

Universität Hamburg

Fakultät Physik

Masterarbeit

Shower shape conversion between electrons and photons in Belle  
II using Cycle Generative Adversarial Network

*Abgabetermin: 08. März 2021*

Verfasser:

Cedric Ly

Rennbahnstraße 125k

22111 Hamburg

Matrikelnummer: 6543166

( Erstgutachter: Dr. Torben Ferber )

( Zweitgutachter: Jun.-Prof. Dr. Gregor Kasieczka )



# Abstract

Simulation is essential to high-energy physics. The application of simulation ranges from development of software to the analysis of physical processes. For example the simulation of physical processes helps to gauge the expected outcome for specific rare decays of B mesons. The simulation in Belle II can be divided into two parts. The first part simulates the physics of the particle with given property like four momentum. The second part simulates the interaction of the simulated particle with the Belle II detectors. The quality of simulation is determined firstly by our understanding of physics and secondly by the consensus with measurements. Therefore, every discrepancy between Monte Carlo simulation (MC) and data could either be caused by a mistake in the measurements or by a lack of understanding of the fundamental processes and structure of the world. In many processes of interest, like ALP-strahlung or the  $B \rightarrow X_q \gamma$  channel, the study is dependent on the quality of the simulation, especially for Belle II as a high precision experiment. Therefore, it is appropriate to look for new ways to avoid and even understand discrepancies between data and simulation. In this thesis a novel approach for simulating electromagnetic shower shapes in the Electromagnetic Calorimeter (ECL) of Belle II has been developed.

A Cycle Generative Adversarial Network (CycleGAN) is a deep neural network architecture, which one could use to convert an element of one class into an element of another class. The overall goal would be to convert a real detected electron shower shape into a shower shape of a photon. The decisive property of a CycleGAN is the correlation between input and output. Meaning the generated photons will have the profile of a photon, but retain many features of the original input electron. In the ideal case, the conversion will retain the particle energy, the distinct background features, and only change (the shower shapes) features by the minimal amount needed, in order to recognize it as an element of the targeted class. An evaluation of the new approach will be presented. This approach may be able to produce calibration samples based on real detected particles and is not only limited by electrons and photons. This approach can be used in theory for any two classes and could enable in many different potential applications beyond the Belle II experiment.



# Kurzfassung

Simulation ist ein wichtiger Bestandteil in der Hochenergie-Physik. Die Anwendungen von Simulation sind divers und helfen bei der Entwicklung von neuer Software, wie auch in der Analyse von physikalischen Prozessen. Zum Beispiel werden Simulationen genutzt um Häufigkeiten von seltenen B Zerfällen abzuschätzen. Simulation in Belle II kann man in zwei Kategorien unterteilen. Die erste Kategorie simuliert die Prozesse und das Verhalten der Teilchen, unter gegebenen Bedingungen, während die andere Kategorie, die Wechselwirkung zwischen Teilchen und Detektor simuliert. Die Qualität der Simulation ist zum einen von unserem Verständnis für die Physik, und zum anderen von der Übereinstimmung mit der Messungen bestimmt. Folglich ist eine Abweichung zwischen der erhaltenen Daten von den Monte Carlo Simulationen (MC) ein Indiz für eine schlechte Messung, oder einen Mangel an Verständnis für die Physik. In vielen interessanten Kanälen, wie ALP-strahlung oder  $B \rightarrow X_q \gamma$ , ist die Erforschung abhängig von der Qualität der Simulation. Für das Belle II Experiment ist die Qualität besonders wichtig, da es im Hochpräzisionssektor agiert. Auf Grund dessen, ist es angemessen neue Methoden zu untersuchen, um Diskrepanzen zwischen Daten und Simulation zu vermeiden und sogar zu verstehen. Diese Abschlussarbeit untersucht eine neue Herangehensweise um "shower shapes" im elektromagnetischen Kalorimeter (ECL) zu simulieren.

Cycle Generative Adversarial Network (CycleGAN) ist ein neuronales Netzwerk mit einer bestimmten Architektur, die einem erlaubt eine Umwandlung zwischen zwei Klassen durchzuführen. Das Ziel der Abschlussarbeit ist es die shower shapes eines echt detektierten Elektrons in die eines Photons umzuwandeln. Die entscheidende Eigenschaft eines CycleGANs ist der Zusammenhang zwischen Input und Output. Das Output ist das äquivalente Teilchen des Inputs in der angestrebten Klasse. Die Abschlussarbeit soll somit die shower shapes von detektierten Elektronen umwandeln, sodass man das Ergebnis zu einem Photon einordnen würde, es jedoch so viele Eigenschaften und Eigenarten (wie Hintergrund, Energieverteilung etc.) des Inputs beibehält. Diese Abschlussarbeit fokussiert sich auf den Versuch eines CycleGANs. Außerdem wird eine Methodik entwickelt einen solchen CycleGAN physikalisch relevant zu evaluieren. Aus dieser Herangehensweise könnte zukünftig eine Kalibrationsprobe aus echt detektierten Teilchen hervorgehen. Prinzipiell ist die Methodik auch nicht auf Belle II beschränkt, weitere Anwendungen wären in der Physik denkbar.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Overview of the standard model . . . . .	1
1.2	Unsolved problems . . . . .	2
1.3	Motivation . . . . .	4
<b>2</b>	<b>SuperKEKB and Belle II</b>	<b>7</b>
2.1	Experimental Setup . . . . .	7
2.2	SuperKEKB . . . . .	8
2.3	Belle II . . . . .	9
2.3.1	Electromagnetic Calorimeter (ECL) . . . . .	10
2.4	Belle II analysis software framework . . . . .	12
2.5	Simulation and reconstruction in the Electromagnetic Calorimeter . . . . .	13
<b>3</b>	<b>Overview: Simulation in Electromagnetic Calorimeter</b>	<b>15</b>
3.1	Simulated data sample . . . . .	16
3.2	Shower shapes variables . . . . .	18
<b>4</b>	<b>Cycle Generative Adversarial Network</b>	<b>27</b>
4.1	Introduction Cycle Generative Adversarial Network . . . . .	27
4.1.1	Assumptions of this method . . . . .	27
4.1.2	General architecture . . . . .	29
4.1.3	Strategy . . . . .	31
4.2	Pre-trained classifier . . . . .	32
4.2.1	Classifier and training parameters . . . . .	33
4.2.2	Results . . . . .	36
4.3	Pairwise metric . . . . .	41
4.3.1	GAN $\Phi_{\text{cluster}}$ . . . . .	42
4.3.2	Constraining term for generator . . . . .	45
4.3.3	Measuring closeness . . . . .	47
4.4	Training parameters and models . . . . .	47
<b>5</b>	<b>Results</b>	<b>51</b>
5.1	Training evolution . . . . .	51
5.2	Pre-trained classifier evaluation . . . . .	57
5.3	Pairwise metric evaluation . . . . .	60
5.4	Conclusion and summary . . . . .	63

<b>6 Outlook</b>	<b>65</b>
6.1 Problems . . . . .	65
6.2 Future steps . . . . .	69

## Thesis guide

This thesis takes the first steps to improve simulation of calorimeter clusters for photons by employing a novel technique to "convert" calorimeter clusters from electrons. The first chapter 1 opens with a general introduction to particle physics and the standard model. The success and the shortcomings of the standard model are addressed. The need for a model beyond the standard model is also discussed. At the end of the first chapter, a connection is drawn between the motivation of this thesis and the shortcomings of the SM.

The second chapter 2 gives a short introduction into the experimental framework. Even if the motivation is not specific to Belle II, the thesis is within the framework of Belle II. This chapter describes the electron-positron collider and various sub-detectors. The electromagnetic calorimeter is of interest to this thesis and is given its own subsection. The second chapter closes with the distinction between the reconstruction and simulation from Belle II. This distinction is important because the thesis uses simulation and information from the reconstruction in order to develop and train the conversion.

The next chapter 3 presents the reconstructed particles, which will be used throughout the master thesis. Some variables are presented in details in this chapter. This chapter defines the initial situation and presents variables in more details.

Chapter four 4 introduces the novel approach of the CycleGAN. The principle and application of the CycleGAN is covered. Additionally, two methods to evaluate the performance of the CycleGAN is discussed in their respected subsection. This chapter finishes with the description of the architecture and training of the CycleGAN.

The next chapter 5 presents and discusses the results of the training. For the discussion the two methods from the previous chapter is used. In the last chapter 6 current problems, which have not been solved in this thesis, and possible solutions are discussed. In addition, this chapter provides next steps and an outlook.

## Definitions

This thesis invents its own variables and use them throughout the master thesis. There are simulation and reconstructed variables for the simulated/reconstructed particles, some will be discussed further in details in chapter 3. There will also be variables for the output of the CycleGANs generators. In table 1 an overview over all used variables is given.

Table 1: Overview of variables used in this thesis.

aliases	description
image	5x5 values of deposited energy in crystals. Center crystal/pixel is local maximum.
$E_{MC}$	Particle energy of the simulated particle.
$\Phi_{MC}$	Azimuthal angle $\Phi = \cos^{-1}(\frac{p_t}{p})$ of the simulated particle.
$\Theta_{MC}$	Polar angle $\Theta = \cos^{-1}(\frac{p_z}{p})$ of the simulated particle.
cluster	Basf2 reconstructed object of the energy deposition and often used as subscript from the reconstructed particle in the electromagnetic calorimeter.
M	Ratio between particle energy within the cluster and the total particle energy of the simulated particle.
$E_{cluster}$	Basf2 reconstructed energy of the cluster.
$\Phi_{cluster}$	Basf2 reconstructed $\Phi$ of the cluster.
$\Theta_{cluster}$	Basf2 reconstructed $\Theta$ of the cluster.
$E_{helix}$	Analytical energy of cluster if the electromagnetic charge of the simulated particle is reversed.
$\Phi_{helix}$	Analytical $\Phi$ of cluster if the electromagnetic charge of the simulated particle is reversed.
$\Theta_{helix}$	Analytical $\Theta$ of cluster if the electromagnetic charge of the simulated particle is reversed.
shower shape	Variables describing the shape of the deposited energy.
GAN image	Output of one of the two generators from CycleGAN.
GAN particle	Combination of GAN image + the helix variables.
reconstructed particle	Image + cluster variables from simulated particle.

# 1 Introduction

## 1.1 Overview of the standard model

The Standard Model (SM) [1–6] is the reigning quantum field theory to explain the interaction between fundamental particles. The SM was able to predict particles like the top quark and a Higgs boson. The current SM model differentiates between fermions and bosons. Fermions have a half integer spin and are the matter content of the SM. There is a unique relationship between these fermions. Every fermion can be put together with two other fermions, which all share the same traits (like electric charge, spin etc.) except for their mass. Physicists categorize these fermions into three generations, with the third generation being the heaviest particle among the particles. One can categorize the fermions into four different types, each with three flavors.

The fermions are summarized in Table 1.1. Hadrons are formed from quarks. For example two up quarks and one down quark form a proton, or one up quark and one anti-bottom quark would form a  $B^+$ -meson. The charge and the mass for each quark is written in Figure 1.1.

The bosons, except for the Higgs boson are representative of one of three fundamental forces. These gauge bosons (see Figure 1.1) could also be called “force carrier”. One interaction of interest in this thesis is the weak interaction. W-bosons can change the flavor of a quark. The possible flavor changing process is strongly suppressed. The suppression of a flavor changing weak interaction is described by the “Cabibbo-Kobayashi-Maskawa matrix” (CKM matrix).

$$\begin{pmatrix} d' \\ s' \\ b' \end{pmatrix} = \begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{pmatrix} \begin{pmatrix} d \\ s \\ b \end{pmatrix}$$

Table 1.1: Fermions types and their flavors.

Fermion Type	$SU(3)_c$	$U(1)_{EM}$ (antimatter reversed sign)	flavor
Up - type quarks	Triplet (3)	+2/3	u,c,t;
Down - type quarks	Triplet(3)	-1/3	d,s,b;
Charged leptons	Singlet (1)	-1	$e, \mu, \tau$
Neutrinos	Singlet (1)	0	$\nu_e, \nu_\mu, \nu_\tau$

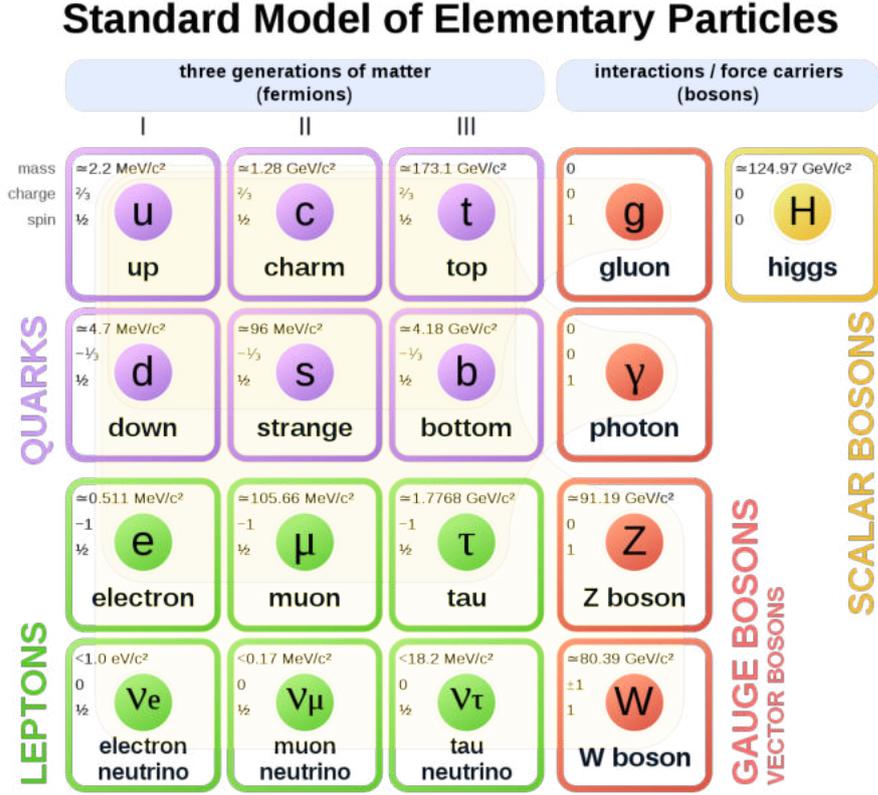


Figure 1.1: Summary of the current standard model [7].

$$|\mathbf{V}_{\text{CKM}}| = \begin{pmatrix} 0.97425 \pm 0.00022 & 0.2252 \pm 0.0009 & 0.00389 \pm 0.00044 \\ 0.230 \pm 0.011 & 1.023 \pm 0.036 & 0.406 \pm 0.0013 \\ 0.0082 \pm 0.0006 & 0.038 \pm 0.0021 & 0.88 \pm 0.07 \end{pmatrix} \quad [8]$$

$\begin{pmatrix} d' \\ s' \\ b' \end{pmatrix}$  is the weak interaction doublet partners of up-type quarks, while  $\begin{pmatrix} d \\ s \\ b \end{pmatrix}$  is the mass eigenstates of the quarks. The element of the CKM matrix ( $\mathbf{V}_{ij}$ ) quantifies the flavor changing weak decay. The probability of transition from a particle of type  $i$  to a particle of type  $j$  is proportional to  $|\mathbf{V}_{ij}|^2$ .

