$, the ratio of the hadron production relative to the muon production rate

$$R(s) = \frac{e^+e^- \to q\bar{q}}{e^+e^- \to \mu^+\mu^-}. \tag{7.2}$$

In a plot of this ratio $R(s)$, the resonances at different center of mass energies $\sqrt{s}$ can be seen (see Fig. 7.2). The cross section of the hadron production is an important ingredient for the theoretical estimation of the hadronic vacuum polarization (see Sec. 2.5), which is required in the estimation of the anomalous magnetic moment of the muon (μ $g-2$). With the present $\approx 3\sigma$ tension of the μ $g-2$ with the SM, it is a viable candidate in the search for new physics.

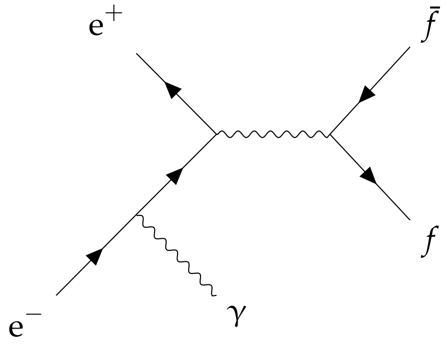


Figure 7.1.: Annihilation of e^+e^- into a fermion pair after emitting a photon from the e^- initial state.

With the high luminosity provided by SuperKEKB, a successful trigger for ISR events would allow Belle II to provide a significant improvement of the measurements of these hadronic cross sections and form factors. Many of these ISR processes have a low track multiplicity in the final state and are thus difficult to handle by the track trigger. This makes them typical candidates where a precise low multiplicity track trigger allows to improve the efficiency. In the following study, the performance of the 3D track finder and the track reconstruction with neural networks is measured in simulated ISR events.

As an example, the Feynman diagram for the initial state radiation event with two fermions in the final state and a single photon radiated from the initial e^- is shown in Fig. 7.1. In general, ISR comprises a series of processes where photons can be radiated from each initial state, electron or proton, and combinations thereof. However, the emission of multiple (soft) photons is suppressed by a factor of α (the fine structure constant) for each additional photon. It is important to note that also final state radiation (FSR) occurs and due to the quantum nature of the interaction only the interference of ISR and FSR can be observed.

7.1.1. Physics Motivation

In ISR events, two main categories of events have to be distinguished: the tagged and the untagged initial state radiation events, where tagging refers to the detection of the ISR photon in the ECL. Since the cross section for ISR events as a function of the polar angle of the photon θ_γ is maximal for vanishing θ angles most of the ISR events are untagged. At a much smaller rate, tagged ISR events can be observed, where the photon has a sufficiently large θ_γ to be seen in the ECL. In these tagged events the photon energy can be reconstructed in the ECL. Therefore, the knowledge on $\sqrt{s}$ of the initial e^+e^- pair is more precise. This reflects in a higher precision on each measured tagged hadronic interaction, however at much lower total rates.

In this study, the Phokhara [111] generator is used to generate the initial state radiation events. Phokhara can simulate initial state radiations with several photons in the initial state and interference of initial state and final state radiations. After the generation, the usual simulation and reconstruction is carried out within basf2. This includes the propagation of the particles through the Belle II detector, to create hits in the subdetectors and a following digitization (simulation of the detector readout). The offline reconstruction provides reconstructed tracks as reference values and the CDC trigger simulation provides all input and output values for the 3D trigger studies.

With the Phokhara generator, the three final states with the largest cross sections are simulated: As the leptonic reference process, the two muon final state $\mu^+\mu^-$ is simulated. The two simulated hadronic final states are $\pi^+\pi^-$ and $\pi^+\pi^-\pi^0$. Due to their large cross section relative to the other hadronic final states, they make up the largest contribution of the error in the hadronic vacuum polarization. In the leptonic final state $\mu^+\mu^-$ the muons are directly produced as final state fermions ($f\bar{f}$ in Fig. 7.1). In contrast, due to the confinement of the quarks, in the hadronic final states the quark pairs $f\bar{f} = q\bar{q}$ are subject to hadronic interactions (multi-meson states) which are measured in the detector.

In the hadronic case, resonances with neutral mesons occur, which have the same quantum numbers as the photon: spin 1 and odd parity (1^-). This is dominantly the ρ^0 meson composed of an up-quark u and a down-quark d and it has the flavor wave function:

$$|\rho^0\rangle = \frac{1}{\sqrt{2}}\left(|d\overline{d}\rangle - |u\overline{u}\rangle\right) \tag{7.3}$$

The ρ^0 meson is a neutral vector meson with spin 1 and odd parity (1^-) and thus it has the same quantum numbers as the photon. Its mass and decay width are [37]

$$m_{\rho^0} = 775.26 \pm 0.25\,\text{MeV} \qquad \Gamma_{\rho^0} = 147.8 \pm 0.9\,\text{MeV}. \tag{7.4}$$

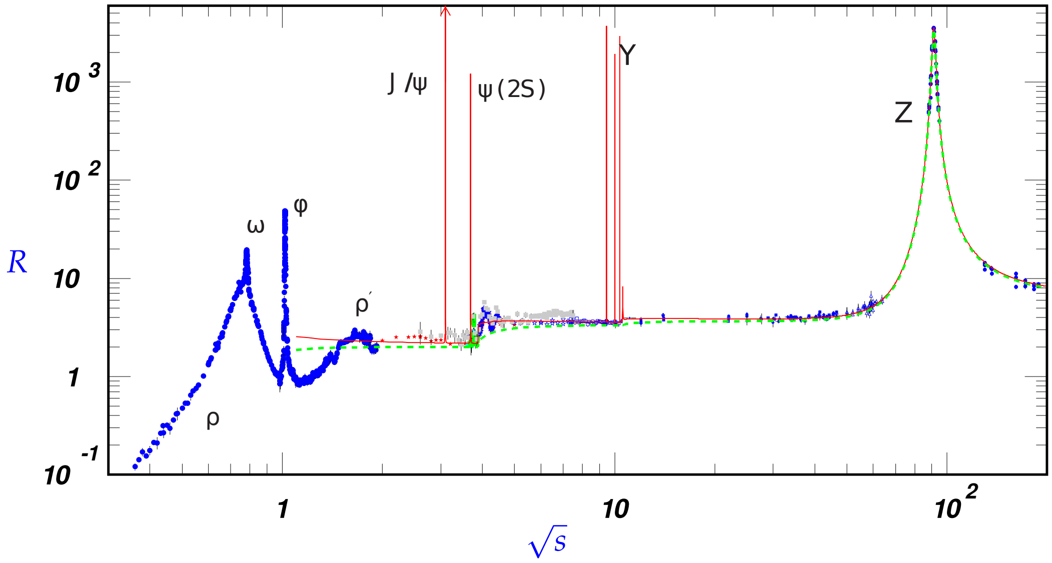


Figure 7.2.: The ratio $R(s)$ published by the particle data group in 2018 [37]. The y-axis shows the ratio R: the hadron production rate $e^+e^- \to q\bar{q}$ divided by the muon production rate $e^+e^- \to \mu^+\mu^-$. The x-axis is the center of mass energy $\sqrt{s}$.

The ρ^0 is the neutral ρ meson of an isospin triplet, where the two charged partners of the same mass and width have the wave functions:

$$|\rho^-\rangle = |d\bar{u}\rangle \qquad |\rho^+\rangle = |u\bar{d}\rangle. \tag{7.5}$$

Besides the ρ^0 also the ω meson can be produced as an intermediate state, with the wave function

$$|\omega\rangle = \frac{1}{\sqrt{2}}\left(|d\bar{d}\rangle + |u\bar{u}\rangle\right). \tag{7.6}$$

This vector meson with spin 1 and odd parity (1^-) is an isospin singlet with mass and decay width [37]:

$$m_\omega = 782.65 \pm 0.12\,\text{MeV} \qquad \Gamma_\omega = 8.49 \pm 0.08\,\text{MeV}. \tag{7.7}$$

With the much smaller width compared to the ρ meson, the ω meson has a larger lifetime. In the $R(s)$ diagram it appears as a narrow peak on top of the broad peak of the ρ^0 meson. The ratio $R(s)$, as published by the particle data group in 2018 [37] is shown in Fig. 7.2.

The final state particles propagating into the detector can then interpreted as the decay products of the intermediate states. The dominant process for the ρ^0 is

$$\rho^0 \to \pi^+\pi^- \tag{7.8}$$

which occurs in almost 100 % of the cases [37]. A small fraction of the ρ^0 has the three pion final state

$$\rho^0 \to \pi^+\pi^-\pi^0 \tag{7.9}$$

which occurs with a fraction of $\left(1.01^{+0.54}_{-0.36} \pm 0.34\right) \times 10^{-4}$ % of the ρ^0 decays [37]. For the ω meson, the final state $\pi^+\pi^-\pi^0$ has the largest partial decay width

$$\omega \to \pi^+\pi^-\pi^0 \tag{7.10}$$

which occurs in (89.2 ± 0.7) % of the ω decays [37]. The final state

$$\omega \to \pi^+\pi^- \tag{7.11}$$

only occurs in $(1.53^{+0.11}_{-0.13})$ % of the ω decays [37]. Hence, ω mesons are the typical intermediate states for the $\pi^+\pi^-\pi^0$ final state and ρ^0 mesons for the $\pi^+\pi^-$ final state.

7.1.2. Analysis Method

The simulated ISR events all have two charged particles in the final state. If both particles propagate through the CDC, these are seen as two track events by the CDC track trigger. So far, the trigger efficiencies were only used for the study of single track events. In order to study full event efficiencies, the single track efficiencies defined in Sec. 6.4.6 are now extended. In the ISR studies, especially the two track extension of the `Gain`-rate is used. For the rate `Gain1`, the events were required to have one neural network track estimate in the event. Out of these one track events, where the MC vertex position is located at the IP, `Gain1` is the rate where the neural network z-estimate is contained within a given cut range (see Sec. 6.4.6).

column	description	default cut
`Gain1`	single track event, estimated in z-cut	z-cut $= 20\,\mathrm{cm}$
`Gain2`	≥ 2 track event, ≥ 1 track with $z_{\mathrm{NN}} \leq z$-cut	z-cut $= 20\,\mathrm{cm}$
`Op60`	two track event, $\theta_{1,2} \leq \theta_{1,2}$-cut (rate `Open`)	$\theta_{1,2}$-cut $= 60^\circ$

Table 7.1.: Quality measure names and descriptions used in the analysis of the neural network trigger performance on simulated ISR events. The column names are used in the summary tables with the default cut values.

In the extension `Gain2`, the event is required to have two estimated neural network tracks, each matching to a MC track. Both MC tracks are required to have their vertices at the IP. Out of these events, the rate `Gain2` is the fraction of events where at least one of the estimated track vertices is contained within a defined z-cut range. The default cut range for the estimation, which is shown in the subsequent result tables, is $|z_{\mathrm{NN}}| \leq 20\,\mathrm{cm}$. Additionally, this section contains plots of the Gain-rates, where the z-cut value is varied in 5 cm steps in the range $z\text{-cut} \in [5, 45]\,\mathrm{cm}$.

Besides the z-cut based rate, with ISR events with hadronic final states, an angular based cut is considered here. With hadrons in the final state where an intermediate resonance state is created, the angles of the particles propagating through the detector are correlated. After the creation of the intermediate state (esp. ρ_0 and ω mesons), this (massive) intermediate meson has a specific energy and momentum before it decays. Due to the asymmetric beam energies of the colliding e^+ (4 GeV) and e^- (7 GeV) beams, the center of mass system in the collisions is boosted in the positive z-direction. Since the initial state radiation photons carry away some fraction of the four-momentum, the boost direction of the intermediate meson may change here, compared to $B\overline{B}$ events with an intermediate $\Upsilon(4S)$ meson.

Energy and momentum conservation in the decay of the intermediate boosted meson dictate a correlation between the two tracks of the hadronic final states. This means, the solid angle between the two tracks must be small. Although the solid angle is composed of the azimuthal angle ϕ and the polar angle θ, the relevant angle to be inspected here is θ. An intermediate particle boosted along the z-axis may have back-to-back ϕ angles, but still a very small difference in the θ angles. Although a calculation of the full solid angle considering the momentum vector of both particles is the complete description of the problem, at the trigger level it is more convenient to use the θ-estimates of the neural network directly.

To study the opening angle between two estimated tracks, the rate Open is introduced here. In an event with two tracks, there are two θ-values: θ_1 and θ_2. Here the magnitude of the θ-difference between the two tracks is defined as

$$\theta_{1,2} = |\theta_1 - \theta_2| \,. \tag{7.12}$$

Similar to the definition of a z-cut, a θ-based cut can be defined on the estimate of $\theta_{1,2}$. By requiring a maximum allowed difference of the $\theta_{1,2}$-cut, events with a smaller $\theta_{1,2}$ can be triggered.

In order to define the new rate Open, two track events are required, with a neural network track estimate for each of the two tracks. Out of these events, Open is the fraction of events where the estimated $\theta_{1,2}$ is smaller than $\theta_{1,2}$-cut. In the summary tables, a fixed cut of $\theta_{1,2}\text{-cut} = 60^\circ$ is used and the column with this value will be called Op60. Additionally, plots are shown with $\theta_{1,2}$-cut varied in 15° steps in the range $\theta_{1,2}\text{-cut} \in [30, 90]\,^\circ$.

Pion Trained Network

Using the Phokhara generator [111], ISR events are simulated with three different final states: $\mu^+\mu^-$, $\pi^+\pi^-$ and $\pi^+\pi^-\pi^0$. For the analysis of the three simulated final states, neural networks with the training data size NNC are used here, where the number of training samples is 150 times the number of weights in the network (see Tab. D.1). In the two hadronic final states, the neural network has to estimate pion tracks rather than the already known muon tracks from the previous section. In order to optimally tune the neural network performance to the case of pion tracks, a retraining of the neural network with pion tracks is used here in addition. For the training with muon tracks, the same network as presented in the sections with size NNC is used. The pion training was carried out with the same track parameter ranges and background types as the muon training, with the only difference in the particle type of the simulated training target tracks.

A possible improvement with the pion network is evaluated in comparison to the muon networks on the simulated ISR events. Muons produce very "clean" tracks in the CDC in the sense that the idealized mathematical description of a helical track is an ideal track model for a muon and the required error corrections are minimal. Pions in contrast appear more fuzzy: they typically produce more secondary particles in multiple scattering interactions with the detector material and they create cluster hits in the ECL via hadronic showers. Thus, pions in the detector produce a sort of background themselves, which makes pion tracks more difficult to reconstruct than muon tracks.

The approach to compensate these errors at the first trigger level used in the following study, is to train optimized neural networks: by using pions in the training data instead of muons, the neural network can learn the fuzziness of the pion data to provide accurate results. An improved resolution could be achieved, leading to a better trigger efficiency for ISR events with pions in the final state. Additionally to the comparison of muon and pion training, in the following studies of the three final states, the results with the 2D finder and the 3D finder are compared with two different background types: bkg0 corresponds to training and testing without background and 15p3 corresponds to training and testing with the simulated Phase 3 background. For compactness, the other background types with lower occupancies known from the previous studies are not shown here.

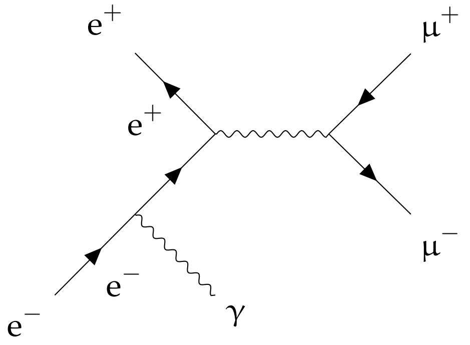


Figure 7.3.: Annihilation of e^+e^- into a $\mu^+\mu^-$ pair after emitting a photon from the e^- initial state.

7.1.3. Final State $\mu^+\mu^-$

The first final state to be tested is the leptonic final state with the $\mu^+\mu^-$ pair in the final state. The Feynman diagram for one such ISR process, with a single ISR photon, is shown in Fig. 7.3. After the radiation of the ISR photon, a virtual photon is created at the interaction vertex of the e^+e^- collision. This virtual photon directly decays into the $\mu^+\mu^-$ pair at the second vertex in the Feynman diagram.

The distributions of reconstructed tracks in the simulated ISR events are shown in Fig 7.4. It shows all reconstructed tracks, i.e. without the requirement of any matching track as well as reconstructed tracks with a matching primary MC particle. Additionally, both cases are tested without background (bkg0) and with the Phase 3 background (15p3). The track counts in each case are shown in the legend.

Without matching, the rate of reconstructed tracks with 15p3 is significantly larger than in the case without background. Fig. 7.4 (a) shows that most of these background tracks have low momentum. After the matching, the reconstructed tracks with and without background show nearly the same distribution. This confirms that the tracking efficiency is not affected by the background.

The matching case shows the track parameter distributions of both final state muon tracks. With 10^5 simulated ISR events, a total of $2 \cdot 10^5$ $\mu^\pm$ tracks is simulated. Due to the angular distribution of the tracks, not all muons traverse the CDC and only about $1.5 \cdot 10^5 \mu^\pm$ of the reconstructed tracks are matched to final state muon tracks. Since the ISR events are simulated signal events, the matched Reco tracks in Fig. 7.4 (d) have their z-vertices at the IP. Prior to the matching, the few reconstructed tracks displaced from the IP can be completely described by the background and the accuracy of the reconstruction. Fig. 7.4 (b) shows a broad peak around $\phi \approx 0^\circ$ also in the case without background and for the matching tracks. In the polar angle θ shown in Fig. 7.4 (c), the tracks are almost distributed flat, with a small additional peak in the forward region ($\theta < 60^\circ$).

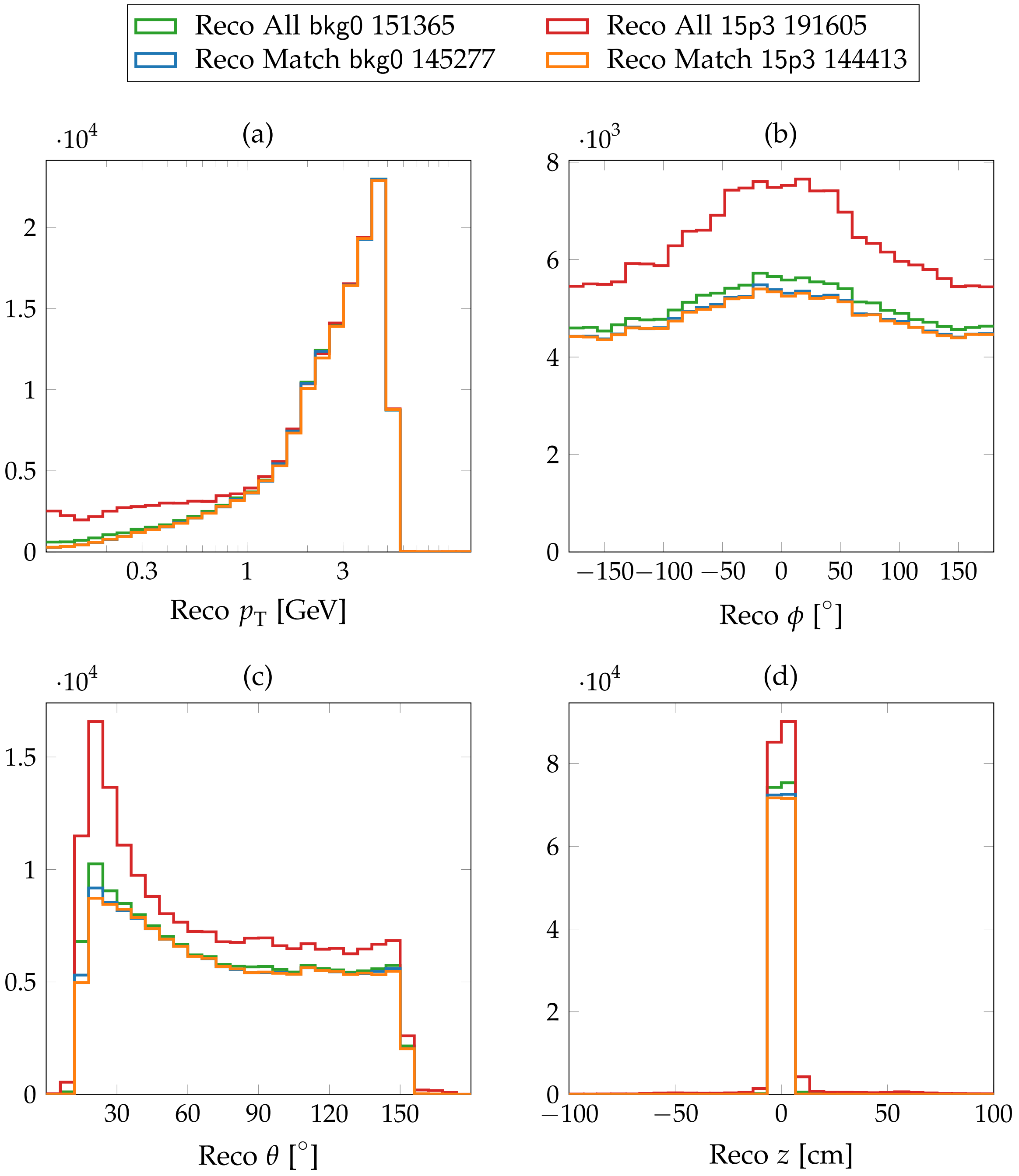


Figure 7.4.: Offline reconstructed track parameter distributions of the simulated ISR event with two muons in the final state ($\mu^+\mu^-$). For the cases without background bkg0 and with the Phase 3 background 15p3 the distributions of all reconstructed tracks as well as the reconstructed tracks related to a primary MC particle are shown.

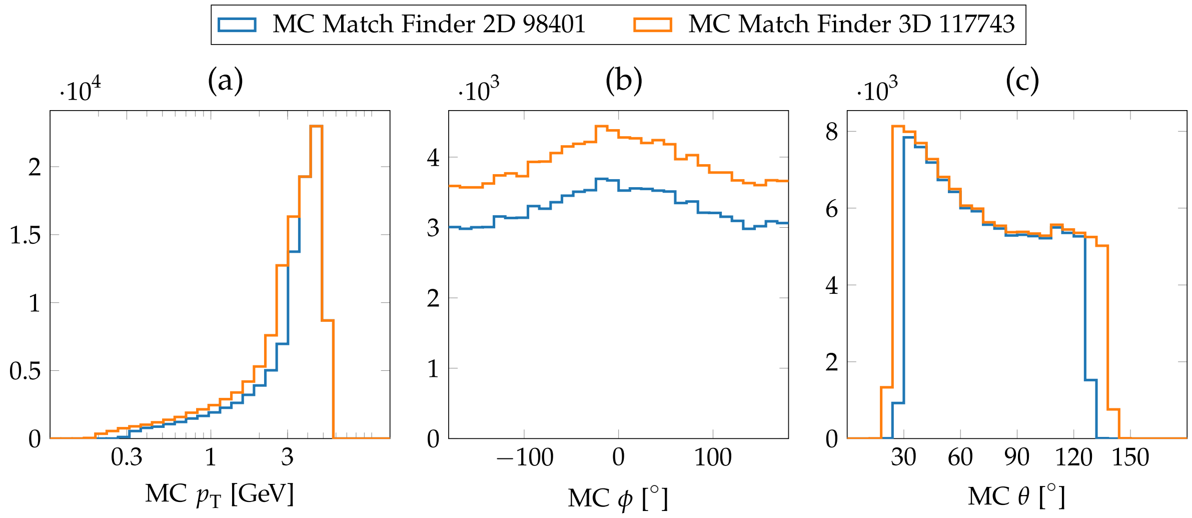


Figure 7.5.: MC track parameter distributions of the primary MC particles with a match to a 2D finder track and with a match to a 3D finder track in $\mu^+\mu^-$ events.

In Fig. 7.5 the MC track parameter distributions for primary MC particles with a match to a finder track (2D finder and 3D finder) are shown. Since the difference between the case with and without background is small after the matching, for clarity only the case without background is shown here. Figure 7.5 (a) shows the improvement of the 3D finder in the low-p_T region. Additionally, the distribution in Fig. 7.5 (c) shows the improvement of the 3D finder for shallow-θ angles. As expected, both finder efficiencies depend on ϕ. This can be seen in Fig. 7.5 (b) in comparison with the reconstructed ϕ-distribution in Fig. 7.4 (b).

Neural Network Results

Bkg$_{\text{test}}$	Train	$\Delta\theta_{\text{MC}}$	Δz_{MC}	$\epsilon_{\text{NN}}^{\text{MC}}$	$\epsilon_{\text{Finder}}^{\text{MC}}$	$\epsilon_{\text{Trg}}^{\text{MC}}$	Gain1	Gain2	Op60
		[°]	[cm]	[%]	[%]	[%]	[%]	[%]	[%]
bkg0	$\mu^\pm$	0.9	1.1	99.9	67.7	67.7	99.5	99.9	72.0
bkg0	$\pi^\pm$	1.0	1.4	99.4	67.7	67.3	99.7	100.0	72.5
15p3	$\mu^\pm$	3.9	4.5	98.8	69.4	68.6	95.8	99.9	68.6
15p3	$\pi^\pm$	4.1	4.8	98.8	69.4	68.6	96.2	99.9	68.7

Table 7.2.: Results of the neural networks with the 2D finder as input for the ISR final state $\mu^+\mu^-$. A training on muon tracks is compared with a training on pion tracks and training with background is compared to training without background. The results are sorted by the resolution Δz.

The results with the $\mu^+\mu^-$ final states and the 2D finder as input are summarized in Tab. 7.2. As expected for muon events, in the case without background an excellent Δz resolution can be achieved: 1.4 cm with the pion network and even 1.1 cm with the muon network. Since the final states are muons, no benefits in the achievable resolution are expected from the pion trained networks here. Although looking at the column for the z-cut rate `Gain1`, the pion network achieves comparable results to the muon network. The rate `Gain2` is almost 100 % in all cases indicating that a z-cut at 20 cm provides a high trigger efficiency for two muon events. The new θ-cut rate with the cut on the opening angle $\theta_{1,2} \leq 60^\circ$ is shown in the column `Op60`. Since the polar angles are not correlated via an intermediate resonance state, only low values of `Op60` $\approx 70\,\%$ are observed.

$\mathtt{Bkg}_{\text{test}}$	`Train`	$\Delta\theta_{\text{MC}}$	Δz_{MC}	$\epsilon_{\text{NN}}^{\text{MC}}$	$\epsilon_{\text{Finder}}^{\text{MC}}$	$\epsilon_{\text{Trg}}^{\text{MC}}$	`Gain1`	`Gain2`	`Op60`
		[°]	[cm]	[%]	[%]	[%]	[%]	[%]	[%]
bkg0	$\mu^\pm$	0.8	1.0	89.1	81.0	72.2	99.5	99.8	69.2
bkg0	$\pi^\pm$	1.0	1.4	89.1	81.0	72.2	99.8	100.0	69.5
15p3	$\pi^\pm$	3.3	4.7	90.4	82.1	74.3	95.5	99.9	71.1
15p3	$\mu^\pm$	3.3	4.7	90.4	82.1	74.3	95.2	99.9	70.9

Table 7.3.: Test results of the neural network with the 3D finder as input for the ISR final state $\mu^+\mu^-$. A training on muon tracks is compared with a training on pion tracks and a training with background is compared to training without background. The results are sorted by the resolution Δz.

The results of the same study carried out with the 3D finder as input are shown in Tab. 7.3. The track finding efficiency ϵ_{Finder} of the 3D finder is larger than $\epsilon_{\text{3D Finder}} \geq 81\,\%$ compared to only $\epsilon_{\text{2D Finder}} \leq 70\,\%$ with the 2D finder. Regardless of the improved efficiency, with the 3D finder equal or slightly improved track parameter resolutions can be achieved by the neural networks. In the case without background, the muon network reaches $\Delta z \approx 1$ cm. This apparent contradiction to the previous single track studies (the resolution here is much better), can be explained by the p_T-distribution. The distribution shown in Fig. 7.4 (a) shows a peak for high momentum tracks with $p_T \geq 4$ GeV. In contrast, in the earlier single track studies a uniform distribution was generated in the inverse transverse momentum (p_T^{-1}), which resulted in a peak at low-p_T values around $p_T \approx 0.3$ GeV (compare e.g. Fig. 6.13 (a)). Due to the high momentum peak, the majority of the muons can be described by straight tracks. In this high momentum region, also the previous single track studies achieved resolutions as good a these.

As in the earlier studies with the 3D finder as input, the efficiency of the neural network is not optimal, with $\epsilon_{\text{NN}} \approx 90\,\%$. In comparison, with the 2D finder as input, an efficiency $\epsilon_{\text{NN}} \approx 100\,\%$ is achieved. The solution to avoid this efficiency loss are the short track expert neural networks, presented in Sec. 6.5.2.

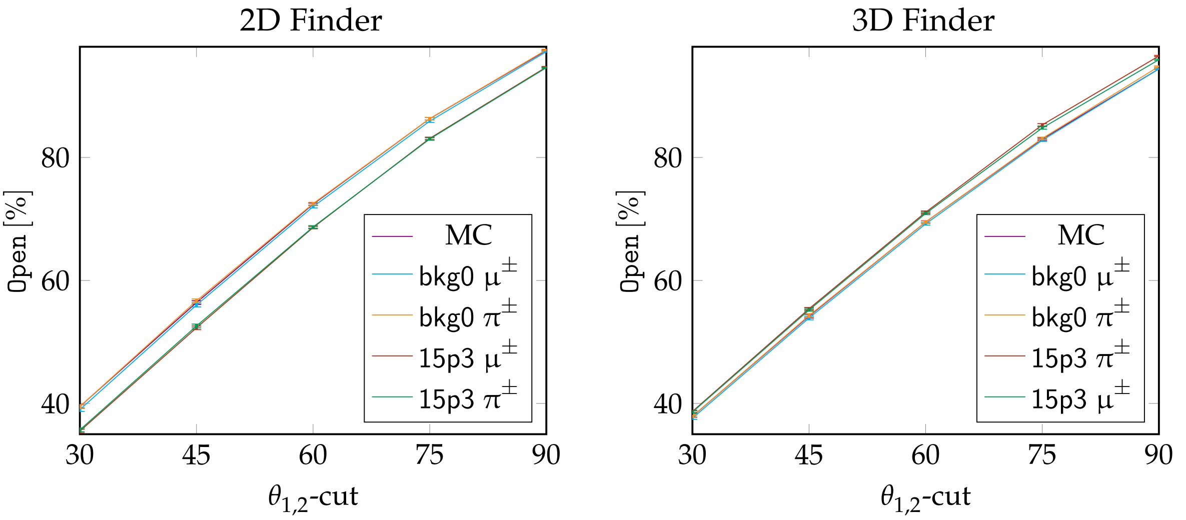


Figure 7.6.: Rate Open of ISR events with an opening angle $\theta_{1,2}$ between the two muons smaller than the cut values $\theta_{1,2}$-cut. For the found track pairs, the MC value of $\theta_{1,2}$ and the estimates of four different neural networks are shown. The networks are trained with muons or pions and with the Phase 3 background 15p3 or without background bkg0.

Figure 7.6 shows the newly introduced rate Open, which describes the rate of events where the θ opening angle of the two tracks is smaller than the value of $\theta_{1,2}$-cut. The linear rise of Open in Fig. 7.6 confirms the absence of the correlation of the θ-angles in the $\mu^+\mu^-$ final state. Therefore, Open is not a suitable criterion to built a trigger for $\mu^+\mu^-$ events. However, the plot is useful as a reference for the following studies with pions in the final state.

A suitable trigger for $\mu^+\mu^-$ events could still be based on the z-cut. This is visualized in Fig. 7.7 for the rates Gain1 and Gain2 and the case with the 2D finder and with the 3D finder as input. Each plot in Fig. 7.7 shows the rates for the four different neural network types. Without background all efficiencies are $\geq 95\,\%$, even for the smallest cut values z-cut $= 5\,\text{cm}$. With background the Gain1-rates are rising slower than the Gain2-rates, which is expected as it is more likely to have at least one track within the defined cut region. In all four plots, the difference between the muon and the pion network is very small, although the muon network achieves slightly better results. This is expected since the test data contains muon tracks. The important take away message here is that the loss in going from muon training to pion training is very small, even if the tested tracks are muons. If the pion net turns out to be significantly better for other event types, this minimal loss in resolution on muon tracks is tolerable.

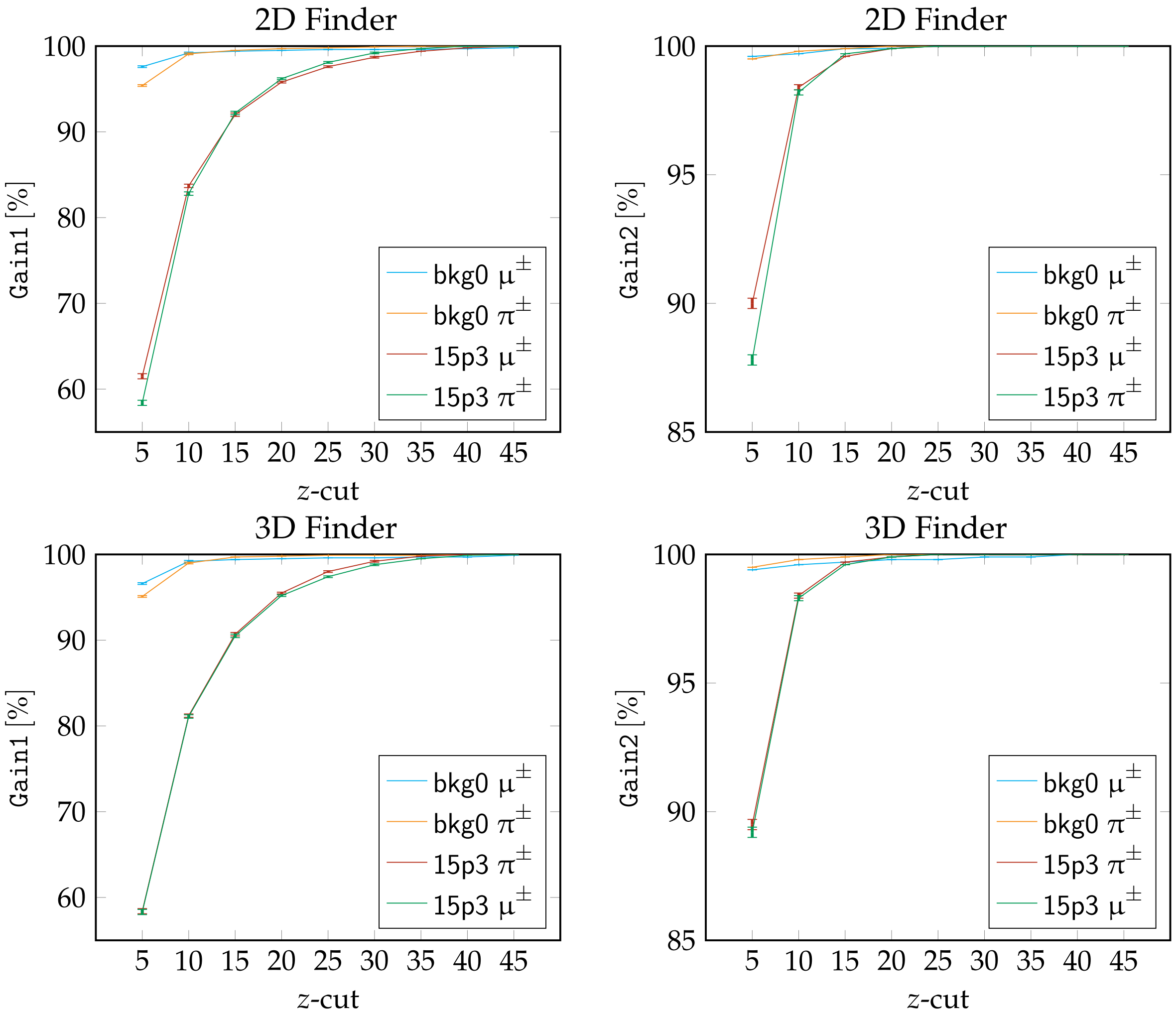


Figure 7.7.: The z-cut based rates `Gain1` and `Gain2` for ISR events with two muons in the final state. In `Gain1` one muon track is processed by the neural network, in `Gain2` two tracks. The results with the 2D finder and with the 3D finder as input are compared.

