Particle Identification at Belle II Using Neural Networks

Xavier Simó for the Belle II collaboration
(xavi.simo@tum.de)

DPG Fühjahrstagung T 10.4, March 20, 2023

MAX PLANCK INSTITUTE FOR PHYSICS

Belle II

Bundesministerium für Bildung und Forschung

Introduction

Belle II

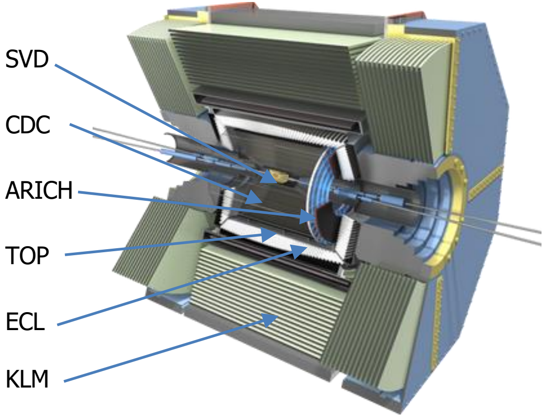
- Located at SuperKEKB
- Asymmetric e^+e^- collider
- At KEK, Tsukuba, Japan
- High-precision tests of the standard model

Objective of particle identification (PID)

- Identify particle species of charged tracks
- Distinguish charged-particle species: $e, \ \mu, \ \pi, \ K, \ p, \ d$

Particle Identification

- 6 subdetectors used for particle identification
- Each provides a likelihood for a given particle species :
  - $\mathcal{L}_h^o$
- 6 subdetectors (o) $*$ 6 particle species (h)
  - In total 36 likelihoods



Pure Likelihood-Based Approach

Current approach at Belle II: pure likelihood-based approach

- Combine detector likelihoods → likelihood for a given particle species:
  - $\mathcal{L}_h = \mathcal{L}_h^{\mathrm{SVD}} \cdot \mathcal{L}_h^{\mathrm{CDC}} \cdot \mathcal{L}_h^{\mathrm{TOP}} \cdot \mathcal{L}_h^{\mathrm{ARICH}} \cdot \mathcal{L}_h^{\mathrm{ECL}} \cdot \mathcal{L}_h^{\mathrm{KLM}}$
- Our goal is to do $K - \pi$ Separation → Binary classification
- $P(K) \equiv \dfrac{\mathcal{L}_K}{\mathcal{L}_K + \mathcal{L}_\pi}$
- $P(\pi) \equiv \dfrac{\mathcal{L}_\pi}{\mathcal{L}_K + \mathcal{L}_\pi}$

Use Neural Network to improve Performance

Limitations of pure likelihood-based approach

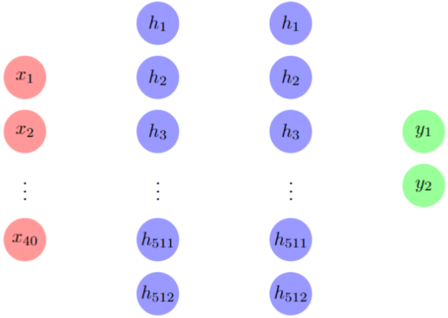
- Computation of likelihoods requires modeling, which requires approximations
- Does not account for correlation
- Challenging to adjust to real world

Goal

- Improve the pure likelihood-based approach by using a neural network

Neural Network for 2 Hypotheses

- 40 inputs:
  - Loglikelihood for the 6 particles hypotheses and 6 subdetectors
  - Magnitude and direction of track momentum
  - Charge
- 2 hidden dense layers of 512 nodes
- 2 outputs → Probabilities for kaon and pion hypotheses



$y_1 = P_{NN}(K)$

$y_2 = P_{NN}(\pi)$

Training Samples

Simulated data: Particle-gun MC (pgMC)

- Generate particles with isotropic momentum distribution
- Detector response simulated using Belle II simulation framework