## 1.2 Unsolved problems

The SM is successful in its experimental prediction and it describes the interaction (except for gravitational interaction) in a coherent theoretical framework. Nevertheless,

there are phenomena the SM cannot explain. Physics beyond the standard model (BSM) may provide answers to questions like what is Dark Matter, why does neutrino oscillation exists, and what cause the asymmetry between matter and antimatter. BSM could extend the theory of the SM and might help to get some hints for a theory of everything.

Dark Matter (DM) is an interesting subject. The existence of DM is postulated in order to explain the rotational velocity from the analysis of the spectral lines [9]. In the study the rotational velocity does not decline with  $\frac{1}{\sqrt{r}}$ , with  $r$  being the distance between object and nucleus, instead the rotational velocity is flat (see figure 1.2). In [11] DM

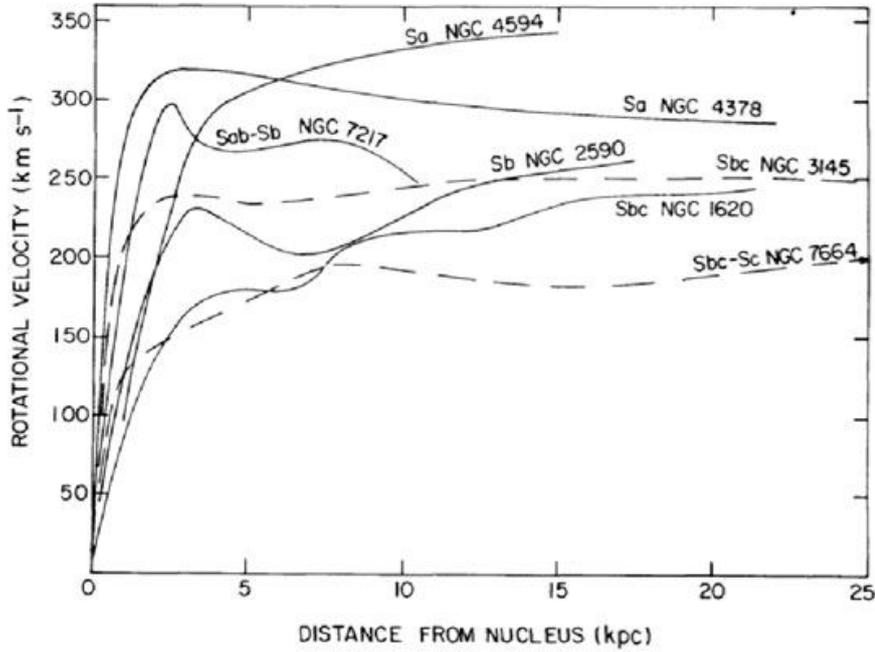


Figure 1.2: Rotational velocity of the galaxies as a function of distance [10].

accounts for  $\approx 27\%$  of the total energy density of the universe and for about 85% of its matter density. The existence of dark matter provide a strong hint for a dark sector, with potentially many particles, which may generally not interact with the SM particles (outside of gravitational force). Some models proposed light dark matter [12]. In those models a “mediator” connects the SM with the dark sector. The mediator is a particle with quantum number from the SM and from the dark sector, so it can interact directly or indirectly with SM particles.

Depending on the model, dark sector particles can be used to explain phenomena, like the discrepancy between measured magnetic moment of the muon and its theoretical SM value [13].

### 1.3 Motivation

Potential mediators would be Axion-like particles (ALPs). The sensitivity of ALPs in Belle II is discussed in [14,15]. A promising channel would be ALPs coupling to photons. The signature of those ALPs in Belle II is dependent on the coupling and the masses of the ALPs (see figure 1.3). The Feynman diagram is shown in figure 1.4. If the ALP

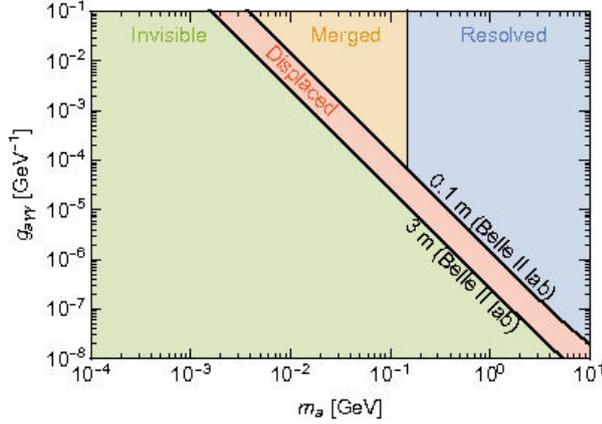


Figure 1.3: Illustration of the different kinematic regimes relevant for ALP decays into two photons with Belle II [14, 16].

does not decay within Belle II or if the ALP decays into particles in the dark sector, the final state will consist of a single photon. The single photon search in Belle II is therefore important. A calibration sample of photons in Belle II can be obtained from radiative Bhabha events  $e^+ e^- \rightarrow e^+ e^- \gamma$ . A calibration sample for low energetic  $\gamma$  in the range of a few MeV is quite challenging, as they are similar to the noise due to pileup and electronics. The Belle II Electromagnetic Calorimeter can detect electromagnetic showers of photons. The lower the particle energy of the  $\gamma$ , the more difficult it is to obtain a calibration sample. For high energy photons ( $\approx$  some GeV) one can estimate the position of the photon from the radiating Bhabha events by studying the angle between the electron and the positron. Figure 1.5 illustrates this behavior.

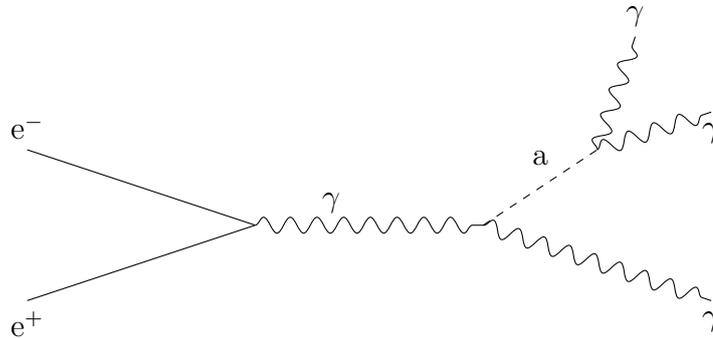


Figure 1.4: Feynman diagram of ALP-strahlung.

The combination of the estimated position and the greater discrimination from noise makes it easier to obtain a calibration sample for high energy photons. These properties do not apply for low energetic photons. The t-channel of a normal Bhabha event is significant and many electrons and positrons scatter with a small angle. Small angles make it difficult to determine the location of the photon if the Bhabha event was radiative. In addition, a radiating Bhabha event with a small angle between the electron and positron has a high probability of hitting the magnet of the detector system, producing various particles such as electrons and photons. These particles can hit the electromagnetic calorimeter and produce electromagnetic showers. The combination of poor location estimation for the photon of the radiating Bhabha event and the additional electromagnetic showers of photons from the interaction between an electron or positron and the magnet, make it difficult to get a calibration sample.

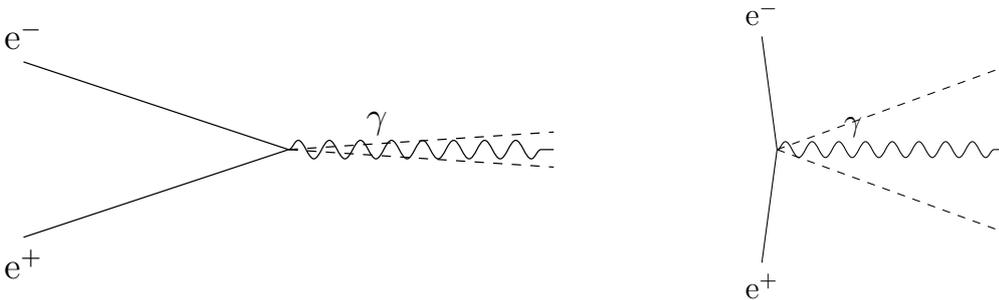


Figure 1.5: Illustration of the location estimate (within dashed line - -) of high energy  $\gamma$  on the left side and of low energy  $\gamma$  on the right side.

A simulation-to-data approach could bridge the gap between a simulated sample and a calibration sample. This gap will be exceedingly interesting for the low energetic photons. In this paper [17], the authors attempt to calculate photon identification efficiencies for photons in a range of transverse momentum without a calibration sample. They used three different data-driven techniques to obtain a sample on which to base their calculation. One of the three different data-driven techniques was to extrapolate the properties of photonic electromagnetic showers from the properties of electronic and positronic electromagnetic showers. The difference between the interaction of electrons with the electromagnetic calorimeter and the interaction of photons with the electromagnetic calorimeter is almost non-existent.

The overall goal is to convert a calibration sample of electrons into a calibration sample of photons by exploiting the similarity between the photonic and electronic electromagnetic showers. A calibration sample of electronic electromagnetic showers is quite easy to obtain, since the electrons in the Belle II Central drift Chamber leave a track that can be assigned to the shower. The only limitation is that the electrons must reach the electromagnetic calorimeter and produce an electromagnetic shower. This limitation is met at  $\approx 200$  MeV.

The goal of this work is to develop a cycle-consistent Generative Adversarial Network (CycleGAN) and to investigate the application of the CycleGAN for the conversion be-

## 1 Introduction

tween electromagnetic shower shapes and photonic electromagnetic shower shapes in the context of Belle II. Projects using similar techniques in the field of generating electromagnetic showers can be found here [18–21].

The approach of CycleGAN is already used in the field of medicine some examples are [22,23]. The objective is to showcase the difficulty and applicability of this approach in high energy physics. This new approach is not limited to Belle II or to the transformation between electrons and photons. The CycleGAN could also be interesting for other particle physics experiments and maybe it will be able to transform different events or channels.

# 2 SuperKEKB and Belle II

## 2.1 Experimental Setup

The Belle II detector is located at the SuperKEKB collider in Tsukuba (Japan). The SuperKEKB accelerator collides  $e^+$  and  $e^-$ . The energy of the collision corresponds to the rest mass of an  $\Upsilon(4S)$  resonance (10.58 GeV). SuperKEKB is also called "B-factory", because  $\Upsilon(4S)$  almost solely decays to a B meson pair. The advantage of the SuperKEKB is that it produces as many B mesons as possible in a clean environment. An overview is shown in figure 2.1.

The whole technical report for this chapter can be read here [24]. The complete physics program of Belle II is summarized in [25].

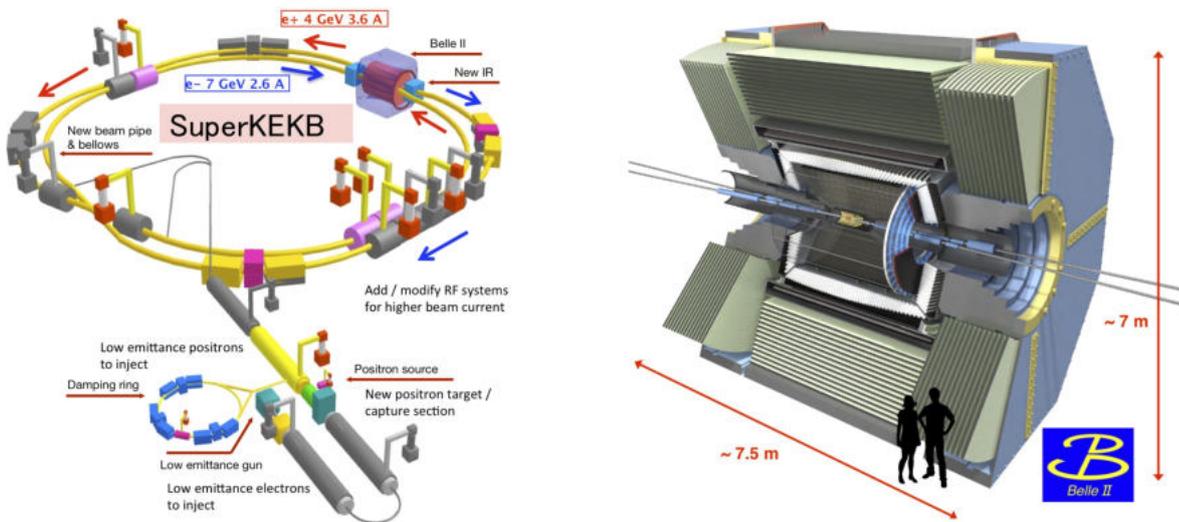


Figure 2.1: SuperKEKB facility (left) provide a 40-fold increase in instantaneous luminosity by exploiting higher beam currents, a large crossing-angle and squeezing the beams down to nanometre scales, in comparison to KEKB. Also the former Belle detector was upgraded to Belle II (right) to cope with the enormous increase in intensity.

## 2.2 SuperKEKB

The SuperKEKB accelerator contains a high energy ring (HER) and a low energy ring (LER). The center of mass energy  $\sqrt{s}$  is slightly above the threshold of two B mesons.

$$\sqrt{s} = \sqrt{4E_{\text{HER}}E_{\text{LER}}} \approx 10.58 \text{ GeV} \text{ with } E_{\text{LER}}=3.5 \text{ GeV and } E_{\text{HER}}=8 \text{ GeV.}$$

Due to the asymmetric energy, the system has a Lorentz-boost in the direction of HER. As a consequence, the B mesons experience an increased lifetime in the laboratory frame. This allows a better measurement of its flight.

One of the specialties of the SuperKEKB is its peak luminosity. The following figure 2.2 compares the SuperKEKB to other existing and future accelerators.

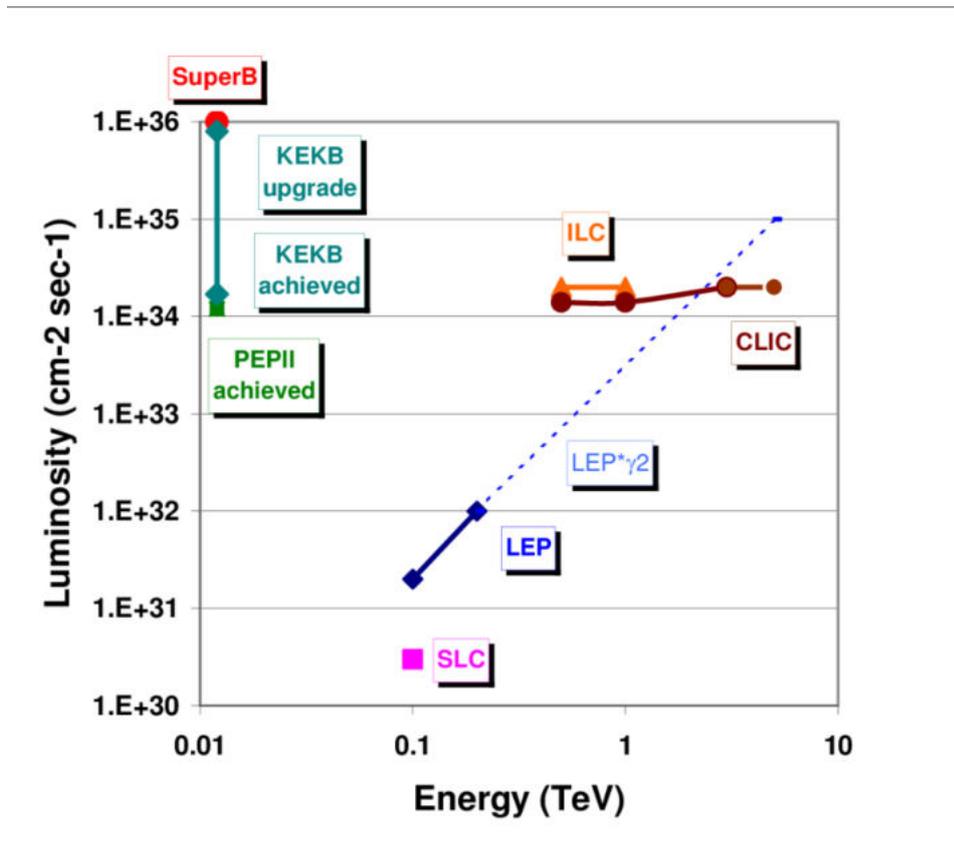


Figure 2.2: Peak luminosity of different accelerators for different cms energy [26].