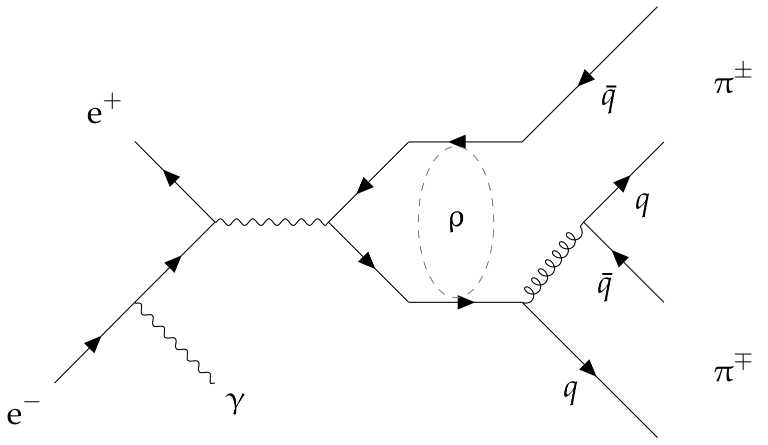


Figure 7.8.: Annihilation of e$^+$e$^-$ into a pion pair $\pi^+\pi^-$ after emitting a photon from the e$^-$ initial state.

7.1.4. Final State $\pi^+\pi^-$

In the next study, ISR events with the final state $\pi^+\pi^-$ are tested. After the initial state radiation, here the created fermion pair are quarks. This quark anti-quark pair can form an intermediate resonance state, which is most likely a ρ_0 meson in this case. This short lived ρ_0 meson decays almost instantly via the strong interaction into the observable charged pion pair. One possible Feynman diagram for this interaction is shown in Fig. 7.8.

The track distributions of the reconstructed tracks in these hadronic ISR events are shown in Fig. 7.9. It shows distributions of all reconstructed tracks as well as the reconstructed tracks matched to primary MC particles (i.e. matched to the final state particles π^+ and π^-). In Fig. 7.9 (a) a large peak appears in the low-p_T region $p_\mathrm{T} \approx 400\,\mathrm{MeV}$, followed by a shallow peak in the high-p_T region around $p_\mathrm{T} \approx 2\,\mathrm{GeV}$. The matching with the MC primary tracks removes the majority of the low-p_T peak in the reconstructed tracks, while the high-p_T peak remains. Since the low-p_T peak without matching is nearly identical in the cases with and without background, it can only be explained by the secondaries created from the pion tracks interacting with the detector material. The large proportion of secondaries is also reflected in the track count: roughly five times more tracks are reconstructed than matched to the primary MC particles. Since the total number of reconstructed tracks ($2.1 \cdot 10^5$) is close to the total number of simulated pions ($2 \cdot 10^5$), on average about one scattered secondary track per missing pion track can be observed.

The ϕ-distribution in Fig. 7.9 (a) has a broad peak around $\phi \approx 0^\circ$, where the pion signal tracks are similarly distributed in ϕ as the already known background tracks. In Fig. 7.9 (c) two peaks of the primary particles into the forward and backward region can be observed (i.e. for shallow polar angles θ). This θ-distribution is the reason for the small number of reconstructed tracks with a match to a MC particle: out of the 10^5 simulated $\pi^+\pi^-$ with a total of $2 \cdot 10^5$ tracks, only $\approx 0.4 \cdot 10^5$ of the primary particles could be reconstructed. The majority of the pions are emitted with even shallower θ angles such that they miss the acceptance region of the Belle II detector. The secondaries are differently distributed, with a broad peak around $\theta \approx 50^\circ$ and only few secondaries with shallow-θ angles. This indicates that these secondaries are more likely created at right angles to the flight direction of the primary pions. In the case with background, an additional peak appears for small values of θ, which must not be confused with the ISR signal events.

The z-distribution in Fig. 7.9 (d) shows a broad peak of secondaries overlaid with the signal tracks from the IP. This broad peak is slightly shifted towards positive values of z, which agrees with the slightly larger peak of the polar angle θ in the forward direction. With the boost in positive z direction, the center of mass system is more likely boosted into the positive z-direction. After the decay of the intermediate state, the boosted $\pi^+\pi^-$ pair creates the scattered secondaries along its flight path. The broad peak of secondaries in z is created by the pions with shallow-θ angles, which are scattered directly in the beampipe, or in the material close to the beampipe.

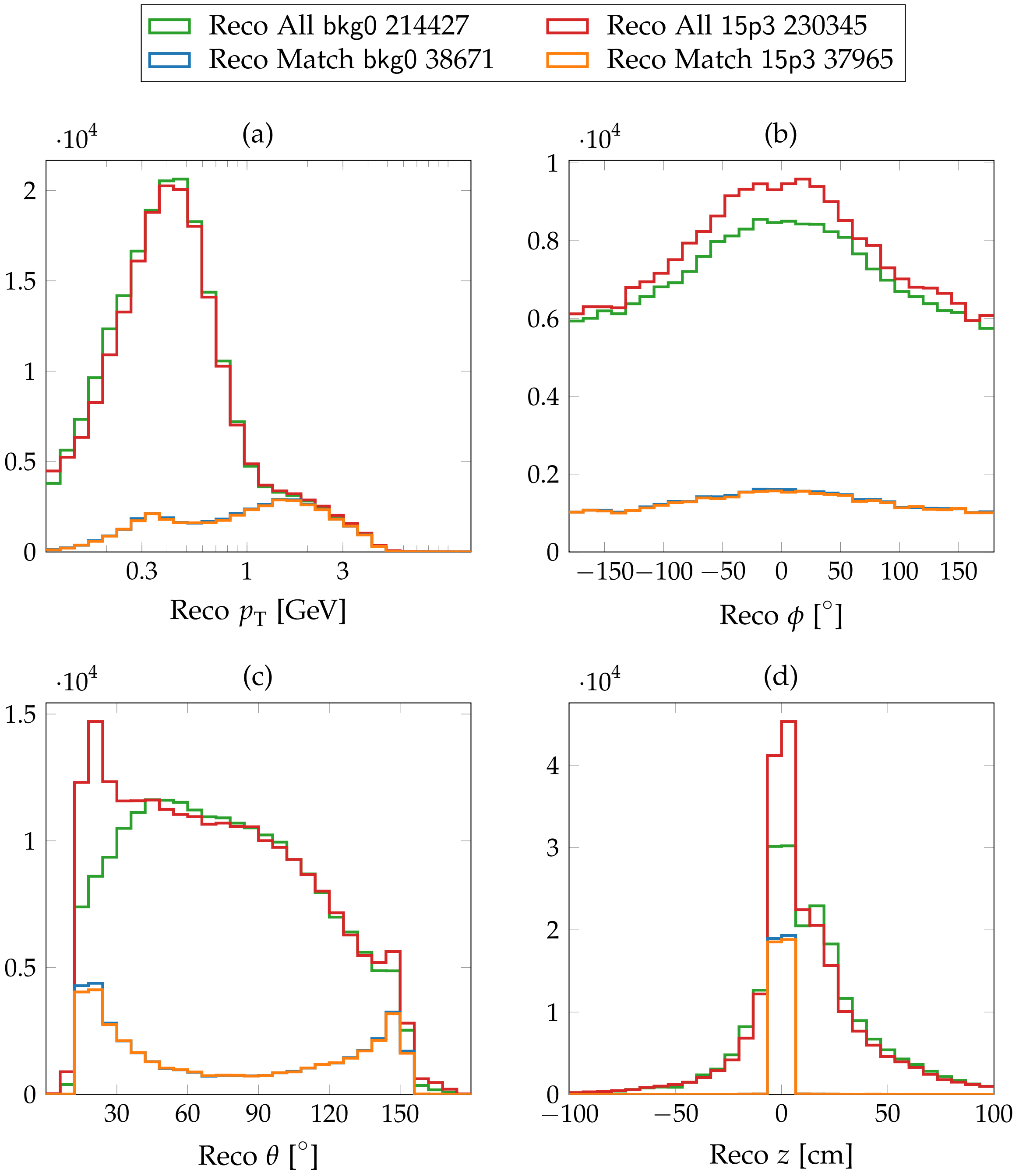


Figure 7.9.: Offline reconstructed track parameter distributions of the simulated ISR events with two pions in the final state ($\pi^+\pi^-$). For the cases without background bkg0 and with the Phase 3 background 15p3 the distribution of all reconstructed tracks as well as the reconstructed tracks related to a primary MC particle are shown.

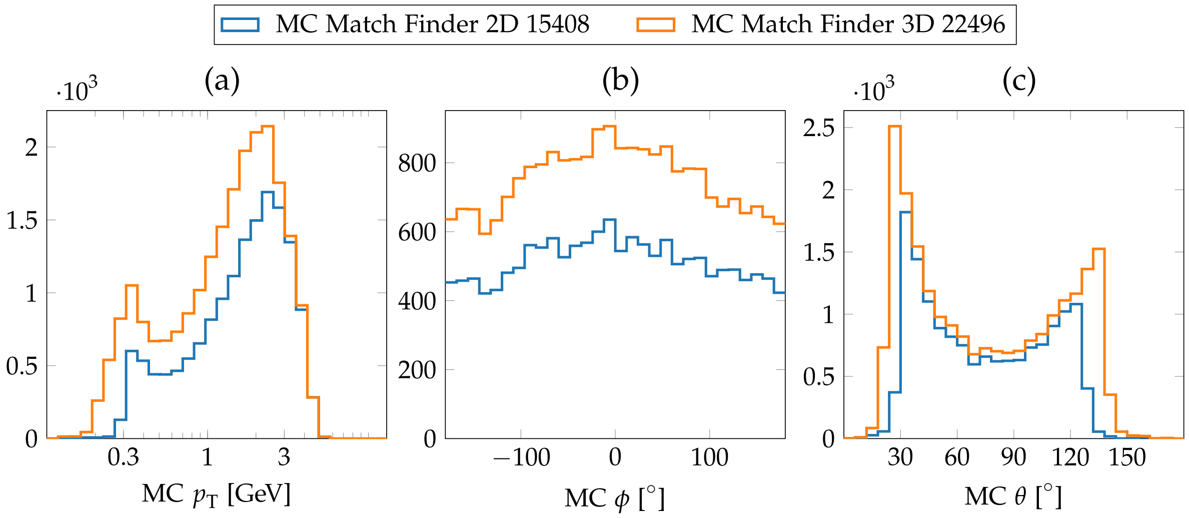


Figure 7.10.: MC track parameter distributions of the primary MC particles with a match to a finder track (2D or 3D) for the final state $\pi^+\pi^-$ in the case without background.

The actual MC track parameter distribution of the input tracks to the neural network, i.e. the MC tracks with a match to a 2D finder or to a 3D finder track, are shown in Fig. 7.10. Since the distributions with and without background are very similar, only the case without background is shown in Fig. 7.10. Characteristic for the 3D finder efficiency is that the total number of found tracks is larger than with the 2D finder. Figure 7.10 (a) shows the improvement in the low-p_T region and Fig. 7.10 (c) shows the improvement at shallow-θ angles. In ϕ no characteristic difference between the two finders is observed. Correctly, both distributions have the same shape as the distributions in Fig. 7.9 (b).

Compared to the muon final state, the two pion final state data set contains way more low-p_T tracks and only very few high-p_T tracks. In contrast, the input tracks to the neural network with the muon final state had a peak in the high-p_T region in Fig. 7.5 (a). Furthermore, the fraction of pion tracks with shallow polar angles is quite large in Fig. 7.10 (c), compared to a relatively flat θ-distribution with the muon final state in Fig. 7.5 (c).

	Bkg$_{\text{test}}$	Train	$\Delta\theta_{\text{MC}}$	Δz_{MC}	$\epsilon_{\text{NN}}^{\text{MC}}$	$\epsilon_{\text{Finder}}^{\text{MC}}$	$\epsilon_{\text{Trg}}^{\text{MC}}$	Gain1	Gain2	Op60
			[°]	[cm]	[%]	[%]	[%]	[%]	[%]	[%]
2D	bkg0	$\pi^{\pm}$	5.3	5.7	96.9	39.9	38.6	89.9	99.4	99.1
	15p3	$\pi^{\pm}$	6.9	8.5	95.2	42.4	40.3	86.6	98.6	97.2
	15p3	$\mu^{\pm}$	7.0	8.9	95.2	42.4	40.3	85.2	97.7	97.2
	bkg0	$\mu^{\pm}$	8.2	10.4	98.0	39.9	39.1	83.7	96.2	97.3
	Bkg$_{\text{test}}$	Train	$\Delta\theta_{\text{MC}}$	Δz_{MC}	$\epsilon_{\text{NN}}^{\text{MC}}$	$\epsilon_{\text{Finder}}^{\text{MC}}$	$\epsilon_{\text{Trg}}^{\text{MC}}$	Gain1	Gain2	Op60
			[°]	[cm]	[%]	[%]	[%]	[%]	[%]	[%]
3D	bkg0	$\pi^{\pm}$	6.1	5.4	79.3	57.6	45.6	92.4	99.0	99.2
	15p3	$\pi^{\pm}$	8.1	8.3	80.8	60.0	48.5	87.2	99.0	98.8
	15p3	$\mu^{\pm}$	7.8	8.5	80.8	60.0	48.5	85.2	98.6	98.6
	bkg0	$\mu^{\pm}$	8.0	11.7	79.2	57.6	45.6	84.8	91.8	97.6

Table 7.4.: Summarized results for the $\pi^+\pi^-$ final state with the 2D finder as input and with the 3D finder as input.

The test results for the $\pi^+\pi^-$ final states and the 2D finder as well as the 3D finder as input are summarized in Tab. 7.4. Compared to the previous resolution in the muon final state, the resolutions Δz are significantly deteriorated. The main reason for the decreased resolution are the different p_{T}- and θ-distributions. With the majority of low-p_{T} and shallow-θ tracks such a decrease in the resolution is expected. Another reason are the produced secondaries due to multiple scattering, ionization and interactions with the detector material. The secondaries itself create additional hits which leads to a similar noise for the track reconstruction of the primary tracks like the noise introduced by an increased background level.

For the two pion final state, a clear benefit from a training with pion tracks over training with muon tracks can be observed. Even in the test run with background, the pion networks achieve a better resolution than the muon trained networks without background (with both finders). By using pion training data, the above mentioned physics effects observed with the pions are already contained in the training data set. This allows the neural network to learn the necessary noise corrections to the helical track model. Among the muon networks, the worst resolution is achieved with the network trained without background. With background, the resolution is getting close to the resolution of the pion trained networks.

The reason for this counter-intuitive tendency of the muon network resolutions is that the background introduces additional fuzziness in the training data. Thus, training with background introduces noise which helps the muon network to recognize the noisy pion events as well. In the case without background, the muon network expects nearly ideal helical tracks without noise, which cannot be found in the pion data.

Comparing the finder efficiencies ϵ_{Finder} in Tab. 7.4 the 2D finder achieves $\epsilon_{\text{Finder}} \approx$

40 % and the 3D finder $\epsilon_{\text{3D Finder}} \approx 60\,\%$ track finding efficiency. While the neural network efficiency with 2D finder input is around $\epsilon_{\text{NN}} \approx 95\,\%$, only $\epsilon_{\text{NN}} \approx 80\,\%$ can be achieved with the 3D finder as input. This known issue can be solved by introducing the short track expert networks for low-p_{T} and shallow-θ tracks (see Sec. 6.5.2).

The rates for the z-cut at 20 cm already indicate the power of the neural network as a z-vertex trigger in two track events: with both finders, values of Gain2 $\approx 99\,\%$ can be achieved. For a one track trigger, with the pion networks a rate of Gain1 $\geq 85\,\%$ can be achieved. Although this rate is not optimal, the possibility to introduce such a one track trigger should be considered. Since there is no one track trigger yet, these events are definitely lost otherwise.

For the $\pi^+\pi^-$ final state the opening angle appears to be a suitable trigger. The column Op60 shows the rate of two track events where the θ-opening angle is smaller than 60°. In the case of the pion network with background, with the 2D finder input Op60 $\geq 97\,\%$ and with the 3D finder input Op60 $\geq 98\,\%$ can be achieved.

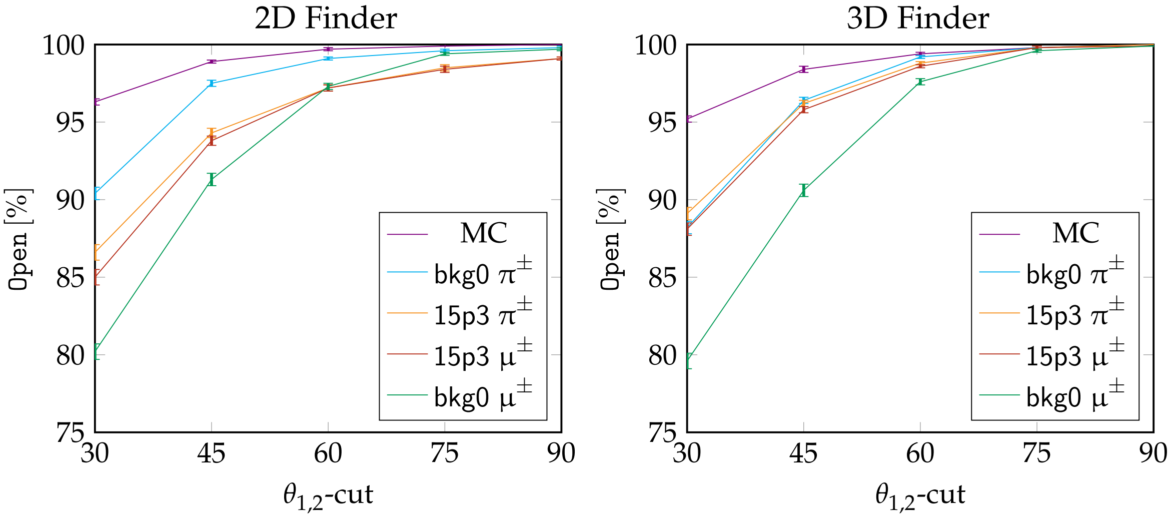


Figure 7.11.: Rate Open of ISR events with an opening angle $\theta_{1,2}$ between the two pions smaller than the values $\theta_{1,2}$-cut. For the found track pairs, the MC value of the rate Open and its estimates with the four different neural networks are shown.

A closer view at the opening angle rate Open, for different values $\theta_{1,2}$-cut is shown in Fig. 7.11. For the 2D finder and for the 3D finder as input, the curves show the Open-rate for the four different networks from Tab. 7.4. Additionally, the MC curve shows the true MC θ opening angle rate for the MC tracks matched to the first neural network (bkg0 $\pi^{\pm}$). Only one MC curve is shown, because they are almost indifferent (the only difference is the track count due to the different network efficiencies).

The MC curve shows that Open $\geq 95\,\%$ of the two track events have an opening angle smaller than $\theta_{1,2}$-cut $= 30^\circ$. Based on the estimates of the networks, the Open-rate for the pion networks at $\theta_{1,2}$-cut $= 30^\circ$ is already close to 90 % and rising for larger values of $\theta_{1,2}$-cut.

In the case of the 2D finder, the muon trained network without background has a steeper rise than the networks with background: for values of $\theta_{1,2}$-cut $\geq 60^\circ$, the training without background appears to be relatively better than the training with background. This crossing point of the curves at $\theta_{1,2}$-cut $= 60^\circ$ is absent in the plot with the 3D finder as input. Therefore, it is a result of the limited 2D finder efficiency and holds only for the case of ideal tracks contained within the full acceptance region of the CDC.

In the case of the 3D finder, the curves for the two pion trained network and the curve for the muon trained network with background are almost identical. Only the curve for the muon network without background has a significantly smaller Open-rate. This confirms the assumption that a training with background has a similar effect to a training with pion tracks. With one type of fuzziness represented in the training data, the neural network can transfer to other kinds of noisy input data. Hence, it demonstrates the robustness of the neural network method for the θ-estimation.

The next studies on the z-cut rates Gain1 and Gain2 with the two pion final state are shown in Fig. 7.12. As expected, the muon trained network without background achieves the worst efficiencies and for larger z-cut values even worse efficiencies than the background networks. In the case without background the networks with pion training data achieve significantly better Gain-rates than the muon networks. In the case with background, the pion trained network is still better, but the improvement is smaller. This reaffirms that adding background in the training data can mimic the noise created by the pions.

As expected for increasing values of z-cut, the rates are increasing but Gain1 is approaching 100 % slower than Gain2. It can be concluded that the z-cut is a very useful trigger for $\pi^+\pi^-$ ISR events. Different values of z-cut might be considered for the cases of single track and of two track events. In order to avoid large fractions of false positive single track background events, i.e. tracks displaced along the z-axis falsely estimated at the IP, a smaller z-cut value for the Gain1-rate might be adequate than for the Gain2-rate.

For the construction of an optimal two track trigger, a combination of Gain2 and Open can be considered: at least one of the tracks is required to have its z-vertex contained within a z-cut region and, additionally, the θ-angle between the two tracks must not exceed the value of $\theta_{1,2}$-cut. In order to find the optimal cut values, further studies with additional physics signal channels are required. Furthermore, the false positive rates have to be measured with simulated or real background events.

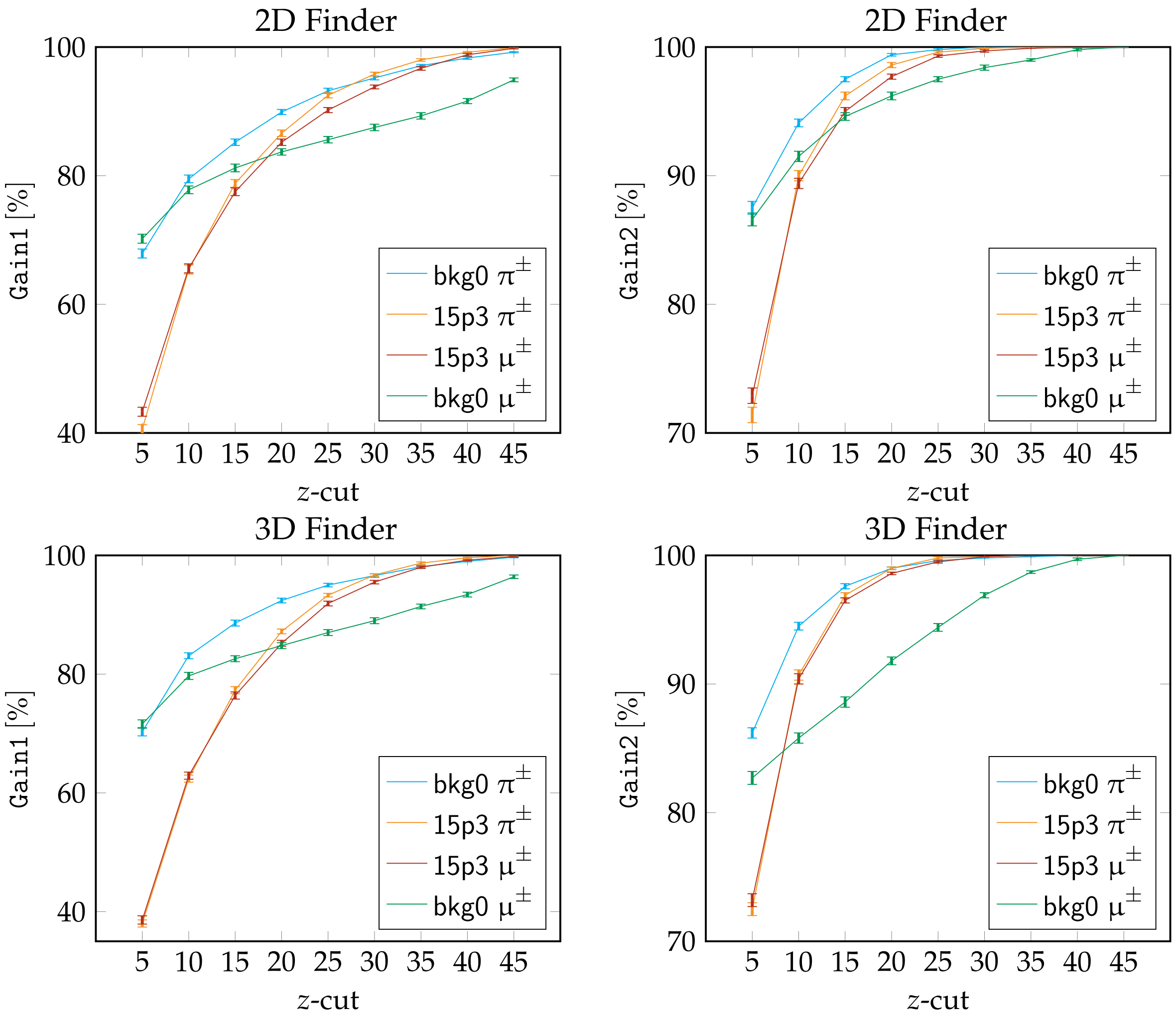


Figure 7.12.: With the 2D finder and the 3D finder as input, the rates `Gain1` and `Gain2` are shown for the ISR events with the two pion final state. In the rate `Gain1` one of the two pions is processed by the neural network and correctly estimated within the *z*-cut region. In the rate `Gain2` both pions are processed by the neural network and at least one of them is correctly estimated within the *z*-cut region.

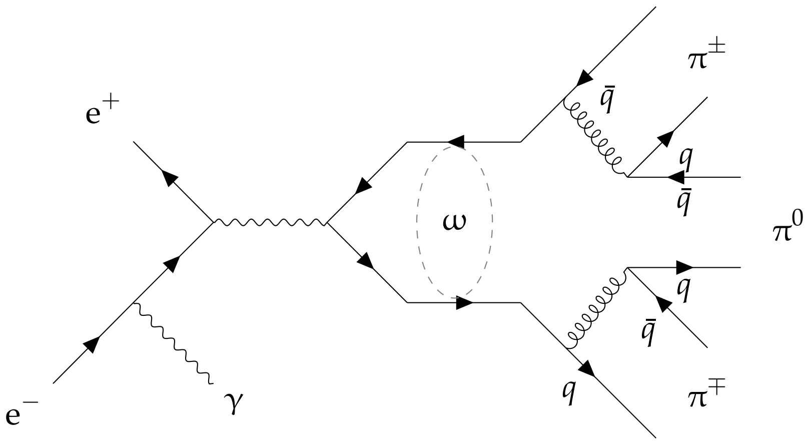


Figure 7.13.: Annihilation of e^+e^- into three pions $\pi^+\pi^-\pi^0$ after emitting a photon from the e^- initial state.

7.1.5. Final State $\pi^+\pi^-\pi^0$

The third final state of the ISR events studied here is the three pion final state $\pi^+\pi^-\pi^0$. With the two charged pions in the final state, two tracks are possibly visible in the CDC. Additionally, the neutral pion will most likely decay into two back-to-back photons, which can be reconstructed from the ECL. An exemplary Feynman diagram for the three pion production in ISR events, with the ω meson as an intermediate resonance state, is shown in Fig. 7.13. The narrow resonance of the ω meson decays in more than 90 % of the cases into the three pion final state. The hadronic process $\omega \to \pi^+\pi^-\pi^0$ is composed of two subsequent hadronic decays. Hence, an additional intermediate state, after the first hadronic decay can be defined as $\omega \to \rho\pi$. The ρ meson continues to decay into a pion pair in the second hadronic decay. In this case, several combinations are possible

$$\omega \to \rho^+\pi^- \qquad (\rho^+ \to \pi^+\pi^0)\pi^- \tag{7.13}$$

$$\omega \to \rho^-\pi^+ \qquad (\rho^- \to \pi^-\pi^0)\pi^+ \tag{7.14}$$

$$\omega \to \rho^0\pi^0 \qquad (\rho^0 \to \pi^+\pi^-)\pi^0 \tag{7.15}$$

which all end up in the same final state $\pi^+\pi^-\pi^0$. In leading order, the observed total decay rate for $\omega \to \pi^+\pi^-\pi^0$ is the sum of these three possible decay paths.

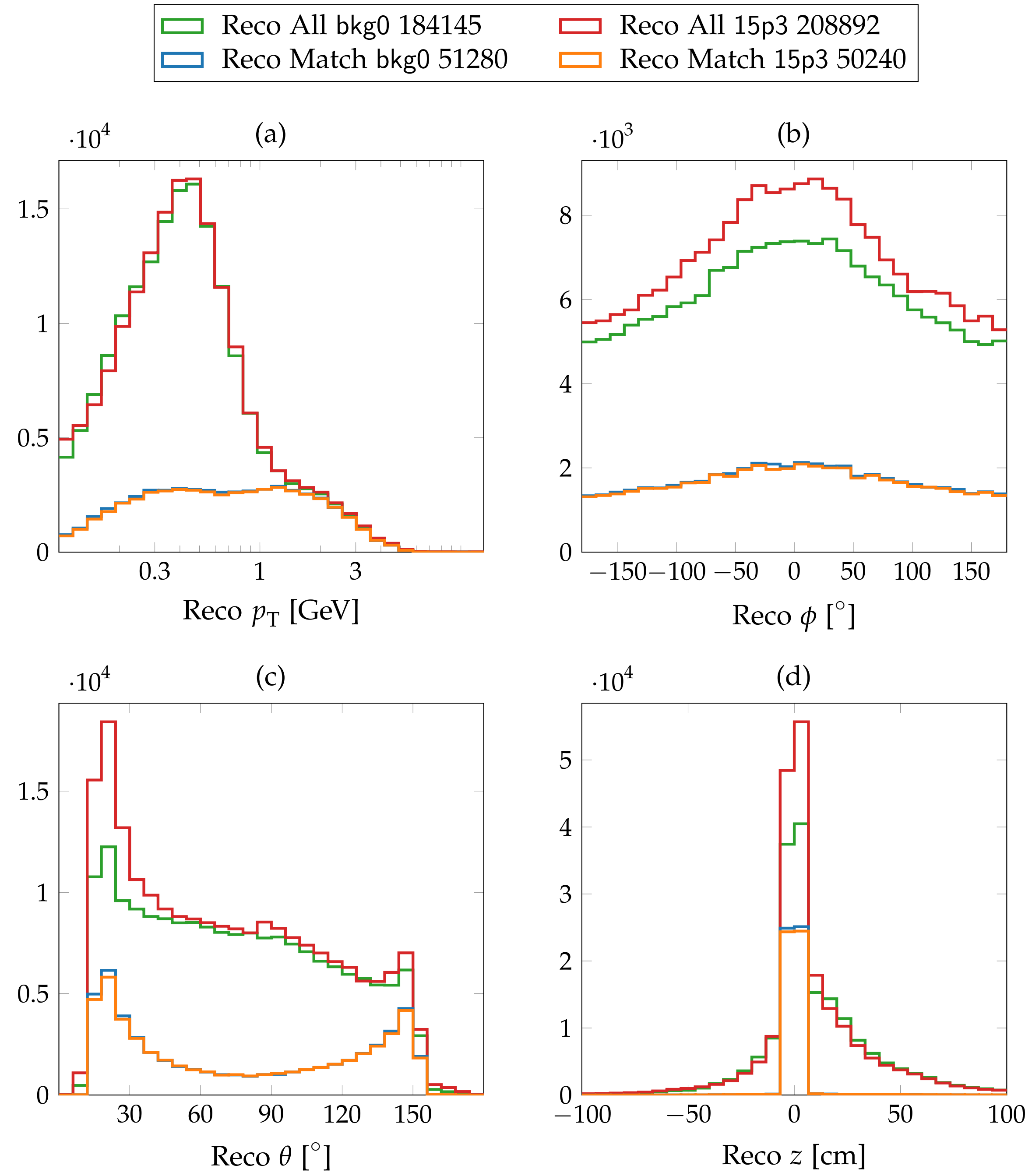


Figure 7.14.: Offline reconstructed track parameter distributions of the simulated ISR event with three pions in the final state ($\pi^+\pi^-\pi^0$). For the cases without background bkg0 and with the Phase 3 background 15p3 the distribution of all reconstructed tracks as well as the reconstructed tracks matched to a primary MC particle are shown.

As before in the two pion final state $\pi^+\pi^-$, only two tracks are visible as charged tracks in the CDC. However, the topology is different since in this three pion final state a fraction of the momentum and energy is carried away by the neutral pion. The track parameter distributions of the reconstructed tracks with the three pion final state are shown in Fig. 7.14. Similar to the final state $\pi^+\pi^-$, a large peak appears in the low-p_T region followed by a shallow peak in the high-p_T region (Fig. 7.14 (a)). After requiring a matched MC tracks, the low-p_T peak mostly disappears. It is replaced by a flat distribution in this log-scale plot in the region of $p_\mathrm{T} \in [0.3, 2]$ GeV. The reason for the flattening is the energy and momentum carried away by the third decay partner (π^0).

The track counts are very different from the $\pi^+\pi^-$ state (compare Fig. 7.9): fewer tracks are reconstructed in total, however, the number of reconstructed tracks with a match to a primary MC particle is larger. Since the same number of events were simulated in both cases, the smaller total number of reconstructed tracks shows that less secondaries are created. The apparently improved reconstruction efficiency is a result of the third decay partner: due to energy and momentum conservation, the $\pi^+\pi^-$ of a three body decay $\pi^+\pi^-\pi^0$ are more likely to be contained within the acceptance region of the CDC. Furthermore, the charged pions with shallow polar angles θ carry a smaller fraction of the energy and are thus less likely to create secondary tracks visible in the CDC.

The distribution in the polar angle θ is shown in Fig. 7.14 (c). For the reconstructed and matched tracks, peaks at shallow-θ angles can be observed. However, they are less pronounced than in the $\pi^+\pi^-$ final state in Fig. 7.9 (c). The θ-distribution of the secondaries (reconstructed tracks not matched to MC particles) is smaller and flatter than for the final state $\pi^+\pi^-$. In the ϕ-distribution shown in Fig. 7.14 (b), the reconstructed tracks have the same broad peak pointing outwards in the accelerator plane around $\phi \approx 0^\circ$. The z-distribution in Fig. 7.14 (d) shows a reduction of the secondary tracks with displaced z-vertices, relative to the number of tracks from the IP.

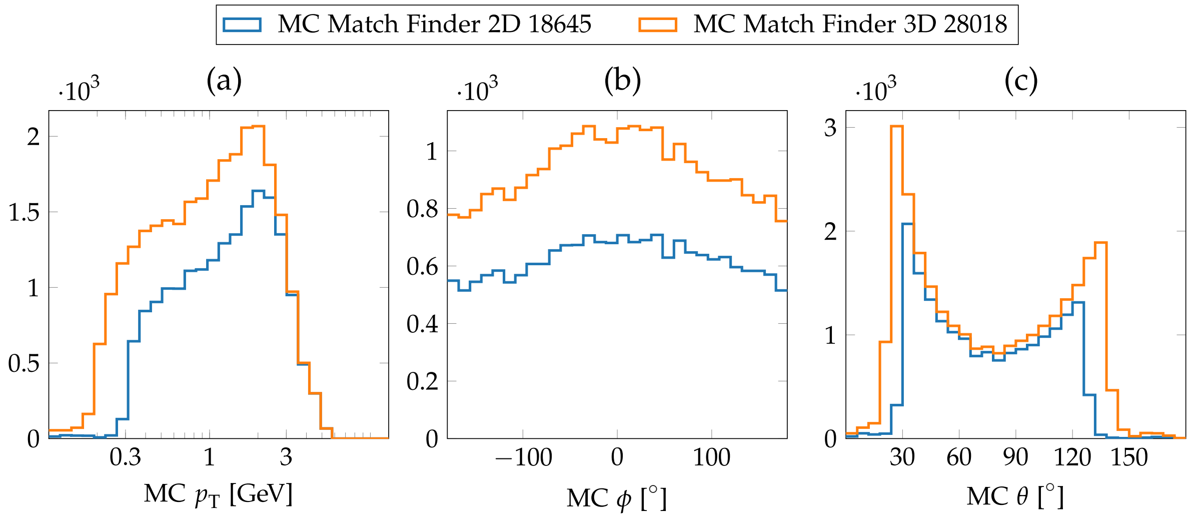


Figure 7.15.: MC track parameter distributions of the charged primary MC particles with a match to a finder track (2D or 3D) for the final state $\pi^+\pi^-\pi^0$.

The distributions of the MC tracks which are found by the track finders, shown in Fig. 7.15, also differ from the previous two pion final state. In agreement with the reconstructed track distributions, in the p_T-distribution in Fig. 7.15 (a), more tracks are found with medium p_T values (i.e. between the peaks at 0.3 GeV and 2 GeV in Fig. 7.10 (a)). Throughout the low-p_T region and at shallow polar angles θ in Fig. 7.15 (c), the improved track finding efficiency of the 3D finder can be observed. As before, in the ϕ-distribution shown in Fig. 7.15 (b), no structural differences between the 2D finder, the 3D finder, and the reconstructed tracks can be observed.

To begin the analysis of the network results for the $\pi^+\pi^-\pi^0$ final state, the results are summarized in Tab. 7.5. With the 3D finder as input, the rows are ordered by the resolution Δz. For better comparability, in the table for the 2D finder the same ordering of entries is selected as for the 3D finder, although the resolution for bkg0 $\mu^\pm$ is slightly better than 15p3 $\mu^\pm$.