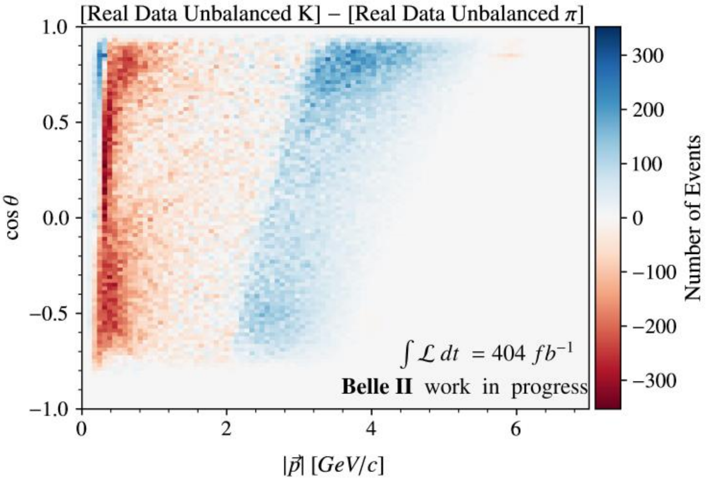
Real data sample

- Sample with known true species without using PID detectors
- Physics process that produces only certain particle species
- Process to obtain clean sample of K and π
  - $D^{*\pm} \to \overline{D}\,\pi^{\pm}$
    - $\overline{D} \to K^{\mp}\pi^{\pm}$

Balancing of Training sample

- Specific sample:
  - $D^{*\pm} \rightarrow \overline{D}\,\pi^{\pm}$
    - $\overline{D} \rightarrow K^{\mp}\pi^{\pm}$

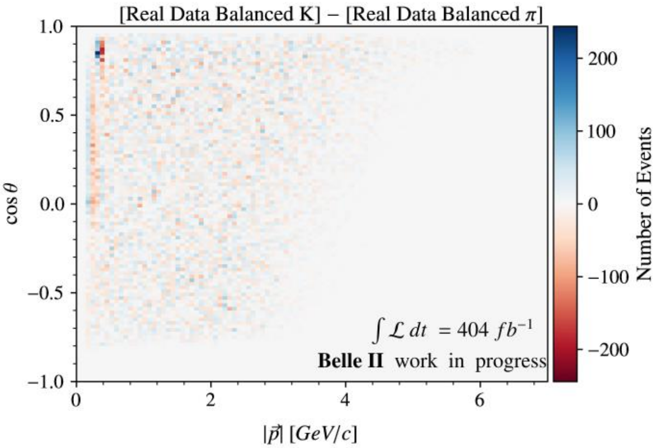
- More K events at high momenta
- More π events at low momenta
- Bias → Neural network should not learn sample-specific features



Balancing

Minimize this bias

- Divide the sample in $\cos(\theta)$ and $|\vec{p}|$ bins and drop tracks according to the imbalance
    - Balanced sample → Used for training



Testing the Particle-Identification Performance

- Testing sample: Real Data
- K efficiency: probability that $K \to K$
- π mis-identification rate: probability that $\pi \to K$

Models overview

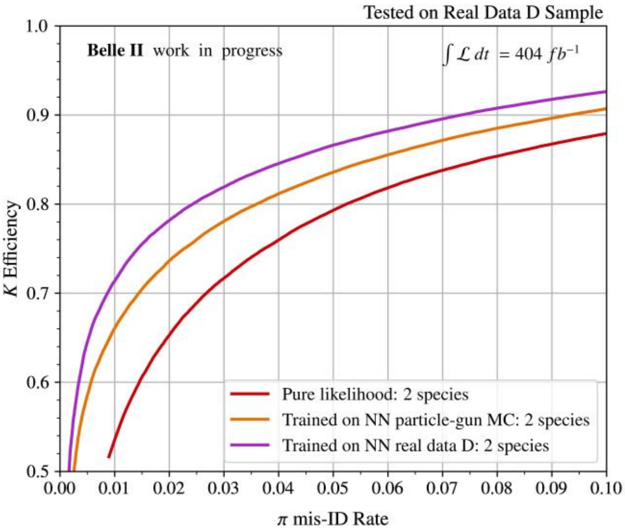
Models

- **Pure likelihood-based**
- **NN trained on real data**
- **NN trained on particle-gun MC for 2 hypotheses**

- Predict probability for pion and kaon hypothesis

Performance

- **Pure likelihood-based**: performs the worst
- **NN trained on particle-gun MC for 2 hypotheses**: has a better performance than pure likelihood-based
- **NN trained on real data**: performs best



Extension: Neural Network for 6 Hypotheses

Motivation

- Identify all 6 hypotheses using a single neural network $(e,\ \mu,\ \pi,\ K,\ \ p,\ d\ )$

Neural Network for 6 hypotheses

- Same inputs
- Same network structure
- 6 outputs → Probabilities for 6 possible hypotheses

Training sample

- Training neural network for 6 hypotheses requires clean training on sample containing all 6 particle species
  → Train on particle-gun MC

Does training on all 6 species decrease the $K-\pi$ separation performance?