The luminosity ( $L$ ) is very important, because it correlates to the interaction rate ( $\frac{dN}{dt}$ )

$$\frac{dN}{dt} = L \cdot \sigma, \quad (2.1)$$

where  $\sigma$  is the SuperKEKB specific cross section. SuperKEKB holds the title for the worlds highest luminosity with  $L = 2.4 \times 10^{34} \text{ cm}^{-2}\text{s}^{-1}$  (on June 21st 2020).

## 2.3 Belle II

Belle II is a complex detector system and can be broken down into sub detectors. Starting from interaction point of  $e^+$  and  $e^-$  to the outer layer of the detector system, Belle II is organized in the following order:

1. Vertexing detectors
2. Tracking detectors
3. Barrel particle ID
4. Forward particle ID
5. Electromagnetig Calorimeter
6.  $K_L^0$  and  $\mu$  detection

This thesis only uses the Electromagnetic Calorimeter and does not include the other sub detectors. The ECL is discussed in detail separately in subsection 2.3.1 and the other subdetectors will only be mentioned in a brief paragraph. If not stated otherwise, a polar coordinate framework will be used with  $\cos(\Theta) = \frac{p_z}{p}$  and respectively  $\cos(\Phi) = \frac{p_t}{p}$ , whereas  $p_z$  is the momentum along the beam axis,  $p_t$  is the transverse momentum and  $p$  the total momentum.

**Vertexing Detector (VXD)** The "Pixel Detector (PXD)" and the "Silicon Vertex Detector (SVD)" are shaping the VXD. Their mutual purpose is to measure precisely the location of charged particle near the interaction point. PXD is the detector nearest to the interaction point and due to the high luminosity of SuperKEKB, it will have an extremely high hit rate. The PXD consists of two layers and are 13 mm and 22 mm away from the beam pipe (10 mm). A strip detector would not be functional, because the large occupancy would make the reconstruction of the B-decay vertices impossible. Strip detector are convenient at a distance of 40 mm. The higher number of channels for one pixel of the PXD reduces the issue of high occupancy near the interaction point. Both detector cover an acceptance region of  $17^\circ < \Theta < 150^\circ$  along the beam pipe.

**Tracking detector: Central Drift Chamber (CDC)** The CDC has three important tasks.

The first task is to reconstruct all charged tracks by analyzing the hits inside the CDC. Secondly, it shall record data like momentum and energy loss  $\frac{dE}{dx}$  for particle identification and lastly the CDC is an efficient and reliable trigger signal of charged particles.

The CDC is filled with 50% helium and 50% ethane. Like the VXD it covers a polar angle of  $17^\circ < \Theta < 150^\circ$ .

**Barrel particle ID: Time-Of-Propagation (TOP)** TOP is the successor of the PID-Barrel from Belle which contains the Aerogel Cherenkov Counter and the Time Of Flight counter (ACC and TOF).

TOP can differentiate charged particles by making use of the cone opening angle, after the particles passes through a radiator material and emits the specific Cherenkov photons. The cone opening angle of the Cherenkov radiation is determined by the velocity and mass of the particle  $\beta$ . At figure 2.3 the principle of the Top is shown schematically.

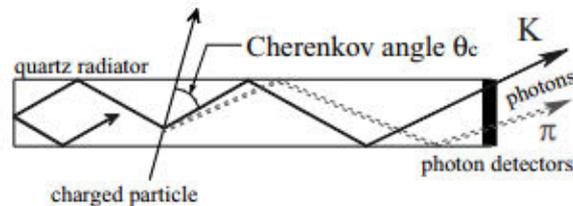


Figure 2.3: The principle of TOP to differentiate particles [27].

**Forward particle ID: Aerogel Ring-Imaging Cherenkov detector (ARICH)** Similar to TOP but located to the forward end cap, ARICH is a pure ring-imaging Cherenkov detector with aerogel radiators. Particles passing the first layer of aerogel will produce Cherenkov photons. The signal from these photons gets enhanced by a photomultiplier. This detector is designed to separate kaons from pions and to identify particles below 1 GeV [28].

**$K_L^0$  and  $\mu$  detector (KLM)** The KLM is a calorimeter with shorten interaction length due to the alternating layers of 4.7cm thick iron plates and resistive plate chambers (RPC). The main task of the KLM is to detect  $K_L^0$  and  $\mu$ .  $K_L^0$  should make an electromagnetic shower in the ECL and an hadronic shower within the iron plates of the KLM, which the RPC detects.  $\mu$  can pass the KLM detector, because of their high penetration prowess and will differ in that way from other particles. The KLM is also used as a flux return for the magnetic field.

### 2.3.1 Electromagnetic Calorimeter (ECL)

[29]

The ECL consists of 8736 thallium doped CsI scintillator crystals with a truncated pyramid shape. One schematically example for an ECL crystal is shown in figure. This material is convenient for electrons or hadrons to deposit their energy as a shower in the form of light, which will be measured with a photodiode attached to the rear of the barrel. The crystal material has a radiation length of  $X_0 = 1.86$  cm and a Moliere radius

of  $R_m = 3.53$  cm [30]. The ECL can be divided into three parts the barrel, the backward endcap (BWD ECL) and the forward endcap (FWD ECL).

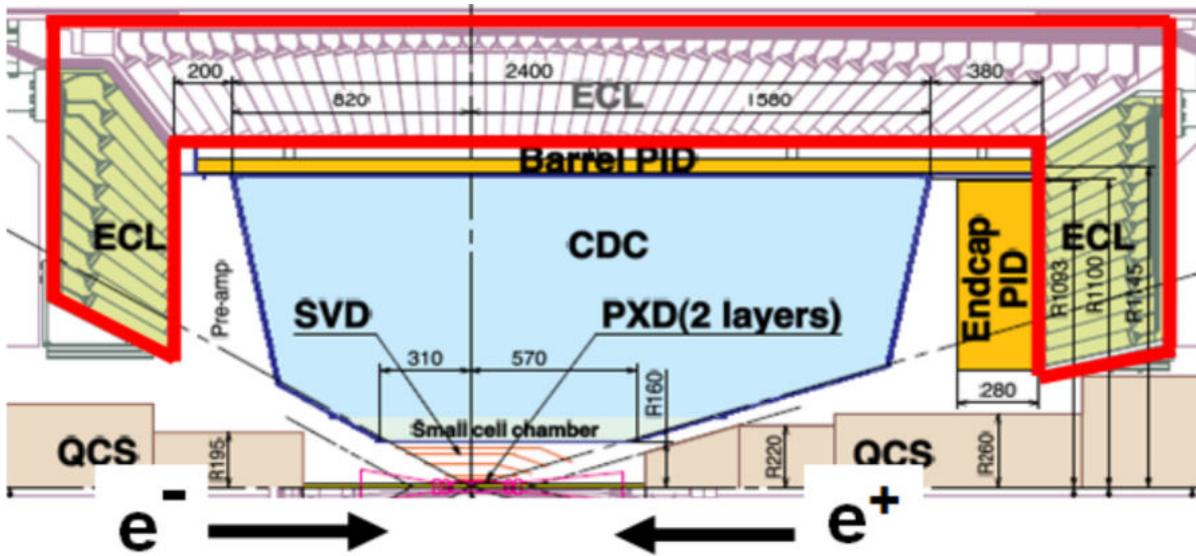


Figure 2.4: ECL crystal arrangement. Violet crystals compose the barrel [31].

The entire ECL comprises of 8736 crystals. 6624 crystals are within the barrel and covers a polar angle of  $12.4^\circ < \theta < 155.1^\circ$ , except for two gaps between the barrel and the BWD ECL, as well as the FWD ECL. The setup of one crystal is schematically shown in figure 2.5. This thesis will focus solely on the barrel. The arrangement in the BWD ECL and FWD ECL is more complex and requires special consideration for its geometry. The inner radius of the barrel accounts to 1250 mm and every crystal is set up to point at the interaction point (IP). The crystal arrangement is shown schematically along the beam pipe in figure 2.4. The ECL measures energy in the form of electromagnetic shower and position. Photons and electrons can produce an electromagnetic shower by pair production and bremsstrahlung. The energy measured in the ECL can be used for kinematic calculation, track reconstruction and particle identification. One task of utmost importance for the ECL is the detection of photons, especially of its energy as well as its location. Otherwise, the photon will likely pass the detector without being detected. Being able to detect photons and neutral particles, the ECL is needed for  $\pi^0$  measurement. The ECL also distinguishes between electrons and hadrons by calculating the ratio of shower energy and track momentum  $\frac{E}{p}$ .

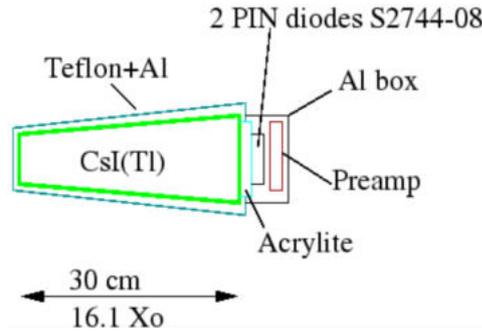


Figure 2.5: ECL crystal setup [32].

## 2.4 Belle II analysis software framework

The evaluation of simulated or recorded data is done with the "Belle II analysis software framework (basf2)" [33]. Basf2 is written in C++ with a Python 3 user interface. For a better understanding of the framework, the framework is divided into packages. Every package deals with a different task of data processing. The user can create a path, which usually starts with the input of data and ends with an output file in Python 3. The path contains so-called "modules" which are provided by the packages from the basf2. Such modules fulfill a certain task like MC generation, detector simulation, reconstruction and many more. By executing such a python file, the path will be built in python and the modules will be executed in C++.

In general the structure looks like:

1. data acquisition (daq),
2. Monte Carlo event generation (generators),
3. general detector simulation (simulation),
4. sub-detector simulation, unpacking and reconstruction (package for each sub-detector),
5. track reconstruction (tracking),
6. visualization of individual events inside the detector (display),
7. physics analysis (analysis).

If the development of a converter for electronic electromagnetic shower shapes to photonic electromagnetic shower shapes was successful, this converter could be added as a module to either the reconstruction or simulation package. The following section explains the difference between simulation and reconstruction within ECL in Belle II.

Table 2.1: Variable of simulated particle.

Aliases	Description
$E_{MC}$	Particle gun created a simulated particle with this particle energy.
$\Theta_{MC}$	Particle gun created a simulated particle with this polar angle $\Theta = \cos^{-1}(p_z/p)$ .
$\Phi_{MC}$	Particle gun created a simulated particle with this azimuthal angle $\Phi = \cos^{-1}(p_t/p)$ .

## 2.5 Simulation and reconstruction in the Electromagnetic Calorimeter

The generation of Monte Carlo samples of simulated Data (MC) is the foundation for this thesis. The simulation package of basf2 is based on Geant4 [34]. Geant4 is a software toolkit. While Geant4 governs the interaction between a simulated particle with the matter of the detectors, one has to create such a simulated particle. The simulated particles were created by a "particle gun". Obviously the creation of a simulated photon will differ from the creation of a simulated electron. The particle gun class needs to have all the physics relevant information of the simulated particle that it wants to create, before passing it to the Geant4 simulation. The particle gun generates a single particle. Generated particles have uniformly distributed energy ranging from [500 MeV, 5 GeV], polar angle  $\Theta$  from [216 mrad, 2707 mrad] and azimuth angle from [-1000 $\Pi$  mrad, 1000 $\Pi$  mrad]. Table 2.1 sums up three variables of the generated particle.

After successfully creating a simulated particle by the particle gun, the simulated particle has to travel through the detector system of Belle II. The propagation and every interaction with each detector will be simulated by the Geant4 package. For this thesis the interaction between the simulated particle and the ECL is primarily of interest. This thesis will not concentrate on the track of a simulated electron in the CDC. The interaction of the simulated particle and the ECL may concentrate at one crystal of the ECL, but the deposited energy can spread into neighboring crystals. The deposited energy is detected. The deposition of the particle energy in the ECL is a shower shape, the basf2 package will define a cluster, which shall represent the shower shape.

A cluster is a reconstructed electromagnetic shower shape object, reconstructed by a basf2 reconstruction module. This module does not differentiate between simulated energy deposition and data energy deposition.

The basf2 reconstruction module for the cluster starts with the readout of all crystals exceeding the threshold of  $\approx 0.2$  MeV. Each crystal with more than 10 MeV will be marked and if the neighboring crystals contains more than 0.5 MeV they will be marked to building connected regions. The connected regions grows iteratively with each neighboring crystal of the connected regions exceeding a threshold of 1.5 MeV. There is an algorithm within the basf2 reconstruction to split overlapping regions into reconstructed showers. There are as many reconstructed showers as local maximum. For each reconstructed shower one can reconstruct the location of the centre of density of the shower and the particle energy with energy leakage correction. The cluster is a different data

## 2 *SuperKEKB and Belle II*

object than the reconstructed shower, but every reconstructed shower with more than 20 MeV will be saved as cluster [35].

### 3 Overview: Simulation in Electromagnetic Calorimeter

This thesis works with a combination of an image and reconstructed variables. The image is like the expanded view on a reconstructed shower. A cluster typically only includes the direct neighboring crystals, but for this thesis the surrounding was expanded to two neighboring crystals. This results in 25 crystals for one image with one cluster within these 25 crystals. Figure 3.1 illustrates a cluster within an image. The local maxima of the image is always located at the center crystal. Additionally to the image reconstructed variables are used. Those variables are summarized in table 3.1. Matchweight ( $M$ ) is a variable only known solely for simulated particles.  $M$  gives the ratio of particle energy deposited within found cluster over the particle energy of the simulated particle. Real detected data will not have any information from table 2.1 or the  $M$  from table 3.1, but the reconstructed variables is the same for simulated and real detected data.

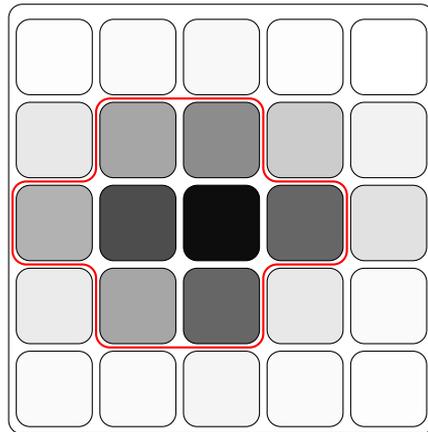


Figure 3.1: Example image of 25 crystals. Each rounded rectangle is a crystal of the ECL, and the color of the crystal represents the amount of deposited energy. Red line is the found cluster.