The averaged resolutions Δz are slightly improved compared to the two pion final state. Due to the smaller number of secondary tracks, the pion tracks appear cleaner in this test data sample. Furthermore, the typical θ-angles of the pion tracks are slightly shifted towards the barrel region, where a better resolution of the track parameters can be achieved. This is also the reason for the slightly improved results for the muon network without background compared to the muon network with background.

The finder efficiencies ϵ_{Finder} are similar to the finder efficiencies in the two pion sample. The 2D finder achieves $\epsilon_{\text{2D Finder}} \approx 39\,\%$ and the 3D finder $\epsilon_{\text{3D Finder}} \approx 57\,\%$. Note that the reconstruction efficiency in this sample is larger than in the $\pi^+\pi^-$ final state. Therefore, the same finder efficiency ϵ_{Finder} still corresponds to a larger total number of found tracks. The neural network efficiencies are similar to the two

	Bkg$_{\text{test}}$	Train	$\Delta\theta_{\text{MC}}$	Δz_{MC}	$\epsilon_{\text{NN}}^{\text{MC}}$	$\epsilon_{\text{Finder}}^{\text{MC}}$	$\epsilon_{\text{Trg}}^{\text{MC}}$	Gain1	Gain2	Op60
			[°]	[cm]	[%]	[%]	[%]	[%]	[%]	[%]
2D	bkg0	$\pi^{\pm}$	4.2	5.1	97.4	36.4	35.4	92.9	99.1	95.8
	15p3	$\pi^{\pm}$	6.1	8.0	95.7	39.1	37.4	89.4	98.8	94.1
	15p3	$\mu^{\pm}$	6.2	8.4	95.7	39.1	37.4	88.5	98.1	94.0
	bkg0	$\mu^{\pm}$	6.2	8.3	98.6	36.4	35.9	88.5	97.4	94.2
	Bkg$_{\text{test}}$	Train	$\Delta\theta_{\text{MC}}$	Δz_{MC}	$\epsilon_{\text{NN}}^{\text{MC}}$	$\epsilon_{\text{Finder}}^{\text{MC}}$	$\epsilon_{\text{Trg}}^{\text{MC}}$	Gain1	Gain2	Op60
			[°]	[cm]	[%]	[%]	[%]	[%]	[%]	[%]
3D	bkg0	$\pi^{\pm}$	5.0	4.8	75.9	54.2	41.1	94.1	99.0	95.8
	15p3	$\pi^{\pm}$	8.0	8.1	78.0	57.4	44.8	88.3	99.0	95.6
	15p3	$\mu^{\pm}$	7.8	8.3	78.0	57.4	44.8	86.6	98.6	95.3
	bkg0	$\mu^{\pm}$	6.6	9.4	75.9	54.2	41.1	88.6	93.3	93.8

Table 7.5.: Summarized results for the $\pi^+\pi^-\pi^0$ final state with the 2D finder and with the 3D finder as input.

pion case as well. In the case with background and the 2D finder $\epsilon_{\text{NN}} \approx 95\,\%$ is achieved and with the 3D finder $\epsilon_{\text{NN}} \approx 78\,\%$. As mentioned earlier, the solution for the limited neural network efficiency is the use of short track expert neural networks presented in Sec. 6.5.2.

With the fixed z-cut at z-cut $= 20\,\text{cm}$, the z-cut rate Gain2 is a very efficient trigger and reaches almost Gain2 $\approx 100\,\%$. The rate Gain1 shows that even in the case of single track events a reasonable efficiency around Gain1 $\approx 90\,\%$ can be achieved. By triggering on Op60 (a cut on the θ opening angle with $\theta_{1,2}$-cut $= 60^{\circ}$) a triggering efficiency of Op60 $\approx 95\,\%$ can be achieved. Since the opening angle is slightly larger, the rate Op60 is not as optimal as for the two pion final state. However, the θ-cut results are still reasonable to be considered in the global trigger decision for the three pion final state. Due to the improved resolution compared to the two pion state, the Gain-rates provide slightly better trigger results here.

To inspect the quality of the cut on the opening angle, Open is plotted for different values of $\theta_{1,2}$ -cut in Fig. 7.16. Comparing the MC curves here with the MC curves in Fig. 7.11 shows that the typical opening angle between the two charged pions in the three pion final state events is indeed larger than in the two pion final state events. Note that the rates for Open are not normalized to the MC values here. Hence, the best rate that can be achieved by the neural network corresponds to the MC curve.

Despite the overall lower Open-rates, the characteristic structure is identical to the previous two pion study. In the test with the 2D finder, the pion network without background achieves better results than the muon network without background. In the case with background, the pion network and the muon network agree within the errors. In the 3D finder plot, only the muon trained network without background is significantly worse than the other networks. The other three networks agree within

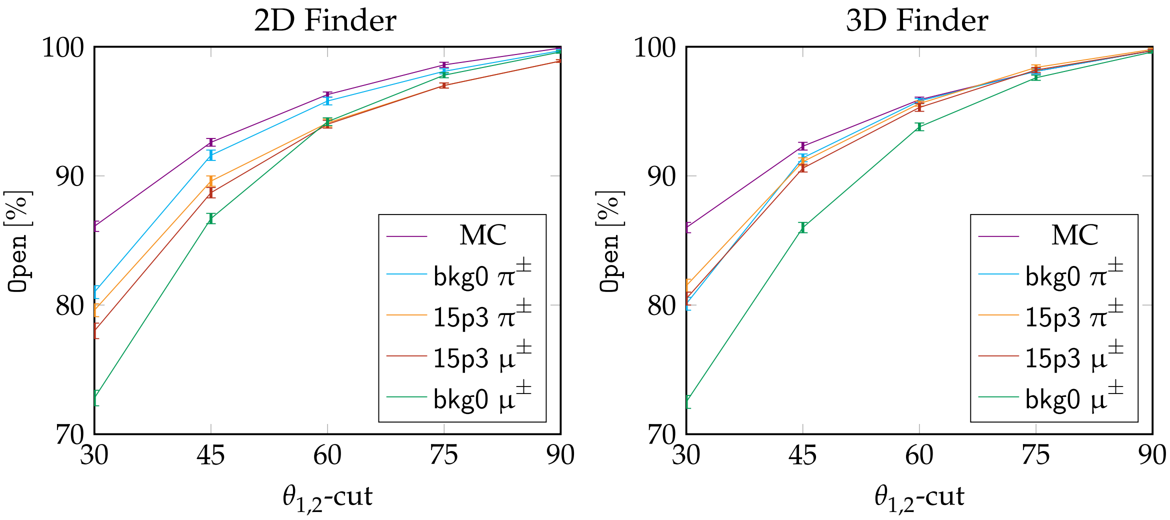


Figure 7.16.: Rate Open of ISR events with an opening angle $\theta_{1,2}$ between the two charged pions smaller than the value $\theta_{1,2}$-cut. For the found track pairs the MC value of the rate Open and its estimates with the four different neural networks are shown.

the errors (pion training without background, pion training with background and muon training with background).

The z-cut rates Gain1 and Gain2 for different values of z-cut are shown in Fig. 7.17. All the observed structures, have also been observed in the two pion case shown in Fig. 7.12. The overall best performance is achieved with the network trained on pions. In the case without background the improvement with the pion network over the muon network is large compared to the case with background.

7.1.6. Summary of the ISR Analysis

The studies with initial state radiation events have demonstrated that efficient trigger logics for these event signatures with low track multiplicities can be built based on the neural network single track estimates. Compared to the case without a 3D trigger, where only three track triggers are feasible, the introduction of two- and one-track triggers is a huge improvement. The whole sector of low multiplicity physics signal events becomes accessible for the precision measurements at Belle II.

The characteristic topology of the hadronic final states $\pi^+\pi^-$ and $\pi^+\pi^-\pi^0$ can be used to construct signal specific event triggers. Especially the cut on the opening angle between the two estimated tracks has turned out to be a suitable new trigger logic. With a cut of $\theta_{1,2}$-cut $= 60^\circ$ efficiencies of $\geq 97\,\%$ for the $\pi^+\pi^-$ final state and $\geq 94\,\%$ for the $\pi^+\pi^-\pi^0$ final state have been measured. In z-cut based triggers, where at least one track is required to be estimated within a cut region of z-cut $= 20\,$cm, an efficiency of $\geq 98\,\%$ for two track events and $\geq 89\,\%$ for one track events can be achieved.

The preliminary cut values have to be further optimized with different signal and background types. Especially the contamination due to false positives has to be studied with a realistic distribution of background tracks displaced along the z-axis. For two track events, it is suggested to use both trigger logics in combination, which means to require a z-cut and a $\theta_{1,2}$-cut. Using a combination of the two logics increases the range of cut values that can be optimized and thus provides more options to achieve an improved efficiency and, at the same time, a reduced false positive rate.

In the final state $\pi^+\pi^-$ and $\pi^+\pi^-\pi^0$ the studies with the 3D finder compared to the 2D finder have shown an increase of the track finding efficiency from $\epsilon_{\text{Finder}} \approx 40\,\%$ with the 2D finder to $\epsilon_{\text{Finder}} \approx 60\,\%$ with the 3D finder. This means, the number of found tracks with the 3D finder is $\approx 50\,\%$ larger than with the 2D finder. The same relative increase in the number of found tracks by $\approx 50\,\%$ was observed in previous studies in Sec. 6.5.1 with simulated single track events from the IP. In particular, the track finding for curved tracks with low transverse momenta and for tracks with shallow polar angles is significantly improved with the 3D finder. As shown in Sec. 6.5.2, this improved 3D finder efficiency can be fully used by the neural network in a setup with experts for short tracks.

The training with pion tracks has turned out to be well suited for the estimation of ISR events with pions in the final state. If these pion tracks are used as training data instead of muon tracks, the neural network can learn to handle the fuzziness in the events created by the pion tracks. However, a training with muon tracks and with background can partially compensate for the resolution loss observed in the pion test data. This means that background can add a fuzziness to the muon training data which mimics the fuzziness in the pion data. Since the observed resolution loss with pion trained networks on muon test data is negligible, it is suggested to use the pion training instead of muon training in combination with the latest background expectations for the future Belle II neural network trigger updates.

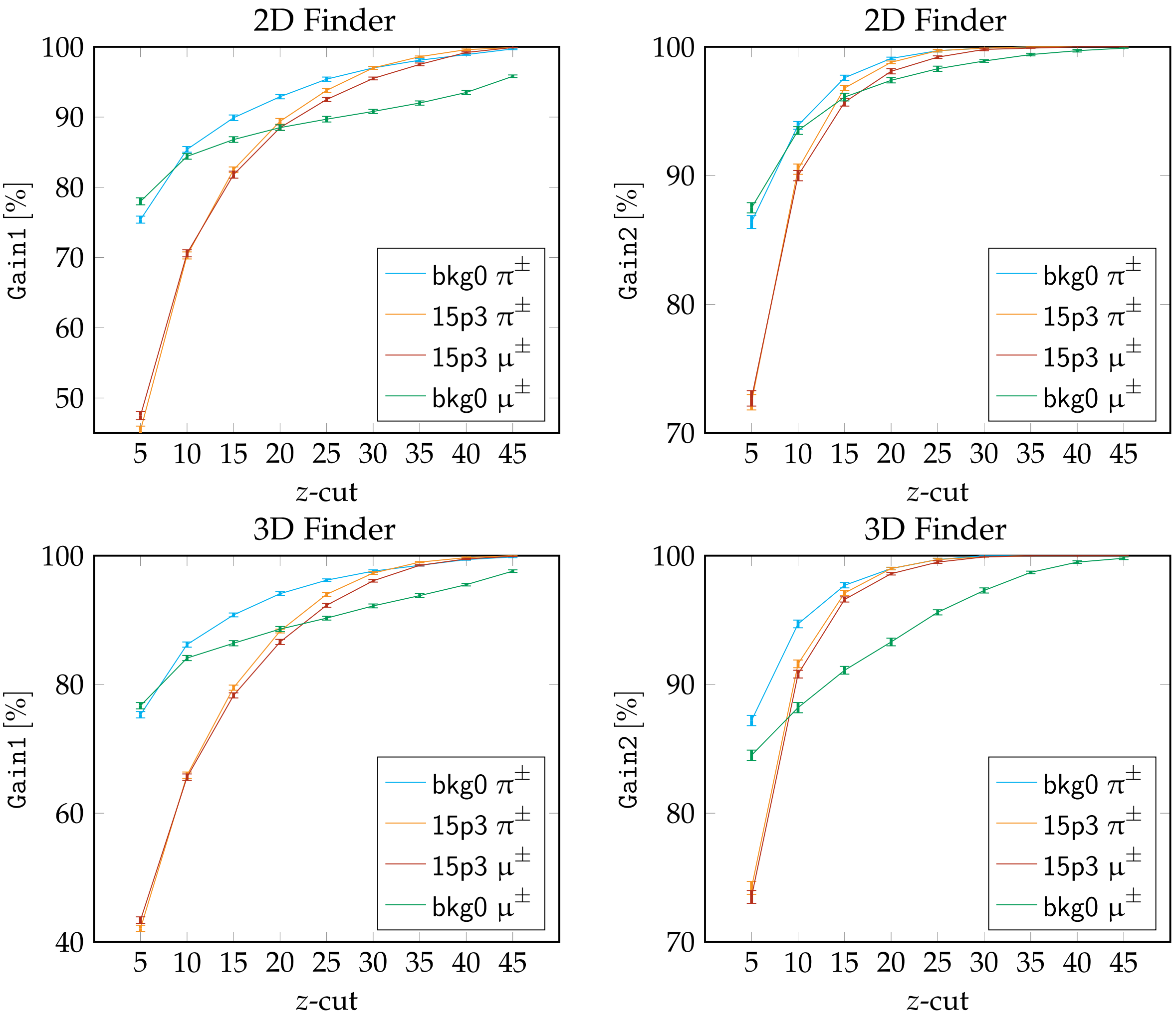


Figure 7.17.: Rates `Gain1` and `Gain2` for ISR with three pions in the final state ($\pi^+\pi^-\pi^0$). In `Gain1` one charged pion is processed by the neural network and estimated within the z-cut range. In `Gain2` both charged pions are processed by the neural network and at least one is estimated within the z-cut range.

7.2. Belle II Run Data

At the time of the writing of this thesis, Belle II is already running in Phase 3 and taking data from first physics runs. However, the luminosity is still much lower than the anticipated design luminosity of Belle II and the background conditions are not yet optimal. During the slow increase of the luminosity, new problems in the subdetector systems can be detected and countermeasures installed. This also applies to the neural network trigger, which is already installed in the electronics hut of Belle II. The summarized trigger information of the hardware neural network is sent to the GDL where the results can be integrated into combined trigger logics. Additionally, a longer format of output information is sent via the B2Link protocol to the data acquisition (DAQ) system of Belle II. In the normal physics runs, this debug information is only recorded with a prescaling factor to reduce the size of the stored data. Additionally, in the online system it is used at the express reco level and in the high level trigger (HLT) to monitor the proper functioning of the hardware. The neural network presently running in the hardware was trained with the background type 15p2 and with an enlarged z-estimation range: $z \in [-100, 100]$ cm.

Besides the studies presented in this thesis, for the purpose of monitoring, an unpacker and a module for data quality monitoring (DQM) have been developed. These are already installed for online monitoring and can additionally be used for offline analysis where the hardware neural networks are compared with software neural networks as well as with the offline reconstruction. Recent hardware studies have shown that the hardware neural network already achieves results comparable to the simulated software network. However, the present bug fixing status will not be discussed here in detail, as it is ongoing work in progress. Overall, the hardware neural network is almost completed. It is expected that the neural network based trigger logics can be fully used during the next data taking Phases of Belle II, with the effect of a significantly improved efficiency for low multiplicity events.

For the studies presented in this section, recorded Belle II data is loaded into the simulation framework basf2. Starting with unpacked CDC hits, the full trigger simulation is run, including a simulated TSF, simulated track finders and simulated neural networks. At the finder level, the 2D finder and the 3D finder are compared on real data. At the neural network level, the optimal training parameters are studied, where in particular the training with different background types is presented here. As an example of the recorded z-distribution at Belle II, Fig. 7.18 (published in [84]) shows the distribution of the z-vertices of offline reconstructed tracks (in experiment 7, run 4048). This distribution was created by dividing the z-histogram of reconstructed tracks, obtained from the DQM module, by the run time in seconds. The luminosity in this run was still very low ($\approx 500\times$ lower than the design luminosity of Belle II). Thus, the rates are only in the order of $\mathcal{O}(\text{Hz})$. This run was taken without data suppression by the HLT. Therefore, a large distribution of background tracks with z-vertices displaced from the IP can be observed.

In the normal data taking mode, the HLT runs the precise offline reconstruction which allows to suppress these displaced tracks prior to the long term offline storage.

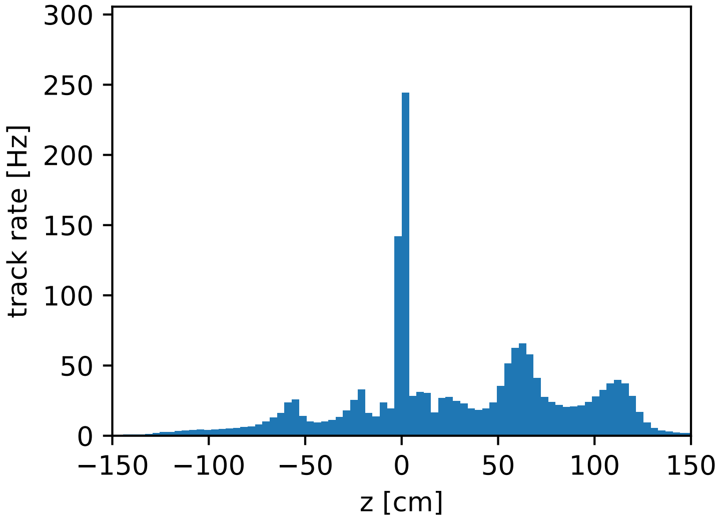


Figure 7.18.: Distribution of the offline reconstructed z-vertex single track rates in an early run of Belle II (exp 7 run 4048) [84].

However, the HLT can only process the events that have passed the L1 trigger with its maximum output rate of 30 kHz. The HLT skimmed data sets are suitable for physics analysis and, in the case presented here, to study the performance of the neural network on physics signal events. However, for the study of the suppression of real background tracks with displaced z-vertices, recorded runs without HLT skimming are required. Therefore, two different run data sets are analyzed in the following studies: a run without HLT skimming (exp 8, run 2413) and a run with HLT skimming (exp 8, run 3092).

7.2.1. Collision Events

The first study with Belle II data is carried of with the recorded data from exp 8 run 3092. In this run, the HLT was actively skimming the events prior to the recording for offline analysis. Therefore, most of the events contained in this data set have tracks with vertices at the IP, where the two beams are colliding. The majority of these IP events are QED interactions with a large cross section. Especially the Bhabha scattering rate (elastic e^+e^- scattering) contributes with a large cross section. However, corresponding to their cross sections, all types of physics interactions can be expected in the collision events. According to the internal Belle II log entry of that run, at the start of the run the beam currents in the HER and the LER were at 0.25 A with 789 bunches in each ring, a luminosity of $2.7 \times 10^{33}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$, and a trigger rate of 1.1 kHz. During a total run time of 21 minutes and 17 seconds an integrated luminosity of about $3.6 \times 10^{36}\,\mathrm{cm}^{-2}$ was collected, with a total of 1458038 events.

After unpacking the hits of all the Belle II subdetectors from the recorded data set, the precise offline reconstruction and the trigger simulation is carried out. Hence, the real CDC hits are used as input to the CDC trigger system. Following the simulation of the TSF, the official 2D finder as well the 3D finder are simulated as input for the neural network. The distribution of the offline reconstructed track parameters for this run is shown in Fig. 7.19. It shows all reconstructed tracks, as well as the reconstructed track parameters for the tracks with a match to a 2D finder or to a 3D finder track.

Comparing the overall rates, the 3D finder can find significantly more tracks than the 2D finder. Fig. 7.19 (a) shows two peaks in the p_T-distribution: a high momentum peak at $p_\mathrm{T} \gtrapprox 3\,\mathrm{GeV}$ and a low momentum peak at $p_\mathrm{T} \approx 500\,\mathrm{MeV}$. At the high momentum end ($p_\mathrm{T} \gtrapprox 4\,\mathrm{GeV}$), both finders nearly achieve full efficiency, i.e. they find all the reconstructed tracks. Towards lower momentum, the 3D finder has a higher efficiency than the 2D finder, however, as expected, it is still less efficient than the precise offline reconstruction. The ϕ-distribution in Fig. 7.19 (b) has the broad distribution around $\phi \approx 0^\circ$, which is expected since the bunches are crossed in the horizontal plane. Furthermore, a small dip appears in the ϕ-distributions of the matched tracks at values of $\phi \approx -50^\circ$ which indicates an inefficiency of the CDC readout in this ϕ-region.

In Fig. 7.19 (c) the improved efficiency of the 3D finder for shallow polar angles θ can be confirmed. The observed θ-distribution of the reconstructed tracks has peaks for shallow-θ angles in the forward and in the backward direction. This means that a large fraction of the tracks are in a region not fully covered by the CDC and thus difficult to handle by the CDC trigger. Such a polar angle distribution is typical for collision events, with a boost in z-direction. The z-distribution in Fig. 7.19 (d) shows a narrow peak in the IP region, with only few displaced tracks. The huge background tails were skimmed by the HLT and are not contained in this data set. Although a few of the tracks displaced in z remain, as can be seen in the quite shallow peak at $z \approx 60\,\mathrm{cm}$.

The shown 2D finder and 3D finder tracks are used as input for the simulated neural networks. These neural networks were trained with the largest training data size (size NNE in Tab. D.1) and with four different background types (see Chapter 4). This allows to search for the optimal simulated background for real detector data and additionally to compare the performance of the 2D finder with the 3D finder. In order to study the resolution for signal tracks from the IP only, an additional cut was used for the calculations of the resolutions and the basic single track trigger efficiencies. Out of the matching pairs of reconstructed tracks with neural network tracks, the following cuts are applied on the z-vertex position and on the transverse vertex displacement d_0:

$$|z_\mathrm{Reco}| \leq 1\,\mathrm{cm} \tag{7.16}$$

$$d_0 \leq 1\,\mathrm{cm} \tag{7.17}$$

Therefore, the small tails visible in Fig. 7.19 (d) do not disturb the averaged resolution estimates and the track trigger efficiencies.

In the result summary tables, additionally the TS-rates in the SLs and the pattern-rates are shown without a cut on the IP. This allows to study the effect of the track vertex positions on the track finders and the neural network. Furthermore, the event based trigger efficiencies will be shown, where the IP cut is only used internally in the respective cut logics. Besides the signal rate `Gain`, also the rates `Fake` and `Loss` are meaningful here: despite the HLT skimming, the data set still contains tracks displaced from the IP. For the z-cut based rates, multi track events are distinguished in: single track events (`Gain1, Fake1, Loss1`), two track events (`Gain2, Fake2, Loss2`) and events with more than two tracks (`Gain3, Fake3, Loss3`). The values shown in the table use the default value of z-cut $= 20\,\mathrm{cm}$. Further details on the definitions of the cuts can be found in Sec. 6.4.6. To test the opening angle distribution, additionally the rate of the polar opening angle `Open` from the previous ISR studies (see Sec. 7.1.2) are added to the summary tables.

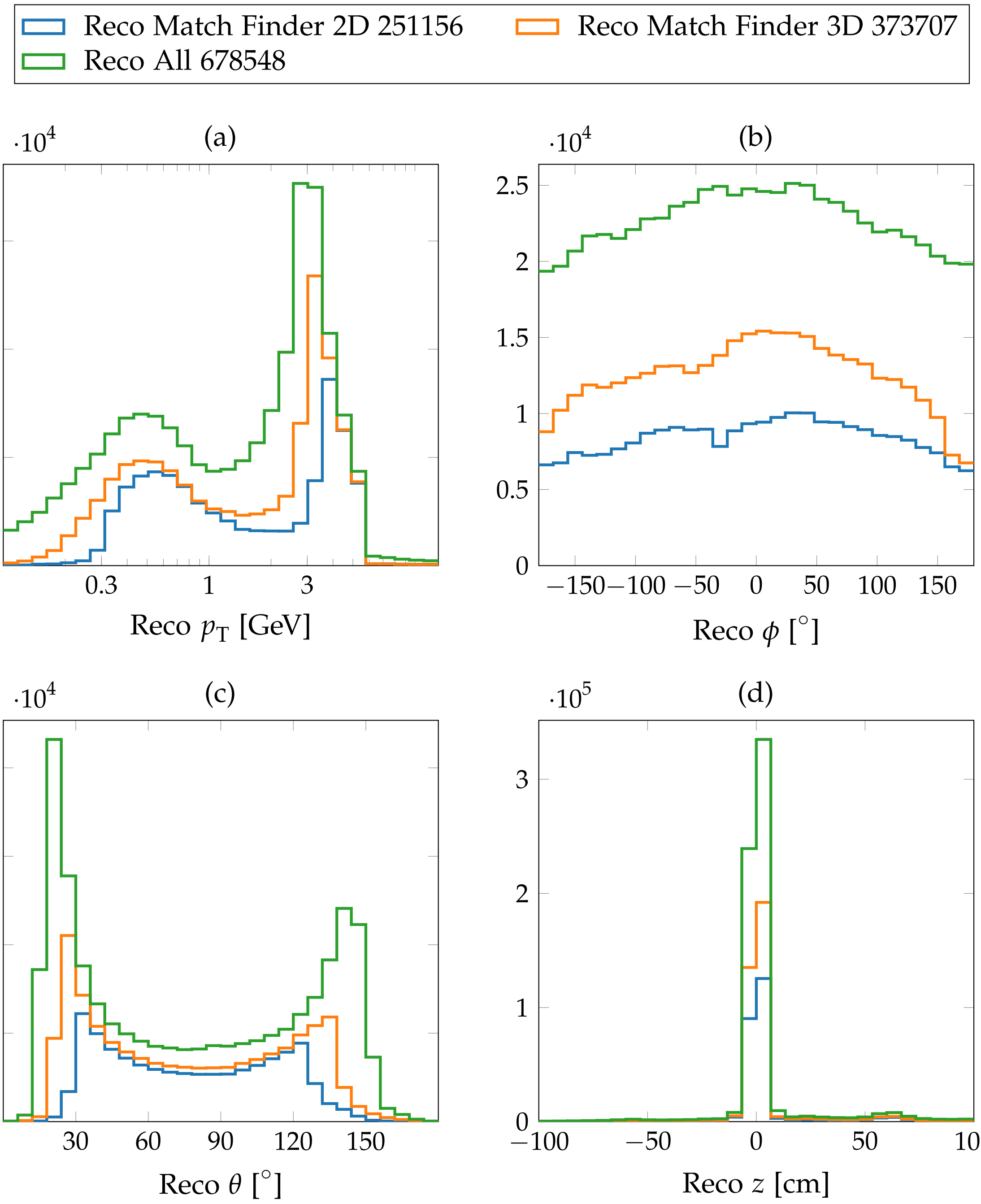


Figure 7.19.: Reconstructed track parmeters in real data from the Belle II detector in a run with HLT skimming (exp 8 run 3092). All reconstructed tracks as well as reconstructed tracks with a match to a 2D finder or a 3D finder track are shown.

2D

	$\text{Bkg}_{\text{train}}$	$\Delta\theta^{\text{Reco}}$	Δz^{Reco}	$\epsilon_{\text{NN}}^{\text{Reco}}$	$\epsilon_{\text{Finder}}^{\text{Reco}}$	$\epsilon_{\text{Trg}}^{\text{Reco}}$	Op60
		[°]	[cm]	[%]	[%]	[%]	[%]
(a)	15p3	4.0	5.8	91.4	39.2	35.9	59.2
	12p3	4.4	5.5	91.4	39.2	35.9	59.2
	15p2	5.0	6.3	91.4	39.2	35.9	59.5
	bkg0	6.2	9.0	91.4	39.2	35.9	57.4

	$\text{Bkg}_{\text{train}}$	SL1	SL3	SL5	SL7	p7	p11	p13	p14	p15
		[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]
(b)	15p3	91.7	96.5	99.0	82.4	8.3	3.5	1.0	17.6	69.6
	12p3	91.7	96.5	99.0	82.4	8.3	3.5	1.0	17.6	69.6
	15p2	91.7	96.5	99.0	82.5	8.3	3.5	1.0	17.5	69.6
	bkg0	91.7	96.5	99.0	82.5	8.3	3.5	1.0	17.5	69.6

	$\text{Bkg}_{\text{train}}$	Gain1	Gain2	Gain3	Loss1	Loss2	Loss3	Fake1	Fake2	Fake3
		[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]
(c)	15p3	96.7	99.9	100.0	3.8	0.2	0.0	16.8	27.6	36.4
	12p3	95.5	99.8	99.9	3.0	0.1	0.1	13.0	22.0	33.3
	15p2	93.5	99.5	99.7	2.4	0.1	0.0	10.3	19.6	29.5
	bkg0	90.8	98.8	98.7	1.5	0.0	0.0	6.0	9.7	14.7

Table 7.6.: Neural network results with the 2D finder as input, tested on HLT skimmed data recorded with Belle II in experiment 8, run 3092.

2D Finder Input

The results with the 2D finder as input are shown in Tab. 7.6. The features analyzed with this data set are the track resolutions and track trigger efficiencies in the first table Tab. 7.6 (a) , the hit rates in the SLs and the pattern rates in the second table Tab. 7.6 (b) and the z-cut based event efficiencies in the third table Tab. 7.6 (c). All results are sorted by the resolution $\Delta\theta$.

The best resolutions are achieved with the Phase 3 backgrounds 12p3 and 15p3, which indicates severe background conditions in this early run. Note that the Phase 3 backgrounds are simulated under the assumption of the full Phase 3 luminosity. Using the Phase 2 background 15p2 in the training, the achieved resolution is still much better than in the case of training without background. The 2D finder efficiency reaches $\epsilon_{\text{2D Finder}} \approx 40\,\%$, similar to the previous ISR studies with the hadronic final states. The neural network efficiency reaches $\epsilon_{\text{NN}} \approx 91\,\%$, while almost 100 % were achieved with the 2D finder in MC events. The efficiencies with the different background types are very similar here, because all networks are tested on the same data set while the different background types were only used during the training. This means that hit selection method, via the training of the ID-range histograms (see e.g. Fig. 6.15), is quite robust to background.

For the case of two track events, the column Op60 shows the rate of events where the opening angle between the two tracks $\theta_{1,2}$, estimated by the neural network, is smaller than the cut value $\theta_{1,2}$-cut $= 60^{\circ}$. With Op60 $\approx 60\,\%$, the opening angle does not appear to be a useful trigger for the general two track collision events tested here.

Table 7.6 (b) shows the hit rates in the SLs and the pattern-rates of the used expert neural networks without the additional IP cut (Eq. 7.16 and Eq. 7.17). Hence, the outliers in z, which are found by the 2D finder and the 3D finder (see Fig. 7.19 (d)) are included in the rates. In agreement with the observed p_{T}-distribution, where many tracks have low-p_{T}, the TS hit rate in the outermost SL (SL7 $\approx 80\,\%$) is smaller compared to TS hit rates in the inner SLs. Although the rates in inner SLs do not reach 100 %, as observed in the previous MC studies. In particular, the pattern-rate p7 $\approx 8\,\%$ indicates that the 2D finder finds tracks with larger displacements in z and d_0, where it is geometrically possible for the first stereo SL not to receive a hit. The non-zero pattern-rates p11 $+$ p13 $\approx 4.5\,\%$ indicate that the inefficiencies in the real CDC are larger than the ones modeled in the Monte Carlo.

The z-cut based event rates Gain, Fake, and Loss for single track events, two track events and for events with more than two tracks are shown in Tab. 7.6 (c). For signal events with two or more tracks, Gain2 and Gain3 turn out to be ideal triggers. Even in the case of single track events, the neural network trained with the background 15p3 reaches an excellent efficiency of Gain1 $\approx 97\,\%$. In this HLT skimmed data set, also the Loss-rates are very low. As expected, the rate Loss reduces with a larger track count in the events: with more tracks estimated at the IP, the chance decreases that none of these estimated tracks truly originates from the IP. In contrast, the rate Fake increases with the number of tracks: with more estimated tracks, which are all truly displaced from the IP, the chance increases that at least one of them is falsely estimated within the region z-cut.

More details on the z-cut rates can be seen in Fig. 7.20. With the 2D finder as input, it shows the three rate types Gain, Fake and Loss for single track events and for two track events. At the default of z-cut $= 20$ cm, which corresponds to the values listed in Tab. 7.6 (c), the best efficiencies (Gain-rates) are achieved by a training with 15p3. Only for the smallest cut value shown, at z-cut $= 5$ cm, the networks trained with lower background levels achieve better results (12p3, 15p2). Although the low efficiency Gain at such a small value of z-cut is not preferred, this indicates that the background level of 15p3 is not yet optimal for the neural network training. Training without background results in the worst Gain-rates.

The rates Fake1 and Fake2 continuously increase towards larger values of z-cut. This is correct, as the cut on the reconstructed z-vertex positions of the tracks to be designated as outliers is at 40 cm (see Sec. 6.4.6). Note that once the z-cut reaches 40 cm as well, a correct estimation of the z-vertex position results in an increase of the Fake-rates. Hence, the cut for true background tracks could be optimized according to the observed background topology. This rate is important to describe the background rate relative to the total number of background events. Correspondingly, the inverse of the Fake-rates, i.e. $100\,\% -$ Fake is a measure of the background suppression rate. The background type 15p3 results in the largest Fake-rates, while

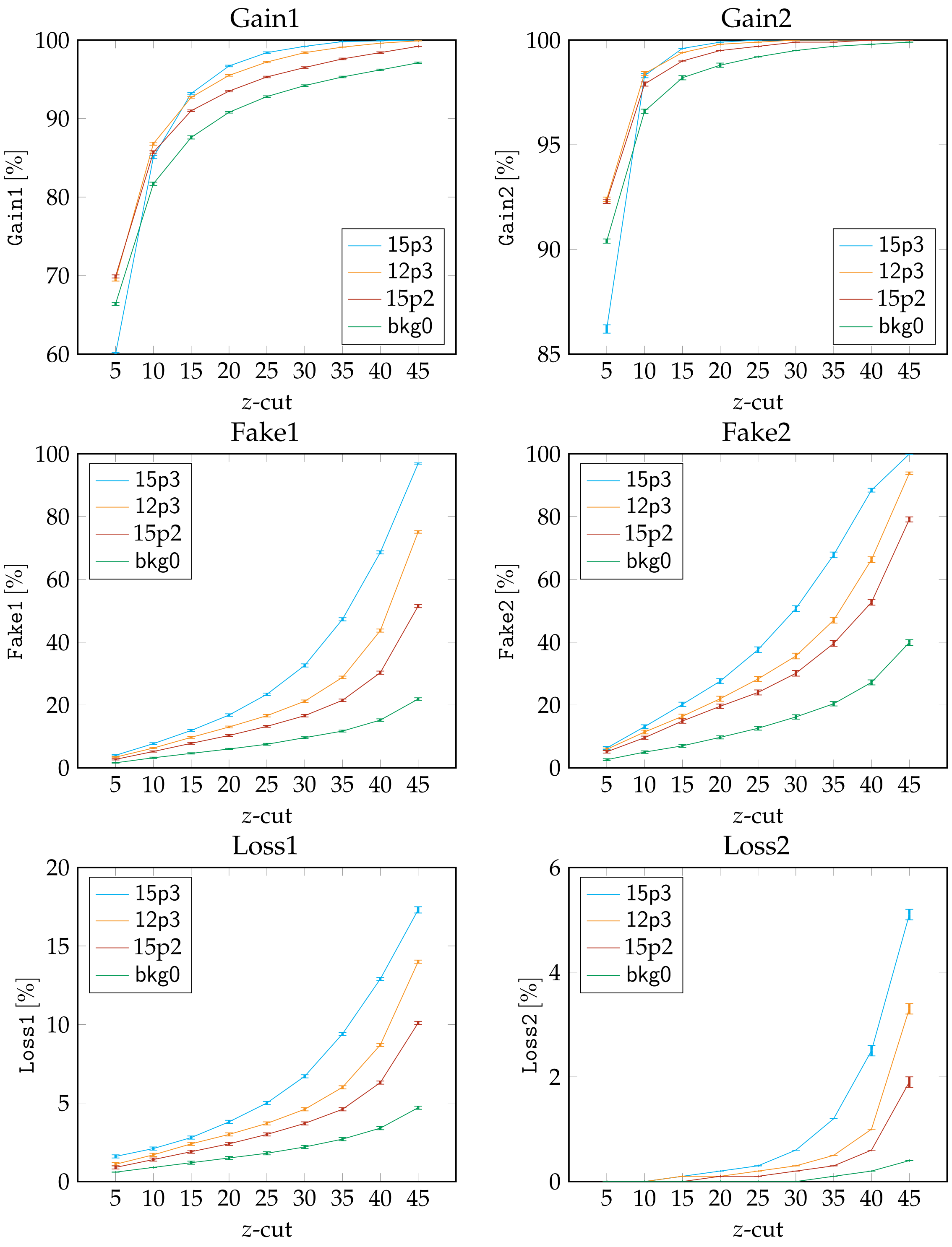


Figure 7.20.: Signal efficiency (`Gain`) and background rejection (`Fake`, `Loss`) for one and two track events with the z-cut based triggers, tested on HLT skimmed Belle II data from exp 8 run 3092.

training without background results in the lowest Fake-rates.