Models overview

Models (for 2 hypotheses)

- **Pure likelihood-based**
- **NN trained on real data**
- **NN trained on particle-gun MC for 2 hypotheses**

- Predict probability for pion and kaon hypothesis

Models (for 6 hypotheses)

- NN trained on particle-gun MC for 6 hypotheses

- Predicts probabilities for all 6 hypotheses

Binary Normalization

$y_1 = P_{NN}(e)$

$y_2 = P_{NN}(\mu)$

$y_3 = P_{NN}(\pi)$

$y_4 = P_{NN}(K)$

$y_5 = P_{NN}(p)$

$y_6 = P_{NN}(d)$

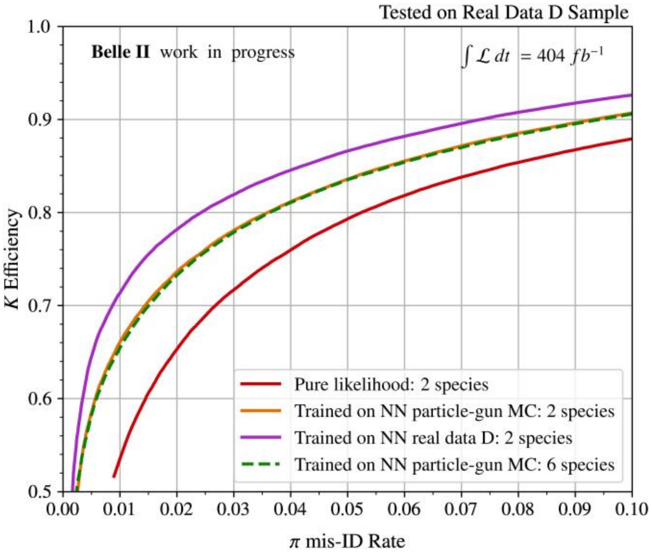
- Normalize probabilities considering only the tested hypotheses, i.e. K and π here:
  - $P'(K) = \frac{P_{NN}(K)}{P_{NN}(K)+P_{NN}(\pi)}$
  - $P'(\pi) = \frac{P_{NN}(\pi)}{P_{NN}(K)+P_{NN}(\pi)}$
- **NN trained on particle-gun MC for 6 hypotheses binary normalization**

Performance

Models

- **Pure likelihood-based**
- **NN trained on real data**
- **NN trained on particle-gun MC for 2 hypotheses**
- **NN trained on particle-gun MC for 6 hypotheses Binary normalization**

- There is no loss in performance for $K-\pi$ separation between training a neural network for 2 or for 6 hypotheses



Conclusions

Models for 2 hypotheses

- Neural networks perform better than pure likelihood-based approach → overcome limitations
- Neural networks performs better when trained or real data than with simulated data
    - Training on real data overcomes imperfections in simulation
- Performance increase of:
    - 30% when trained on real data
    - 20% when trained on simulated data

Models for 6 hypotheses

- Neural networks can be trained for multiclass classification without losing the performance for binary classification

Outlook

- Test the 6 hypotheses neural network for other particle species, e.g. e or μ
- Go beyond likelihood inputs

Backup

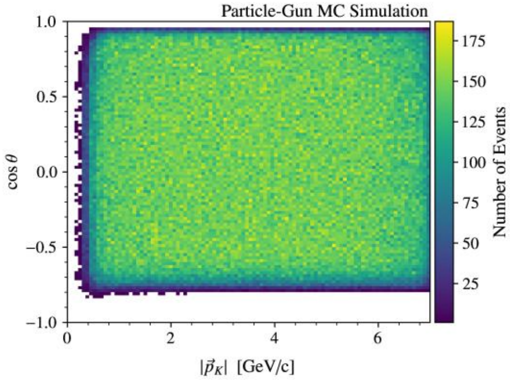
Real Data and Particle-Gun MC

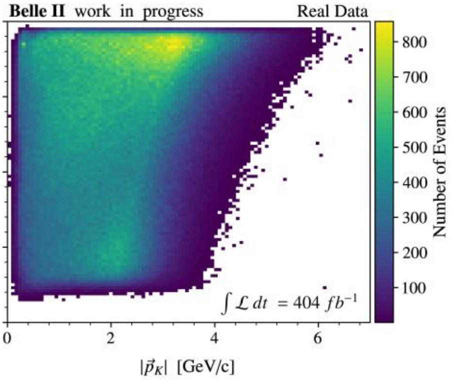
particle-gun MC

- Covers full kinematic range
- Particle-gun MC sample designed to impose minimal bias from sample distributions