Table 3.1: Variable of the reconstructed particle.

Aliases	Description
$E_{\text{cluster}}$	Reconstructed energy leakage corrected energy by reconstruction package of basf2.
$\Theta_{\text{cluster}}$	Polar angle $\Theta = \cos^{-1}(p_z/p)$ by reconstruction package of basf2.
$\Phi_{\text{cluster}}$	Azimuthal angle $\Phi = \cos^{-1}(p_t/p)$ by reconstruction package of basf2
image	25 crystals with detected energies and the center being a local maximum.
M	$(\text{intersection between total energy of found cluster and } E_{\text{MC}})/\text{mcEnergy}$

### 3.1 Simulated data sample

For the CycleGAN simulated electrons and photons were used. The particle gun was used one million times for each particle. In order to have a good quality sample, the simulated data requires passing some basic filters. The following basic filters was used:

- correct electromagnetic charge,
- correct cluster type (having a track or not),
- $E_{\text{MC}} > 0$ .

After filter the simulated data sample accounts to 921821 [92.18 %] simulated electrons and 912313 [91.23 %] simulated photons. The difference between the MC information (see table 2.1) and the basf2 reconstructed cluster information (see table 3.1) for the filtered simulated sample of electrons and photons are shown in figure 3.2.

In the first row of figure 3.2, one can see the behavior of the reconstructed particle energy ( $E_{\text{cluster}}$ ) to the simulated particle energy ( $E_{\text{MC}}$ ). For lower energetic simulated photons ( $E_{\text{MC}} \approx 500$  MeV) the reconstruction seem to over correct ( $E_{\text{cluster}} > E_{\text{MC}}$ ), while at the end of the scale ( $E_{\text{MC}} \approx 5$  GeV) the difference between  $E_{\text{MC}}$  and  $E_{\text{cluster}}$  tends to grow, where some simulated particles get reconstructed with too less and some with too much energy. This effect seems to be more prominent for simulated electrons (upper right), which could be explained through the wider shower shape. The high peaks in the  $\Theta$  distribution (middle row) is due to the gap at the edge of the ECL-barrel. Energy vanishes in those gaps, causing a lack of information for the reconstruction software to estimate  $\Theta$  correctly. There is also a dip around  $\approx 1570$  mrad. This dip is at  $90^\circ$  perpendicular to the interaction point and the beam axis. It does not hit a crystal in the center, but rather hits at a transition between two crystals. The  $\Phi$  distribution (bottom row) is flat for the simulated photons, as well as for the simulated electrons. There also seems to be no biases in the reconstruction of  $\Phi$ .

The ECL crystals in Belle II have an exceptionally good energy absorption efficiency. In figure 3.3 the energy absorption efficiency is displayed for the simulated photons and simulated electrons. The efficiency is M, because it compares the detected energy with the particle energy  $E_{\text{MC}}$ . Lower efficiency for the electrons could be explained through their curved trajectory. Simulated electrons will hit the crystal not orthogonal, but with a smaller incident angle, due to its curved trajectory. The simulated electrons are there-

### 3.1 Simulated data sample

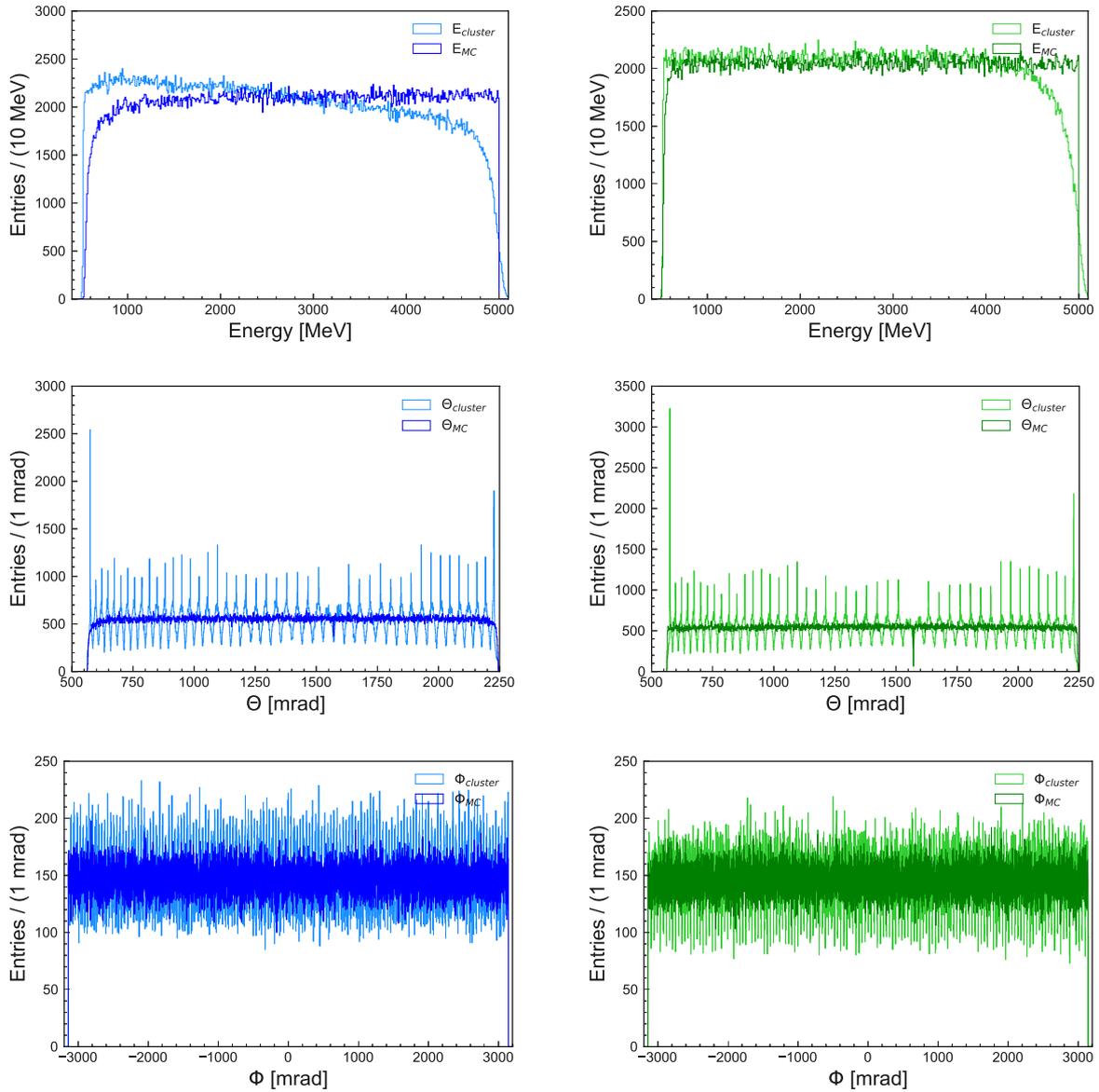


Figure 3.2: Distribution comparison between MC information and the basf2 reconstructed cluster information. Left side is the simulated electrons and the right side is the simulated photons.

fore more likely to distribute their energy over multiple crystals than the photons. Each transition between crystals of the simulated particle will result in a drop in efficiency. In the following section the shower shapes for simulated photons and simulated electrons will be shown.

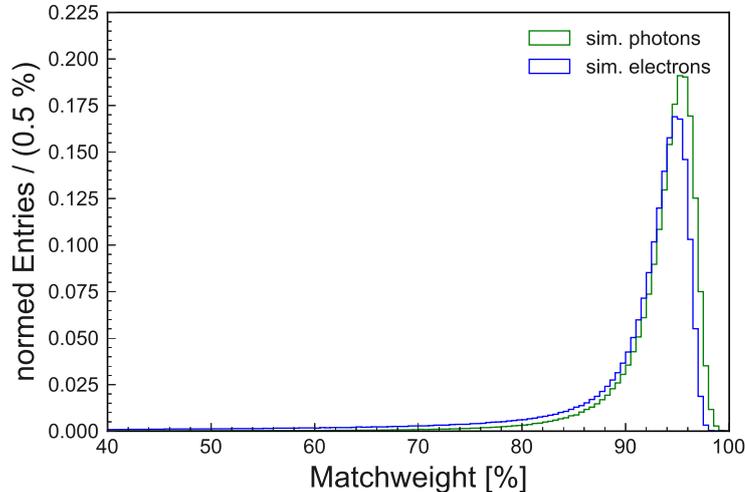


Figure 3.3: ECL efficiency for the simulated photons and simulated electrons.

## 3.2 Shower shapes variables

The propagation of a simulated particle through the ECL and the deposition of energy in some crystals is simulated. Based on the simulated deposition of energy in the crystal the reconstruction package define a cluster, as described in section 2.5. The average image for the simulated photons and for the simulated electrons with an  $E_{MC} < 1$  GeV is shown in figure 3.4. From this figure a slight difference in the  $\Phi$  axis is observable. This is expected due to the negative charge of the simulated electrons and the magnetic field, which curves the trajectories of the simulated electrons along  $\Phi$ . In general, the shower shape between simulated photons and simulated electrons are very close and individually almost not distinguishable. This section will introduce the different shower shape variables. The shower shape variables look at the energy distribution in the image from each simulated particles. Physically speaking, those shower shape variables describe the detection profile of the simulated particle. The shower shape variables are summed up in table 3.2.

In figure 3.5 the distribution for simulated photons and simulated electrons is compared for each shower shape variable. The left side of figure 3.5 is consistent with the first row of figure 3.2. The difference between the simulated photons and simulated electrons is greater in the ratio variables (E1E9, E1E25, E9E25) as seen on the right side. The drop at 45% in the shower shape variables E1E9 and E1E25 (right side, middle

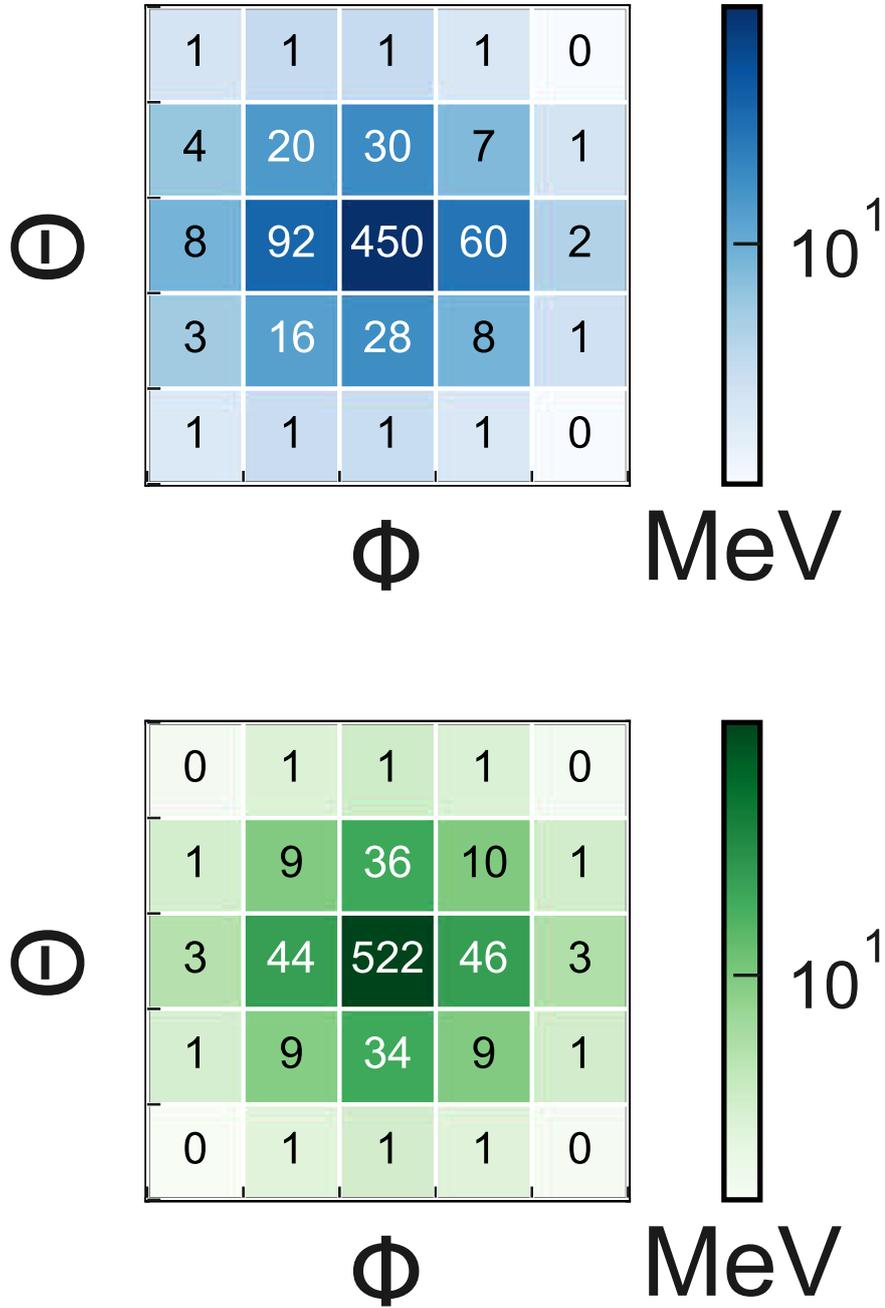


Figure 3.4: Average image of simulated electrons (top) and simulated photons (bottom) with an  $E_{MC}$  below 1 GeV.

Table 3.2: Shower shape variables.

Name	Description
E1	Value of the center pixel
E9	Sum of the inner 9 pixel of the image
E25	Sum of the whole image
E1E25	ratio E1/E25
E1E9	ratio E1/E9
E9E25	ratio E9/E25

and bottom row) is displayed in figure 3.6 and highlights the MC information at which the drop occurs ( $<45\%$ ). This might be due to the simulated particle depositing its energy more to the edge of a crystal. This can be seen for the second plot. One can see the favored  $\Theta$ , which happens to be the  $\Theta$  between the crystals. The same effect can be seen in the polar angle of  $\Phi$  for simulated photons. The stronger curvature in  $\Phi$  evokes a smaller impact angle with the crystal, which further results in a higher probability of energy deposition over more crystals. Respectively the distribution of  $\Phi$  for the simulated electrons are more flat, but this is again due to the stronger curvature effect. The first plot of this figure is expected due to the stronger curvature of lower energetic simulated electrons.

Figure 3.5 only represents the distribution of the simulated particles over the shower shape variables. In order to get a better understanding of the profile of the simulated particles, figure 3.7 is displayed. The plots are in agreement with the described divergent shower shape of simulated electrons and also observed in figure 3.4. The width of the shower shape of the simulated electrons can be seen in the first two lines (left side) of the figure 3.7 by the slightly larger angle between the distribution and the identity line (black line). The distribution from the simulated photons (right side) are slightly closer to the identity line. The effect can be seen more prominent in the last row, where the energy at the outer ring of the image for simulated electrons deposits more energy, than for the simulated photons.