Out of the number of events with one or two tracks estimated within the z-cut region, the fraction of events where all these tracks are false positives are shown as the rates Loss1 and Loss2 in Fig. 7.20. At the z-cut $= 20\,\mathrm{cm}$ for two track events excellent results are achieved with Loss2 $\lessapprox 0.2\,\%$. Even in the single track case, rates strictly lower than Loss1 $< 4\,\%$ are achieved. The structure of the z-cut dependence is similar to the structure of the Fake-rates, however with different absolute values. Here it is important to note that these low rates reflect the fact that the data set is HLT skimmed. Therefore, the majority of events has vertices at the IP, which automatically reduce the Loss-rates and an increase is expected in the following study without HLT skimming in Sec. 7.2.2. In contrast, the Fake-rates are expected to be similar in Sec. 7.2.2, because the sample size increases without HLT skimming but not the relative proportions.

The neural network resolutions with the 2D finder input are shown in Fig. 7.21. The resolution $\Delta z(p_\mathrm{T})$ in Fig. 7.21 (a) shows a good resolution of $\Delta z \approx 4\,\mathrm{cm}$ in the high-p_T region, followed by an expected loss in resolution towards low-p_T. While the network trained with 15p3 achieves the best Δz for values of $p_\mathrm{T} < 1.5\,\mathrm{GeV}$ it achieves the worst resolution in the high-p_T region. Especially in the low-p_T region, training with background is significantly better than training without background. Both Phase 3 backgrounds provide similar resolutions, which are significantly better than the resolution with the Phase 2 background 15p2. The resolution $\Delta\theta(p_\mathrm{T})$ in Fig. 7.21 (b) has a similar structure as $\Delta z(p_\mathrm{T})$.

Distributed by the polar angle ($\Delta z(\theta)$ in Fig. 7.21 (c)) the network trained without background shows a broad peak around $\theta \approx 90^\circ$ with better resolutions towards shallow-θ angles. However, in the shallowest shown θ-bins the resolutions decrease again. With larger background levels $\Delta z(\theta)$ becomes almost flat. This means, noise from background improves the resolution for tracks fully contained in the barrel region. In the central region ($\theta \in [50, 100]\,^\circ$) similar resolutions are achieved with both Phase 3 background types, while slightly improved resolutions can be achieved with 12p3 for shallow-θ angles. The resolution $\Delta\theta(\theta)$ in Fig. 7.21 (d) is similarly distributed as $\Delta z(\theta)$ in Fig. 7.21 (c) with some minor differences: in the case without background, the resolution $\Delta\theta(\theta)$ is equal or better than in the case with the Phase 2 background 15p2. Almost in the full θ region, the best $\Delta\theta(\theta)$ can be achieved by training with 15p3. The improvement of the resolution for shallow-θ angles is plausible due to the stereo angle of the CDC wires. Since the stereo angles are quite small (see Sec. 3.2.5), a slanting in the polar angle θ (w.r.t. 90°) has a similar effect as an increase of the stereo angles in the CDC. Only if the number of stereo TSs per track is reduced due to very shallow-θ angles, the resolutions decrease.

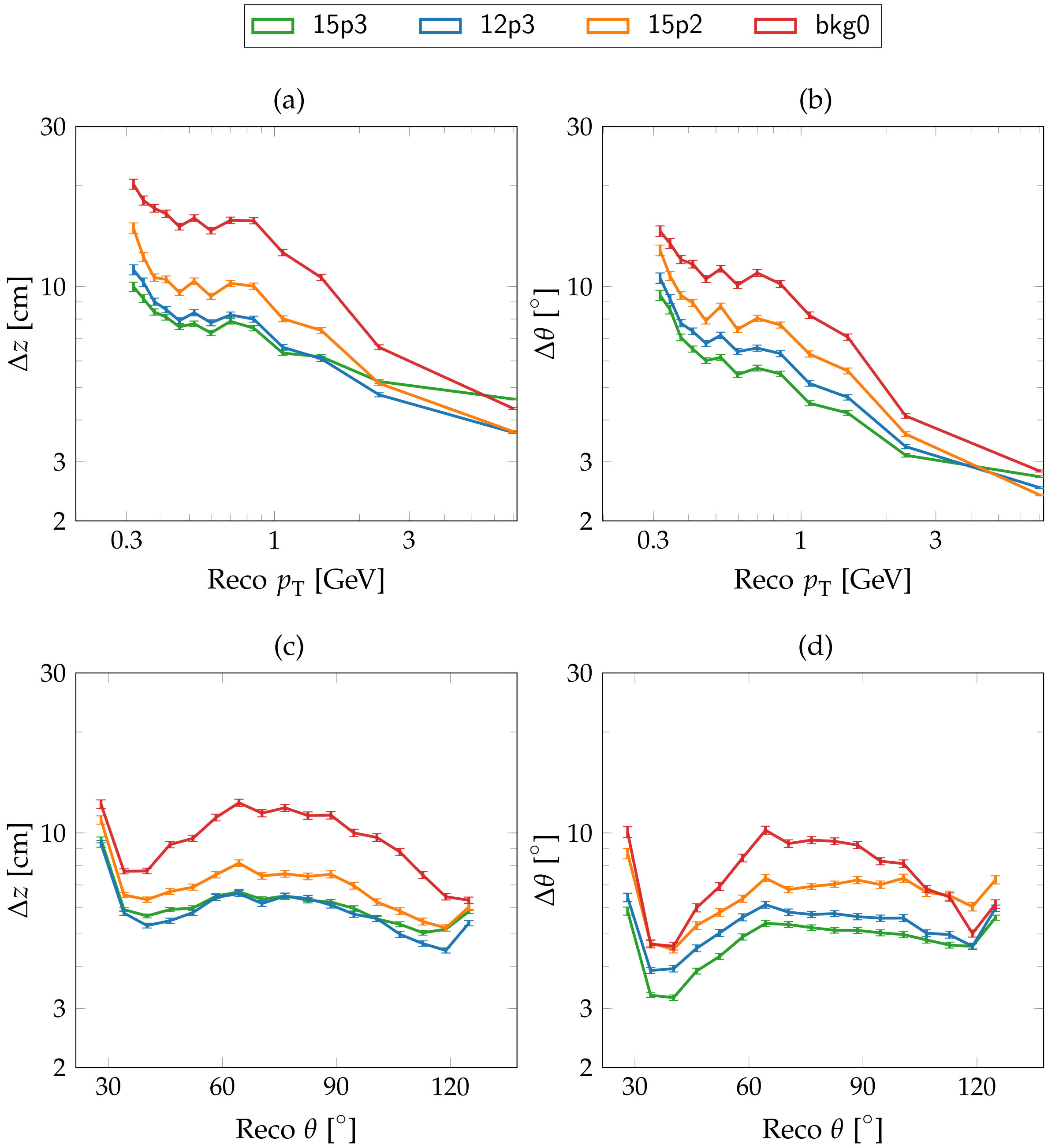


Figure 7.21.: Simulated neural network track parameter resolutions with the 2D finder as input and tested on recorded data from the Belle II experiment with HLT skimming (exp 8 run 3092). A cut was applied on the reconstructed track parameters to select only tracks from the IP (cut: $|z| \leq 1\,\mathrm{cm} \wedge d_0 \leq 1\,\mathrm{cm}$).

3D

	Bkg$_{\text{train}}$	$\Delta\theta^{\text{Reco}}$	Δz^{Reco}	$\epsilon^{\text{Reco}}_{\text{NN}}$	$\epsilon^{\text{Reco}}_{\text{Finder}}$	$\epsilon^{\text{Reco}}_{\text{Trg}}$	Op60
		[°]	[cm]	[%]	[%]	[%]	[%]
(a)	15p3	4.5	5.9	71.5	59.7	42.6	52.7
	12p3	5.0	6.0	71.5	59.7	42.6	53.4
	15p2	5.8	7.1	71.5	59.7	42.6	54.0
	bkg0	6.6	8.0	71.7	59.7	42.8	53.5

	Bkg$_{\text{train}}$	SL1	SL3	SL5	SL7	p7	p11	p13	p14	p15
		[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]
(b)	15p3	94.7	96.9	97.7	69.2	5.3	3.1	2.3	30.8	58.5
	12p3	94.7	96.9	97.7	69.2	5.3	3.1	2.3	30.8	58.5
	15p2	94.7	96.9	97.7	69.2	5.3	3.1	2.3	30.8	58.5
	bkg0	94.6	96.8	97.6	70.8	5.4	3.2	2.4	29.2	59.9

	Bkg$_{\text{train}}$	Gain1	Gain2	Gain3	Loss1	Loss2	Loss3	Fake1	Fake2	Fake3
		[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]
(c)	15p3	95.3	99.9	100.0	4.6	0.4	0.1	34.2	50.2	69.1
	12p3	91.4	99.7	99.9	4.2	0.3	0.0	29.9	43.8	60.6
	15p2	89.3	99.5	99.8	3.9	0.2	0.0	26.6	39.4	52.5
	bkg0	88.3	99.2	99.6	3.1	0.1	0.0	20.5	30.6	41.6

Table 7.7.: Neural network results on recorded HLT skimmed data from Belle II (exp 8 run 3092) using the 3D finder as input.

3D Finder Input

The results of the same study with the 3D finder as input are summarized in Tab. 7.7. Although the efficiency of the 3D finder is significantly larger, $\epsilon_{\text{3D Finder}} \approx 60\,\%$, the resolutions of the neural networks are quite similar as with the 2D finder as input. The same hierarchy of the background types as with the 2D finder can be observed, with the best resolutions for the Phase 3 background 15p3 and the worst resolution without background. As known from the simulation studies, the increased 3D finder efficiency results in a reduced neural network efficiency, of only $\epsilon_{\text{NN}} \approx 70\,\%$. However, the observed total trigger efficiency ϵ_{Trg} is still better than with the 2D finder as input. Further improvements are possible on the neural network side, by using the short track expert neural networks presented in Sec. 6.5.2.

The SL hit rates in Tab. 7.7 (b) show a smaller rate SL7 than with the 2D finder. This is plausible as the 3D finder has an improved low-p_{T} efficiency. Interestingly, the rate in SL1 is larger than with the 2D finder as input. This demonstrates an additional benefit of the 3D finder: it is optimal only for tracks from the IP, where a hit in the first SL is expected. Geometrically, the TS hit in the first stereo SL can only be missing for tracks with vertices displaced from the IP in longitudinal z-direction or with transverse offsets d_0. As shown in the simulation studies in Fig. 6.23 in Sec. 6.5.1, the high efficiency of the 3D finder for IP tracks decreases for tracks displaced along the z-axis. In contrast, the 2D finder efficiency is passably flat in z, i.e. it cannot provide any 3D preselection of the tracks like the 3D finder.

Correlated with the hit rates in the SLs, the majority of events are processed by the expert networks with the patterns p14 and p15. However, the experts with missing intermediate SLs are not negligible: the experts p11 and p13 combined process about 5.4 % of the events. This is similar to the previous case with the 2D finder in Tab. 7.6 (b), where the experts p11 and p13 processed about 4.5 %. Therefore, the case of missing TSs in SL3 or SL5 should not be neglected in the hardware studies. The reason for missing inner TSs, which is not seen in the MC simulations (see Tab. 6.18 in Sec. 6.5.1), has to be searched in the detector inefficiencies and unexpected background types not represented in the present simulation. For the training of the neural network, this indicates that the number of expert networks cannot be reduced by skipping the experts for the missing inner SLs without a loss in efficiency. Therefore, further studies with the short track expert networks using all eight experts should be pursued.

The z-cut based rates with the 3D finder as input are shown in Tab. 7.7 (c). In the rate Gain1, the high efficiency of Gain1 $= 95.1\,\%$ in single track events is achieved with the background 15p3. Although this is slightly smaller than the peak value achieved with the 2D finder as input, this has to be seen in face of the improved 3D finder efficiency. The loss in Gain1 with the other background types, especially compared to the Phase 3 background 12p3, is larger than observed with the 2D finder. This indicates that an adequate background model becomes even more important for the additional tracks found due to the increased 3D finder efficiency. In two track events, excellent efficiencies of Gain2 $\gtrapprox 99\,\%$ can be achieved with all background types. This tendency also holds for events with three or more tracks, with Gain3 $\gtrapprox 99.6\,\%$. The Loss-rates are slightly larger than before with the 2D finder, but still in an ideal region of Loss1 $< 5\,\%$ and Loss2 $< 0.5\,\%$ for the network trained with the largest background level. Due to the improved track finding efficiency, the observed Fake-rates are almost twice as large as with the 2D finder. Therefore, the value of z-cut has to be chosen as a tradeoff between signal efficiency and background suppression. Especially the topology of the background distribution is an important ingredient to the optimization of the z-cut value.

The plots for the z-cut based rates with the 3D finder as input are shown in Fig. 7.22. The plot Gain1 shows an improvement with the background 15p3 compared to the other Phase 3 background 12p3 in the range z-cut $\in [15, 35]$ cm. The structure of the curves is similar to the structure with the 2D finder as input. This includes the inversion effect for very small values of z-cut $= 5$ cm, where the worst Gain-rates are achieved with 15p3.

The increase of the rates Fake1 and Fake2 for larger values of z-cut is nearly linear with the 3D finder. In contrast, with the 2D finder (see Fig. 7.20) a sweet spot of Fake-rates for low values of z-cut is followed by an almost quadratic increase of the Fake-rates for larger values of z-cut. This difference in the structure is a result of the 3D track pre-selection by the 3D finder combined with its improved efficiency: the 3D finder is less efficient for tracks displaced along the z-axis. In turn, the neural

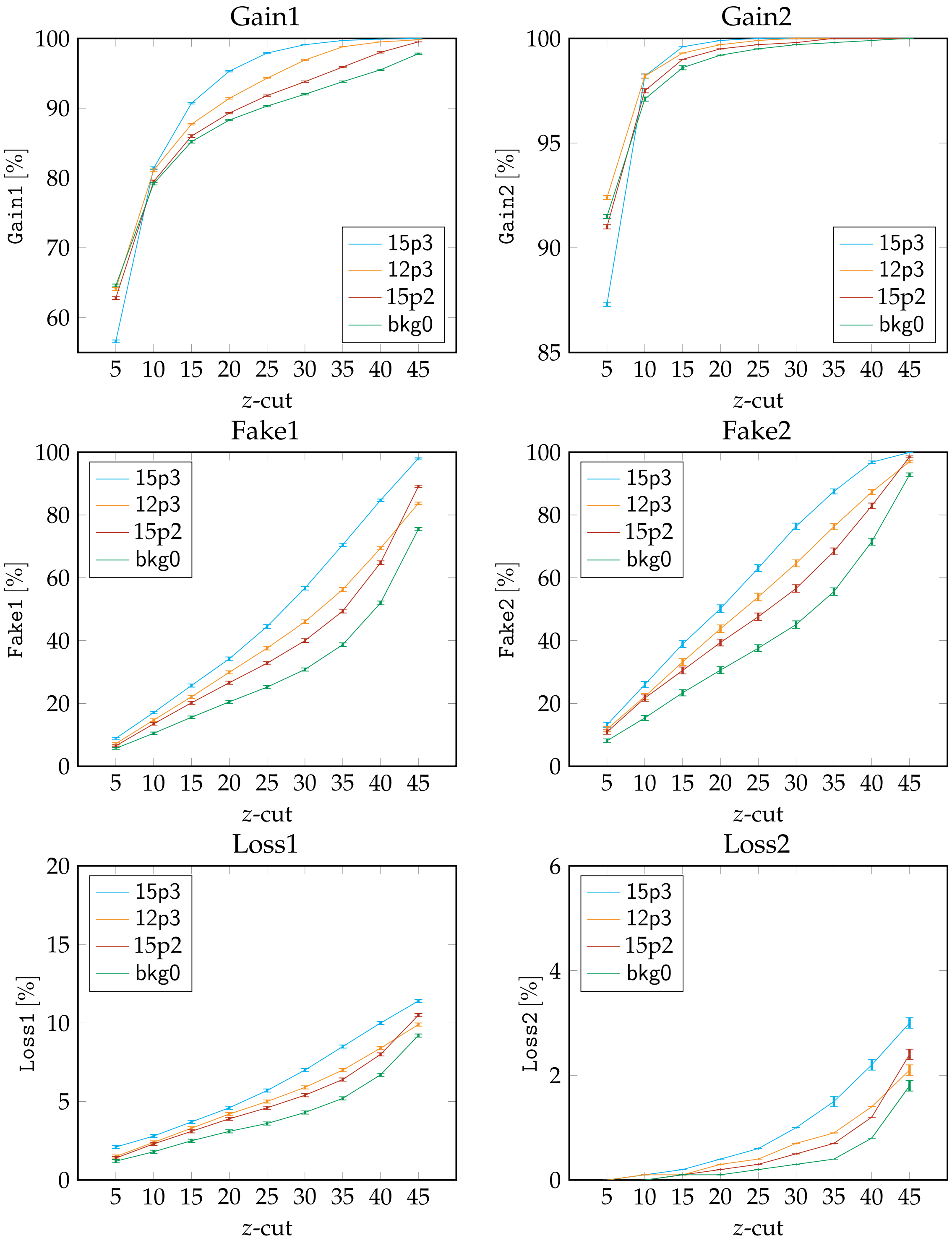


Figure 7.22.: Neural network trigger rates with the 3D finder as input. For one and two track events the z-cut based trigger cuts are measured on the HLT skimmed Belle II data from experiment 8 run 3092.

network has to process only those tracks, which already passed through the 3D finder.

In the Loss-rates a more linear increase with the z-cut can be observed as well. Compared to the 2D finder Loss-rates, this results in larger Loss-rates for small values of z-cut while for larger values of z-cut lower Loss-rates are achieved. This is advantageous if the z-cut is set to large values.

The resolutions of the neural networks as a function of the reconstructed track parameters p_T and θ are shown in Fig. 7.23. The same tendencies as with the 2D finder as input can be observed (compare Fig. 7.21). The best resolutions are achieved in the high-p_T region (see Fig. 7.23 (a) and (b)) of about $\Delta z \lessapprox 5\,\mathrm{cm}$ with all background types. Although the network trained with 15p3 achieves the best resolutions for low-p_T tracks, it achieves a slightly worse resolution than the other background types in the high-p_T region. In the low-p_T region, all resolutions are better than in the case with the 2D finder. The reason is the improved precision of the 2D track parameter estimates p_T and ϕ provided by the 3D finder in the low-p_T region (see Sec. 5.3.3). As these parameters are used to linearize the 3D tracking problem in the neural network preprocessing, an improved precision simplifies the 3D track reconstruction task for the neural network.

The z- and θ-resolutions as a function of θ are shown in Fig. 7.23 (c) and (d). Compared to the 2D finder input, the peak of the resolutions without background for tracks with central polar angles θ is less pronounced. In $\Delta z(\theta)$ with both Phase 3 backgrounds comparable resolutions are achieved. Although in $\Delta\theta(\theta)$ and especially in the forward region (small values of θ), the training with the background 15p3 turns out to provide the best resolutions.

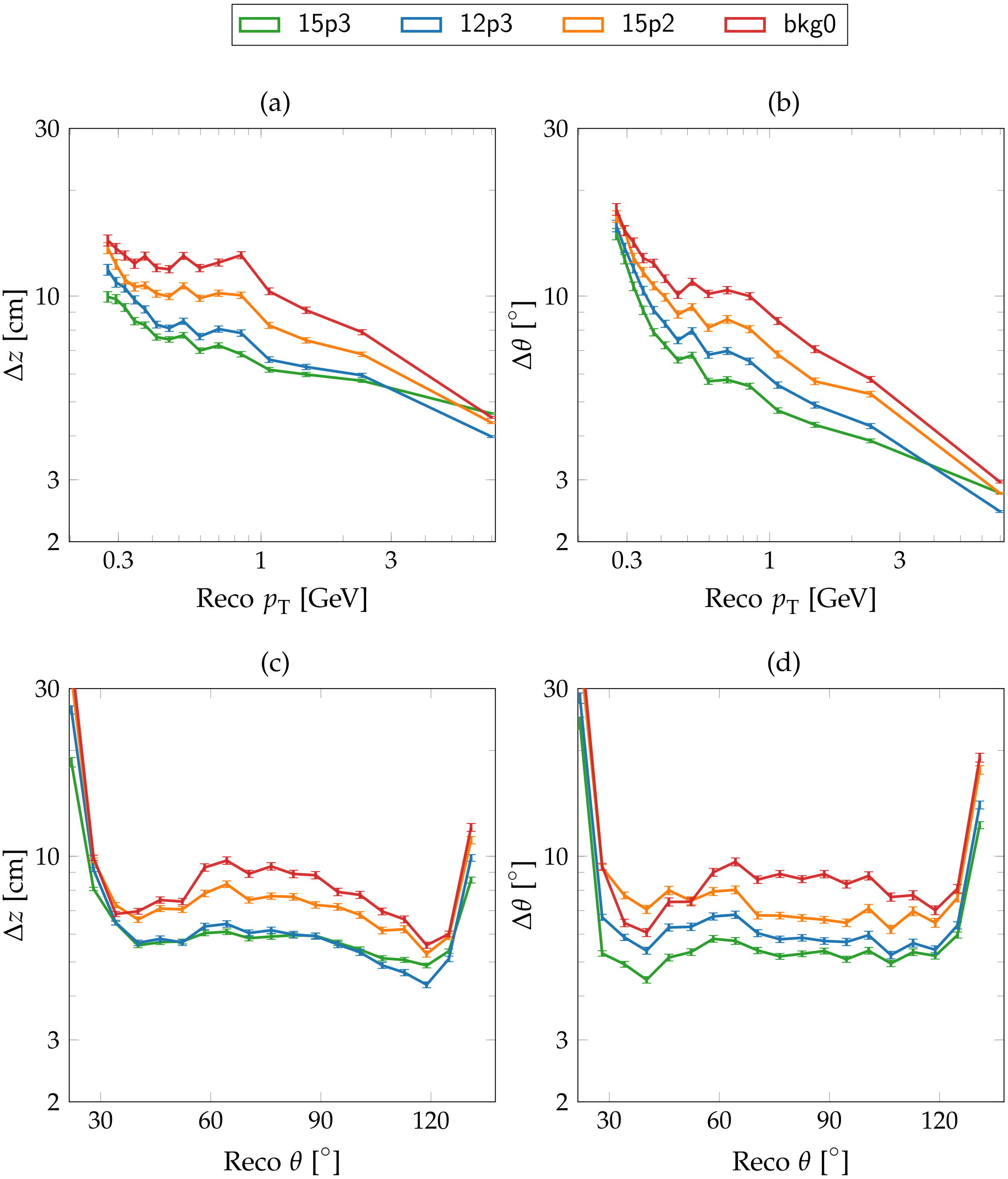


Figure 7.23.: Neural network 3D track parameter resolution on HLT skimmed Belle II data from experiment 8 run 3092 with the 3D finder as input. An additional cut was applied on the reconstructed track parameters to select only tracks from the IP.

7.2.2. Background Events

The trigger performance on real background tracks can be studied best using a data set without HLT skimming. So far, the previous study with HLT skimmed data from Belle II mainly demonstrated the efficiency of the neural network based trigger cuts for recorded physics signal events. This is now extended by measurements of the background suppression performance in a data set with a comparably large number of background tracks with displaced z-vertices.

The Belle II data selected for this study without HLT skimming is taken from the Belle II experiment 8 run 2413. In principle, this was a luminosity run as well, however without the background suppression by the HLT. Hence, the majority of the events in this data set are pure background events, with a small admixture of physics signal events. According to the internal Belle II log entry of that run, at the start of the run the beam currents in the HER and the LER were at 0.35 A with 1576 bunches in each ring, a luminosity of $1.97 \times 10^{33}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$, and a trigger rate of 3.1 kHz. During a total run time of 2 minutes and 33 seconds an integrated luminosity of about $287 \times 10^{33}\,\mathrm{cm}^{-2}$ was collected, with a total of 447653 events. In the same way as in the previous study of the physics signal events, the recorded CDC hits are used as input to the trigger simulation. Additionally, the full offline reconstruction is carried out in order to obtain the reconstructed tracks as reference values.

The distributions of the reconstructed track parameters are shown in Fig. 7.24. The p_T-distribution in Fig. 7.24 (a) has a large peak for low momentum tracks at $p_\mathrm{T} \approx 600\,\mathrm{MeV}$ and only a very small peak in the high-p_T region. Although an improved efficiency for low-p_T tracks can be observed with the 3D finder, the 2D finder now achieves larger total track finding rates. This means that the 2D finder finds a larger number of background tracks than the 3D finder. The ϕ-distributions in Fig. 7.24 (b) are almost flat. Since for the luminosity related background tracks from the IP a broad peak around $\phi \approx 0^\circ$ is expected, this indicates larger levels of machine background as well as backscattering background tracks than represented in the present background simulation. The distribution in the polar angle θ in Fig. 7.24 (c) shows more tracks in the backward region than in the forward region. Due to the lower total efficiency of the 3D finder for displaced background tracks, the improvements for small θ-angles are visible in the forward and background region, however they appear to be smaller than in the previous study with IP tracks in Fig. 7.19 (c).

Figure 7.24 (d) shows a strong background peak in the z-distribution of the reconstructed tracks at $z \approx 60\,\mathrm{cm}$ and a small background peak at $z \approx -60\,\mathrm{cm}$. These large background structures were not visible in the HLT skimmed data set in Fig. 7.19 (d). The background peak in positive z-direction nicely agrees with the correlated large θ-peak in the backward direction. While the track finding efficiency of the 3D finder is larger for tracks from the IP, background tracks with large displacements are found more efficiently with the 2D finder. This is best seen in Fig. 7.19 (d) in the region $z \geq 70\,\mathrm{cm}$ and at the peak located at $z \approx -60\,\mathrm{cm}$. Note that this apparent loss in efficiency for background tracks is actually a substantial improvement, as it implements a first level of 3D background suppression prior to the neural network.

The shape of the z-distribution in Fig. 7.24 (d) is similar to the z-distribution in Fig. 7.18. Although the size of the peaks are different, which indicates different background conditions in those two runs. This means that a continuous monitoring of the background during the Belle II runs is crucial for the optimization of the neural networks to the changing background conditions.

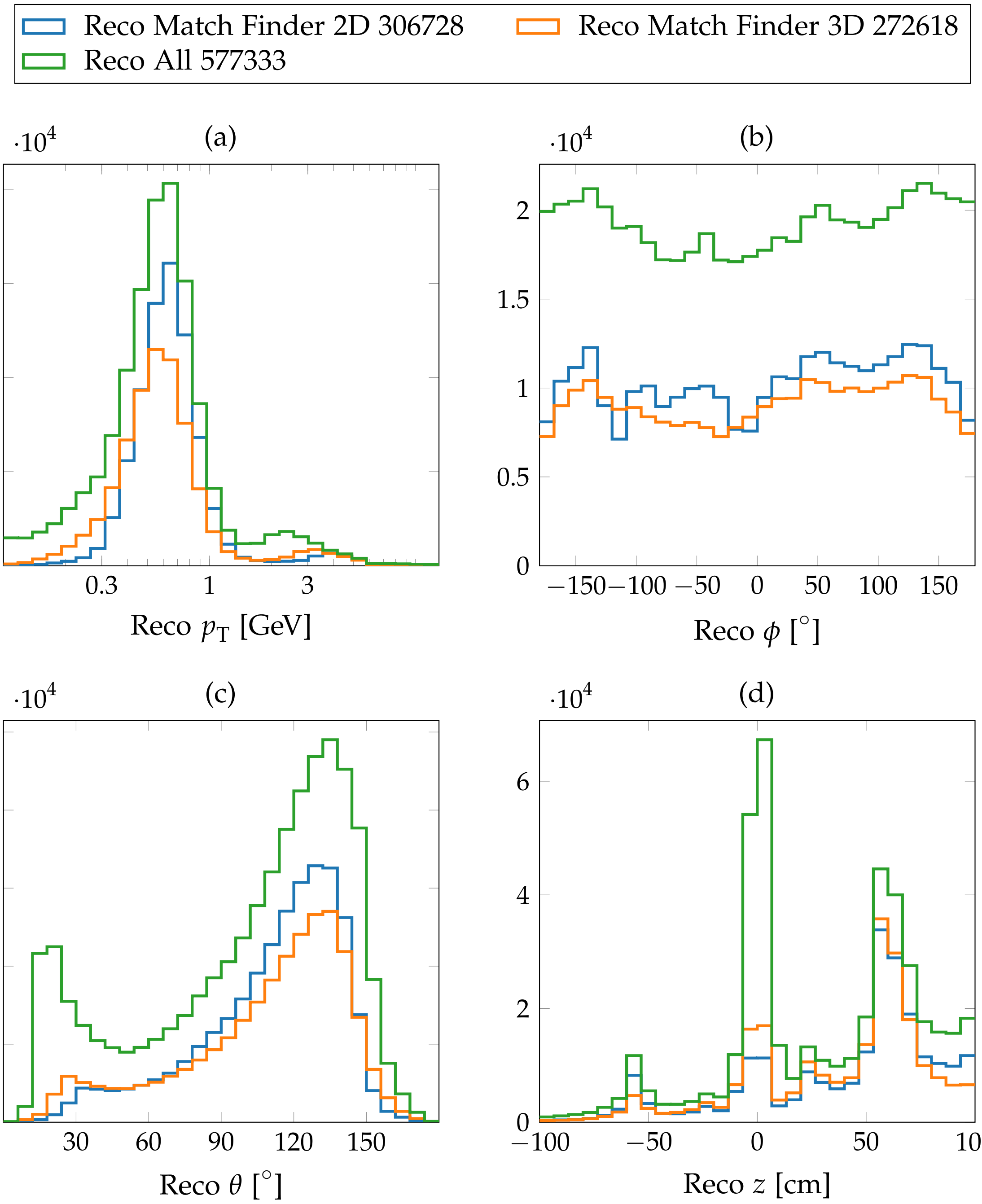


Figure 7.24.: Distribution of the reconstructed track parameters in the data recorded by Belle II in experiment 8 run 2413. No HLT skimming was used in this run such that the topology of the background can be studied. All reconstructed tracks as well as the reconstructed tracks matching to the 2D finder and to the 3D finder tracks are shown.

2D

	$\text{Bkg}_{\text{train}}$	$\epsilon_{\text{NN}}^{\text{Reco}}$	$\epsilon_{\text{Finder}}^{\text{Reco}}$	$\epsilon_{\text{Trg}}^{\text{Reco}}$	Op60
		[%]	[%]	[%]	[%]
(a)	15p3	84.4	53.1	44.9	91.8
	12p3	84.4	53.1	44.9	90.6
	15p2	84.4	53.1	44.8	90.0
	bkg0	84.4	53.1	44.8	86.4

	$\text{Bkg}_{\text{train}}$	SL1	SL3	SL5	SL7	p7	p11	p13	p14	p15
		[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]
(b)	15p3	76.0	98.1	99.0	83.1	24.0	1.9	1.0	16.9	56.2
	12p3	76.0	98.1	99.0	83.1	24.0	1.9	1.0	16.9	56.2
	15p2	76.0	98.1	99.0	83.1	24.0	1.9	1.0	16.9	56.2
	bkg0	76.0	98.1	99.0	83.1	24.0	1.9	1.0	16.9	56.2

	$\text{Bkg}_{\text{train}}$	Gain1	Gain2	Loss1	Loss2	Fake1	Fake2
		[%]	[%]	[%]	[%]	[%]	[%]
(c)	15p3	93.2	99.9	67.4	16.8	21.9	35.5
	12p3	90.3	99.3	63.5	12.2	17.3	29.5
	15p2	87.1	98.6	56.6	8.4	12.0	22.9
	bkg0	80.9	96.9	46.4	3.7	6.6	12.9

Table 7.8.: Neural network results in the background dominated data set from Belle II without HLT skimming (exp 8 run 2413). The 2D finder is used as input.

Neural Network Results

The focus of this study are the efficiencies and the background suppression rates. Therefore, the resolutions for tracks with large displacements are not studied here and no additional cuts are applied to select tracks from the IP, as it was done previously with the data HLT skimmed data. Table 7.8 is a summary of the results with the 2D finder as input. This time the entries are sorted by the rate Loss2 in Tab. 7.8 (c). Since the true topology of the background tracks enters in this study, the most important new results are the Loss-rates. In the rates Gain and Fake, only fluctuations due to different test samples are expected.

The efficiency of the 2D finder in Tab. 7.8 (a) is now larger, with $\epsilon_{\text{2D Finder}} \approx 53\,\%$ compared to the 2D finder efficiency for signal tracks from the IP in the previous study of $\epsilon_{\text{2D Finder}} \approx 39\,\%$ (see Tab. 7.6). This is a result of the different θ-distribution in this data set: the peak in the backward region is shifted towards the barrel region (less shallow-θ angles) where the 2D finder is more efficient (see Fig. 7.24 (c)). However, the larger 2D finder efficiency is accompanied by a reduced neural network efficiency of only $\epsilon_{\text{NN}} \approx 84\,\%$ compared to $\epsilon_{\text{NN}} \approx 92\,\%$ for the IP tracks in Tab. 7.6. Similar to the 3D finder efficiency for displaced tracks, this reduced neural network efficiency corresponds to a background suppression of displaced tracks skipped by the neural network preprocessing. Hence, this is an advantage for the track trigger and countermeasures are not required. The large values of the rate `0p60` in Tab. 7.8 (a) show that two track events in this data sample have correlations in their polar angle. This means that these types of background tracks have to be carefully distinguished from boosted physics signal events (compare ISR events in Sec. 7.1).

The hit rates in the SLs and the pattern rates are shown in Tab. 7.8 (b). The large fraction of tracks with a missing stereo TS in the outermost SL, indicated by the low `SL7`-rate and the correlated contribution of the pattern-rate `p14`, are typical for the tracks with low-p_T. However, compared to the tracks from the IP, the rate of tracks with the innermost stereo SL missing is now much larger as well, which can be seen as a low value of $\texttt{SL1} \approx 76\,\%$ and a large rate of the correlated expert with $\texttt{p7} \approx 24\,\%$. This is plausible considering that this data set contains many tracks displaced from the IP, where the chance for a track to miss the innermost stereo SL is increased. With the 2D finder as input, the different background types used in the training have no effect on the selected expert networks.

As expected, the `Gain`-rates and `Fake`-rates in Tab. 7.8 (c) are consistent with Tab. 7.6 (c). The best signal efficiency `Gain` is still obtained by using the background type 15`p`3 in the training: the cut at $z\text{-cut} = 20\,\text{cm}$ is very efficient for two track events with $\texttt{Gain2} \approx 99.9\,\%$. At the same time, a reasonable level of impurity can be observed for two track events with a value of $\texttt{Loss2} \approx 16.8\,\%$ with the background type 15`p`3. However, since the `Loss1`-rate for single track events is quite large, smaller values of z-cut should be preferred in the case of single track triggering.

The corresponding results with the 3D finder as input are shown in Tab. 7.9. With $\epsilon_\text{3D Finder} \approx 47\,\%$ the total track finding rate is now lower than the track finding rate with the 2D finder. In contrast, the neural network efficiencies with $\epsilon_\text{NN} \approx 73\,\%$ are even slightly improved compared to the case for IP tracks (see Tab. 7.7 (a)). This confirms the background suppression capabilities of the 3D finder compared to the 2D finder, where a loss ϵ_NN is observed in this data set without HLT skimming. Using the 3D finder, a large fraction of the displaced tracks are not presented to the neural network as input such that the neural network can process a larger fraction of the received input tracks. Furthermore, this demonstrates the stability of the neural network efficiency with the 3D finder as input. As mentioned earlier, to make full use of the improved 3D finder efficiency for IP tracks, an extension of the neural network setup to short track expert networks is required (see Sec. 6.5.2).

This 3D finder background suppression manifests itself in the SL hit rates, shown in Tab. 7.9 (b): with $\texttt{SL1} \approx 90\,\%$ the fraction of tracks displaced from the IP is smaller than with the 2D finder with its $\texttt{SL1} \approx 76\,\%$ in Tab. 7.8 (b). This agrees with the pattern rates, where the expert for the innermost stereo SL missing is only used in $\texttt{p7} \approx 10\,\%$ of the processed tracks. As before with the HLT skimmed data, the experts for missing TS in the inner SLs should not be neglected, as they process about $\texttt{p11} + \texttt{p13} = 4.6\,\%$ of the tracks with the 3D finder and $\texttt{p11} + \texttt{p13} = 2.9\,\%$ with the 2D finder.