Real data

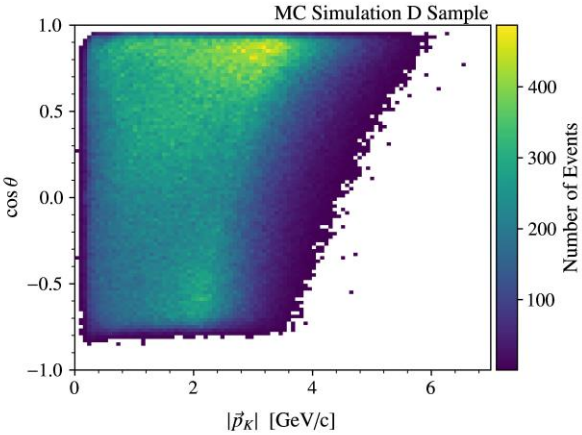
- Covers limited kinematic range $|\vec{p}| < 4.5$ GeV/ c
  - Neural Network cannot be used outside this range

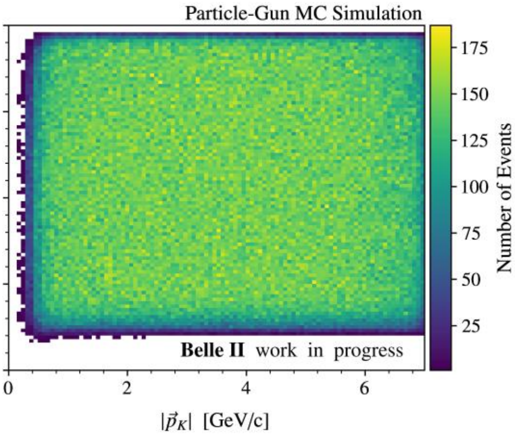




MC D Sample vs Particle-Gun MC

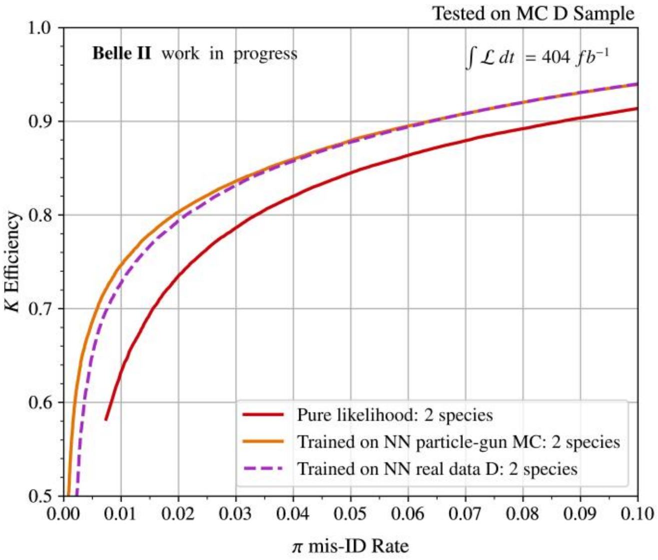
- MC simulating the physical decay of D
- Particle-gun MC: generating isotopically particles





Performance: Tested on MC D Sample

- Both **NN trained on real data** and **NN trained on particle-gun MC for 2 hypotheses**: have a better performance than **pure likelihood-based**



Test On MC: Results

- Neural network trained on real data performs similarly than neural network trained on particle-gun MC for 2 hypotheses
  - It confirms that Training in Real Data overcomes imperfections in simulation and it is not something related to D* sample

Sample

- Sample where we know the true specie without PID: $D^{*\pm}$
- $D^{*+} \to D\,\pi^{+}$
  - $D \to K^{-}\pi^{+}$

or

- $D^{*-} \to \overline{D}\,\pi^{-}$
  - $\overline{D} \to K^{+}\pi^{-}$