In order to emphasize the outer crystals more, every neural network will not get the MeV for each crystal per image, but the  $\log_{10}$  scale of it. An offset of 1 MeV was added to each pixel to avoid mathematical problems. On the one hand scaling the energy with  $\log_{10}$  will indeed complicate the physical interpretation of the shower shape variables. On the other hand there are a few benefits by using  $\log_{10}$  scale. One benefit is that the central crystal is not as overly dominant in  $\log_{10}$  scale. Another benefit is that this scaling helps stabilizing the training, because the features do not carry that great anymore. The  $\log_{10}$  scaling is therefore an effective normalizing step. The colorcode of figure 3.4 is also in  $\log_{10}$  scale. If the color of the figure would not be in  $\log_{10}$  the image would only be dark in the center and the other pixel would not be visible colored (as seen in the appendix). The shower shape variables for the  $\log_{10}$  scaled simulated

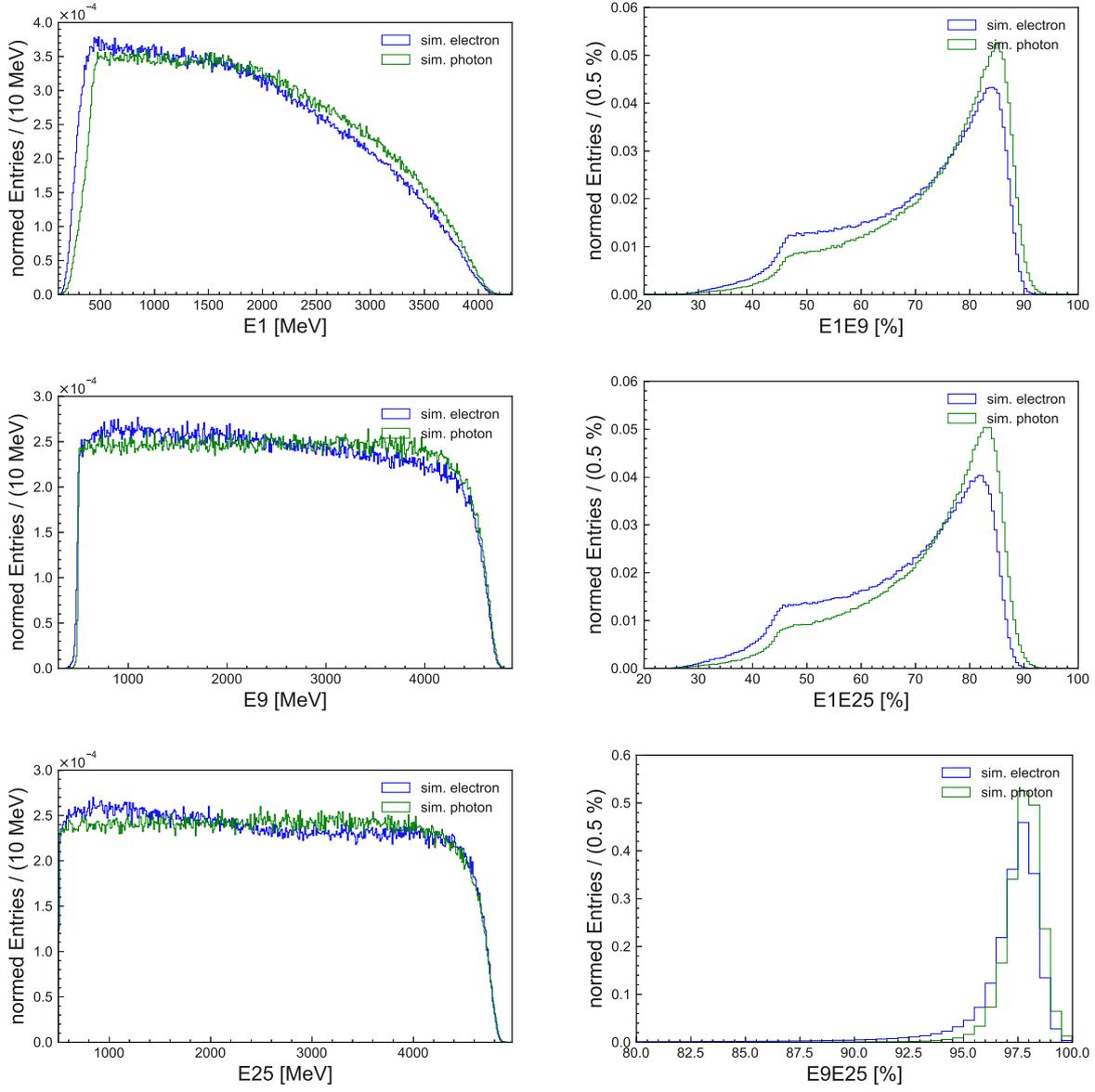


Figure 3.5: Shower shape comparison between simulated photons and simulated electrons.

### 3 Overview: *Simulation in Electromagnetic Calorimeter*

particles is shown in figure 3.8 and its profile in figure 3.9.

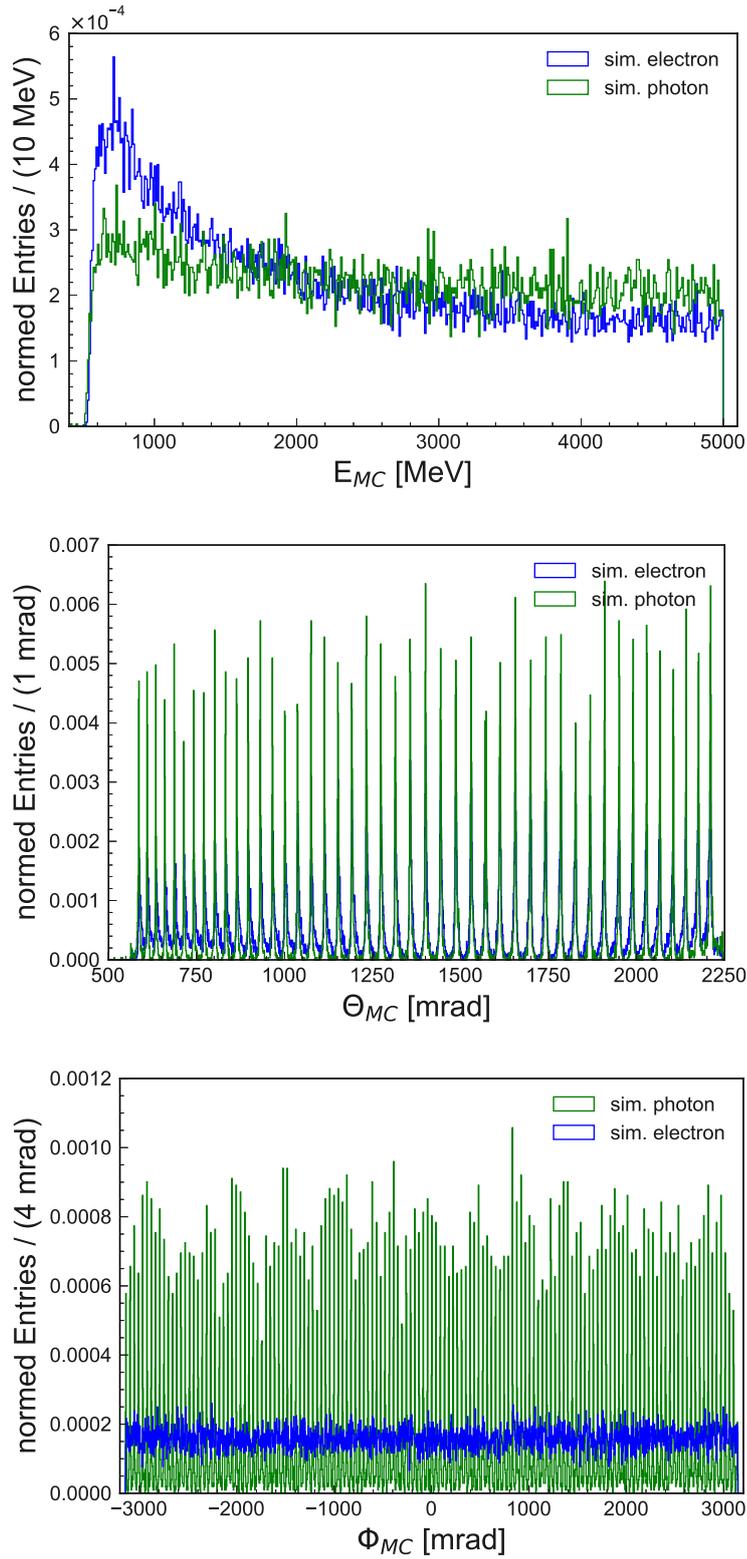


Figure 3.6: MC information of simulated particles with an  $E1E9 \leq 45\%$ .

### 3 Overview: Simulation in Electromagnetic Calorimeter

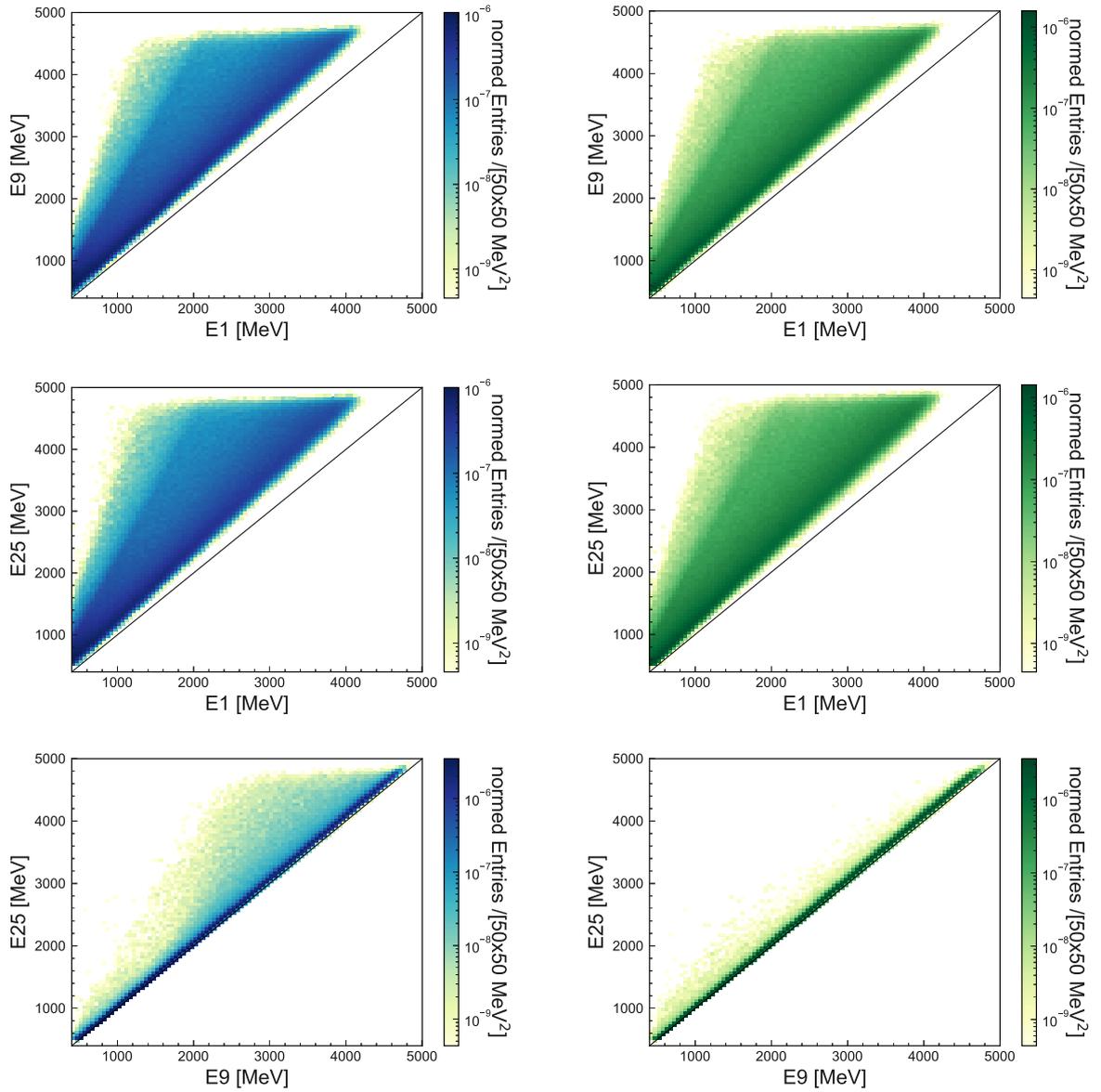


Figure 3.7: 2D Shower shape variables of simulated photons [right side] and simulated electrons [left side]. Black line is the identity line, where the compared shower shape variables are equal.

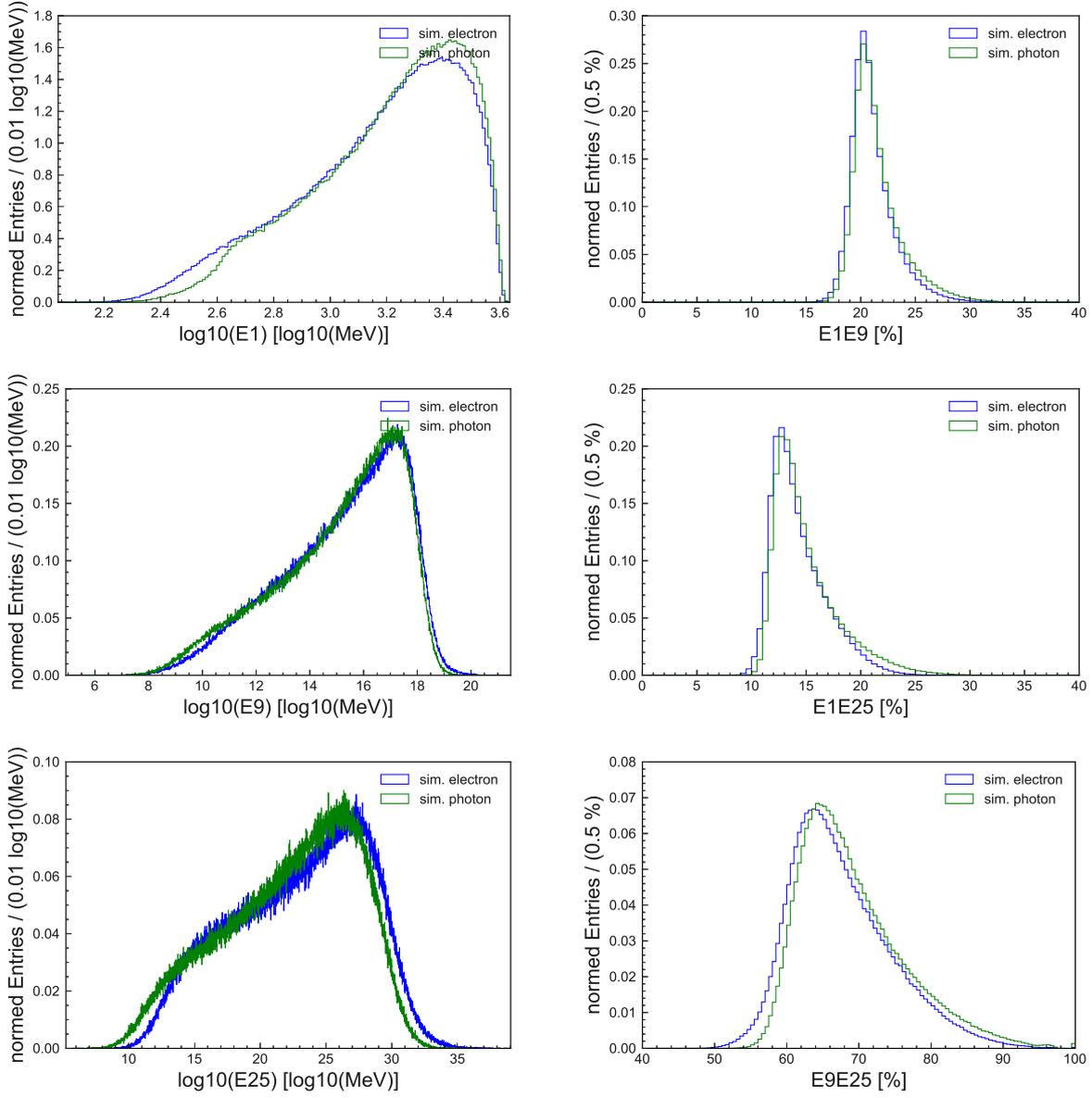


Figure 3.8: Log10 scaled shower shape comparison between simulated photons and simulated electrons.