The `Gain`-rates in Tab. 7.9 (c) are slightly smaller than the 2D finder results. Since

3D

	$\text{Bkg}_{\text{train}}$	$\epsilon_{\text{NN}}^{\text{Reco}}$	$\epsilon_{\text{Finder}}^{\text{Reco}}$	$\epsilon_{\text{Trg}}^{\text{Reco}}$	Op60
		[%]	[%]	[%]	[%]
(a)	15p3	73.4	47.2	34.6	89.0
	12p3	73.4	47.2	34.6	87.6
	15p2	73.3	47.2	34.6	87.7
	bkg0	73.7	47.2	34.8	85.1

	$\text{Bkg}_{\text{train}}$	SL1	SL3	SL5	SL7	p7	p11	p13	p14	p15
		[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]
(b)	15p3	89.3	98.5	96.9	67.6	10.7	1.5	3.1	32.4	52.3
	12p3	89.3	98.5	96.9	67.6	10.7	1.5	3.1	32.4	52.3
	15p2	89.3	98.5	96.9	67.6	10.7	1.5	3.1	32.4	52.2
	bkg0	89.1	98.5	96.6	68.9	10.9	1.5	3.4	31.1	53.1

	$\text{Bkg}_{\text{train}}$	Gain1	Gain2	Gain3	Loss1	Loss2	Loss3	Fake1	Fake2	Fake3
		[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]	[%]
(c)	15p3	90.8	99.6	100.0	65.8	21.5	7.0	35.3	56.1	76.4
	12p3	85.7	99.1	100.0	63.6	18.3	6.0	29.5	48.5	68.7
	15p2	81.3	98.6	99.6	62.4	16.3	6.0	26.1	44.2	62.9
	bkg0	79.0	97.7	99.4	57.2	11.4	3.9	19.2	34.3	53.8

Table 7.9.: Neural network results with the 3D finder as input in the background dominated data set from Belle II without HLT skimming (experiment 8 run 2413).

the 3D finder has a larger efficiency for the IP tracks, this still corresponds to a larger total signal efficiency. Due to the correct background topology in this study, the Loss-rates observed here are significantly larger compared to the HLT skimmed sample of IP tracks. For the network trained with the background 15p3, where the best Gain-rates are achieved, the Loss-rates with the 3D finder and with the 2D finder are similar. In particular for the triggering of two track events, the cut at z-cut $= 20\,\text{cm}$ turns out to be well suited for the 3D finder input as well: in the training with the background type 15p3 a tolerable impurity of Loss2 $\approx 21.5\,\%$ can be achieved, while excellent signal efficiencies of Gain2 $= 99.6\,\%$ can be achieved. This means that nearly all the two track IP events are correctly recognized as signal, while only about 20 % are false positives. This corresponds to a purity of about 80 % for the z-cut at 20 cm.

The Fake-rates in Tab. 7.9 (c) are of the same order as for the HLT skimmed events and roughly twice as large as with the 2D finder input. Note that in this study a rather small cut value for displaced tracks of 40 cm is used here, which can be optimized in future studies by setting it to the observed peak position of the reconstructed background tracks. In particular, this means to use the large peak position of background tracks at 60 cm, observed in the z-distribution in Fig. 7.24 (d). In the calculation of the Fake-rates, only those tracks found by the track finder enter in

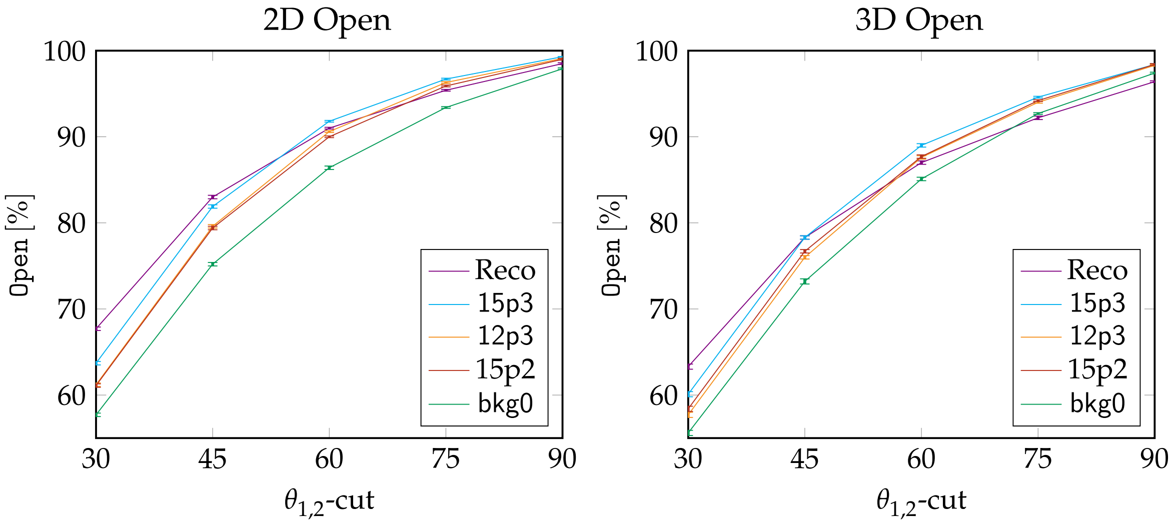


Figure 7.25.: Rate Open for two track events based on the neural network estimates with the 2D finder and with the 3D finder as input. The rate Open shows the fraction of events where the θ opening angle between the two tracks is smaller than $\theta_{1,2}$-cut.

the calculation. By using the 3D finder, the tracks received by the neural network are already pre-selected based on the stereo TSs which resulted in the larger neural network efficiencies ϵ_{NN} with the 3D finder on the data set with displaced tracks. This means that, some of the background tracks correctly rejected by the neural network with the 2D finder as input, can already rejected by the 3D finder prior to the neural network. Hence, the Fake-rates are biased by the track finder efficiencies and should not be used for an absolute comparison. However, large Fake-rates indicate that a further confinement of the cut values might be required if the absolute background track rates exceed the limits of the first level trigger. Unbiased background rates, relative to the absolute number of background tracks, could be constructed by considering products of the 3D finder background rejection of displaced tracks and the z-cut based Fake-rates of the neural network. Here the background rates relative to the absolute number of estimated tracks is considered to be sufficient for a comparison of the 2D finder with the 3D finder in the neural network input.

With both finders as input, the polar angles of the two track events seem to have a larger correlation compared to the previous study with IP tracks. Therefore, the rate of tracks with different opening angle cuts are shown in Fig. 7.25. Comparing these curves to the case of the simulated ISR events with the $\pi^+\pi^-$ final state in Fig. 7.11 shows that the correlation is much lower. These background events do not object to triggers based on the opening angle by showing the same characteristic opening angle signature. Hence, Open should still be considered as a trigger for ISR events.

The z-cut based neural network trigger rates with the 2D finder as input are shown in Fig. 7.26. Comparing the rates Gain2 with Fake2 and Loss2 shows that there is

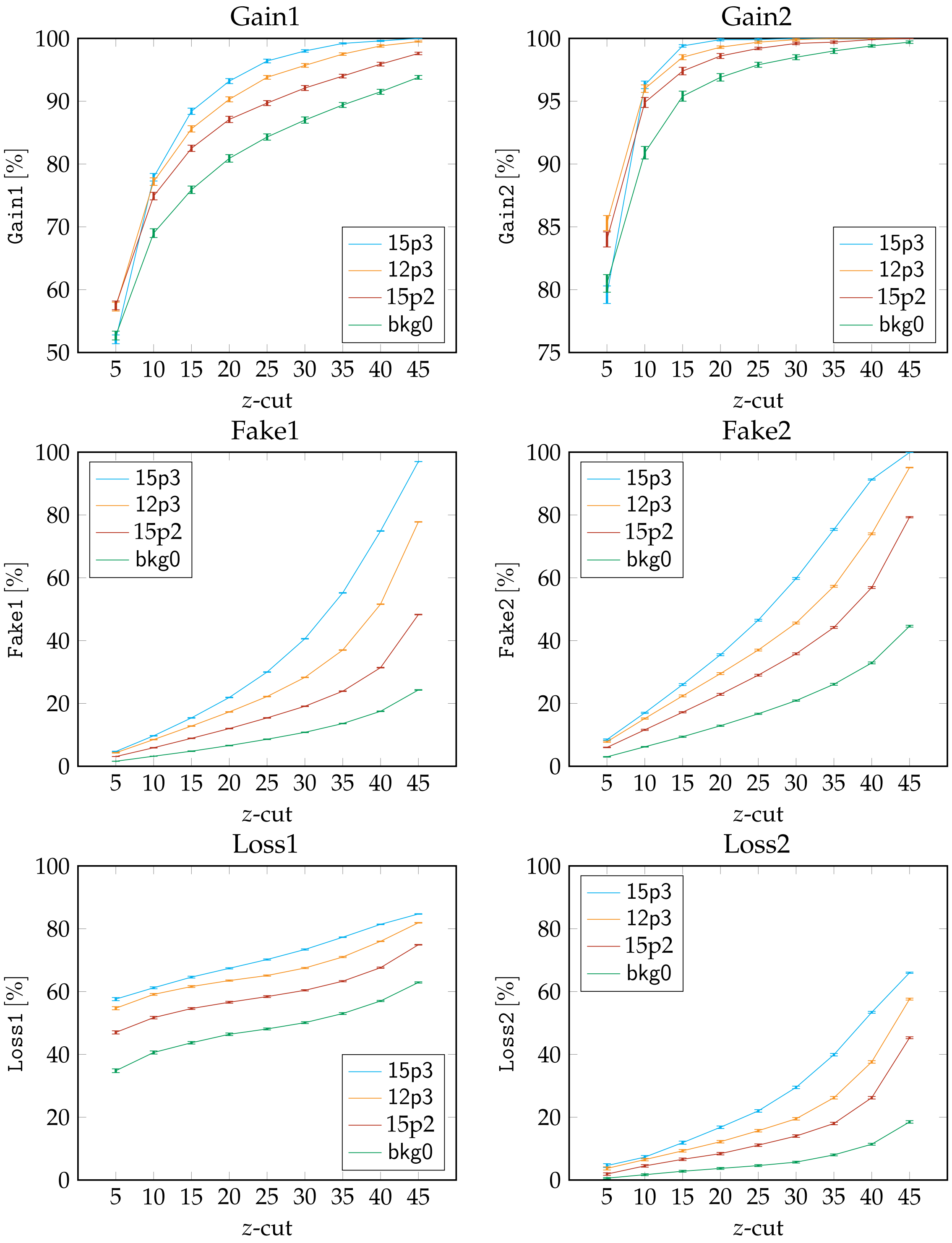


Figure 7.26.: Neural network z-cut based trigger rates for one and two track events with the 2D finder as input. The z-cut based triggers are tested on Belle II data from experiment 8 run 2413 without HLT skimming.

room for larger levels of background suppression. This might become important if the rates Loss2 $\approx$ 17 % or Fake2 $\approx$ 35 % exceed the trigger capabilities. The observed Loss2-rate is already at a tolerable level, however it can also be further reduced with smaller values of z-cut. It has to be considered that the Fake2-rate expresses the background rate relative to the topology of background tracks found by the track finder, while the rate Loss2 is the fraction of background contamination in the triggered events. For example, by reducing the z-cut to z-cut $=$ 10 cm with the background 15p3 the signal efficiency Gain2 is still larger than 95 %, while the Loss2-rate and the Fake2-rate fall by almost a factor of two.

In particular, the single track Loss1-rate is significantly larger compared to the HLT skimmed sample. While Loss1 is rising almost linearly, its starts out at 60 % for the smallest value of z-cut $=$ 5 cm. Due to the large number of background tracks in this data set, the false positive rate of a single track trigger (Loss1) is quite large. This rate can only improve with different background topologies or with a higher precision of the track parameters estimated by the neural network. Although the background levels relative to the luminosity will improve during the runtime of the Belle II experiment, due to the continuous increase of the luminosity more severe background conditions are expected in the future of the Belle II experiment. Hence, the single track triggers can only be installed with pre-scaling factors. If such a single track trigger is pursued, the optimal z-cut should be selected as the point of the best signal to background ratio. Using the latest background topology, this can be extracted from a receiver operator characteristic (ROC) like curve in a tradeoff between Gain1 and Loss1.

The z-cut based trigger efficiencies using input tracks from the 3D finder are shown in Fig. 7.27. Considering the two track triggers, the reduction of the z-cut value to z-cut $=$ 10 cm has a similar effect as with the 2D finder input: the signal efficiency Gain2 is still larger than 95 % and the Loss2-rate and Fake2-rate are reduced by a factor of two. In particular the maximum total Belle II L1 trigger rate of 30 kHz has to be considered for the fine tuning of the z-cut based triggers.

In general, the 3D finder turns out to be a viable alternative to the 2D finder. It achieves improved signal efficiencies, especially for tracks with shallow polar angles θ and for tracks with low transverse momenta p_T. The similar Loss-rates with the 3D finder and with the 2D finder are notable: this means that the purity of the z-cut based triggers is only slightly affected by a change of the track finder. Especially in the Fake-rates, where the main unit corresponds to those background tracks preselected by the 3D finder, the efficiency improvement appears as larger Fake-rates. This means that a larger fraction of the displaced background tracks found by the track finder are falsely estimated within the z-cut region. A two track trigger seems feasible with both finders. However, for a single track triggers the false positive rate Loss1 is relatively large compared to the signal efficiency Gain1.

With the 3D finder, just as with the 2D finder, the best signal efficiencies (Gain) of the neural network are obtained with the background 15p3, however this is accompanied by the worst Loss- and Fake-rates. Therefore, optimizing the used background type is a suitable measure to tune the neural network performance. In particular,

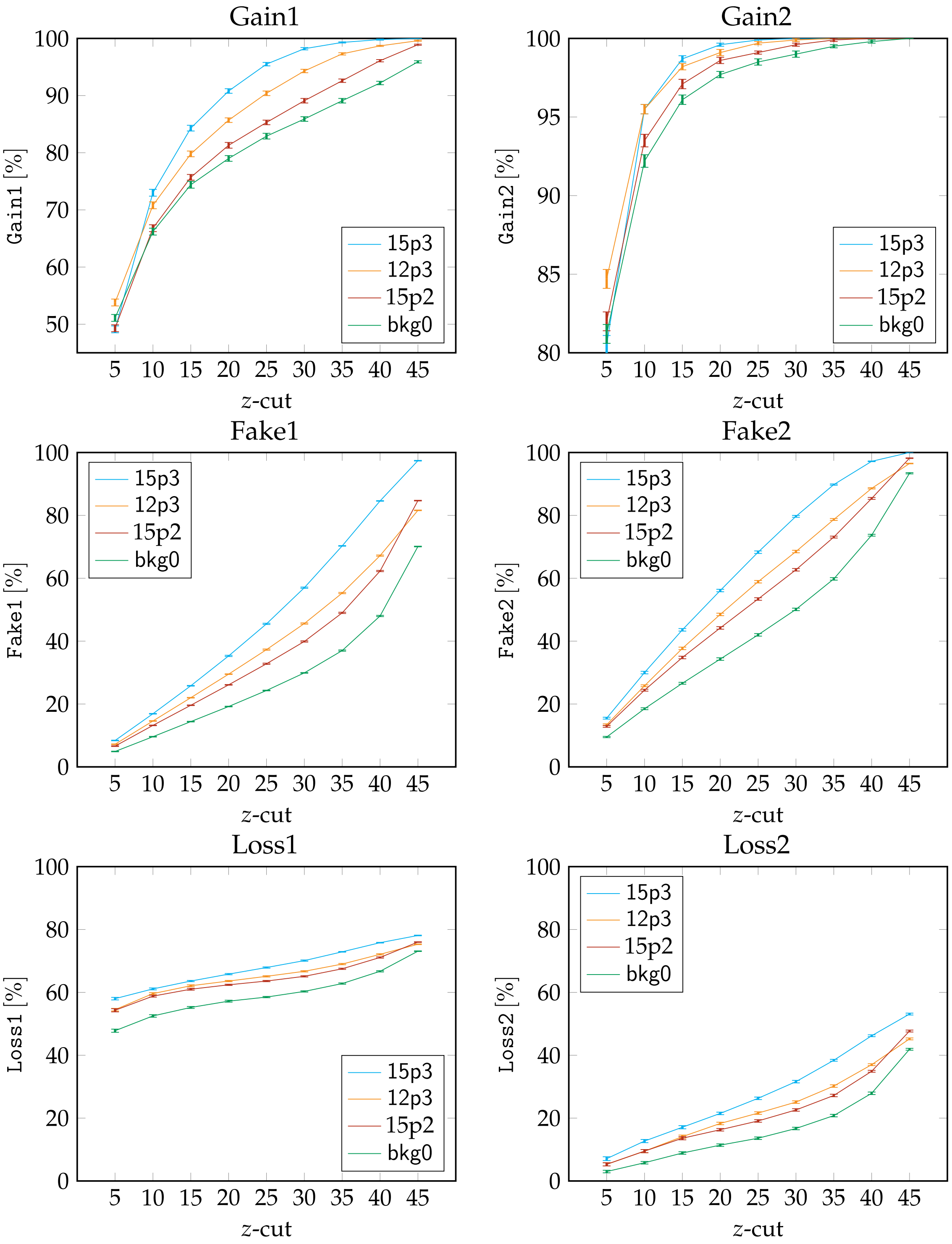


Figure 7.27.: Neural network z-cut based trigger rates for one and two track events with the 3D finder as input. The z-cut based triggers are tested on Belle II data from experiment 8 run 2413 without HLT skimming.

after a retraining of new neural networks, the used trigger cuts have to be revised in order to obtain the optimal trigger performance.

7.2.3. Summary of the Belle II Data Analysis

The simulated neural network tests with real data from Belle II as input confirmed the successful operation of the neural network trigger. From the existing hardware implementation of the neural network trigger, with the 2D finder as input to a five-expert network setup, substantial benefits for the whole trigger system can be expected. Based on the neural network estimates, event trigger logics can be constructed in order to significantly improve the signal efficiency for low multiplicity events. Using the presented networks, with both finders an efficient z-cut trigger for two track events can be realized using z-cut $= 20\,\mathrm{cm}$: by requiring that at least one out of two estimated tracks has a z-estimate within this z-cut region, signal efficiencies of Gain2 $\gtrapprox 99\,\%$ can be achieved while the purity of the estimated neural network tracks is in the order of $80\,\%$. If a reduction of the trigger rate is required, the excellent signal efficiency allows to further confine the z-cut value.

The 3D finder can successfully be used for triggering real Belle II events with a significantly better track finding efficiency compared to the existing 2D finder. The observed $50\,\%$ larger number of IP tracks found by the 3D finder, combined with the similar Gain- and Loss-rates, are a strong arguments for a future realization of the presented 3D finder in hardware. Furthermore, the observed $50\,\%$ increase in the found IP track by the 3D finder is in agreement with the previous single tracks studies as well as with the previous studies on simulated initial state radiation events. An extension to expert networks for short tracks is important in a setup with the 3D finder as input since many of the additional tracks found by the 3D finder can otherwise not be processed by the neural network. Due to the resolution loss with the short track experts, the z-cut values should be set dynamically with different values depending on the number of related stereo TS. In order to realize tighter z-cut values in a future hardware implementations of the 3D finder, the z-cut values might additionally be defined as functions of the p_{T}-estimates of the track finder.

The neural networks trained with the background level with the highest occupancy turned out to provide the best precision for physics signal events, however they also result in the largest background rates. In particular, the background samples simulated with the full Phase 3 luminosity (15p3) perform optimally on signal events in the real data sample. This indicates larger than expected background conditions in the studied early Belle II runs. Hence, the background models have to be further optimized and ideally a retraining using real recorded background should be anticipated. In consideration of the results of the previous sections, it is recommended that the training is performed on pion tracks rather than muons. Since most of the background and signal tracks are not muons, a change of the particle type in the training data will reflect more realistic conditions. This improvement can add up to the usage of the correct background model in the training. Since many of the back-scattered background particles are protons [74] it is recommended to carry out studies with protons as training target and compare the results to a training on pion tracks.

Chapter 8.

Conclusion

In this thesis, the development and validation of the machine learning algorithms for the online first level track trigger have been presented: the model independent 3D track finder with a significantly improved track finding efficiency and the neural network trigger for the precise estimation of the polar angles θ and the longitudinal vertices z. The developed algorithms have successfully been validated, using Monte Carlo background simulations in combination with simulated initial state radiation signal events as well as with recorded data from early collisions in the Belle II experiment. Due to the upcoming high luminosity, Belle II will have to face severe levels of beam background tracks with vertices displaced along the beamline. Based on the neural network estimates, background tracks with displaced z-vertices can be suppressed already at the first trigger level such that the remaining trigger logics can be strongly relaxed. Especially for physics decay channels with a low track multiplicity in the final states (e.g. τ pair production or initial state radiation events) the efficiencies can be significantly increased. To date, the neural network trigger is already implemented on FPGA hardware and integrated into the first level trigger pipeline of the Belle II experiment where it receives TS hits and track estimates from the 2D finder as input.

The 3D track finder is a model independent Hough finder, where the track-to-hit relations are learned from a training data set. In the presented setup, a density based clustering algorithm is used to identify the tracks in the binned 3D Hough space. By using the additional stereo TS hits, the track finding efficiency and track parameter resolutions of the present 2D finder are significantly improved and additionally an estimate of the polar angle θ is provided. At the same time, this 3D finder already realizes an active 3D background suppression of tracks displaced along the z-axis. This 3D finder extends the CDC acceptance region of the first level trigger for tracks with low transverse momenta p_T as well as shallow polar angles θ. Since the cross section of the physics interactions in the Belle II detector typically increase towards shallow polar angles, a significant improvement in the signal efficiencies can be achieved with the 3D finder. An improvement of the track finding efficiency of about 50 % has been confirmed threefold: in simulation studies with single tracks and initial state radiation events as well as with early data from the Belle II experiment.

A 3D fitter has been implemented as a reference model for the 3D track reconstruction problem during the development of the neural network trigger. Its linearization concepts of an ideal 3D helical track entered in the preprocessing of the neural network. In this way, the neural network can learn the nonlinear corrections in the 3D track reconstruction. The neural network trigger has been developed as an ensemble of expert networks, where the present state of the art is using expert networks for the case of single missing stereo TS hits in the input. In combination with the 3D finder, an efficiency optimized neural network configuration of the pool of expert networks has been suggested. By training short track expert networks for the case of two missing stereo TS hits the additional 3D finder tracks with low-p_{T} or with shallow polar angles θ can be efficiently processed.

Trigger cuts based on the z- and θ-estimates of the neural network have been studied, just as they could be implemented in the GDL of the first level trigger. In particular, with a z-cut at 20 cm that has to be fulfilled by at least one track, excellent two track efficiencies of more than 99 % can be achieved in simulated ISR events as well as in recorded Belle II data with generic two track events from the IP. A corresponding rate of false positives in the order of only 20 % has been measured in a study with Belle II data containing displaced tracks. The observed distribution of the recorded Belle II data shows large background peaks in the region $|z| \geq 50\,\mathrm{cm}$. This agrees with the z-distribution in the background simulations, where a large fraction of these peaks corresponds to backscattering tracks at the material in the crotch part created initially by the luminosity background types. Due to the large z-displacement of the observed background peaks, there is plenty of room for the fine tuning of the neural network trigger cuts in favor of improved efficiencies or reduced background rates.

Appendix A.

Error Estimation

The two most important quantities in the analysis of the experimental results are track parameter resolutions and rates like efficiencies or fake rates. The used formulas for these quantities and their errors are documented in this appendix. As a reference model of the ideal histogram resolution, the binning error introduced by a histogram of a given bin-width is calculated. Furthermore, the method of track matching in the CDC is discussed, where pairs of different track types are formed based on their relation to CDC hits.

A.1. Statistical Errors

The trimmed RMS90 is used as a stable estimator of the resolution, where outliers are trimmed before calculating the variance. Using error propagation, the method of calculating the variance of the variance is used to obtain an estimator of the error of the resolution. The most important rates used in this thesis are the efficiencies and the fake rates. A general estimator for the error of a rate can be propagated from the standard error of the histogram size.

A.1.1. Gaussian Error Propagation

Let the quantity f be a function composed of N components x_i, $f(x_1, \ldots, x_N)$. An error Δf of f can be expressed in terms of the errors Δx_i of the components x_i. Assuming a Gaussian and independent distribution of the Δx_i, the contribution of each error Δx_i is weighted by the derivative $\partial f / \partial x_i$. The error contributions are added quadratically and the square root is taken to obtain the total statistical error.

$$\Delta f = \sqrt{\sum_{i=1}^{N} \left(\frac{\partial f}{\partial x_i} \Delta x_i \right)^2} \tag{A.1}$$

The Gaussian error propagation, is derived from the identity of Bienaymè for uncorrelated errors:

$$\mathrm{Var}\left[\sum_{i=1}^{n} x_i \right] = \sum_{i=1}^{n} \mathrm{Var}[x_i] \tag{A.2}$$

A.1.2. Moments and Variance

A random variable x that is distributed according to a Probability Density Function (PDF) can be described by its moments. The n-th moments μ_n of the distribution are defined by

$$\mu_n = \left\langle (x - \langle x \rangle)^n \right\rangle \tag{A.3}$$

where $\langle y \rangle$ denotes the expectation value of y. In a Gaussian distribution, all higher moments ($n > 2$) vanish. Commonly, only the first two moments are used and the distribution is described by its mean $\langle x \rangle$ and its variance. The variance can be written as

$$\mathrm{Var}[x] = \mu_2 = \left\langle (x - \langle x \rangle)^2 \right\rangle = \left\langle x^2 - 2x\langle x \rangle + \langle x \rangle^2 \right\rangle = \langle x^2 \rangle - \langle x \rangle^2 \tag{A.4}$$

This variance is the expectation value of the squared difference of the outcomes x to the mean $\langle x \rangle$. The standard deviation is defined as the square root of the variance

$$\sigma = \sqrt{\mathrm{Var}[x]} \tag{A.5}$$

To define an error of the variance, the same definition of the variance can be used to get the related variance to this altered random variable. Substituting $y_i = (x_i - \langle x \rangle)^2$ for the squared distance of each element x_i to the mean $\langle x \rangle$, the expectation value of y is the variance of x

$$\mathrm{Var}[x] = \langle y \rangle = \left\langle (x - \langle x \rangle)^2 \right\rangle \tag{A.6}$$

The variance of y is a measure used in the calculation of the variance of the variance of x:

$$\begin{align} \mathrm{Var}[y] &= \langle y^2 \rangle - \langle y \rangle^2 \tag{A.7} \\ &= \left\langle (x - \langle x \rangle)^4 \right\rangle - \left\langle (x - \langle x \rangle)^2 \right\rangle^2 \tag{A.8} \\ &= \mu_4 - \mu_2^2 \tag{A.9} \end{align}$$

where μ_4 is the fourth moment of the distribution, known as kurtosis. In a Gaussian distributions, all higher moments ($n > 2$) vanish. Non-zero higher moments can be interpreted as a measure of the deviation from an ideal Gaussian distribution.

A.1.3. Standard Error

The standard error of the arithmetic mean $\langle x \rangle$ in a sample of size N is also derived from the Bienaymè identity (Eq. A.2). Using the definition of the mean, the standard error follows:

$$\Delta\langle x \rangle = \sqrt{\mathrm{Var}\left[\langle x \rangle\right]} = \sqrt{\frac{\mathrm{Var}[x]}{N}} \tag{A.10}$$

Using the previously introduced notation of the variance as mean of the squared differences, also the error of the variance is directly defined:

$$\begin{aligned}\Delta\mathrm{Var}[x] &= \Delta\langle y\rangle &\text{(A.11)}\\ &= \sqrt{\frac{\mathrm{Var}[y]}{N}} &\text{(A.12)}\\ &= \sqrt{\frac{1}{N}\left(\mu_4-\mu_2^2\right)} &\text{(A.13)}\end{aligned}$$

Standard Deviation Error

The standard deviation is defined as the square root of the variance

$$\sigma = \Delta x = \sqrt{\mathrm{Var}[x]} = \sqrt{\mu_2} \tag{A.14}$$

An error of the standard deviation can be propagated from the previously defined standard error of the variance:

$$\begin{aligned}\Delta\sigma &= \left|\frac{\partial\sigma}{\partial\mu_2}\Delta\mu_2\right| &\text{(A.15)}\\ &= \left|-\frac{1}{2\sqrt{\mu_2}}\sqrt{\frac{1}{N}\left(\mu_4-\mu_2^2\right)}\right| &\text{(A.16)}\\ &= \frac{1}{2\sigma}\sqrt{\frac{1}{N}\left(\mu_4-\sigma^4\right)} &\text{(A.17)}\end{aligned}$$

A.1.4. Root Mean Square

The RMS (Root Mean Square) of a sample of random variables x is defined as the square root of the Mean Square Error (MSE)

$$\mathrm{RMS}[x] = \sqrt{\mathrm{MSE}[x]} = \sqrt{\langle x^2\rangle} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2} \tag{A.18}$$

By setting $x = z_{\mathrm{true}} - z_{\mathrm{estimate}}$, the RMS is a useful quantity to measure the accuracy of an estimator. If the expectation value of x is zero, $\langle x\rangle = 0$, the estimator is unbiased and the RMS is identical to the square root of the variance (standard deviation). A non-zero expectation value $\langle x\rangle$ is called bias. In general, the RMS can be decomposed into the variance and the bias

$$\mathrm{RMS}[x] = \sqrt{\mathrm{Bias}^2[x] + \mathrm{Var}[x]} = \sqrt{\langle x\rangle^2 + \langle x^2\rangle - \langle x\rangle^2} \tag{A.19}$$

where $\mathrm{Bias}[x] = \langle x\rangle$. The error of the RMS can be propagated from the error of the MSE, where the MSE denotes the expectation value of the squares $\mathrm{MSE}[x] = \langle x^2\rangle$.

Its error is obtained from the rule of the MSE of the mean

$$\mathrm{MSE}\left[\langle x^2\rangle\right] = \frac{\mathrm{MSE}[x^2]}{N} = \frac{\langle x^4\rangle}{N} \qquad \mathrm{RMS}\left[\langle x^2\rangle\right] = \sqrt{\frac{\mathrm{MSE}[x^2]}{N}} = \sqrt{\frac{\langle x^4\rangle}{N}} \tag{A.20}$$

The error of the RMS is then propagated from the error of the MSE (similar to the error calculation of the standard deviation):

$$\Delta\mathrm{RMS}[x] = \left|\frac{\partial\mathrm{RMS}[x]}{\partial\langle x^2\rangle}\Delta\langle x^2\rangle\right| = \tag{A.21}$$

$$= \frac{1}{2\cdot\mathrm{RMS}[x]}\sqrt{\frac{\langle x^4\rangle}{N}} \tag{A.22}$$

$$= \frac{1}{2}\sqrt{\frac{\langle x^4\rangle}{N\cdot\langle x^2\rangle}} \tag{A.23}$$

Trimmed Root Mean Square

In real distributions, where some outliers on top of a central Gaussian distribution would dominate the RMS, a better estimator for the resolution can be defined by the trimmed RMS. Before calculating the RMS, the tails of the distribution are cut off and only the central fraction of $p\%$ of the values are kept.

To this end, a number of elements to be cut off at each end is defined as

$$c = \left\lfloor\frac{(1-p)\cdot N}{2}\right\rfloor \tag{A.24}$$

where p is the percentage of events that should be kept and N is the number of elements. After sorting the samples x_i by size, the c smallest and largest values are trimmed. The resulting trimmed sample of elements x_i' has $N-2c$ elements.

As a measure of the resolution, the RMS90 can be calculated in the previously described way using the trimmed sample with $p = 90\,\%$. The error of this trimmed RMS is calculated in the same way as the error of the RMS but using the trimmed sample. Due to the trimming of outliers, the RMS90 as well as its error is better than the RMS and its error on the full sample. Compared to the full RMS it is a quite stable estimator that provides an estimate of a central Gaussian distribution.

Finite Sample

In a discrete sample with N elements, the mean is explicitly defined as

$$\langle x\rangle = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{A.25}$$

The sample variance is the expectation value of the squared difference of the x_i to their mean $\langle x \rangle$.

$$\mathrm{Var}[x] = \frac{1}{N-1} \sum_{i=1}^{n} (x_i - \langle x \rangle)^2 \tag{A.26}$$

where the normalization with $N-1$ is a correction in a finite sample size to avoid underestimating the variance. One degree of freedom is already fixed by the calculation of the mean. With a sufficient size of the sample, this correction is only a minor effect and the sample corrections can be neglected if the sample size N is large.

For small sample sizes, the previously defined σ can be corrected to the sample standard deviation S

$$S = \frac{N}{N-1}\sigma = \frac{N}{N-1}\sqrt{\left(\langle x^2 \rangle - \langle x \rangle^2\right)} \tag{A.27}$$

$$= \frac{N}{N-1}\sqrt{\mu^2} \tag{A.28}$$

For this corrected estimator of the sample standard deviation, the variance can be calculated by [112]:

$$\mathrm{Var}[S^2] = \frac{1}{N}\left(\mu_4 - \frac{N-3}{N-1}\mu_2^2\right) \tag{A.29}$$

which can be used to calculate the error of the sample standard deviation.

$$\Delta S = \frac{1}{2S}\sqrt{\frac{1}{N}\left(\mu_4 - \frac{N-3}{N-1}\mu_2^2\right)} \tag{A.30}$$

A.1.5. Efficiency and Rate Errors

Rates can be defined by dividing a set of samples into two distinct classes. The rate is then defined as the fraction of occurrences of the samples in one class in the total sample. By labelling the two subsets of distinct classes N and F, the total sample set is $N + F$. Efficiencies are special rate type, which measure the fraction of the correct class samples out of a number of samples (e.g. in an efficiency, N corresponds to the correct class and F to the false class). The rate ϵ can be expressed as

$$\epsilon = \frac{N}{N+F} \tag{A.31}$$

To define an error of the rate, the simple variance of the sample size for N and F are used here:

$$\sigma_N = \sqrt{N} \qquad \sigma_F = \sqrt{F} \tag{A.32}$$

In the error propagation, the contributions of the errors are proportional to the gradients:

$$\frac{\partial \epsilon}{\partial N} = \frac{F}{(N+F)^2} \qquad \frac{\partial \epsilon}{\partial F} = -\frac{N}{(N+F)^2} \tag{A.33}$$

Using the formula (Eq. A.1), the errors σ_N^2 and σ_F^2 are propagated to obtain the error $\Delta\epsilon$ of the rate ϵ:

$$\Delta\epsilon = \sqrt{\left(\frac{\partial\epsilon}{\partial N}\right)^2 \sigma_N^2 + \left(\frac{\partial\epsilon}{\partial F}\right)^2 \sigma_F^2} \tag{A.34}$$

$$= \sqrt{\frac{N \cdot F}{(N+F)^3}} \tag{A.35}$$

A.2. Histogram Binning Error

Due to its bin-width, a histogram introduces an intrinsic error to calculations of mean values based on its bin contents. This is especially relevant for the track finder, where the track parameters are calculated as mean values over the bins in the Hough space. For the simple case of a 1D histogram for a single binned value which is uniformly distributed, the RMS introduced solely by the binning is calculated here.

With M bins for a variable x in the range $x \in [x_{\text{min}}, x_{\text{max}}]$, the bin widths w are

$$w = \frac{x_{\text{max}} - x_{\text{min}}}{M} \tag{A.36}$$

The binned value x_b, at the center of the bins, is

$$x_b = \left\lfloor \frac{x}{w} \right\rfloor w \tag{A.37}$$

where $\lfloor\rfloor$ is the floor function.

The difference of x and its binned value x_b is equal do the modulo of x with the bin width w

$$d = x - x_b = x\%w \tag{A.38}$$

where % is the modulo operator.

With N samples x_i, the expectation value of d is

$$\langle d \rangle = \frac{1}{N}\sum_i^N d_i \underbrace{=}_{\text{uniform}} \frac{w}{2} \tag{A.39}$$

where the second equality uses the assumption that the x are uniformly distributed. The variance can then be calculated as

$$\text{Var}[d] = \frac{1}{N}\sum_j^N (d_j - \langle d \rangle)^2 \tag{A.40}$$

Due to the assumption of a uniform distribution, the variance can be rearranged as follows

$$\mathrm{Var} = \frac{1}{N}\sum_{j}^{N}\left(d_j - \frac{w}{2}\right)^2 \tag{A.41}$$

$$= \frac{1}{N}\sum_{j}^{N}\left(d_j^2 - d_j w + \frac{w^2}{4}\right) \tag{A.42}$$

$$= \frac{1}{N}\sum_{j}^{N} d_j^2 - \frac{1}{N}\sum_{j}^{N} d_j w + \underbrace{\frac{1}{N}\sum_{j}^{N}\frac{w^2}{4}}_{\frac{w^2}{2}} \tag{A.43}$$

where the third summand is a trivial sum over a constant.

In order to calculate the first and second summands, let d_j be ideally uniformly distributed by rewriting it as:

$$d_j = \frac{jw}{N} \tag{A.44}$$

The first sum can then by calculated by using the formula for the sum of the first N quadratic number:

$$\frac{1}{N}\sum_{j}^{N} d_j^2 = \frac{w^2}{N^3}\sum_{j}^{N} j^2 = \frac{w^2}{N^3}\frac{N(N+1)(2N+1)}{6} \tag{A.45}$$

The second sum is

$$\frac{1}{N}\sum_{j}^{N} d_j w = w\langle d\rangle = \frac{w^2}{2} \tag{A.46}$$

Putting it all together, the variance is

$$\mathrm{Var} = \frac{w^2}{N^3}\frac{N(N+1)(2N+1)}{6} - \frac{w^2}{2} + \frac{w^2}{4} \tag{A.47}$$

$$= \frac{w^2}{12}\left(\frac{N(N+1)(2N+1)}{6N^3} - 3\right) \tag{A.48}$$

$$= \frac{w^2}{12}\left(\frac{N^3 + 6N^2 + 2N}{N^3}\right) \tag{A.49}$$

$$\tag{A.50}$$

In the limit of large N, the term in brackets approximates one. Thus, the variance induced solely by the binning is

$$\mathrm{Var} = \frac{w^2}{12} \tag{A.51}$$

the square of the bin width divided by twelve.

A.3. Track Matching

For two different sets of track types, the matching identifies which tracks describe the same physical particle. This is essential for the evaluation of the novel track trigger methods presented in this thesis. Resolutions can only be calculated for matched track pairs and efficiencies are calculated as the fractions of matched tracks. Since a reference set containing the correct matching assignment does not exist, the efficiency of the matching itself cannot be determined independently. Only relative ratios can be defined that describe how well the tracks in each class are represented by each track in the other class.

Within the basf2 framework, dedicated modules are implemented for the trigger track matching, which create relations between trigger tracks and reference tracks based on the coincidence of TS hits. The matching module for the MC track to trigger track relations was developed in analogy to the matching used for offline tracking by the Belle II tracking group as presented in [67]. In particular the labels for the different classes of unmatched tracks are chosen in accordance with the offline tracking.

The principle idea in the track matching is that tracks that should be matched have relations to the same hits. Using a confusion matrix of the track-to-hit relations, a match is created for the track pair with the largest coincidence of hits. As an additional measure of the coincidence, two ratios are calculated for each possible track pair. These ratios are used to define cuts during the matching.

As reference tracks for the matching, simulated MC tracks as well as reconstructed tracks can be used. Any type of trigger track can be considered as an estimated track, which includes 2D finder tracks, 3D finder tracks, and neural network tracks. The reference tracks are assumed to have correct relations to the hits, while larger errors in the track-to-hit relations are assumed for the estimated trigger tracks. Especially in events with background many hits are not related to any of the reference tracks. Since relations of hits to MC tracks are unique, the matching of MC tracks to trigger tracks is carried out asymmetrically: a trigger track is only matched to a reference track if the reference track can explain the trigger track better than the background. Hits unrelated to any MC track are assumed to be created by the background. In the matching algorithm, this is realized by adding a fallback reference track which is related to all the hits without relations to any reference track. If an estimated trigger track has the maximal coincidence of hits with this fallback track, it is not related to any of the reference tracks. For single track simulations presented in this thesis this has the effect that more than 50 % of the hits in the trigger track have to be related to the reference track.

However, the relations of reconstructed tracks to the hits is not unique as in the case of MC tracks. This means, multiple reconstructed tracks can be related to the same TS hits. Therefore, the set of reference tracks is not extended by a fallback track catching unrelated hits. Hence, the matching with reconstructed tracks is completely symmetric.

A.3.1. Matching Algorithm

In detail, the matching algorithm carries out the following steps: Two classes of tracks are going to be matched: a set of reference tracks $\mathcal{A}$ and a set estimated tracks $\mathcal{B}$. Additionally, a set of hits $\mathcal{H}$ is required, with relations to both classes of tracks. In the matcher for the CDC trigger, this set of hits are the TSs in the event. However, since each TS has the relations to their contained CDC hits, the method can also be applied directly to the CDC hits.

The core logic of the matcher is based on a confusion matrix $\mathbf{C}$ of the track-to-hit relations in both classes. In order to construct this matrix, the following concept is used: let $K = |\mathcal{H}|$ be the number of hits $H_i \in \mathcal{H}, i \in [1, \ldots, K]$, $L = |\mathcal{A}|$ the number of tracks $A_j \in \mathcal{A}$ in the set of reference tracks $\mathcal{A}$ with $j \in [1, \ldots, L]$, and $M = |\mathcal{B}|$ the number of tracks $B_k \in \mathcal{B}$ in the class of estimated trigger tracks $\mathcal{B}$ with $B_k \in [1, \ldots, M]$. For each track class, a binary relation matrix, $\mathbf{R}^{(\mathcal{A})}$ and $\mathbf{R}^{(\mathcal{B})}$, of the track-to-hit relations is created, with a row for each hit and a column for each track. The superscripts $(\mathcal{A})$ and $(\mathcal{B})$ are subsequently used to distinguish several objects related to the respective track class. These matrices are binary in the sense that each track is related to each hit maximally once:

$$R_{i,j} \begin{cases} 1 \text{ if hit } i \text{ is related to track } j \\ 0 \text{ otherwise} \end{cases} \tag{A.52}$$

These binary matrices $\mathbf{R}^{(\mathcal{A})}$ and $\mathbf{R}^{(\mathcal{B})}$ are illustrated in Fig. A.1, together with the row-sum and the column-sum. The row-sum corresponds to the number of hits per track and the column-sum to the number of tracks per hit. In the case of estimated tracks, the hit-to-track relation is not unique. This means that a hit can be related to several tracks and therefore several columns of the binary matrix are filled with a 1. In the case of the reference tracks $\mathcal{A}$, the previously mentioned column for the fallback track is added. This has the effect that each hit is related to at least one reference track. Hence, the matrix $\mathbf{R}^{(\mathcal{B})}$ has dimension $\dim\left(\mathbf{R}^{(\mathcal{B})}\right) = (K \times M)$ while the matrix $\mathbf{R}^{(\mathcal{A})}$ has dimension $\dim\left(\mathbf{R}^{(\mathcal{A})}\right) = (K \times (L+1))$. The matrices $\mathbf{R}^{(\mathcal{A})}$ and $\mathbf{R}^{(\mathcal{B})}$ simply provide an interface to query hit relations in the virtually extended set of tracks. Both matrices are filled in a loop over all the existing track-to-hit relations in the event. Fallback relations $R^{(\mathcal{A})}_{i,L+1}$ are created for those hits H_i without any relation to a track in the set $\mathcal{A}$.

The confusion matrix $\mathbf{C}$ of the common hits in the track classes $\mathcal{A}$ and $\mathcal{B}$ can be created as the matrix product of the relation matrices $\mathbf{R}^{(\mathcal{A})}$ and $\mathbf{R}^{(\mathcal{B})}$. Using reference tracks as rows and estimated tracks as columns, this confusion matrix is given by

$$\mathbf{C} = \mathbf{R}^{(\mathcal{A})^{\mathsf{T}}} \cdot \mathbf{R}^{(\mathcal{B})} \tag{A.53}$$

where $\mathbf{R}^{(\mathcal{A})^{\mathsf{T}}}$ is the transposed matrix $\mathbf{R}^{(\mathcal{A})}$. The confusion matrix $\mathbf{C}$ has the dimension $\dim(\mathbf{C}) = ((L+1) \times M)$. The elements of $\mathbf{C}$ can be expressed as the sum over

$\mathbf{R}^{(\mathcal{A})}$

	A_1	$\dots$	A_L	A_{L+1}	$\sum_{L+1}$
H_1	$R_{1,1}$	$\dots$	$R_{1,L}$	$R_{1,L}$	1
H_2	$R_{2,L}$	$\dots$	$R_{2,L}$	$R_{2,L}$	1
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
H_K	$R_{K,L}$	$\dots$	$R_{K,L}$	$R_{K,L}$	1
$\mathbf{q}^{(\mathcal{A})}$	$q_1^{(\mathcal{B})}$	$\dots$	$q_L^{(\mathcal{A})}$	$q_{L+1}^{(\mathcal{A})}$	

$\mathbf{R}^{(\mathcal{B})}$

	B_1	$\dots$	B_M	$\sum_M$
H_1	$R_{1,1}$	$\dots$	$R_{1,M}$	≥ 0
H_2	$R_{2,1}$	$\dots$	$R_{2,M}$	≥ 0
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
H_K	$R_{K,1}$	$\dots$	$R_{K,M}$	≥ 0
$\mathbf{q}^{(\mathcal{B})}$	$q_1^{(\mathcal{B})}$	$\dots$	$q_M^{(\mathcal{B})}$	

Figure A.1.: Illustration of the binary matrices $\mathbf{R}^{(\mathcal{A})}$ and $\mathbf{R}^{(\mathcal{B})}$ used to hold the relations between a set of estimated tracks $\mathcal{B}$ and a set of reference tracks $\mathcal{A}$. The set of reference tracks is extended by the fallback column A_{L+1}, which is related to background hits without any relation to a reference track. In the case of MC reference tracks, each hit is related to exactly one track in $\mathcal{A}$, which means the column sum is 1 for each hit. The row-sums corresponding to the number of hits per track are shown as $\mathbf{q}^{(\mathcal{A})}$ and $\mathbf{q}^{(\mathcal{B})}$.

the product of the respective binary matrices $\mathbf{R}^{(\mathcal{A})}$ and $\mathbf{R}^{(\mathcal{B})}$:

$$C_{j,k} = \sum_{i=1}^{K} R_{i,j}^{(\mathcal{A})} \cdot R_{i,k}^{(\mathcal{B})} \tag{A.54}$$

where the product $R_{i,j}^{(\mathcal{A})} \cdot R_{i,k}^{(\mathcal{B})}$ is 1 if the hit i is related to both, the reference track j and the estimated track k, and 0 otherwise. Alternatively, $\mathbf{C}$ could be filled like a 2D histogram in a loop over all the hits $H_i \in \mathcal{H}$, where each field $C_{j,k} \in \mathbf{C}$ is incremented if the hit has a relation to both tracks. This resulting confusion matrix is illustrated in Fig. A.2.

For the calculation of the matching, two reciprocal ratios are defined that describe how well each tracks from one class is represented by each tracks of the other class. The representation of a specific track $A_j \in \mathcal{A}$ is by an estimated track $B_k \in \mathcal{B}$ is called efficiency; reciprocally the representation of an estimated track $B_k \in \mathcal{B}$ by a reference track $A_j \in \mathcal{A}$ is called purity. To this end, additionally the total number of hits for each track is required which are described by the vectors $\mathbf{q}^{(\mathcal{A})}$ and $\mathbf{q}^{(\mathcal{B})}$. Using the matrices $\mathbf{R}^{(\mathcal{A})}$ and $\mathbf{R}^{(\mathcal{B})}$, the elements of these two vectors are obtained by row-wise summing up the hit axes in the relation matrices:

$$q_j^{(\mathcal{A})} = \sum_{i=1}^{K} R_{i,j}^{(\mathcal{A})} \qquad q_k^{(\mathcal{B})} = \sum_{i=1}^{K} R_{i,k}^{(\mathcal{B})}. \tag{A.55}$$

This summation in the binary matrices is indicated in Fig. A.1 and their relation to the confusion matrix is indicated in Fig. A.2.

$\mathbf{C}$

	B_1	B_2	$\dots$	B_M	$\mathbf{q}^{(\mathcal{A})}$
A_1	$C_{1,1}$	$C_{1,2}$	$\dots$	$C_{1,M}$	$q_1^{(\mathcal{A})}$
A_2	$C_{2,1}$	$C_{2,2}$	$\dots$	$C_{2,M}$	$q_2^{(\mathcal{A})}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
A_L	$C_{L,1}$	$C_{L,2}$	$\dots$	$C_{L,M}$	$q_L^{(\mathcal{A})}$
A_{L+1}	$C_{L+1,1}$	$C_{L+1,2}$	$\dots$	$C_{L+1,M}$	$q_{L+1}^{(\mathcal{A})}$
$\mathbf{q}^{(\mathcal{B})}$	$q_1^{(\mathcal{B})}$	$q_2^{(\mathcal{B})}$	$\dots$	$q_M^{(\mathcal{B})}$	

Figure A.2.: Confusion matrix used to match a number of M estimated tracks in $\mathcal{B}$ to a number of L reference tracks in $\mathcal{A}$. The additional row A_{L+1} of the matrix $\mathbf{C}$ is the fallback row collecting hits without a relation to any reference track. The column $\mathbf{q}^{(\mathcal{A})}$ is the total number of hits for each reference track and the row $\mathbf{q}^{(\mathcal{B})}$ is the total number of hits for each estimated track.

All fields in the confusion matrix $\mathbf{C}$ correspond to a hypothetical matching track pair of a track in $\mathcal{A}$ to a track in $\mathcal{B}$. In order to obtain an optimal matching, the best matching partner is searched by looking for a maximum coincidence of hits. This means to search for the row-wise and column-wise largest number of common hits in the matrix $\mathbf{C}$ and noting the index with the arg max function as additional vectors $\mathbf{l}^{(\mathcal{A})}$ and $\mathbf{l}^{(\mathcal{B})}$. At this crucial point resides the asymmetry between the otherwise symmetric matching mechanism. In the loop over the estimated tracks, i.e. the column in Fig. A.2, the maximum for each column is selected from the rows in the range $[1, L+1]$. This means, the fallback reference track A_{L+1} is included. In contrast, in the loop over the reference tracks, i.e. the rows in Fig. A.2, the maximum for each row is selected from the columns in the range $[1, M]$. The indices of the vectors with arg max indices of the row-wise and column-wise maximum coincidences are given by

$$l_j^{(\mathcal{A})} = \underset{k\in[1,\dots,M]}{\arg\max}\,(C_{j,k}) \qquad l_k^{(\mathcal{B})} = \underset{j\in[1,\dots,L+1]}{\arg\max}\,(C_{j,k}). \tag{A.56}$$

The vectors $\mathbf{l}^{(\mathcal{A})}$ and $\mathbf{l}^{(\mathcal{B})}$ can be interpreted as preliminary directed relations: a track A_j is related to the track B_k with $k = l_j^{(\mathcal{A})}$; vice versa, a track B_k is related to the track A_j with $j = l_k^{(\mathcal{B})}$. In ideal events containing only correct track-to-hit relations for both track classes the matching is directly given by these vectors.

However, reconstruction errors and background contamination pose difficulties for a unique matching of track pairs based solely on the unidirectional maximum hit coincidence criteria. For this reason, two ratios are reciprocally defined for each

	B_1	B_2	$\dots$	B_M	$\mathbf{q}^{(\mathcal{A})}$	$\mathbf{l}^{(\mathcal{A})}$	max–efficiency
A_1	$C_{1,1}$	$C_{1,2}$	$\dots$	$C_{1,M}$	$q_1^{(\mathcal{A})}$	$l_1^{(\mathcal{A})}$	$\varepsilon_1^{(\mathcal{A})}$
A_2	$C_{2,1}$	$C_{2,2}$	$\dots$	$C_{2,M}$	$q_2^{(\mathcal{A})}$	$l_2^{(\mathcal{A})}$	$\varepsilon_1^{(\mathcal{A})}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
A_L	$C_{L,1}$	$C_{L,2}$	$\dots$	$C_{L,M}$	$q_L^{(\mathcal{A})}$	$l_L^{(\mathcal{A})}$	$\varepsilon_L^{(\mathcal{A})}$
A_{L+1}	$C_{L+1,1}$	$C_{L+1,2}$	$\dots$	$C_{L+1,M}$	$q_{L+1}^{(\mathcal{A})}$	$l_{L+1}^{(\mathcal{A})}$	$\varepsilon_{L+1}^{(\mathcal{A})}$
$\mathbf{q}^{(\mathcal{B})}$	$q_1^{(\mathcal{B})}$	$q_2^{(\mathcal{B})}$	$\dots$	$q_M^{(\mathcal{B})}$			
$\mathbf{l}^{(\mathcal{B})}$	$l_1^{(\mathcal{B})}$	$l_2^{(\mathcal{B})}$	$\dots$	$l_M^{(\mathcal{B})}$			
max–purity	$\varepsilon_1^{(\mathcal{B})}$	$\varepsilon_2^{(\mathcal{B})}$	$\dots$	$\varepsilon_M^{(\mathcal{B})}$			

Figure A.3.: Confusion matrix side by side with the additional columns used by the track finding algorithm. $\mathbf{l}^{(\mathcal{A})}$ and $\mathbf{l}^{(\mathcal{B})}$ denote the indices of the row-wise and column-wise maxima. Calculated from the respective maxima positions, the row-wise max–efficiency and the column-wise max–purity is shown. The additional row A_{L+1} of the matrix $\mathbf{C}$ is the fallback row collecting hits without a relation to any reference track. The column $\mathbf{q}^{(\mathcal{A})}$ is the total number of hits for each reference track and the row $\mathbf{q}^{(\mathcal{B})}$ is the total number of hits for each estimated track.

field of the confusion matrix $\mathbf{C}$:

$$\text{efficiency}_{j,k} = \frac{C_{j,k}}{q_j^{(\mathcal{A})}} \qquad \text{purity}_{j,k} = \frac{C_{j,k}}{q_k^{(\mathcal{B})}} \tag{A.57}$$

The coefficients for these ratios can be read off Fig. A.2.

In principle, the efficiencies and purities for all track to track relations can be represented as a matrix with the same dimension as the confusion matrix $\mathbf{C}$. However, in the subsequently described matching algorithm only the efficiencies and purities of the tracks with the largest coincidence are used. This is done by selecting the respective maximum fields using the previously defined arg max-vectors $\mathbf{l}^{(\mathcal{A})}$ and $\mathbf{l}^{(\mathcal{B})}$. The coefficients $\varepsilon_j^{(\mathcal{A})}$ of the row-wise max–efficiency vector and the coefficients $\varepsilon_k^{(\mathcal{B})}$ of the column-wise max–purity vector are given by:

$$\varepsilon_j^{(\mathcal{A})} = \text{efficiency}_{j,k}; k = l_j^{(\mathcal{A})} \tag{A.58}$$

$$\varepsilon_k^{(\mathcal{B})} = \text{purity}_{j,k}; j = l_k^{(\mathcal{B})} \tag{A.59}$$

So far all elements required for the matching algorithm have been defined. An extended illustration of the confusion matrix, including additionally the row-wise and column-wise analysis vectors, is shown in Fig. A.3.

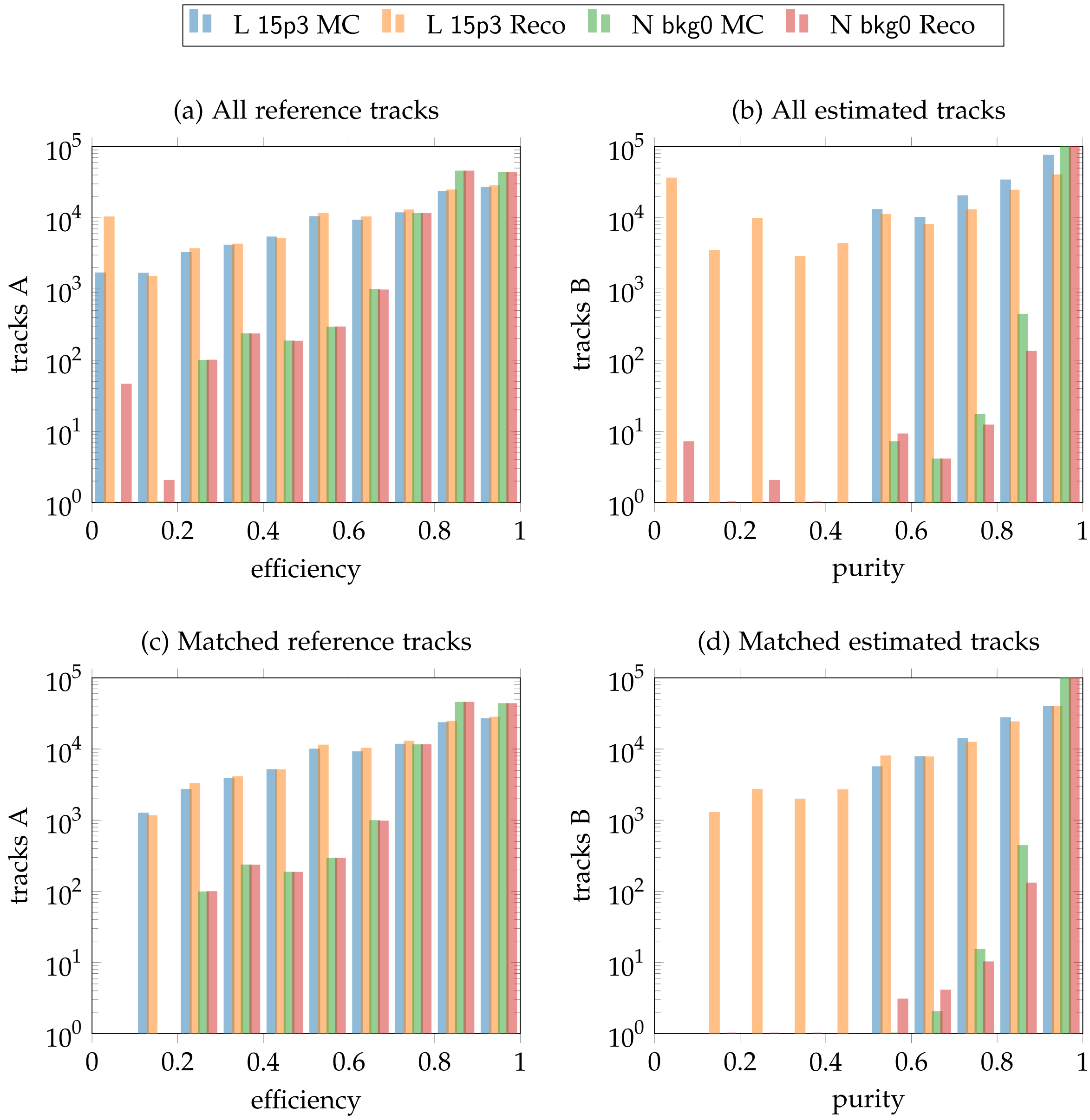


Figure A.4.: Histograms of the efficiencies and purities for the matching of 3D finder tracks to MC tracks and to reconstructed tracks using ten bins. The same four test data sets are used as shown subsequently in Tab. A.2. The following values are shown: (a) efficiencies of all reference tracks; (b) purities of all estimated tracks; (c) efficiencies of the reference tracks matched to estimated tracks; (d) purities of the estimated tracks matched to reference tracks.

parameter	value
`min_efficiency`	0.1
`min_purity`	0.1

Table A.1.: The two cut parameters used by the matching algorithm.

Matching Logic

The actual matching uses the two cut parameters listed in Tab. A.1. Typical distributions of all efficiencies and purities as well as efficiencies and purities in the matched tracks are shown in Fig. A.4.

In a loop over the estimated tracks $\mathcal{B}$, the relations are possibly created to the reference tracks with a minimum purity. For a trigger track B_k, the explicit purity check is carried out by requiring

$$\varepsilon^{(\mathcal{B})} > \texttt{min\_purity}. \tag{A.60}$$

This means a cut is to be applied on the distribution shown in Fig. A.4 (b).

However, since the reference track with the maximum coincidence is selected, the minimal possible purity is limited by the use of a fallback column. In the worst case, where all the tracks share hits, the purity is implicitly limited to

$$\text{purity} \geq \frac{1}{L+1} \tag{A.61}$$

which corresponds to 50 % in single track events with $L = 1$. This limit can be seen in Fig. A.4 (b) in the distributions of the purities for the matching to MC tracks.

Subsequently the reference track A_j with the largest coincidence is selected using $j = l_k^{(\mathcal{B})}$. A matching is possible if this reference track is not the background fallback track. In turn this reference track is required to have the largest coincidence with the current trigger track. This is achieved by selecting $k' = l_j^{(\mathcal{A})}$ and requiring $k \equiv k'$. If this condition is fulfilled, a directed match is created from the trigger track to the reference track. This means that the directed cuts from estimated tracks to MC tracks are solely based on the maximum coincidence criterion and the cut parameters are not used at all. In contrast, in the case of matching to reconstructed tracks where the matching is carried out without a fallback column, the `min_purity` cut is used. This is the set of tracks used as the basis for the efficiency and resolution studies throughout this thesis. The purities after the match are shown in Fig. A.4 (d).

Independently, the matching in the inverse direction is created in a loop over the reference tracks. The main requirement for a reference track A_j is to make it through the efficiency cut which means

$$\varepsilon_j^{(\mathcal{A})} > \texttt{min\_efficiency}. \tag{A.62}$$

	L 15p3 MC	L 15p3 Reco	N bkg0 MC	N bkg0 Reco
A missed	1652	43409	2	132
A merged	2352	1347	4	5
A matched	91884	98701	99964	99974
B ghost purity	0	35420	0	7
B fallback	47453	0	18	0
B clone	7546	15979	72	73
B ghost efficiency	3101	106	0	0
B matched	92368	98774	99964	99974
A all	95888	143457	99970	100111
B all	150468	150279	100054	100054
skipped events	4112	4217	30	30

Table A.2.: Exemplary numbers of matched and unmatched tracks with the 3D finder using the matching parameters of `min_efficiency` = 0.1 and `min_purity` = 0.1. For reconstructed tracks (Reco) and MC tracks, two typical single track data ranges from the experimental sections are shown. The test data set range N contains tracks within the full CDC acceptance region and the range L includes low-p_T and shallow-θ tracks (see Tab. 5.7 in Sec. 5.3.1). Within each range 10^5 single track are simulated for the cases with background (15p3) and without background (bkg0).

An exemplary set of the efficiencies prior to the application of this cut is shown in Fig. A.4 (a). Additionally, the related estimated track B_k is selected using $k = l^{(\mathcal{A})}$. The related estimated track in turn must have the largest coincidence with the current reference track. This is tested by selecting $j' = l^{(\mathcal{B})}$ and requiring $j \equiv j'$. If this condition is fulfilled, the relation from the reference track A_j to the trigger track B_k is created. Figure A.4 (c) shows the efficiencies of the matched tracks in the example data sets.

Classes of Unmatched Tracks

There are different reasons for a track not to be matched which are distinguished by the matching algorithm. A list of the track counts in all the different track classes is shown in Tab. A.2. The following total numbers in Tab. A.2 correspond to the total numbers of the previously discussed distributions shown in Fig. A.4: "A all" is the total number of all reference tracks in Fig. A.4 (a); "B all" is the total number of all estimated tracks in Fig. A.4 (b); "A matched" is the total number of all matched reference tracks in Fig. A.4 (c); "B matched" is the total number of all matched estimated tracks in Fig. A.4 (d).

The estimated tracks are either classified as matched or one of ghost-purity, background, or ghost-efficiency. Estimated tracks with insufficient purity, i.e. purity $\leq$ `min_purity`, are classified as ghost-purity track. As explained above, with the setting `min_purity` $= 0.1$ this can only happen for Reco track matching without the fallback track. If an estimated track ends up to have the largest coincidence of hits with this fallback track, it is classified as background track. If the maximum coincidence is with a real reference track, this real track is further inspected. If it has the maximum coincidence with the current track, the current track is classified as matched. However, if the reference track redirects back to a different column, two unmatched cases are distinguished depending on the efficiency of the reference track. With efficiency $\geq$ `min_efficiency` the estimated track is classified as a clone. In the case of a small efficiency of efficiency $<$ `min_efficiency` the estimated track is estimated as the second ghost type: a ghost-efficiency track.

In principle the same classes of non-matching tracks are distinguished for the reference tracks, however with a different interpretation. In a loop over the rows of the confusion matrix **C**, each reference track is classified into one of the categories: missing, merged, matched. A reference track with efficiency $<$ `min_efficiency` is classified as missing track. For reference tracks with a sufficient efficiency of efficiency $\geq$ `min_efficiency`, the estimated track with the largest coincidence is selected from the confusion matrix **C**. If the best matching estimated track refers back to a different MC track, the current reference track is classified as merged. The merged tracks in the set of reference tracks are structurally equivalent to the clone tracks in the set of estimated tracks. If the estimated track points back to the current MC track, this track is classified as match.

Appendix B.

Appendix Background

B.1. Review of the Background Simulation

In the majority of simulated background events, the primary background tracks are not visible in the CDC, as most of the tracks are not contained in the geometrical acceptance region of the CDC. Only secondary particles, created via scattering interactions with the detector material, increase the hit occupancy in all subdetectors and may become visible in the detector. Although the overall track and hit rates can be measured from the official background campaign data (see Sec. 4.4.2), the lack of track statistics does not yet allow to extract the detailed topology of the background tracks. Using a newly developed method of the machine background simulation, a high statistics background simulation was carried out [74]. This extends the work of the Belle II background group, from a per hit analysis to a per track analysis.

The study in [74] was carried out during the development of background campaign 15. Therefore, the same generator-, detector-, collider-parameters were used in order to describe the same physics processes as in background campaign 15. Especially, the same mask settings were used in the machine simulation with SAD. Different to the studies with a background mixer shown in Sec. 4.4.2 , in [74] the background types were inspected separately without mixing. This means, the real time equivalents of each background type were directly used to calculate the track rates for each background type separately without mixing. The mixing of background in Sec. 4.4.2 creates additional fake tracks composed of hits from different background tracks. In contrast, the direct background simulation without mixing, used in [74], allows to study the true MC vertex positions of correct background tracks.

B.1.1. SAD Machine Simulation

According to the physics models of the background processes Touschek, Coulomb and Brems, deviations from the nominal trajectory can be applied to the particles in the bunches. Due to limited computational resources, not all particles in the bunch can be tracked (there are $O(10^7)$ particles in a bunch). Instead, only a small number of particles (250 in background campaign 15) are tracked per bunch. The initial deviations $\delta p/p$ for each tracked particle w.r.t. the nominal momentum p of the par-

parameter	range
`npar`	250 - 1000
`maxturns`	100 - 1000

Table B.1.: Parameters to control the backgronud generation

ticles on the beam trajectory are drawn randomly within the range of a probability density function for the background model. Hence, each initial background particle is assigned with a weight. Only the weighted distribution of the initial background particles corresponds to the physics model of the whole bunch.

These rated particles in the bunches are tracked for several turns ($O(1000)$) in the SuperKEKB HER and LER rings, while observing their transverse elongation d. If a particle has $d > d_{\max}$ the particle hits material of the beam wall and it is lost from the beam. The output of SAD are the four-vectors at the loss positions s of the lost, rated particles, where s is the parameter along the SuperKEKB collider ring. Knowing the initially assigned weight, the rate of this particle loss process can be estimated. The four-vectors of the lost particles in the vicinity of the detector ($|s| < 4\,\mathrm{m}$) are used as input to the detector simulation within basf2, where they are tracked through the Belle II material using Geant4 [113, 71]. Note that in the vicinity of Belle II, the parameter s along the $\approx 2\,\mathrm{km}$ approximately corresponds to the Cartesian z-axis.

SAD is an older software project written in Fortran that had been used successfully in the simulation of the predecessor KEKB with Belle. Due to its age, there are challenging issues with the code; new developers do not know Fortran (poor maintainability); multi threaded execution was not foreseen (poor parallelization); computer memory was expected much smaller than it is today (small workload per thread). Instead of rewriting the whole accelerator simulation, one can cope with these challenges by starting many small jobs with different random seeds and combining the results. Using new compilers, the memory per job could be increased, which allows to increase the number of tracked particles. The main parameters controlling a SAD run are `npar` and `maxturns` shown in B.1. The parameter `npar` is the number of rated particles to be tracked. The parameter `maxturns` is the maximum number of turns the particles are tracked in the detector.

Consistent with the probability distribution of a deviation in the initial state, the finally lost particle is equipped with a rate. In order to obtain unrated background events, a real time equivalent T is chosen; according to their rate R_i, each rated particle i is then repeated N_i^{real} times as input to Geant4:

$$N_i^{\mathrm{real}} = R_i \cdot T \tag{B.1}$$

where N_i^{real} is the number of occurrences of the particle loss i within the real time interval T. This produces the correct, unrated, physical distribution of the lost particles. But the re-use of the same particles many times as input to the simulation of the material interaction is susceptible to underrepresent the full background event topologies. Especially if a neural network training with background tracks is pursued, overfitting to the few discrete loss positions in a small background sample has to be avoided, where the networks do not generalize to the underlying continuous background distribution.

B.1.2. Large Scale SAD Simulation

A direct method to increase the SAD statistics of each single job, is increasing the memory size for each SAD job. This allows to track `npar` = 1000 particles per job, instead of only `npar` = 250. The number of SAD particles lost varies with `npar`, but the total number of lost particles, after counting each SAD particle with its rate, stays constant. This method was used in the background simulation in [74].

Parallelization of the SAD simulation can be realized by a starting several SAD jobs with different random seeds. In each run, the rate is used to rescale the number of SAD particles to the number of real particles per run. The output of several SAD runs with different random seeds can be merged by adjusting the rates: Let N_s be the number of lost SAD particles in the run with random seed s and $R_{s,i}$ the corresponding rates of each rated particle i in the run s. For a list of different random seeds S with $s \in S$, the outputs are merged by rescaling the rates:

$$R'_{s,i} = R_{s,i} \cdot \frac{N_s}{\sum\limits_{s \in S} N_s} \tag{B.2}$$

This has the effect that the total number of real particles (SAD particles multiplied by the rates) stays approximately constant.

After the generation of SAD particles, with loss rates at various positions along the beamline, the particles in the vicinity of the detector are selected as input for the detector simulation with Geant4. In the background campaigns, event samples with a real time equivalent were selected at the generator level, before the detector simulation. A new approach to address the issue of low statistics at this level, is to carry out the Geant4 simulation with rated events; all rated events are simulated N times with Geant4, with a fixed N, while rescaling the rates by $\frac{1}{N}$. This has two advantages: 1.) all generated SAD events are used in the simulation, not only the ones with high rates. 2.) The scattering in Geant4 is calculated several times with different random seeds resulting in an average response of the detector structure to each generated lost particle.

To get realistic distributions of the observables, the rates are applied directly when filling the plots. This results in a higher statistics and a smoother distribution of the background topology. Most particles seen by the trigger are actually secondary particles originating from a scattering process with the detector material. Because the multiple scattering is different in the N Geant4 runs, this gives a more realistic picture of the average effect of each rated lost particle. With this approach, the computational time cane be used efficiently: the runtime for the SAD generator level is

background	process	2D trigger rate
TwoPhoton	$e^+e^- \underbrace{\to}_{e^+e^-\gamma\gamma} e^+e^-e^+e^-$	6 kHz
Bhabha-S		20 kHz
Bhabha-M	$e^+e^- \to e^+e^-\gamma$	52 kHz
Bhabha-L		26 kHz
Touschek	intra bunch scatt.	2 kHz
Coulomb	$e^\pm N \to e^\pm N$	15 kHz
Brems	$e^\pm N \to e^\pm N\gamma$	1 kHz

Table B.2.: Simulated background types with their production processes and observed 2D track trigger rates [74].

$\approx 10\times$ larger than the simulation time with Geant4. Therefore, in this study [74] the number of Geant4 runs N is set to $N = 10$ to spent a similar amount of computational cost into both steps.

B.1.3. Background Topology

A rate of simulated background tracks found by the 2D finder was measured in [74]. A found track rate of about 121.5 kHz are expected [74], which exceeds the capabilities of the L1 trigger with the maximum trigger rate of 30 kHz. In [74], the capabilities of the neural network trigger to suppress these background tracks was successfully demonstrated. The rates of each separately simulated background type are listed in Tab. B.2. The main results of [74] are shown in Fig. B.1.

Using the full MC information during the simulation, the background particle types resulting in correct background tracks could be identified in [74]. Obtained from the matched[1] MC information for each found 2D finder track, the particle types resulting in the background tracks are shown in Fig. B.1 (b). Especially a large fraction of proton tracks was observed, which are created as secondary scattering particles.

Fig. B.1 (a) shows the 2D distribution of the background vertices on top of the detector structure [74] in the r, z-plane, where r is the transverse projection of the vertex positions

$$r = \sqrt{x^2 + y^2}. \tag{B.3}$$

Although the initial background particles are either located at the IP (luminosity background) or distributed along the beam wall (machine background), after simulating the material interaction of the background particles with the detector, the

[1] The matching is described in Appendix A.3.