### 3 Overview: Simulation in Electromagnetic Calorimeter

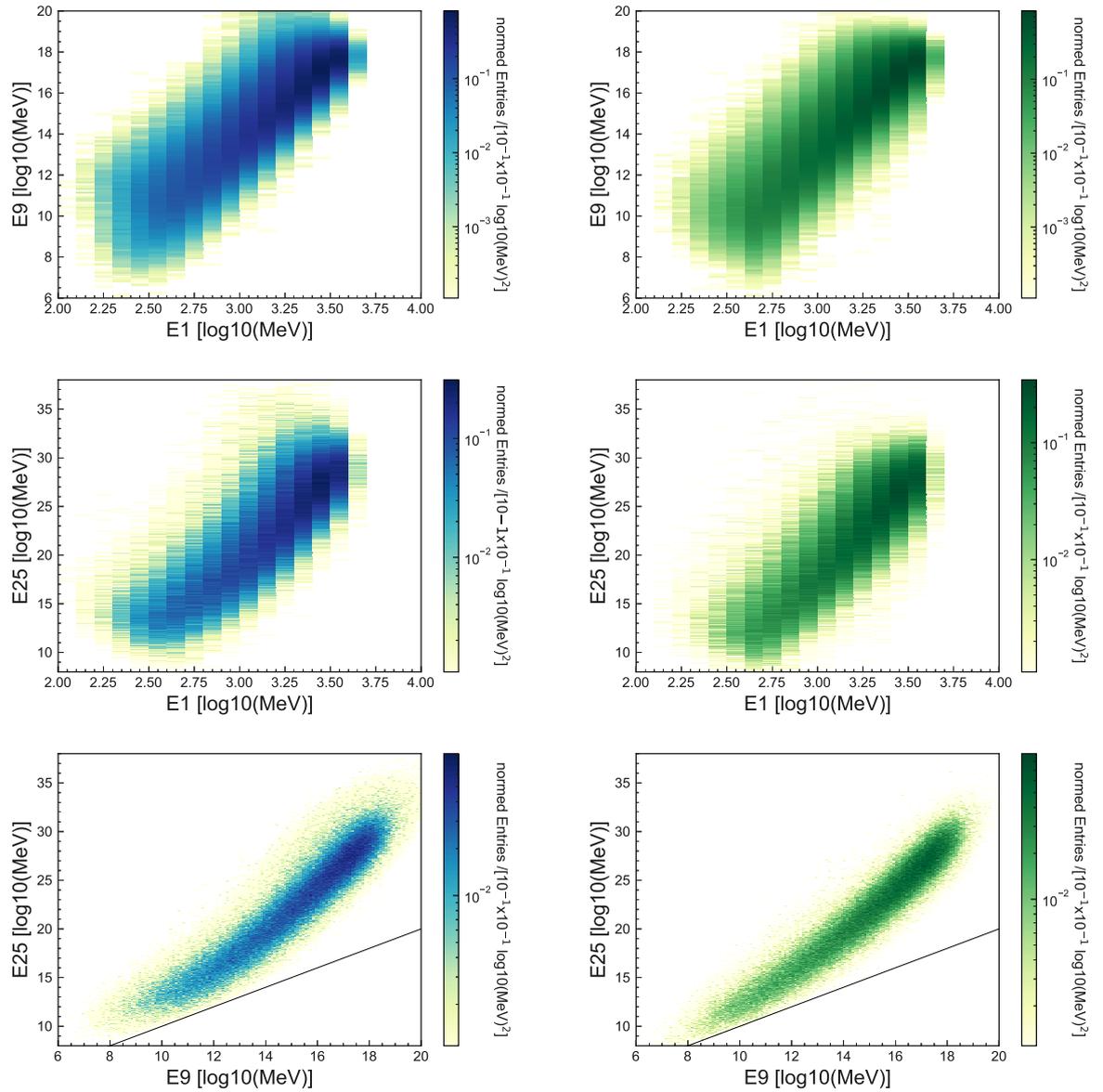


Figure 3.9: 2D log<sub>10</sub> scaled Shower shape variables of simulated photons [right side] and simulated electrons [left side]. Black line is the identity line, where the the compared shower shape variables are equal.

# 4 Cycle Generative Adversarial Network

## 4.1 Introduction Cycle Generative Adversarial Network

The Cycle Generative Adversarial Network (CycleGAN) is a deep neural network architecture, which learns the mapping between two classes. Normally a training set of aligned image pairs is required to solve an image-to-image conversion. The CycleGAN is an approach to solving an image-to-image pair conversion in the absence of an aligned training set [36]. The CycleGAN is a unpaired supervised image-to-image converter. An image-to-image conversion means that the input and output of the conversion are matched and can be counted as one pair. On the one hand, features of the input must change in order to achieve conversion to the target class. On the other hand, these characteristics should change only as much as necessary. The central idea for the future is to use this mapping in order to generate a photon cluster based on a converted true detected electron cluster.

### 4.1.1 Assumptions of this method

The CycleGAN will be trained on basf2 reconstructed data, but the interest lies in applying it on real detected electrons. This central idea is exemplified with faces in figure 4.1. The top row shows the training phase and the bottom row shows the application of the trained CycleGAN to true detected electrons. The drawings of faces will stand for reconstructed particles and the photographs of faces will be real detected particles. We have two classes in the figure, the female face and the male face. The female faces will represent electrons and the male faces photons. In the training phase the CycleGAN trains the conversion exclusively with reconstructed particles. After concluding the training phase, the trained CycleGAN can be tested on real detected particles.

Maintaining the performance of the CycleGAN on real detected particles is based on some assumptions. These assumptions are of pivotal importance, because they ensure the application of the trained CycleGAN on real detected particles. Those two assumptions are.

For one, it is assumed that the mapping from reconstructed electrons to reconstructed photons is the same as the mapping from real detected electrons to real detected photons. For another, the adoption of background from the input to the output is assumed for the

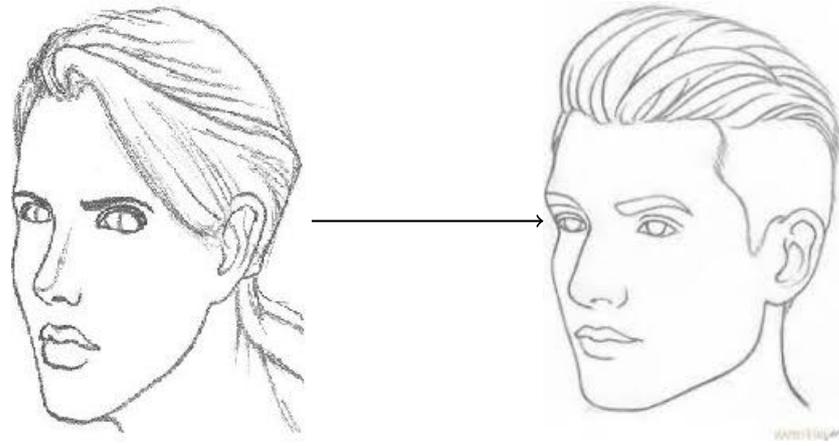


Figure 4.1: The idea of generating GAN particles visualized as an example of faces. Here the two classes are male and female faces. Train for the mapping on drawings of faces and apply a trained CycleGAN to do the mapping on photographs of faces.

Thanks to my sister and "FaceApp" which most likely used a StyleGAN to do the conversion.

CycleGAN. This can be tested by adding some random noise to an image of an electron and apply it on the CycleGAN. If the added noise can be extracted from the generated photon this assumption is solid.

These assumptions are reasonable but was not tested in this thesis. A concept for the development and training of such a CycleGAN is needed prior to the verification of these assumptions legitimizing the use of the CycleGAN on real detected particles. This thesis will only provide these concepts.

### 4.1.2 General architecture

The CycleGAN is a quite complex neural network architecture. There are in total four neural networks, which interact directly or indirectly with each other.

One can divide the CycleGAN into two smaller Generative Adversarial Networks (GANs). Each GAN will manage one direction of the conversion between the reconstructed photons and reconstructed electrons. Each GAN is composed of two neural networks, the **discriminator** and **generator**. The generator and the discriminator pursue adversarial goals. The following paragraphs will depict the objectives of the generator and the discriminator.

**The generator** The generator will get an image ( $\text{image}_{\text{input}}$ ) and the cluster information ( $E_{\text{cluster}}$ ,  $\Theta_{\text{cluster}}$  and  $\Phi_{\text{cluster}}$ ) of a reconstructed particle ( $x \in A$ ) as an input. The generator tries to generate from  $x$  an image ( $\text{image}_{\text{generated}}$ ). Let  $G$  be the generator and  $x$  the input, than ( $G(x) = \tilde{y} \in \tilde{B}$ ) will be called GAN image. The combination of the generator output and analytical calculated cluster information, which takes energy and electromagnetic charge into account, will be called GAN particle. The objective of the generator ( $G$ ) is to generate  $\tilde{y}$ , which fools the discriminator ( $D$ ) to categorized  $\tilde{y}$  as a particle of the target class ( $y \in B$ ). The discriminator is therefore essential for the generator.

$$\text{Loss}_{\text{generator}} = - \text{mean}(\ln(D[G[x]])). \quad (4.1)$$

**The discriminator** The discriminator will get a set. The set comes either from a reconstructed particle ( $y \in B$ ) or from a GAN particle ( $\tilde{y} \in \tilde{B}$ ). The set contains one image plus the cluster information. The cluster information for the GAN particle was calculated analytically. Details about the cluster information of GAN particles are in section 4.3. The discriminator will score the different inputs. Like a common classifier problem, the discriminator tries to separate  $y$  and  $\tilde{y}$ . For such a task it is common to use the binary cross entropy. The set of a simulated particle  $y$ , the  $y_{\text{label}}$  is one and for  $\tilde{y}$  it is zero. The general loss function for binary cross entropy is defined as:

$$\text{Loss}_{\text{BCE}} = -\text{mean}(y_{\text{label}} \cdot \ln(\text{classifier}(y)) + (1-y_{\text{label}}) \cdot \ln(1-\text{classifier}(y))). \quad (4.2)$$

As mentioned before, the CycleGAN consists of two GANs. Each GAN will try to get an equilibrium between its two components. The loss functions for the GANs can be

#### 4 Cycle Generative Adversarial Network

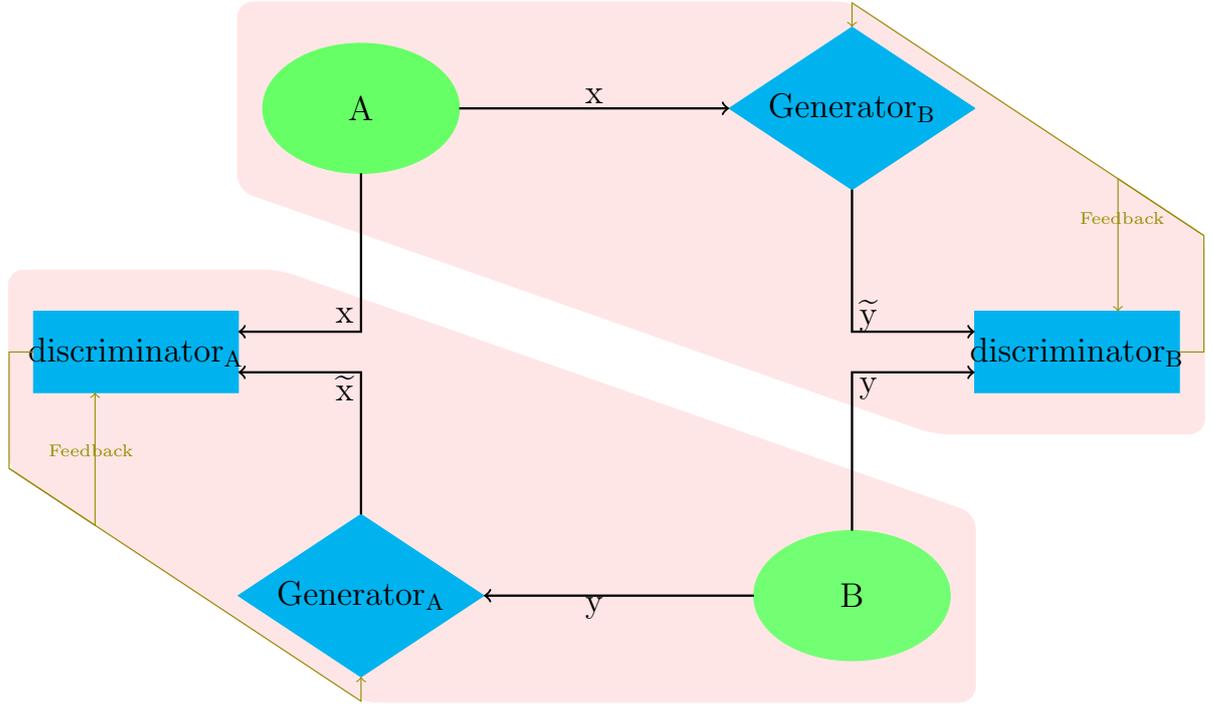


Figure 4.2: Setup of a generic CycleGAN, with  $x \in A$  and  $y \in B$ . Each red box frames one GAN.

summarised as

$$\begin{aligned} L_{\text{GAN}_{\text{electron}}} &= \min_{G_{\text{electron}}} \max_{D_{\text{electron}}} \mathbb{E}[\log D_{\text{electron}}(x)] + \mathbb{E}[\log(1 - D_{\text{electron}}(G_{\text{electron}}(y)))] \\ L_{\text{GAN}_{\text{photon}}} &= \min_{G_{\text{photon}}} \max_{D_{\text{photon}}} \mathbb{E}[\log D_{\text{photon}}(y)] + \mathbb{E}[\log(1 - D_{\text{photon}}(G_{\text{photon}}(x)))] \end{aligned} \quad (4.3)$$

$L_{\text{GAN}_{\text{electron}}}$  will generate images of GAN electrons from reconstructed photons and respectively  $L_{\text{GAN}_{\text{photon}}}$  will generate images of GAN photons from reconstructed electrons. The figure 4.2 shows schematically the setup of the CycleGAN plus the direct interactions between the neural networks.

The loss functions 4.3 with the setup shown in figure 4.2 will not ensure the production of pairwise images, like seen in figure 4.1. There must be an additional constraint that limits the amount of changes that the generators can produce. The constraint responsible for the pairwise results is called the cycle consistency. Cycle consistency is achieved if there is no difference between reconstructed particles  $x$  and cycle particle  $\hat{x}$ . A cycle particle is the transformation of the GAN particle  $\hat{x} = G_A(\tilde{y}) = G_A(G_B(x))$ .

$$\text{Loss}_{\text{Cycle}} = \mathbb{E}[\|G_A(G_B(x)) - x\|_1] + \mathbb{E}[\|G_B(G_A(y)) - y\|_1]. \quad (4.4)$$

Equation 4.4 puts a constraint on both generators. The generators must already satisfy the discriminators and with the addition of cycle consistency, they are even more constrained in their conversion. Figure 4.3 illustrates the cycle consistency.

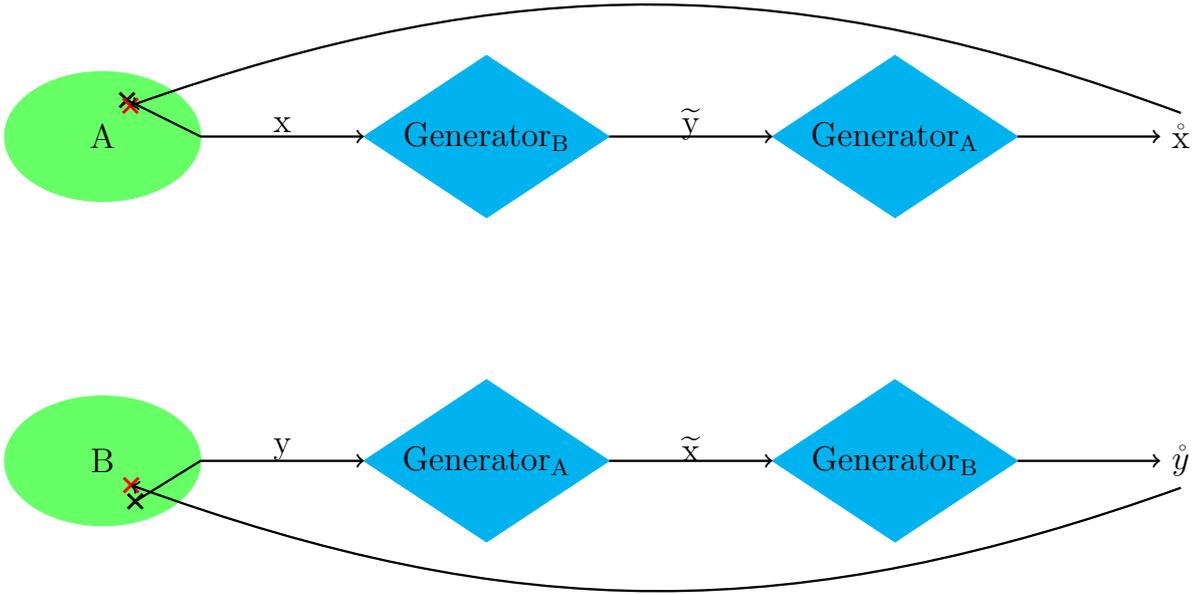


Figure 4.3: A schema for cycle consistency with  $x \in A$  and  $y \in B$ . Red cross is the cycle particle, while the inputted reconstructed particle is the black cross.  $\text{Loss}_{\text{Cycle}}$  tries to get both red crosses as close to the black crosses as possible.

### 4.1.3 Strategy

There are many possibilities for the CycleGAN to fail or to under perform. Reconstructed photons and reconstructed electrons are very similar as seen in figure 3.9. Consequently, there are a many images and cluster information ( $E_{\text{cluster}}$ ,  $\Theta_{\text{cluster}}$  and  $\Phi_{\text{cluster}}$ ), which can stem from reconstruct electrons or reconstructed photons. The similarities between reconstructed electrons and reconstructed photons complicates the task of training and evaluating the CycleGAN. The already complex architecture of a CycleGAN paired with the challenging task of converting between two almost indistinguishable classes increases the difficulty significantly.

Even though the overall motivation of this thesis is to get a calibration sample, the main focus have to be exploring the unknown territory and developing a concept for this approach in the field of high energy physics. Understanding the performance and results of the CycleGAN requires a meticulous approach.

Firstly, one has to answer what can be considerate as success for the CycleGAN and how it can be measured. The next two sections will explain the evaluation metrics and define successful CycleGAN performance. There are two aspects which have to be inspected. Section 4.2 will focus on the conversion and evaluate how good the GAN particles are. A good conversion of a GAN particle is achieved, if the GAN particle is not distinguishable to a reconstructed particle of the targeted class anymore.

Whereas, section 4.3 concentrates on the similarities between the GAN particles and

their respected reconstructed input particles. For the desired GAN performance the output and the input of the conversion needs to be similar. A successful CycleGAN will perform well in both evaluation metrics.

Secondly, the CycleGAN has many hyper parameters, which can be changed and tested. The purpose of the last section of this chapter is to display all hyper parameters used in the training session. It will present the architecture of the discriminators and generators along with their loss functions. All the other potential hyper parameters one could use, will be mentioned in chapter 6. The results of the CycleGAN will be presented in a separate chapter 5.

### 4.2 Pre-trained classifier

CycleGAN's training focuses on reconstructed electrons and reconstructed photons. As can be seen in Figure 3.4, the average values of reconstructed electrons and reconstructed photons are similar. For individual reconstructed particles, it is almost impossible to assign the correct particle by eye. The pre-trained classifier should learn to correctly classify a given reconstructed particle. This is schematically shown in figure 4.4.

The pre-trained classifier is important because it can be used as a tool to benchmark the effectiveness of the conversion. A pre-trained classifier is used to assess the differences between a GAN particle and a reconstructed input particle or respectively the similarity between GAN particle and reconstructed particle of the target class. Let  $\tilde{y}=G(x)$ , with  $G$  being the generator and  $x$  the input of the reconstructed particle. The idea is to check whether the pre-trained classifier will categorize the GAN particle as one particle of the target class. A successful conversion will fool the pre-trained classifier, so it classifies  $\tilde{y}$  as a particle of class B.

In summary, these are the steps:

1. Build a classifier to distinguish reconstructed photons from reconstructed electrons,
2. Train classifier,
3. Check classifier performance,
4. Use classifier to evaluate conversion effectiveness of CycleGAN output.

The pre-trained classifier is not to be equated with the particle identification of basf2. The pre-trained classifier is restricted by the image and the cluster information. The particle identification of basf2 however takes additional information from other sub detectors into account. Track information from CDC is for example essential to distinguish an electron cluster from a photon cluster. In addition to the importance of the pre-trained classifier as an evaluation metric, the pre-trained classifier also gives insight into the task of converting reconstructed electron image into reconstructed photon image. If there is no classifier able to distinguish between the reconstructed electron image from reconstructed photon image, with their respected cluster information, the conversion between those two reconstructed particles will not be necessary. If the pre-trained classifier can

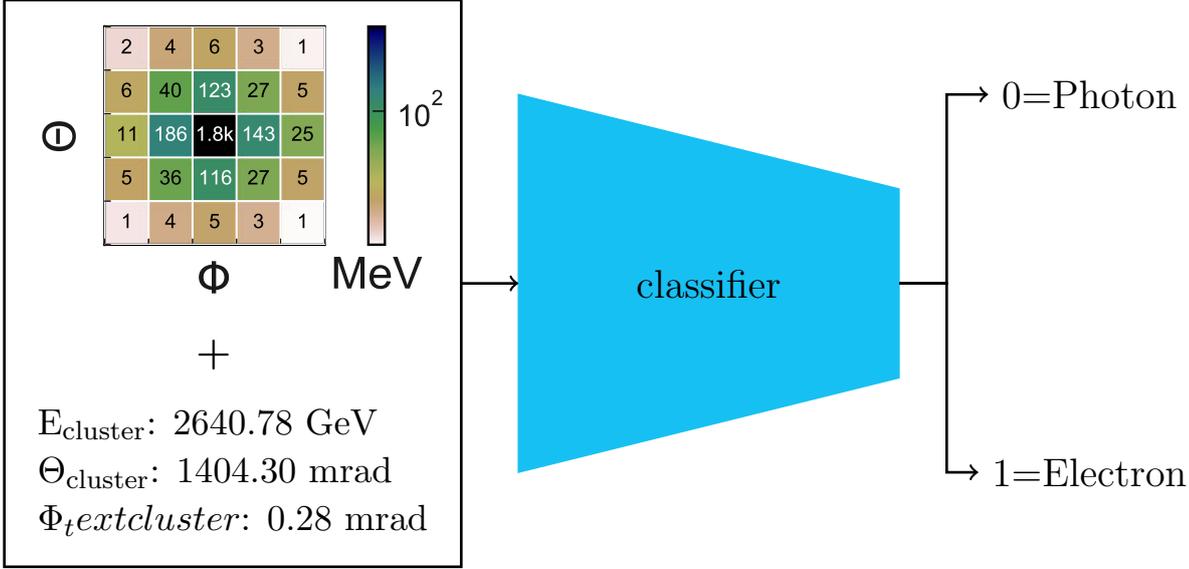


Figure 4.4: Task of the pre-trained classifier.

separate the reconstructed particles perfectly, the conversion will consequently be easy to observe. The next subsection tests a set of classifier. The best classifier will be crowned as the pre-trained classifier.

### 4.2.1 Classifier and training parameters

The classifier will get as input a pair of an image of a reconstructed particles and its cluster information ( $E_{\text{cluster}}$ ,  $\Theta_{\text{cluster}}$  and  $\Phi_{\text{cluster}}$ ). The combination of cluster information with the image will forward inside the classifier. The classifier will output a score within the range of  $[0,1]$ . The goal of the classifier is to output zero, if the input stems from a reconstructed photon and to output one if the input stems from a reconstructed electron.

The loss function for the classifier was already described at equation 4.2. In the case of a classifier which can not distinguish between reconstructed photons and reconstructed electrons the output will always be 0.5. In this case  $\text{Loss}_{\text{BCE}}$  will be  $\approx 0.7$ .

The classification between reconstructed photons and reconstructed electrons is particularly difficult due to their similarities. Therefore, it is expected that a neural network with only a few hidden layers will not be sufficient. This chapter presents a number of neural networks. Every neural network will have the same set of input data. The input is composed of random 80% of the total reconstructed particles introduced in subsection 3.1. The remaining 20% will be used to calculate the validation loss.

The first classifier starts with a basic model and all the following models will build on it. Each additional classifier will be more sophisticated in its approach. The image of the reconstructed input particle are 25 energy values in  $\log_{10}$  scale. So they range from  $[0,3.4]$  instead from  $[1,5000]$ . The  $E_{\text{cluster}}$  information is on the other hand still in MeV

Table 4.1: Number of neurons for each layer in basic model.

28	128	256	128	64	32	1
input	hidden layers					output

and  $\Phi_{\text{cluster}}$  and  $\Theta_{\text{cluster}}$  are both in mrad.

**Basic model** The basic model is a simple model which takes the image and the cluster information raw and forward it to five hidden layers. An activation function of LeakyReLU and a dropout of 0.25 were used between the hidden layers. Between the input layer and the first hidden layer, the same activation function and dropout occurs. The number of neurons for each layer is summarized in table 4.1. The output layer uses a Sigmoid function in order to restrict the value from zero to one.

**Basic meta model** The basic meta model extends the basic model, by applying some preprocessing steps to the input. The preprocessing steps try to solve two issues. For one it addresses the scaling issue between cluster information and the image values and for another it tries to extract information from the edge of the image.

The scaling issue is explained by the different ranges within one input. On the one hand the value of the image ranges from  $[0,3.4]$  and on the other hand the cluster information can range from  $[-2000\pi,2000\pi]$  for  $\Phi_{\text{cluster}}$  or from  $[500 \text{ MeV}, 5000 \text{ MeV}]$  in the case of  $E_{\text{cluster}}$ .

In order to counteract the scaling issue, one has to scale down the cluster information. The concept of an embedding layer is applied for each cluster information for this purpose. An embedding layer takes one label and map the label to a tensor with a predefined size. The basic meta model will have a preprocessing step, where the cluster information will forward through one or two layers. The output will replace the cluster information as input. In this process an activation function of Tanh is used. This activation function will cause the tensor representation to adopt smaller values.  $E_{\text{cluster}}$  will be passed into one layer with four neurons and an activation layer of Tanh. The variable  $E_{\text{cluster}}$  will therefore be represented by a tensor with the size of four.  $\Phi_{\text{cluster}}$  and  $\Theta_{\text{cluster}}$  will be passed to another layer and will be forwarded to a layer with 16 neurons. Tanh is the activation function. These 16 neurons will be further passed forward to a layer with four neurons. The combination of  $\Phi_{\text{cluster}}$  and  $\Theta_{\text{cluster}}$  will now be represented by another tensor of four neurons. Tanh was chosen, because it is a smooth function ranging from  $[-1,1]$ . The eight neurons from the preprocessing step will now be passed to the basic model instead of the raw cluster information.

The advantages of this preprocessing step on the cluster information are, on the one hand, the associated scaling effects and, on the other hand, the additional parameters that depend exclusively on the cluster information.

The representation of the image border is interesting for the distinction between reconstructed electrons and reconstructed photons. The main difference between the shower

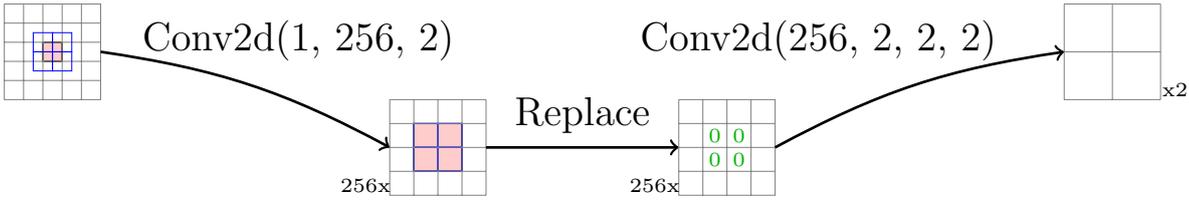


Figure 4.5: Constructing inputs to represent edge of the image.

Table 4.2: Number of neurons for each layer in basic\_meta model.

25+4+4+8	128	256	128	64	32	1
input	hidden layers				output	

shapes and images of the reconstructed particles are not located at the center of the image as seen in figure 3.4. Like the idea of appending a representational tensor for the cluster information to the input, a representational tensor for the image edge will be created by another pre-processing step. There are three steps to get a representation of the outer pixels of the image.

1. Using a convolutional layer with 256 filters and a kernel size of 2 on the 5x5 image.
2. Replacing the middle of the output from before with zero. These neurons contain the center pixel, which may dominate.
3. Using a convolutional layer on the output with 2 filters, a kernel size of 2 and a stride of 2.