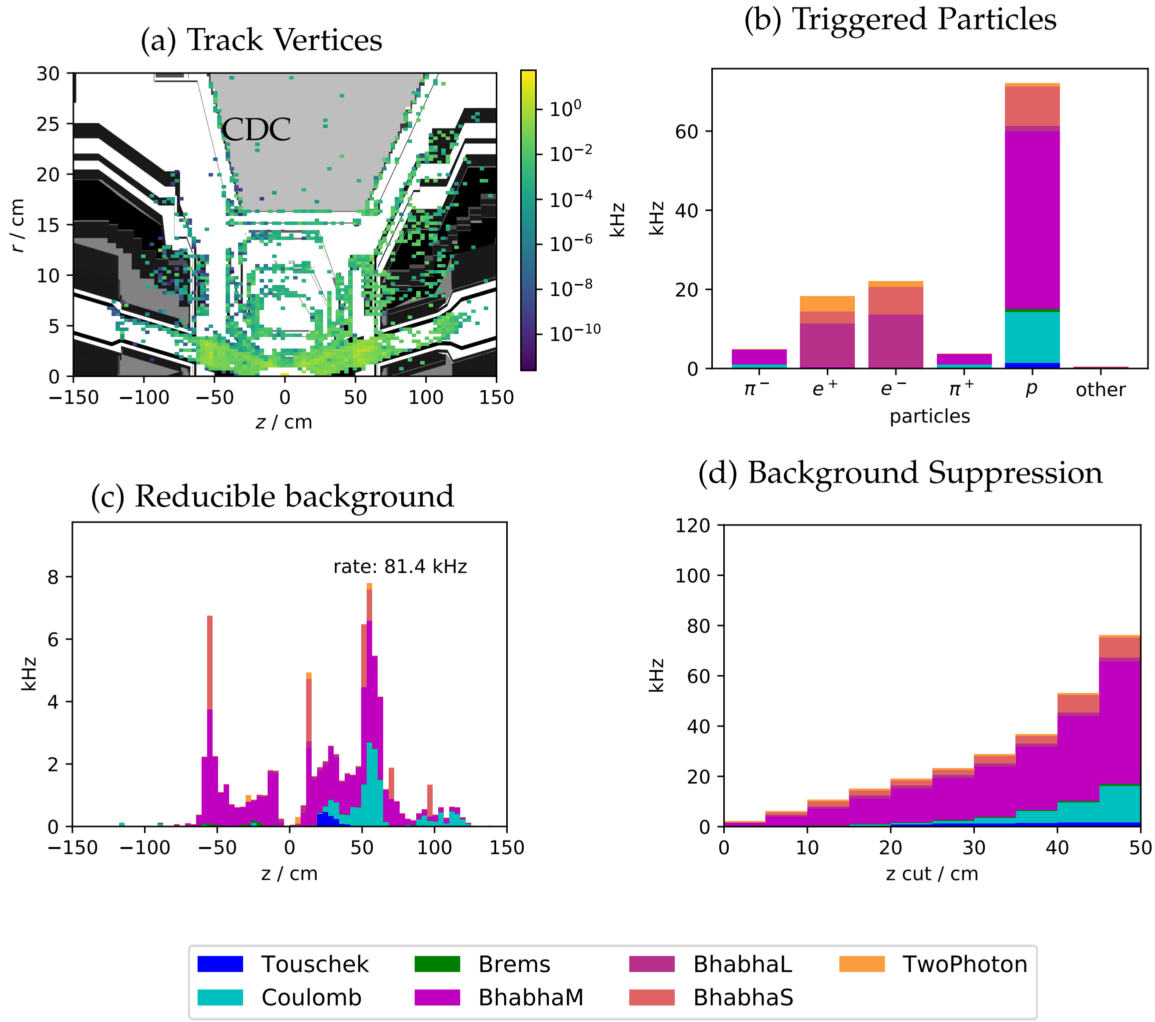


Figure B.1.: Results of the high statistics background simulation [74]. At the scattering positions shown in (a), the particle types shown in (b) create tracks visible by the 2D track finder in the CDC trigger. The distribution of reducible tracks is shown in (c) and the possible suppression of those tracks with the neural network trigger in (d).

vertices of the particles matched to the 2D trigger tracks are distributed on the detector material.

Fig. B.1 (c) shows the matched MC z-vertex distribution of the particles displaced from the IP ($|z| \geq 1\,\mathrm{cm}$) with a matching 2D finder track. It is the one-dimensional histogram, corresponding to the z-axis in the 2D histogram in Fig. B.1 (a).

Via scattering not only the machine background but also luminosity background types produce secondary tracks with vertices displaced from the IP. Due to their large cross section, the largest source of background tracks with displaced track vertices are backscattered particles from Bhabha events. This explains the huge contribution of the BhabhaM background with proton tracks in Fig. B.1 (b) as well as the BhabhaM peaks in Fig. B.1 (c) displaced from the IP.

The background appears to be separated into two different classes, which can be handled by different measures in the L1 trigger [74]. The first class are the e^+e^- particle tracks in Fig. B.1 (b) with a total rate of $\approx 40\,\mathrm{kHz}$. These are tracks from the IP which can only be suppressed using a Bhabha veto, which identifies electrons and positrons by matching the CDC tracking information with the ECL cluster information. The second class with a rate of $\approx 80\,\mathrm{kHz}$ are the secondary scattering particles in Fig. B.1 (c). These background tracks with vertices displaced from the IP can effectively be suppressed by a z-cut on the estimate of the neural network trigger, as shown in Fig. B.1 (d).

Appendix C.

Appendix 3D Finder

C.1. Pseudo Code

C.1.1. Training

Creating the Hit Curve

```
for track in tracks:
  trackBin, trackMod = track.getLocal()
  trackBinAxial = trackBin[:-1]
  for hit in hits(track):
    hitBin, hitMod = hit.getLocal()
      distMod = (hitMod - trackMod) % neighbors
      if hit.isAxial():
        address = hitBin + trackBinAxial + (distMod,)
        A[ address ] += 1
      elif hit.isStereo():
        address = hitBin + trackBin + (distMod,)
        S[ address ] += 1
```

Figure C.1.: Pseudo code for the histogram filling. Tracks and their related hits are transformed from a global to a local frame (ϕ-modulo-sector). Then two separate histograms are filled (A for axial and S for stereo hits).

C.1.2. Track Finding

Track Finding with a Clustering Algorithm

In this appendix the pseudo code of the modified DBSCAN algorithm used during the track finding is listed in the following figures. This algorithm is derived from the DBSCAN [85] algorithm and, therefore, has the same general structure. The difference is a replacement of the determination of the local density via spatial distance measurements. Instead, the weights in a binned space are interpreted as measurements of the local density.

```
def findClusters():
  clusters = []
  candidates = where(houghmap>minweight)
  for cell in candidates:
    if visit(cell):
      newNeighbors = regionQuery(cell)
      if len(newNeighbors) >= minpts:
        newcluster = [cell]
        expandCluster(newNeighbors, newcluster)
        clusters.append(newcluster)
  return clusters
```

Figure C.2.: Pseudo code for the main function to find clusters in the Hough plane. The houghmap and the parameters `minweight`, `minpts` are considered to be global.

Python like pseudo code of the used clustering algorithm is shown in the subsequent figures. The details of the object structure are not shown and the following objects are considered to be global: the configuration parameters `minweight`, `minpts`, `diagonal`, `var_cyclic` and the houghmap. The global houghmap is the 3D array of the track parameter weight space after the summation of all the hits in the event.

The main function for the clustering is listed in Fig. C.2. In a loop over all the elements in the houghmap, the cells where the weight is larger than the minimum weight are selected as candidate cells. If the candidate cell is not yet visited, the function regionQuery() (see Fig. C.3) is called in order to select the neighboring cells of that candidate cell and to test if the candidate cell is also a core cell. If so, the function expandCluster() (see Fig. C.4) is called in order to build a new cluster.

The function regionQuery(), listed in Fig. C.3, selects the cells in the neighborhood where the weight is larger than `minweight`. This is done in a recursive loop over the dimensions of the houghmap, by checking the left and right neighbors along each axis. The functions before() and after() return the neighbor along a given axis of the 3D houghmap. This includes the correct handling of the continuous variable ϕ. If

```
def check(neighbors, cell, dim):
  if cell is not None:
    if visit(cell) and houghmap[cell] > minweight:
      neighbors.append(cell)
    if diagonal and dim > 0:
      blockcheck(neighbors, cell, dim-1)

def blockcheck(neighbors, cell, dim):
  left = before(cell, dim)
  check(neighbors, left, dim)
  right = after(cell, dim)
  check(neighbors, right, dim)
  if dim > 0:
    blockcheck(neighbors, cell, dim-1)

def regionQuery(cell):
  neighbors = []
  blockcheck(neighbors, cell, dimsize-1)
  return neighbors
```

Figure C.3.: Pseudo code for the functions regionQuery(), blockcheck(), and check(). regionQuery() returns the neighborhood of a cell by calling the recursive function blockcheck.

the parameter `diagonal` is not set, regionQuery() tests only the six direct neighbors of a cell in the 3D houghmap. If `diagonal` is set, all the 26 neighbors are tested (see Fig. 5.19).

```
def expandCluster(neighborCells, clusterCells):
  while len(neighborCells) > 0:
    cell = neighborCells.pop()
    if visit(cell):
      if houghmap[cell] >= minweight:
        newNeighbors = regionQuery(cell)
        if len(newNeighbors) >= minpts-1:
          neighborCells.extend(newNeighbors)
        clusterCells.append(cell)
```

Figure C.4.: Pseudo code for the expandCluster function. This function is used to add neighboring cells to an initial cluster seed.

Finder	Data	$p_{\mathrm{T}}^{-1}[10^{-1}\,\mathrm{GeV}^{-1}]$	$\phi[^\circ]$	$\theta[^\circ]$
2D	P1	1.183 ±0.005	0.895 ±0.003	
3D	P1	0.833 ±0.003	0.568 ±0.002	4.105 ±0.011
2D	W1	1.022 ±0.004	0.858 ±0.003	
3D	W1	0.675 ±0.002	0.479 ±0.001	3.205 ±0.008
2D	P0	0.773 ±0.003	0.549 ±0.002	
3D	P0	0.702 ±0.002	0.500 ±0.001	3.247 ±0.008
2D	W0	0.722 ±0.002	0.541 ±0.001	
3D	W0	0.600 ±0.002	0.424 ±0.001	2.750 ±0.007

Table C.1.: Resolutions values (RMS90) of the trained 3D finders and of the 2D finder in test data sets with the different track parameter ranges (see Tab. 5.8).

The function expandCluster(), shown in Fig. C.4, constructs a cluster starting with a candidate as seed. It adds cells to the cluster as long as neighboring cells are found fulfilling the `minweight` criterium (candidate cells). The neighborhood search is expanded to those cells also fulfilling the `minpts` criterium (core cells).

C.2. Resolution Plots at the Separate Boundary Regions

Resolutions at the Separate Low-p_{T} and Shallow-θ Boundaries

Additional plots for Sec. 5.3.4. The table with the average resolution values in the test data sets at the separate boundary regions P and W are shown in Tab. C.1. The corresponding resolution plots for the range P are shown in Fig. C.5 and the resolution plots in the range W are shown in Fig. C.6.

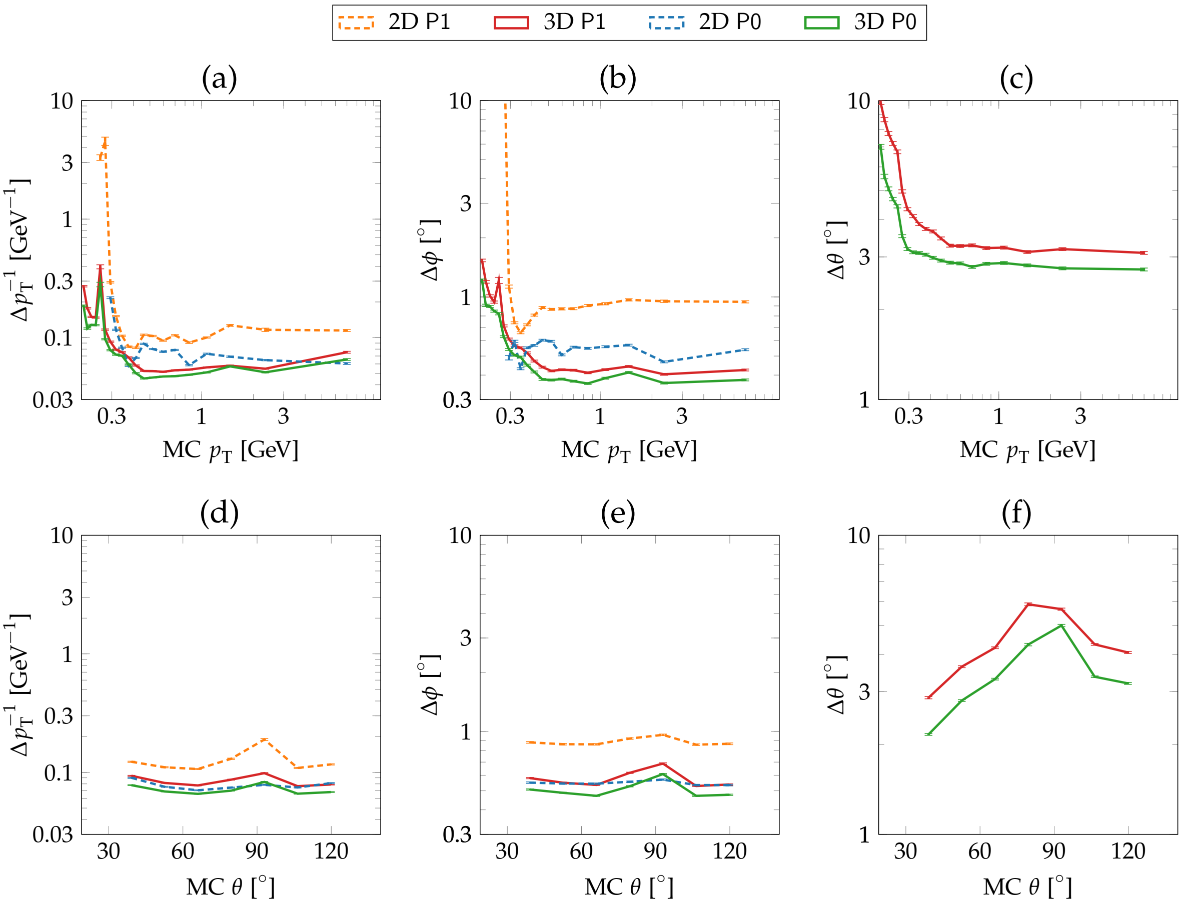


Figure C.5.: Resolution of the finders on the test data sets P0 and P1 (see Tab. 5.8). In the two lowest-p_T bins, the 2D finder often estimates the curl-back track instead of the outwards going track. The data points for the 2D finder ϕ-resolution in these low-p_T bins, with values of $\Delta\phi \approx 30°$, are beyond the scale of the plot (b).

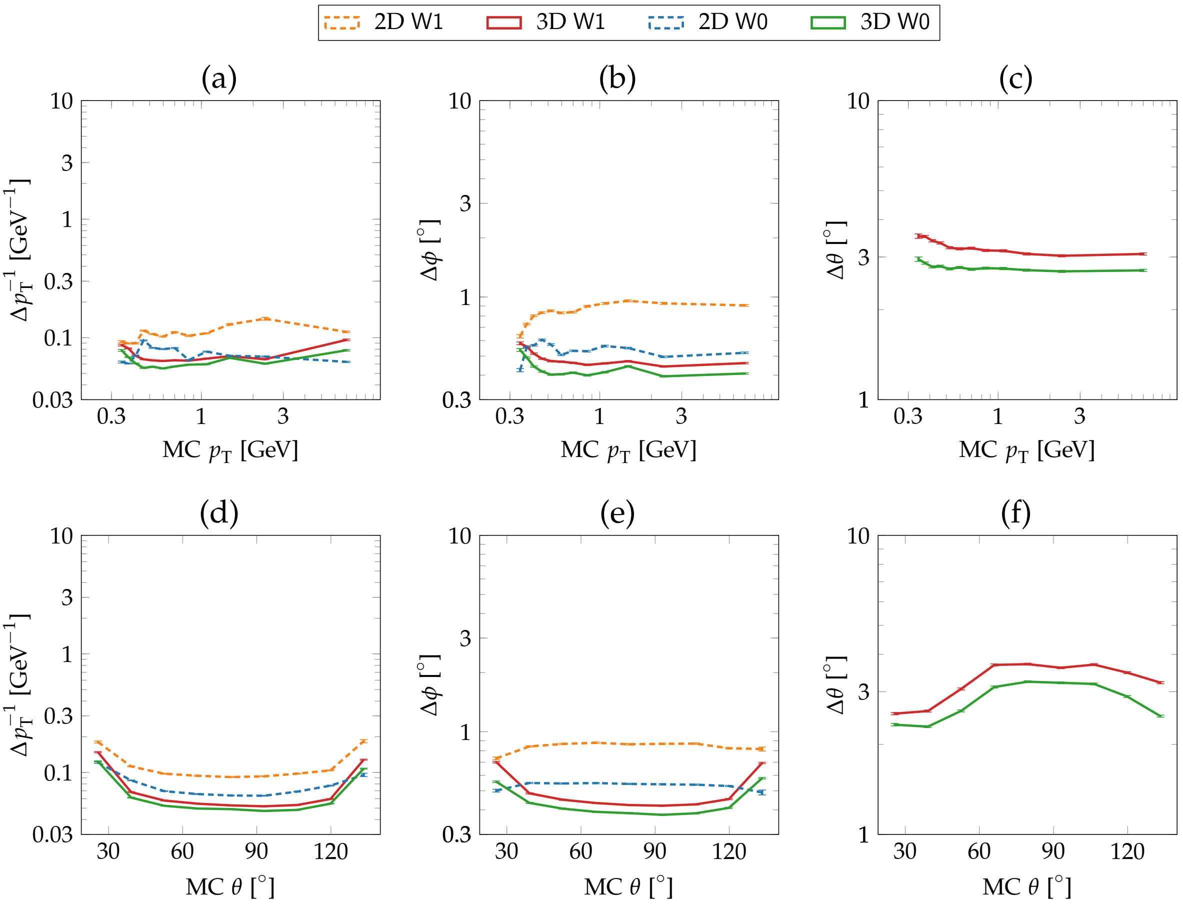


Figure C.6.: Resolution of the finders tested on the data sets W0 and W1 (see Tab. 5.8).

Appendix D.

Appendix Neural Network

D.1. Studies of the Neural Network Training

D.1.1. Sample Size of the Training Data

The weights in the neural network are internal degrees of freedom, which are optimized by the training algorithm based on the training data set. With a larger training data size, the average fluctuations in the training set can be reduced. This in turn reduces the risk of overfitting the network to the random fluctuations in the training data. A network is said to overfit if its weights are optimal only for the used training data set, but if it loses the ability to generalize to different test sets. A different countermeasure for overfitting is the already presented method of cross validation, where an independent test data set is used to stop the training in the epoch with the optimal generalization capabilities. While cross validation can only reduce the problem of overfitting, increasing the sample size can improve the accuracy as the function to be learned has a better representation in the training data. Thus, the size of the training data set is expected to have an effect on the neural network performance. In the following, a study with different sizes of the training data set is presented.

NNpar	nTrain	nValid	nTest
NNA	10	5000	5000
NNB	50	50000	50000
NNC	150	50000	50000
NND	500	50000	50000
NNE	1000	50000	50000

Table D.1.: Definition of a set of neural network training parameters (NNpar) with different training data sizes. The number of events in the training data set is given by the number of weights (2432) in the network multiplied with the factor given in the column nTrain. The columns nValid and nTest are directly the number of events used for the training.

All the neural networks in this experiment have the same architecture and training parameters with the only exception being that different sizes of the training data set are used. These neural networks with their different training data sizes are identified by the labels NNpar, which are listed in Tab. D.1. The basic network presented in the previous studies corresponds to the network with the largest training data size: NNE. The networks NNB, NNC and NND only differ in the size of the training data set, while the same size of the validation and test set is used. In the smallest network NNA, with the factor nTrain of only 10, the number of training samples is only 24320 tracks. This means, for each internal degree of freedom of the network, the training data set contains 10 samples. To avoid validation and test samples larger than the training sample, the values of nValid and nTest are reduced for the smallest network NNA. During the training, the main difference between the networks is their required training time. While the NNA network can be trained within an hour, it takes about four days to train one of the NNE networks (using massively parallel processing on a computing cluster).

$\mathrm{Bkg}_{\mathrm{test}}$	$\Delta\theta_{\mathrm{MC}}$	Δz_{MC}	NNpar	$\epsilon_{\mathrm{NN}}^{\mathrm{MC}}$	$\epsilon_{\mathrm{Finder}}^{\mathrm{MC}}$	$\epsilon_{\mathrm{Trg}}^{\mathrm{MC}}$
	[°]	[cm]		[%]	[%]	[%]
bkg0	1.6	1.4	NND	99.5	55.2	55.0
bkg0	1.6	1.4	NNC	99.5	55.2	55.0
bkg0	1.6	1.4	NNA	99.5	55.2	55.0
bkg0	1.6	1.5	NNE	99.5	55.2	55.0
bkg0	1.6	1.6	NNB	99.5	55.2	55.0
15p2	1.9	1.9	NNE	99.5	55.5	55.2
15p2	1.9	2.0	NND	99.5	55.5	55.2
15p2	1.9	2.0	NNC	99.5	55.5	55.2
15p2	1.9	2.2	NNB	99.5	55.5	55.2
15p2	2.0	2.2	NNA	99.5	55.5	55.2
12p3	2.7	2.9	NNE	99.4	56.4	56.1
12p3	2.7	3.0	NNC	99.4	56.4	56.1
12p3	2.7	3.0	NND	99.4	56.4	56.1
12p3	2.8	3.1	NNB	99.4	56.4	56.1
12p3	2.9	3.3	NNA	99.4	56.4	56.1
15p3	4.6	5.8	NNE	98.1	59.3	58.2
15p3	4.7	5.8	NND	98.1	59.3	58.2
15p3	4.7	5.9	NNC	98.1	59.3	58.2
15p3	4.7	6.0	NNB	98.1	59.3	58.2
15p3	5.1	6.6	NNA	98.1	59.3	58.2

Table D.2.: Neural networks with different sample sizes of the training data are compared. The results are sorted by the z-resolution Δz_{MC}.

Table D.2 shows a summary of the neural network performance with the different sizes of the training data sample. All the results are sorted by the resolution Δz, which implicitly induces the results to be sorted by the background types: the best resolution is achieved without background ($\Delta z \approx [1.4 - 1.6]\,\mathrm{cm}$), the second best with the lightweight Phase 2 background 15p2 ($\Delta z \approx [1.9 - 2.2]\,\mathrm{cm}$). The Phase 3 backgrounds result in the worst resolution, with ($\Delta z \approx [2.9 - 3.3]\,\mathrm{cm}$) with 12p3 and ($\Delta z \approx [5.8 - 6.6]\,\mathrm{cm}$). In the neural network efficiency ϵ_{NN} no difference can be observed within each background type, which is plausible as it could only change due to different ID-ranges. Since the cutoff value and the sizes for the ID-range histograms are the same for all networks, the only possibility for an efficiency change with the same background type are random fluctuations in the ID-range histograms. This suggests that the resolution of the neural network in particular can be improved by a larger training data sample.

Within all non-zero background types, the network with the smallest training data set (NNA) has the worst resolution and the network with the largest training sample (NNE) has the best resolution. Furthermore, the networks with background appear to be sorted according to their training data size. Only in the case without background, networks with small training seem to achieve a slightly better resolution. Due to the smallness of the improvement in the case without background, this is likely a random fluctuation, which could be compensated by a retraining of the bigger networks. This also means that increasing the training data sample is especially helpful for training runs with background and indicates that the neural network can learn the special conditions of the background samples. For the case without background a smaller training data sample is sufficient, as only the reconstruction of the ideal helical track has to be learned. Anyway, the case without background is not realistic and optimizations of the neural network to this idealized case are not of importance.

Figure D.1 shows the resolution of the network NNA (smallest sample) compared to the network NNE (largest sample). For clarity, the intermediate training data sizes are not shown in the plot. In agreement with Tab. D.2, a systematic improvement with the larger training data set can be observed with all background types. Furthermore, with larger background levels the improvement becomes larger. This systematic improvement does not appear to depend on the track parameter ranges in Fig. D.1.

This leads to the conclusion that large training data sets are an efficient measure to improve the resolution of the neural network. Since the required time for the training is intense, it makes sense to carry out other optimization studies - not related to the training data size - at first with smaller training data sets. Once other optimized parameters are identified and selected, it is worthy to carry out a retraining with a substantially larger training data set.

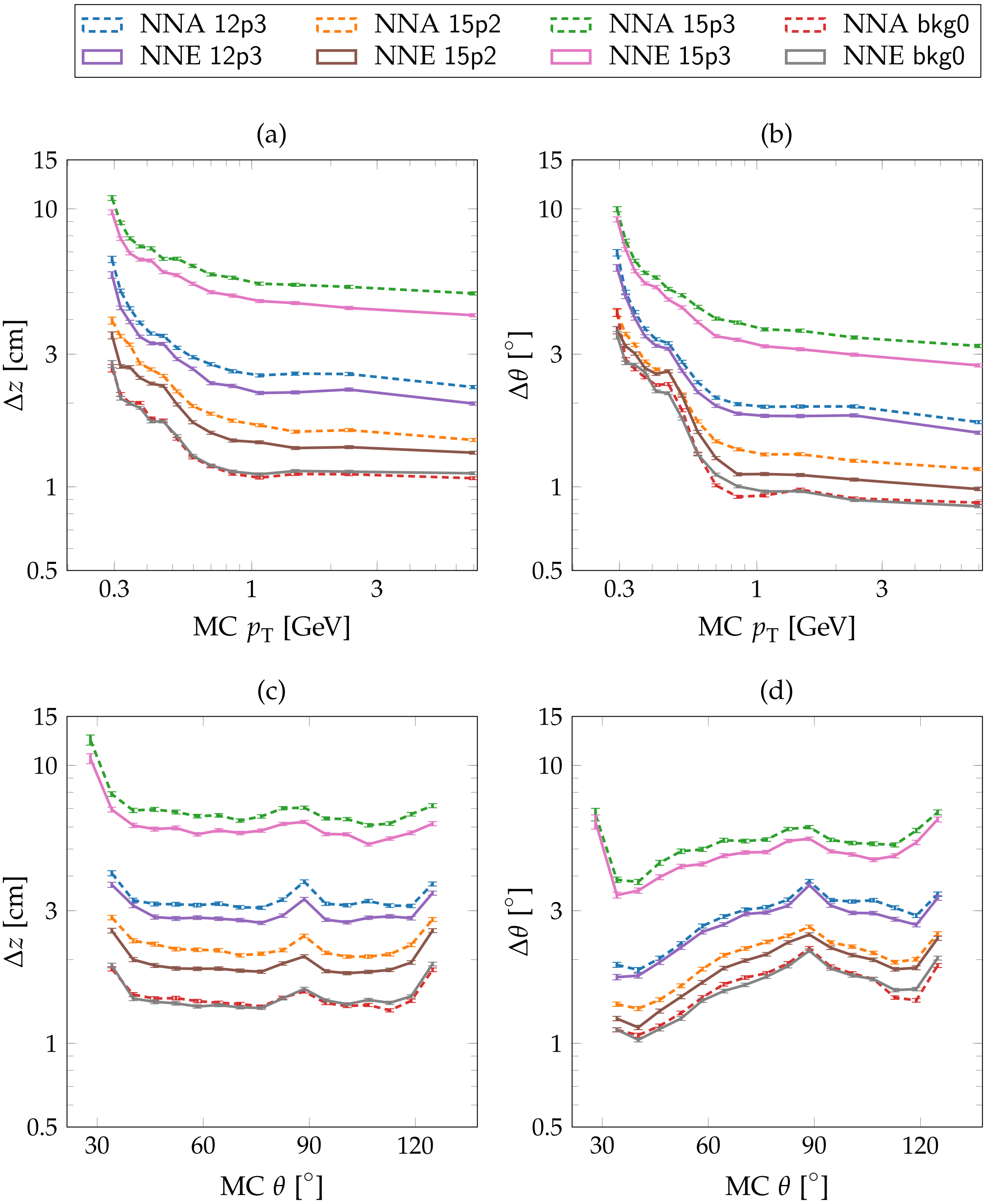


Figure D.1.: Resolutions of the neural network with two different sizes of the training data set. The results are shown for the four background types as functions of the track parameters p_T and θ.

D.1.2. Training Target with Reconstruction vs MC

In this study the training with the MC track parameter values as training targets is compared to a training with the reconstructed track parameter values. Since the 2D finder tracks are required to calculate the neural network input, during the training the target values are either selected as primary MC particles with a matching 2D finder track or as reconstructed track (Reco) with a matching to a 2D finder track. In the setup used for this experiment, a matching of MC particles with Reco tracks is not required during the training. Therefore, by using a training data set which includes background tracks, the Reco training data set contains background tracks as training targets while the MC training data set does not. Hence, it can be expected that the training on Reco better generalizes to real data in the detector. On the other hand, it may have a worse performance on idealized simulated MC tracks.

(a)

Bkg$_{\text{test}}$	$\Delta\theta_{\text{MC}}$	Δz_{MC}	`Target`	$\epsilon_{\text{NN}}^{\text{MC}}$	$\epsilon_{\text{Finder}}^{\text{MC}}$	$\epsilon_{\text{Trg}}^{\text{MC}}$
	[°]	[cm]		[%]	[%]	[%]
bkg0	1.6	1.5	MC	99.5	55.2	55.0
bkg0	1.5	1.5	Reco	99.5	55.2	55.0
15p2	1.9	1.9	MC	99.5	55.5	55.2
15p2	1.9	2.1	Reco	99.5	55.5	55.2
12p3	2.7	2.9	MC	99.4	56.4	56.1
12p3	2.7	3.0	Reco	99.4	56.4	56.1
15p3	4.6	5.8	MC	98.1	59.3	58.2
15p3	4.8	5.9	Reco	98.0	59.3	58.1

(b)

Bkg$_{\text{test}}$	$\Delta\theta^{\text{Reco}}$	Δz^{Reco}	`Target`	$\epsilon_{\text{NN}}^{\text{Reco}}$	$\epsilon_{\text{Finder}}^{\text{Reco}}$	$\epsilon_{\text{Trg}}^{\text{Reco}}$
	[°]	[cm]		[%]	[%]	[%]
bkg0	1.6	1.4	MC	99.3	48.8	48.4
bkg0	1.5	1.5	Reco	99.3	48.8	48.4
15p2	1.9	1.9	MC	99.3	48.7	48.4
15p2	1.9	2.2	Reco	99.3	48.7	48.3
12p3	2.8	3.0	MC	99.1	35.5	35.2
12p3	2.8	3.1	Reco	99.1	35.5	35.2
15p3	5.2	6.4	MC	97.3	42.2	41.0
15p3	5.4	6.5	Reco	97.3	42.2	41.0

Table D.3.: Training of the neural networks with Reco compared MC tracks as `Target` using a test data with ranges ZIP (see Tab. 6.6). In (a) MC values are used in the test; in (b) Reco values. The results are sorted by the z-resolution Δz_{MC} in (a) and Δz_{Reco} in (b).

Since the networks are trained with two different training targets in this study, the networks are also tested twice each; once with Reco values and once with MC values for the calculation of the efficiencies and the resolutions. Table D.3 shows the summarized results of the (Reco, MC)-comparison with a test data set with the ranges ZIP (see Tab. 6.6). The results with the MC test values are shown in Tab. D.3 (a) and with Reco test values in Tab. D.3 (b). In both tables, the additional column `Target` indicates the type of target tracks which were used in the training. As explained in Sec. 6.4.4, the reference tracks for the tests with MC values (in Tab. D.3 (a)) are selected as primary MC particles with matching reconstructed tracks. In contrast, in Tab. D.3 (b) with the Reco test values, the reference tracks are only obtained from the reconstruction, without requiring a matching to the MC tracks. Hence, pure background tracks enter the results shown in Tab. D.3 (b).

The entries in the table are sorted by Δz which results again in the implicit ordering by the background types. It can be observed that for all background types the training with MC target values results in an equal or slightly improved resolution in Δz. With small background levels, the resolutions with MC test values in Tab. D.3 (a) are very similar to the resolutions with Reco test values in Tab. D.3 (b) ($\Delta z_{\mathrm{MC}} \approx \Delta z_{\mathrm{Reco}}$ and $\Delta\theta_{\mathrm{MC}} \approx \Delta\theta_{\mathrm{Reco}}$). However, with higher background levels, the differences between $(\Delta z_{\mathrm{MC}}, \Delta\theta_{\mathrm{MC}})$ and $(\Delta z_{\mathrm{Reco}}, \Delta\theta_{\mathrm{Reco}})$ become larger. This can be explained by the accuracy of the reconstruction, in combination with the clean muon tracks used in this experiment. By using $\mu^{\pm}$ as simulated particles with vertices at the IP, the multiple scattering is minimal such that the tracks are usually not deflected (i.e. the MC values are correct). In contrast, the reconstruction has to cope with the background, which decreases the resolution of the reconstructed values. Hence, by comparing the network estimates to Reco values, the estimation error of the reconstruction adds up to the estimation error of the neural network. This is expected to change with different particle types, like $\pi^{\pm}$ or $\tau^{\pm}$ tracks, where multiple scattering in the detector is more likely such that the comparison with the reconstructed values becomes more accurate in all cases.

Comparing the finder efficiencies shows an increase of $\epsilon_{\mathrm{Finder}}$ with the background level in Tab. D.3 (a) but a decrease in Tab. D.3 (b). This is caused by the different selection of the tracks that should be found. For the MC tracks with matching Reco tracks in Tab. D.3 (a), the additional occupancy in the background increases the chance for the tracks to be found. In contrast, in the evaluation with Reco test values in Tab. D.3 (b) all reconstructed tracks are used as main unit, which includes background tracks that are more difficult to be found by the track finder.

Figure D.2 shows the resolutions of the networks trained with reconstructed tracks and with MC tracks as training target as functions of the MC track parameter values p_{T} and θ. Since the distributions with Reco test values are quite similar, they are not shown here. In all track parameter ranges, the difference between the training with Reco or MC is quite small. Only with larger background levels (especially 15p3), a slightly improved resolution of the training with MC target values becomes visible in the high-p_{T} regions in Fig. D.2 (a) and (b) and the regions with shallow polar angles in Fig. D.2 (c) and (d).

In conclusion, the difference of training with reconstructed values compared to the training with MC values is not very different. This is excellent, as it allows retraining with real data without loss in resolution. The small deviations observed can be explained by the errors of the reconstruction, which add up to the estimation error of the neural network. Even in other simulation experiments, where particle types other than muons are used, the benefits of using reconstructed tracks as training target are expected to be evident.

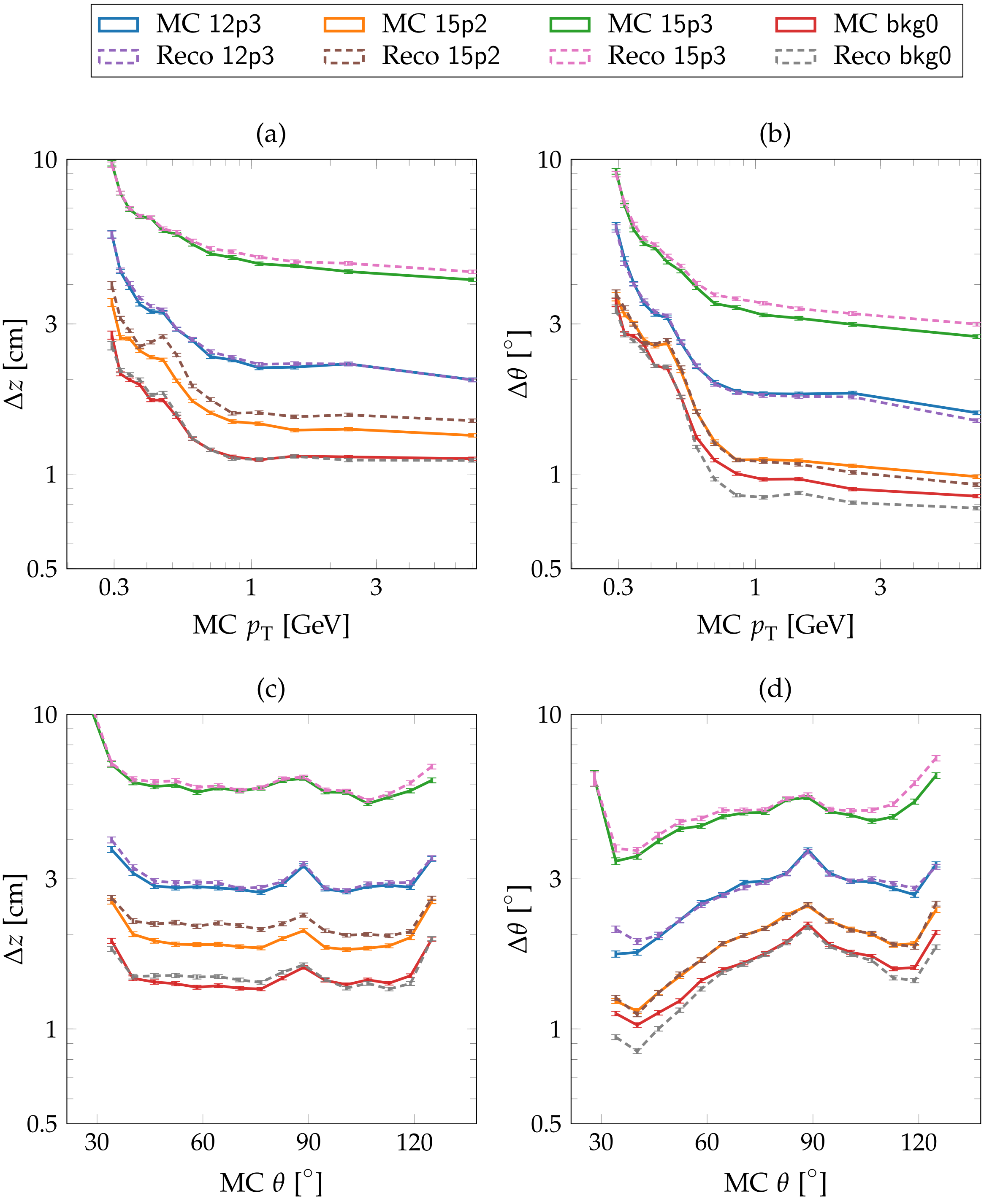


Figure D.2.: Neural networks trained on Reco and MC track parameters as training target are tested on IP tracks with the ranges ZIP (see Tab. 6.6). The resolutions are calculated by comparing the neural network output values with the true MC values.

D.1.3. Background in the Training and Testing

In the studies so far, the background type mixed into the test data set was identical to the background type mixed into the training data set. However, it is also possible to train a network with one background type and test it on another background type. Since the background in the real experiment will always be different to the background used in the training, it is important to study the generalization of background trained neural networks to different background types in the test data. Therefore, this experiment is carried out to find the optimal training background type for each test background type.

To this end, two background columns are shown in the result summary table: $\mathtt{Bkg}_{\text{test}}$ is the background type in the test data set and $\mathtt{Bkg}_{\text{train}}$ is the background type in the training data set. The test is carried with the single $\mu^{\pm}$ tracks from the IP using the range ZIP (see Tab. 6.6), with four background types, each for training and testing (bkg0, 12p3, 15p2, 15p3).

$\mathtt{Bkg}_{\text{test}}$	$\mathtt{Bkg}_{\text{train}}$	$\Delta\theta_{\text{MC}}$ [°]	Δz_{MC} [cm]	$\epsilon^{\text{MC}}_{\text{NN}}$ [%]	$\epsilon^{\text{MC}}_{\text{Finder}}$ [%]	$\epsilon^{\text{MC}}_{\text{Trg}}$ [%]
bkg0	bkg0	1.6	1.5	99.5	55.2	55.0
bkg0	15p2	1.6	1.6	99.5	55.2	55.0
bkg0	12p3	1.7	1.8	99.5	55.2	55.0
15p2	bkg0	2.0	1.9	99.5	55.5	55.2
15p2	15p2	1.9	1.9	99.5	55.5	55.2
15p2	12p3	2.0	2.1	99.5	55.5	55.2
12p3	12p3	2.7	2.9	99.4	56.4	56.1
bkg0	15p3	2.2	2.9	99.6	55.2	55.0
12p3	15p2	2.8	3.0	99.4	56.4	56.1
15p2	15p3	2.3	3.1	99.5	55.5	55.2
12p3	bkg0	3.4	3.6	99.4	56.4	56.1
12p3	15p3	2.8	3.7	99.4	56.4	56.1
15p3	15p3	4.6	5.8	98.1	59.3	58.2
15p3	12p3	5.5	6.0	98.0	59.3	58.1
15p3	15p2	6.3	6.6	98.0	59.3	58.1
15p3	bkg0	7.5	9.4	98.0	59.3	58.1

Table D.4.: Neural network test results, including tests where the network was trained with a background type different to the background type present in the test data. The results are sorted by Δz_{MC}.

The results are summarized in Tab. D.4. As expected, using the same background for training and testing gives the best resolutions. The Phase 2 background 15p2 is comparable to the situation without background bkg0. This means, using the 15p2 trained network on the test data bkg0 gives almost as good results as using the network trained with bkg0 on the test data bkg0. Vice versa, the network trained without background achieves almost the same resolution on 15p2 as the network trained with 15p2.

Similarly, with the Phase 3 background 15p3, the network trained with the other Phase 3 background type 12p3 achieves almost the same resolution. Interestingly, the situation is different in the test with the background 12p3; the second best resolution can be achieved by the network trained with 15p2, the third best in the training without background and the training with 15p3 gives the worst resolution. The reason for this result is the high background hit occupancy in the 15p3. This results in a robuster network, which can handle a high occupancy, but the average resolution in idealized test samples decreases.

In summary, the best results are in general achieved by using the same background type in the training and in testing. However, in the real experiment the actual background cannot be used for the training. Experimental tests should indicate which background type is optimal for the conditions in the experiment. If smaller background levels are sufficient, they should be preferred as the average resolution improves. Once a sufficient amount of background tracks is recorded by the Belle II experiment, these can be used instead of the suboptimal simulated background model used here. Studies on the optimal networks under real data conditions can be found later in Sec. 7.2.

D.1.4. Track Segment Finder Left/Right Information

The TSF uses a lookup table (LUT) of hit patterns within the TSs (called TS-LUT here) to estimate whether a track passed left or right of the priority wire. This Left/Right information (LR) is important for the correct use of the drift time information as a directed distance measure of the track to the priority wire in the neural network input. Without the LR information, only the distance of the track to the wire is known, but not its direction.

In the CDC, the drift charges are accelerated by an electric field from the field wires to the sense wires. Within a drift cell, the electric field is almost radially accelerating the charges towards the field wire (with small corrections in higher order). Due to the viscosity of the gas admixture in the CDC, the charges reach an almost constant drift velocity such that drift time is proportional to the shortest distance of the track to the wire. In general, for a found track and a related wire hit, two solutions are possible for a direction vector which is perpendicular to the track and connects the track with the wire (compare position of the drift circles relative to the 2D track in Fig. 6.3). The length of this direction vector is proportional to the drift time t, which is obtained from the timing measurement at the CDC sense wires by subtracting an event t_0 (see Sec. 3.2.5).

In the TSF, many possible patterns of wire hits within the active TSs hit are stored in the TS-LUTs. Based on these patterns, the TSF tries to decide the LR information, with three possible outcomes: left, right or undecided. In the neural network input,

$\mathrm{Bkg}_{\mathrm{test}}$	$\Delta\theta_{\mathrm{MC}}$	Δz_{MC}	$\epsilon_{\mathrm{NN}}^{\mathrm{MC}}$	$\epsilon_{\mathrm{Finder}}^{\mathrm{MC}}$	$\epsilon_{\mathrm{Trg}}^{\mathrm{MC}}$	TS-LUT
	[°]	[cm]	[%]	[%]	[%]	
bkg0	1.5	1.4	99.6	55.2	55.0	LUTv22
bkg0	1.6	1.5	99.5	55.2	55.0	LUTbkg
15p2	1.9	1.9	99.5	55.5	55.2	LUTbkg
15p2	2.0	2.1	99.5	55.5	55.2	LUTv22
12p3	2.7	2.9	99.4	56.4	56.1	LUTbkg
12p3	3.2	3.6	99.4	56.4	56.1	LUTv22
15p3	4.6	5.8	98.1	59.3	58.2	LUTbkg
15p3	5.7	7.5	98.1	59.3	58.2	LUTv22

Table D.5.: Neural networks using different TS-LUTs. The same TS-LUT was used in the training and in the testing. Again, the output is sorted by Δz_{MC}.

this LR information is used as the sign of the drift time in the input vector. In the case of unknown LR information, this precise distance information is not available for the neural network: in this case the respective drift time inputs are set to zero in the preprocessing (see Sec. 6.2.4), which results in a massive loss in resolution with the neural network. Hence, the neural network benefits from a high efficiency of a correctly determined LR information.

Presently several different TS-LUTs are tested and implemented within the Belle II simulation basf2. The current simulation default is the TS-LUT LUTbkg which is optimized by training the hit pattern TS-LUTs from a data set including simulated background hits [67]. This reduces the rate of TS with undecided LR information, since more patterns including additional background hits enter the TS-LUT. The TS-LUT LUTv22 presently loaded into the hardware is generated without considering background hits and thus has a larger rate of TSs with undecided LR information. These two TS-LUTs are the subject of this study as they are the presently existing candidates to be used in Belle II. The other experimental TS-LUTs implemented in basf2 are not considered here.

Table D.5 shows the summarized results of the neural network with the two different TS-LUTs: LUTbkg and LUTv22. Tested are single track $\mu^{\pm}$ events with vertices at the IP and with four different background types. The results are sorted by Δz_{MC}. Only in the case without background, the resolution with LUTv22 is slightly better. In all cases with background, LUTbkg has an improved resolution, which becomes significant with higher background levels. With the background 15p3, the averaged resolution in Δz can improve from $\Delta z = 7.5\,\mathrm{cm}$ with LUTv22 to $\Delta z = 5.8\,\mathrm{cm}$ with LUTbkg.

The accuracies for the different TS-LUTs are illustrated in Fig. D.3. Fig. D.3 (a) and (b) shows that the improvement achievable with the background optimized TS-LUT is almost constant in all regions of large p_{T}, although at low-p_{T} values, where the resolution is already worse, the improvement seems to decrease. Similarly, Fig. D.3 (c) and (d) show an almost constant improvement in the barrel region and smaller improvements for shallow polar angles θ. It can be concluded that using a background optimized `TS-LUT` is effective for the optimization of the resolution. Considering the continuously updated background models and the availability of recorded background from the Belle II experiment, further optimization studies of the `TS-LUT` should be carried out in the future for the optimal fine tuning of the Belle II trigger.

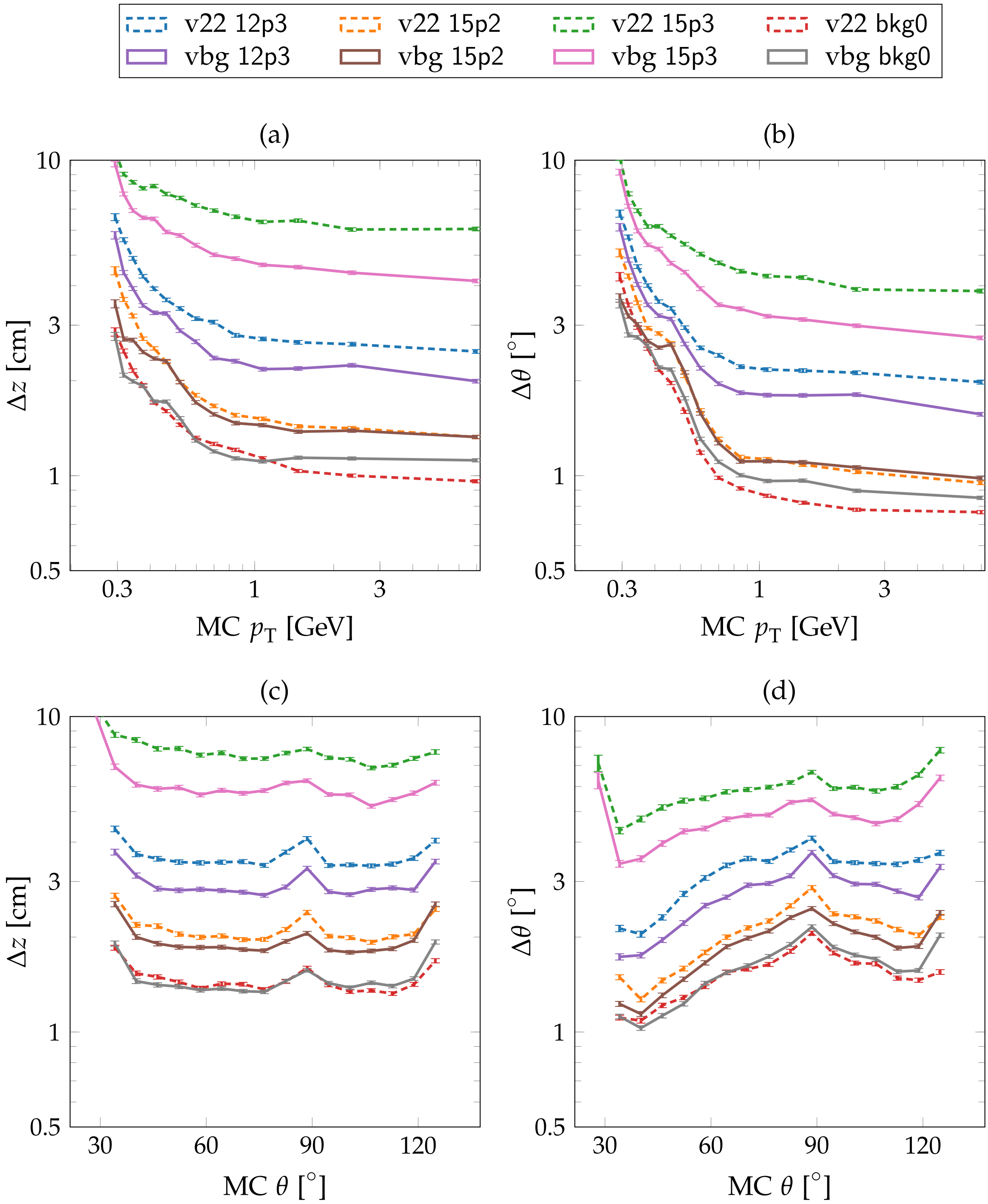


Figure D.3.: Resolution of the neural network with two different TS-LUTs: LUTbkg and LUTv22. Tested are tracks from the IP with four different background types.

Bibliography

[1] A. D. Sakharov, "Violation of $\mathcal{CP}$ invariance, $\mathcal{C}$ asymmetry, and baryon asymmetry of the universe," *JETP Lett.*, vol. 5, no. 1, pp. 24–27, 1967. [Online]. Available: http://cds.cern.ch/record/721864

[2] M. Kobayashi and T. Maskawa, "CP-violation in the renormalizable theory of weak interaction," *Progress of theoretical physics*, vol. 49, no. 2, pp. 652–657, 1973.

[3] R. P. Feynman and M. Gell-Mann, "Theory of the fermi interaction," *Phys. Rev.*, vol. 109, pp. 193–198, Jan 1958. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRev.109.193

[4] S. L. Glashow, "The renormalizability of vector meson interactions," *Nuclear Physics*, vol. 10, pp. 107 – 117, 1959. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0029558259901968

[5] S. Weinberg, "A model of leptons," *Phys. Rev. Lett.*, vol. 19, pp. 1264–1266, Nov 1967. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.19.1264

[6] A. Salam, "A. salam and jc ward, nuovo cimento 11, 568 (1959)," *Nuovo cimento*, vol. 11, p. 568, 1959. [Online]. Available: https://link.springer.com/content/pdf/10.1007%2FBF02726525.pdf

[7] H. Fritzsch, M. Gell-Mann, and H. Leutwyler, "Advantages of the color octet gluon picture," *Physics Letters B*, vol. 47, no. 4, pp. 365 – 368, 1973. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0370269373906254

[8] W. Hollik, "Quantum field theory and the standard model," *arXiv preprint arXiv:1012.3883*, 2010.

[9] Wikipedia. Wikipedia standard model. [Online]. Available: https://en.wikipedia.org/wiki/Standard_Model

[10] S. L. Glashow, J. Iliopoulos, and L. Maiani, "Weak interactions with lepton-hadron symmetry," *Phys. Rev. D*, vol. 2, pp. 1285–1292, Oct 1970. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevD.2.1285

[11] P. W. Higgs, "Broken symmetries and the masses of gauge bosons," *Phys. Rev. Lett.*, vol. 13, pp. 508–509, Oct 1964. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.13.508

[12] N. Cabibbo, "Unitary symmetry and leptonic decays," *Phys. Rev. Lett.*, vol. 10, pp. 531–533, Jun 1963. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.10.531

[13] J. H. Christenson *et al.*, "Evidence for the 2π Decay of the K_2^0 Meson," *Phys. Rev. Lett.*, vol. 13, pp. 138–140, Jul 1964. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.13.138

[14] K. Olive, "Review of particle physics," *Chinese Physics C*, vol. 40, no. 10, p. 100001, oct 2016. [Online]. Available: https://doi.org/10.1088%2F1674-1137%2F40%2F10%2F100001

[15] C. Jarlskog, "Commutator of the quark mass matrices in the standard electroweak model and a measure of maximal $\mathcal{CP}$ nonconservation," *Phys. Rev. Lett.*, vol. 55, pp. 1039–1042, Sep 1985. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.55.1039

[16] Y. Grossman, "Introduction to flavor physics," in *Flavianet School on Flavour Physics Karlsruhe, Germany, September 7-18, 2009*, 2014, pp. 111–144, [,73(2014)].

[17] "CKM Fitter group." [Online]. Available: http://ckmfitter.in2p3.fr/

[18] A. Bevan *et al.*, "The physics of the b factories," *The European Physical Journal C*, vol. 74, no. 11, p. 3026, 2014.

[19] I. I. Bigi and A. I. Sanda, *$\mathcal{CP}$ violation*. Cambridge University Press, 2009.

[20] E. Kou *et al.*, "The Belle II Physics book," *arXiv preprint 1808.10567*, 2018.

[21] T. Abe *et al.*, "Belle II Technical Design Report," *ArXiv e-prints*, Nov. 2010. [Online]. Available: http://arxiv.org/abs/1011.0352

[22] G. Bertone, *Particle dark matter: observations, models and searches*. Cambridge University Press, 2010.

[23] R. Strauss *et al.*, "Energy-dependent light quenching in cawo4 crystals at mk temperatures," *The European Physical Journal C*, vol. 74, no. 7, p. 2957, 2014.

[24] G. Angloher *et al.*, "Results from 730 kg days of the cresst-ii dark matter search," *The European Physical Journal C*, vol. 72, no. 4, p. 1971, 2012.

[25] S. Weinberg, "A new light boson?" *Physical Review Letters*, vol. 40, no. 4, p. 223, 1978.

[26] F. Wilczek, "Problem of strong p and t invariance in the presence of instantons," *Physical Review Letters*, vol. 40, no. 5, p. 279, 1978.

[27] F. Wilczek, "Problem of strong $\mathcal{P}$ and $\mathcal{T}$ invariance in the presence of instantons," *Phys. Rev. Lett.*, vol. 40, pp. 279–282, Jan 1978. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.40.279

[28] P. W. Graham *et al.*, "Experimental searches for the axion and axion-like particles," *Annual Review of Nuclear and Particle Science*, vol. 65, pp. 485–514, 2015.

[29] P. Brun *et al.*, "A new experimental approach to probe qcd axion dark matter in the mass range above 40 ev," *The European Physical Journal C*, vol. 79, no. 3, p. 186, 2019.

[30] "Superkamiokande." [Online]. Available: http://www-sk.icrr.u-tokyo.ac.jp/sk/index-e.html

[31] "Sudbury neutrino observatory." [Online]. Available: https://sno.phy.queensu.ca/

[32] Z. Maki, M. Nakagawa, and S. Sakata, "Remarks on the unified model of elementary particles," *Progress of Theoretical Physics*, vol. 28, no. 5, pp. 870–880, 1962.

[33] G. Aad *et al.*, "Combined measurement of the higgs boson mass in pp collisions at $\sqrt{s} = 7$ and 8 TeV with the atlas and cms experiments," *Phys. Rev. Lett.*, vol. 114, p. 191803, May 2015. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.114.191803

[34] T. R. S. A. of Sciences, "Nobel prize web," 2018. [Online]. Available: https://www.nobelprize.org

[35] S.-K. Choi *et al.*, "Observation of a narrow charmoniumlike state in exclusive $B^{\pm} \rightarrow K^{\pm}\pi^{+}\pi^{-}J/\psi$ decays," *Phys. Rev. Lett.*, vol. 91, p. 262001, Dec 2003. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.91.262001

[36] B. Holdom, "Two $\mathcal{U}(1)$'s and Epsilon Charge Shifts," *Phys. Lett.*, vol. 166B, pp. 196–198, 1986.

[37] M. Tanabashi *et al.*, "Review of particle physics," *Phys. Rev. D*, vol. 98, p. 030001, Aug 2018. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevD.98.030001

[38] "Muon $g-2$ experiment." [Online]. Available: http://muon-g-2.fnal.gov/

[39] A. Abashian *et al.*, "The Belle detector," *Nucl. Instrum. Methods A*, vol. 479, no. 1, pp. 117–232, 2002.

[40] B. Aubert *et al.*, "The BABAR detector," *Nucl. Instrum. Methods A*, vol. 479, no. 1, pp. 1–116, 2002.

[41] Y. Iwasaki *et al.*, "Level 1 Trigger System for the Belle II Experiment," *Nuclear Science, IEEE Transactions on*, vol. 58, no. 4, pp. 1807–1815, Aug. 2011.

[42] Belle II Collaboration. Belle II Website. [Online]. Available: https://www.belle2.org

[43] M. Bona *et al.*, "SuperB: A High-Luminosity Asymmetric e^+e^- Super Flavor Factory. Conceptual Design Report," *arXiv preprint arXiv:0709.0451*, 2007.

[44] P.Raimondi, "Status of the SuperB effort." [Online]. Available: http://www.lnf.infn.it/conference/superb06/talks/raimondi1.ppt

[45] S. Hashimoto *et al.*, "Letter of intent for KEK Super B Factory," High Energy Accelerator Research Organization, Tech. Rep., 2004. [Online]. Available: http://superb.kek.jp/documents/loi/img/LoI_accelerator.pdf

[46] KEK, "SuperKEKB TDR," High Energy Accelerator Research Organization, Tech. Rep., 2014. [Online]. Available: https://kds.kek.jp/indico/event/15914/

[47] Y. Ohnishi *et al.*, "Accelerator design at SuperKEKB," *Progress of Theoretical and Experimental Physics*, vol. 2013, no. 3, 2013. [Online]. Available: http://ptep.oxfordjournals.org/content/2013/3/03A011.abstract

[48] H. Wiedemann, *Particle accelerator physics*. Springer, 2015.

[49] KEK. SuperKEKB Website. [Online]. Available: http://www-superkekb.kek.jp

[50] H. Nakayama *et al.*, "Beam background and MDI design for SuperKEKB/Belle-II," in *Proc. of IPAC 2012, New Orleans, Louisiana, USA*, 2012, pp. 1825–1827, TUPPR007.

[51] H. Nakayama *et al.*, "Small-beta collimation at SuperKEKB to stop beam-gas scattered particles and to avoid transverse mode coupling instability," in *Proc. of IPAC 2012, New Orleans, Louisiana, USA*, 2012, pp. 1104–1106, tUOBC02.

[52] N. Braun *et al.*, "Study of the collision point properties." May 2018, BELLE2-NOTE-PH-2018-006.

[53] K. A. Olive *et al.*, "Review of Particle Physics," *Chin. Phys.*, vol. C38, p. 090001, 2014.

[54] P. Billoir and S. Qian, "Fast vertex fitting with a local parametrization of tracks," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 311, no. 1, pp. 139 – 150, 1992. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0168900292908593

[55] P. Lewis *et al.*, "First measurements of beam backgrounds at superkekb," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 914, pp. 69–144, 2019. [Online]. Available: https://arxiv.org/pdf/1802.01366

[56] N. Taniguchi, "Central drift chamber for Belle-II," *Journal of Instrumentation*, vol. 12, no. 06, p. C06014, 2017. [Online]. Available: http://stacks.iop.org/1748-0221/12/i=06/a=C06014

[57] J. Townsend, "Motion of electrons in gases," *Journal of the Franklin Institute*, vol. 200, no. 5, pp. 563–590, 1925.

[58] O. Alonso *et al.*, "Depfet active pixel detectors for a future linear collider," *IEEE Trans. Nucl. Sci*, vol. 60, pp. 1457–1465, 2012.

[59] F. Müller, "Characterization and Optimization of the Prototype DEPFET Modules for the Belle II Pixel Vertex Detector," Ph.D. dissertation, Faculty of Physics, Ludwig-Maximilians-Univ., 2017.

[60] A. Moll, "Comprehensive study of the background for the pixel vertex detector at Belle II," Ph.D. dissertation, Faculty of Physics, Ludwig-Maximilians-Univ., 2015.

[61] J. . Shiu, "The level 1 trigger system for Belle II CDC," in *2016 IEEE-NPSS Real Time Conference (RT)*, June 2016, pp. 1–3.

[62] S. Lee *et al.*, "Belle-II High Level Trigger at SuperKEKB," *Journal of Physics: Conference Series*, vol. 396, no. 1, p. 012029, 2012. [Online]. Available: http://stacks.iop.org/1742-6596/396/i=1/a=012029

[63] R. Itoh *et al.*, "Data flow and high level trigger of Belle II DAQ system," *IEEE Transactions on Nuclear Science*, vol. 60, no. 5, pp. 3720–3724, Oct 2013.

[64] B. Spruck *et al.*, "The Belle II pixel detector data acquisition and reduction system," *IEEE Transactions on Nuclear Science*, vol. 60, no. 5, pp. 3709–3713, 2013.

[65] C. Li, "The data acquisition and trigger system of the Belle II experiment," *PoS*, vol. 234, p. 258, 2015. [Online]. Available: https://pos.sissa.it/234/258/pdf

[66] S. Skambraks, S. Neuhaus, and C. Kiesling, "Background Suppression with the Belle II Neural Network Trigger," 2017. [Online]. Available: https://indico.cern.ch/event/567550/contributions/2629586/attachments/1510648/2355661/acat17NeuroTrigger.pdf

[67] S. Pohl, "Track reconstruction at the first level trigger of the Belle II experiment," Ph.D. dissertation, Faculty of Physics, Ludwig-Maximilians-Univ., 2017.

[68] E. Won, J. B. Kim, and B. R. Ko, "Three-dimensional fast tracker for the central drift chamber based level-1 trigger system in the Belle II experiment," *Journal of the Korean Physical Society*, vol. 72, no. 1, pp. 33–37, Jan 2018. [Online]. Available: https://doi.org/10.3938/jkps.72.33

[69] T. Kuhr *et al.*, "The Belle II core software," *Computing and Software for Big Science*, vol. 3, no. 1, p. 1, 2019.

[70] A. Moll, "The software framework of the Belle II experiment," in *Journal of Physics: Conference Series*, vol. 331, no. 3. IOP Publishing, 2011, p. 032024.

[71] S. Agostinelli *et al.*, "Geant4–a simulation toolkit," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 506, no. 3, pp. 250 – 303, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168900203013688

[72] KEK Collaboration. Strategic Accelerator Design. [Online]. Available: http://acc-physics.kek.jp/SAD

[73] A. Piwinski, "The Touschek effect in strong focusing storage rings," *arXiv preprint physics/9903034*, 1998. [Online]. Available: http://arxiv.org/abs/physics/9903034

[74] S. Skambraks, S. Neuhaus, and C. Kiesling, "Background Suppression with the Belle II Neural Network Trigger," *Journal of Physics: Conference Series*, vol. 1085, no. 4, p. 042026, 2018. [Online]. Available: http://stacks.iop.org/1742-6596/1085/i=4/a=042026

[75] R. Kleiss and H. Burkhardt, "BBBREM Monte Carlo simulation of radiative Bhabha scattering in the very forward direction," *Computer Physics Communications*, vol. 81, no. 3, pp. 372 – 380, 1994. [Online]. Available: http://www.sciencedirect.com/science/article/pii/001046559490085X

[76] S. Jadach, W. Płaczek, and B. Ward, "BHWIDE 1.00: $\mathcal{O}(\alpha)$ YFS exponentiated Monte Carlo for Bhabha scattering at wide angles for LEP1/SLC and LEP2," *Physics Letters B*, vol. 390, no. 1-4, pp. 298–308, 1997.

[77] E. Nedelkovska, "Measurement of the branching fraction of the decay $\mathrm{B}^0 \to \Psi(2\mathrm{S})\pi^0$ and studies of the luminosity-dependent background for the Belle II experiment at the future accelerator SuperKEKB," Ph.D. dissertation, Faculty of Physics, Ludwig-Maximilians-Univ., 2014.

[78] F. A. Berends, P. Daverveldt, and R. Kleiss, "Complete lowest-order calculations for four-lepton final states in electron-positron collisions," *Nuclear Physics B*, vol. 253, pp. 441–463, 1985.

[79] C. Bernardini *et al.*, "Lifetime and Beam Size in a Storage Ring," *Phys. Rev. Lett.*, vol. 10, pp. 407–409, May 1963.

[80] P. V. C. Hough, "Machine Analysis Of Bubble Chamber Pictures," in *HEACC, Proc. 2nd Int. Conf. on High-Energy Accelerators and Instrumentation*, vol. C590914, 1959, pp. 554–558. [Online]. Available: http://inspirehep.net/record/919922/files/HEACC59_598-602.pdf

[81] P. E. Hart, "How the hough transform was invented [dsp history]," *IEEE Signal Processing Magazine*, vol. 26, no. 6, pp. 18–22, November 2009.

[82] M. Hansroul, D. Savard, and H. Jeremie, "Fast circle fit with the conformal mapping method," *Nucl. Instrum. Methods Phys. Res., A*, vol. 270, no. CERN-DD-88-20, pp. 498–501, 1988.

[83] S. Skambraks, S. Neuhaus, and C. Kiesling, "The neuro-z-vertex trigger of the Belle II experiment," in *EPJ Web of Conferences*, vol. 127. EDP Sciences, 2016, p. 00016.

[84] S. Skambraks *et al.*, "A 3D track finder for the Belle II CDC L1 trigger," 2020, accepted for publication by IOP in their Journal Of Physics: Conference Series. [Online]. Available: https://indico.cern.ch/event/708041/papers/3269727/files/9527-ACAT3DFinderProceeding-v3.pdf

[85] M. Ester *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[86] S. Skambraks, "Use of Neural Networks for Triggering in Particle Physics," Diplomarbeit, Inst. for Informatics, Ludwig-Maximilians-Univ., München, 2013. [Online]. Available: https://publications.mppmu.mpg.de/2013/MPP-2013-339/FullText.pdf

[87] Human Brain Project. human brain project. [Online]. Available: https://www.humanbrainproject.eu/en/

[88] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *The Journal of physiology*, vol. 117, no. 4, pp. 500–544, 1952.

[89] R. Beale and T. Jackson, *Neural Computing-an introduction.* CRC Press, 1990.

[90] W. Maass, "Liquid computing," in *Conference on Computability in Europe*. Springer, 2007, pp. 507–516.

[91] R. Gütig and H. Sompolinsky, "The tempotron: a neuron that learns spike timing–based decisions," *Nature neuroscience*, vol. 9, no. 3, p. 420, 2006.

[92] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[93] S. Skambraks, "Use of Neural Networks for Triggering in the Belle II Experiment," M.S. thesis, Faculty of Physics, Ludwig-Maximilians-Univ., München, 2013. [Online]. Available: https://publications.mppmu.mpg.de/2013/MPP-2013-377/FullText.pdf

[94] C. M. Bishop *et al.*, *Neural networks for pattern recognition.* Oxford university press, 1995.

[95] R. Frühwirth, A. Strandlie, and W. Waltenberger, "Helix fitting by an extended riemann fit," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 490, no. 1-2, pp. 366–378, 2002.

[96] S. Neuhaus *et al.*, "A neural network z -vertex trigger for Belle II," *Journal of Physics: Conference Series*, vol. 608, no. 1, p. 012052, 2015. [Online]. Available: http://stacks.iop.org/1742-6596/608/i=1/a=012052

[97] A. Strandlie and R. Fruhwirth, "Track and vertex reconstruction: From classical to adaptive methods," *Rev. Mod. Phys.*, vol. 82, pp. 1419–1458, 2010.

[98] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation: theory algorithms and software.* John Wiley & Sons, 2004.

[99] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *IJCNN, IEEE Int. Joint Conf. Neural Networks*, vol. 1, 1989, pp. 593–605.

[100] S. McCarney, "Optimization of the z-Vertex Neural Network Trigger for the Belle II Experiment," M.S. thesis, Faculty of Physics, Ludwig-Maximilians-Univ., München, 2019.

[101] C. Igel and M. Hüsken, "Improving the Rprop Learning Algorithm," in *Proc. 2nd Int. Symp. Neural Computation*, 2000, pp. 115–121.

[102] C. Igel and M. Hüsken, "Empirical evaluation of the improved Rprop learning algorithms," *Neurocomputing*, vol. 50, pp. 105–123, 2003.

[103] S. Nissen *et al.*, "Implementation of a fast artificial neural network library (fann)," *Report, Department of Computer Science University of Copenhagen (DIKU)*, vol. 31, p. 29, 2003. [Online]. Available: ttp://fann.sf.net

[104] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural networks for perception.* Elsevier, 1992, pp. 65–93.

[105] R. A. Jacobs *et al.*, "Adaptive Mixtures of Local Experts," *Neural Compution*, vol. 3, no. 1, pp. 79–87, 1991.

[106] S. Skambraks *et al.*, "A z -vertex trigger for Belle II," *Nuclear Science, IEEE Transactions on*, vol. 62, no. 4, pp. 1732–1740, Aug 2015.

[107] S. Neuhaus, S. Skambraks, and C. Kiesling, "Track vertex reconstruction with neural networks at the first level trigger of Belle II," in *EPJ Web of Conferences*, vol. 150. EDP Sciences, 2017, p. 00009.

[108] S. Baehr *et al.*, "A neural network on FPGAs for the z-vertex track trigger in Belle II," *Journal of Instrumentation*, vol. 12, no. 03, p. C03065, 2017. [Online]. Available: http://stacks.iop.org/1748-0221/12/i=03/a=C03065

[109] S. Skambraks *et al.*, "A Machine Learning based 3D Track Trigger for Belle II," 2020, connecting the Dots. [Online]. Available: https://indico.cern.ch/event/831165/contributions/3717108/attachments/2024886/3386928/CTD2020_NeuroTrigger_Skambraks.pdf

[110] F. Jegerlehner, *The anomalous magnetic moment of the muon*. Springer, 2017, vol. 274.

[111] G. Rodrigo *et al.*, "Radiative return at nlo and the measurement of the hadronic cross-section in electron–positron annihilation," *The European Physical Journal C-Particles and Fields*, vol. 24, no. 1, pp. 71–82, 2002.

[112] E. Cho and M. J. Cho, "Variance of sample variance," *Section on Survey Research Methods–JSM*, vol. 2, pp. 1291–1293, 2008.

[113] Geant4 Collaboration, "Geant4." [Online]. Available: https://geant4.web.cern.ch

Acknowledgements

Hereby I would like to express my gratitude for all the support I received during the writing of this thesis. Foremost, I want to thank Prof. Christian Kiesling for his guidance throughout my thesis and the thorough discussions of the newly developed algorithms and their application to the trigger system. Without him the whole neural network trigger project would not have been possible and I could not have whished for of a better teacher and mentor. I want to thank my Belle II working group at the Max-Planck-Institute for all the interesting discussions on the physics at Belle II and on the technical details of the trigger system. Many thanks to the head of our group Hans-Günther Moser for letting me be a part of this experience. I like to thank all my colleagues for the inspiring and nice time at the Max-Planck-Institute.

I want to thank Prof. Jochen Schieck, director of the institute for high energy physics of the OAW, for introducing me to the Belle II group and the helpful support during the seeding phase of the neural network trigger project. Many thanks to Prof. Alois Knoll, chair of robotics, artificial intelligence and real-time systems at the TUM department of informatics, Prof. Stephan Paul, head of the group on hadronic structure and fundamental symmetries at the TUM department of physics and from the Excellence Cluster Universe, and Prof. Jürgen Becker from the Institut für Technik der Informationsverarbeitung (ITIV) at the Karlsruher Institute of Technology (KIT), for the supervision, initiation and continuation of the neural network trigger project.

Many thanks to Steffen Bähr from the KIT who realized the hardware implementation of the presently running neural network trigger algorithm. Furthermore, I like to thank the trigger group at Belle II for the fruitful collaboration and the inspirations during the development of the algorithms. I want to thank Sara Pohl for the thorough discussions during the early development phases of the neural network trigger and for the inspirations on the typesetting. I like to thank Sara McCarney for the supportive advice on the Englisch spelling and grammer. Last but not least, many thanks to my family, my friends, and especially my life partner Lina-Marie Bannenberg for the emotional support.