Those steps will leave a 2 times 2x2 outputs. The flatten output will have 8 neurons which can be appended to the input. The steps are schematically shown in figure 4.5. This extension will be called the basic\_meta model, because it tries to improve every meta variables, which includes the cluster information and the edge of the image. After the two preprocessing steps for the cluster information and for the image border, the process is the same as for the original basic model. They only differ with 41 neurons in the input layer instead of 28 (see table 4.2).

**Resnet model** The last extension is already an improvement, which is shown in the next subsection. However due to the difficult task an even more sophisticated neural network is required. A more sophisticated model would be a model with more hidden layers. In order to have a deeper neural network, which is also stable in its training, a technique described in [37] is used. In principle this technique reintroduces the input after some layers, by adding the neurons values, before propagating further in the neural network. The layers between the input and the reintroduced input is a residual block. The purpose of a residual block is to change the input in a way, which makes it easier to distinguish between reconstructed photons from reconstructed electrons.

The input for this model will be the same as for the basic\_meta model. The resnet model

Table 4.3: Number of neurons for each layer in residual block.

41	128	256	128	64	41
input	hidden layers				output

will inherit the preprocessing steps from basic\_meta model . 8 neurons representing the cluster information, 8 neurons representing the image edge and the 25 values of the image will again be the input. The setup for a residual block is summarized in table 4.3. Like the basic model the residual block have LeakyReLU as activation function between the layers and a dropout of 0.25. The integration of the residual block in the whole resnet model is shown in figure 4.6.

**Training parameters** Every model was trained at least with 200 epochs. The optimizer was Adam with an initial learning rate of 0.000002. The learning rate decreases with an factor of 0.8 at selected epochs. The selected epochs are 50,100,150 and 175. In a second step the resnet model was further trained to 500 epochs because the learning curve was still in decline. The learning rate further decayed with the same factor at epoch 250, 350, 400 and 450.

## 4.2.2 Results

The three models, basic model, basic\_meta model and the resnet model follows the same calculation for its loss value (see equation 4.2). The loss value should decrease over the epochs and the loss over 200 epochs for the three models is shown in figure 4.7.

The black line serves as a benchmark line. If the loss value is below the black line the model starts to differentiate between reconstructed photons from reconstructed electrons correctly. It is noticeable, that the basic model is not sufficient enough. It seems to converge to the black line. A convergence to the black line would mean that in average the model does not predict the wrong label but also not the right one. In average the basic model outputs 0.5.

The extension model basic\_meta model is a significant improvement in comparison to the basic model. Except for the first few epochs the loss value is always below the black line and the curve seems to further decrease if trained longer.

The resnet model is a considerable amount better than the basic\_meta model. This may support the hypothesis that the task is difficult and needed a more sophisticated model in order to produce better results. The promising resnet model was further trained to 500 epochs and the loss between the training and validation set of reconstructed particles was plotted on each other in figure 4.8.

For such a task the case of overtraining has to be checked. Overtraining occurs if the model does not have to predict the outcome anymore, because it instead remembers the label by the input. An indication for overtraining would be a decreasing loss value for the training sample, but an increasing loss value for the validation sample. Overtraining

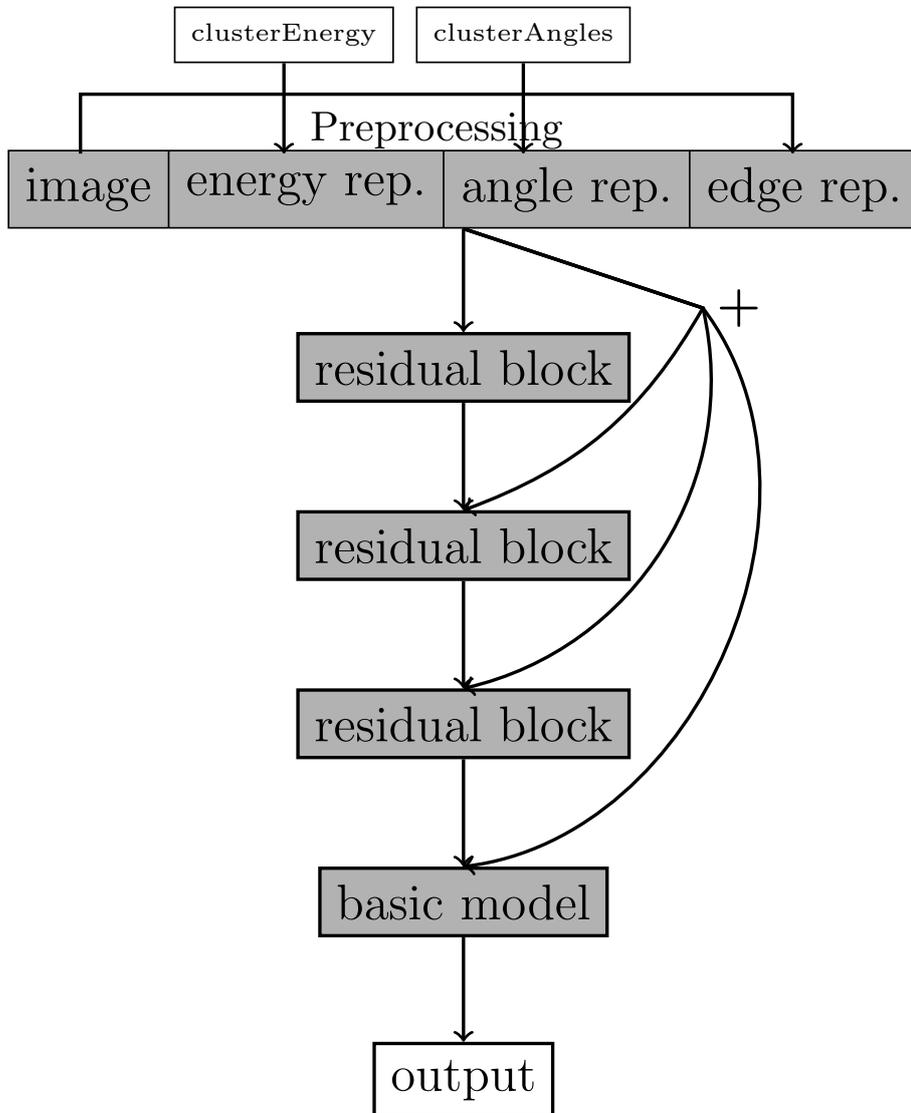


Figure 4.6: Total setup of the resnet model.

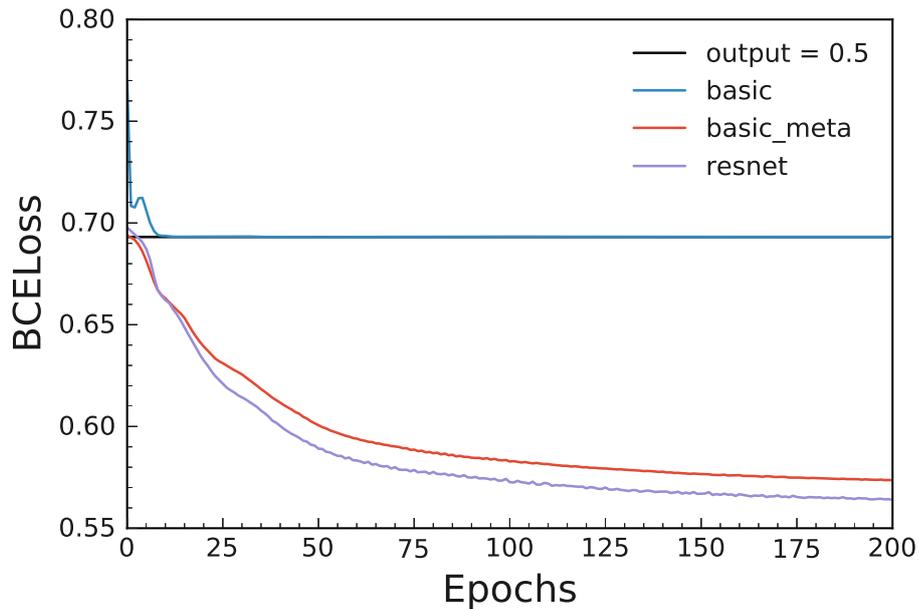


Figure 4.7: Comparison of different models for the classifier task. The black line is a benchmark and is the loss value if models always outputs 0.5.

was not observed for the resnet model.

After 500 epochs the resnet model will be referred in the rest of the thesis as the pre-trained classifier. To have a better understanding of the pre-trained classifier performance the output of 500000 reconstructed photons and 500000 reconstructed electrons will be plotted in figure 4.9.

A separation of reconstructed photons and reconstructed electrons is realized. There are some reconstructed electrons which the pre-trained classifier can categorized with confident. This is realized as a narrow peak at 1. Around 0.5 there are the reconstructed electrons which the pre-trained classifier could not assign the particle to. There is no narrow peak at 0. This indicate that the pre-trained classifier did not found a photon specific feature.

The pre-trained classifier works better for some particles. In order to understand the performance of the pre-trained classifier better, reconstructed electrons and reconstructed photons will be studied separately. The reconstructed particles are divided into unique and ambiguous reconstructed particles. The unique reconstructed electrons have a pre-trained classifier score higher or equal to 0.8. Consequently the ambiguous reconstructed electrons have a pre-trained classifier score lower than 0.8. The same is true for the reconstructed photons, with unique reconstructed photons having a pre-trained classifier value less than or equal to 0.2. Figure 4.10 tries to visualize and compare the cluster information profile between unique and ambiguous reconstructed particles with respect to  $E_{\text{cluster}}$ .

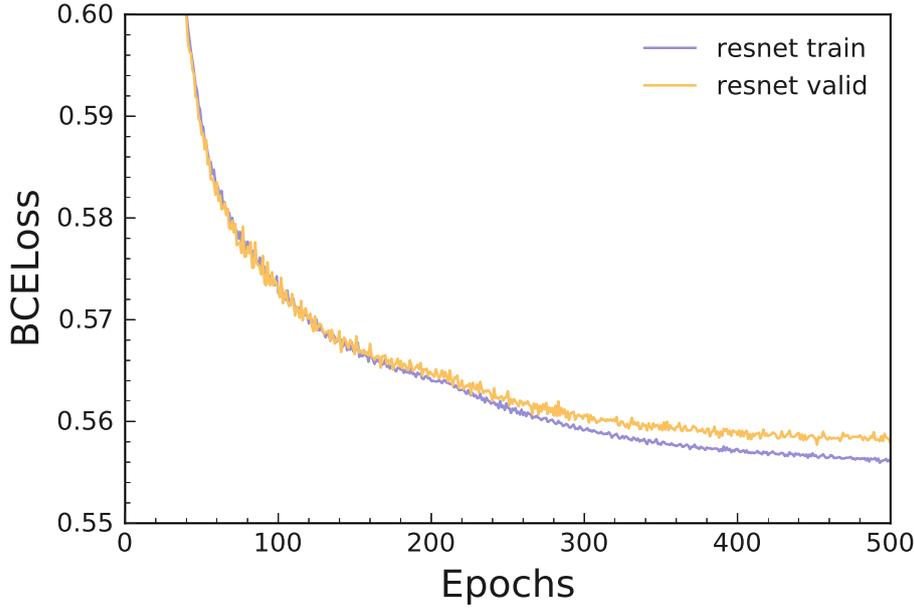


Figure 4.8: Loss value for the resnet model on training and validation sample. The black line is a benchmark and is the loss value if models always outputs 0.5.

Unique reconstructed particles are predominantly low in  $E_{\text{cluster}}$ . This behavior is also expected, considering a high energetic electron will effectively propagate straight like a photon (see figure 4.10).

Figure 4.11 compares polar angle profile between unique and ambiguous particles. The top row of the figure contains only the unique reconstructed particles and the bottom row contains only the ambiguous reconstructed particles. If a peak exists only in the top row and not in the bottom row, the pre-trained classifier will have a better performance for this particular combination of  $\Phi_{\text{cluster}}$  and  $\Theta_{\text{cluster}}$ . In this case no such peak is observed for reconstructed electrons (left side) or reconstructed photons (right side). This suggests a performance independence of  $\Phi_{\text{cluster}}$  and  $\Theta_{\text{cluster}}$ . The distribution over  $\Phi_{\text{cluster}}$  and  $\Theta_{\text{cluster}}$  is for the unique and ambiguous reconstructed particles uniformly. The comparison between the top and bottom row does not reveal anything.

Improvement regarding the pre-trained classifier will be discussed in the last chapter 6.

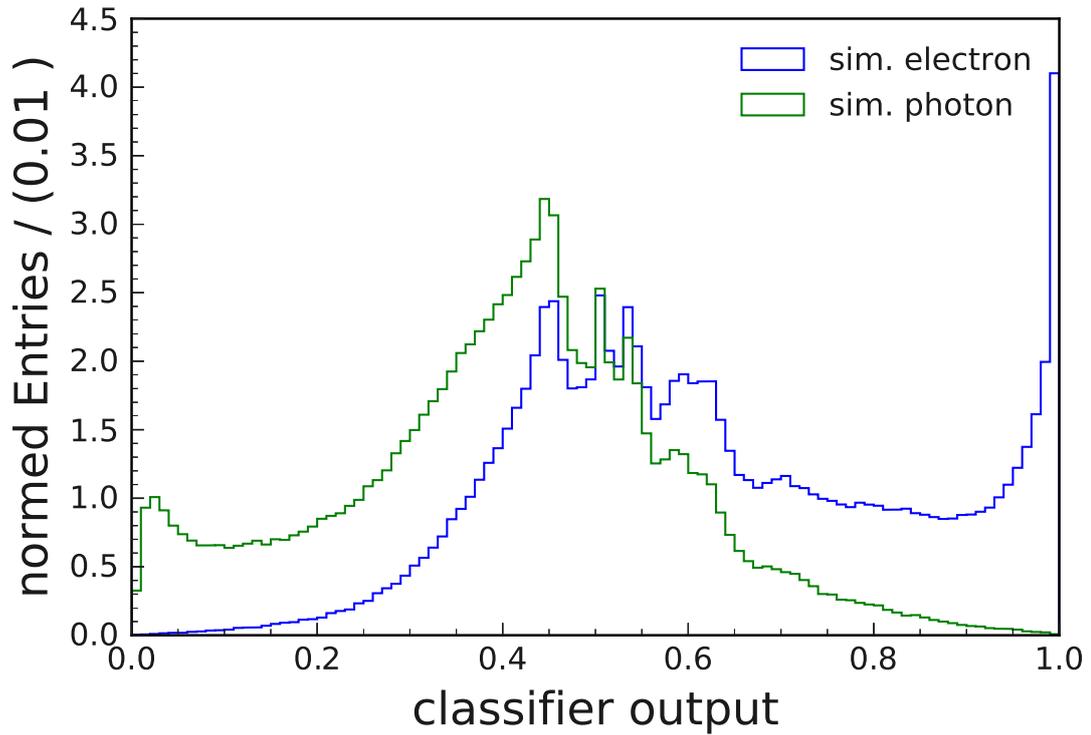


Figure 4.9: Pretrained classifier output for 1 000 000 reconstructed photons and reconstructed electrons.

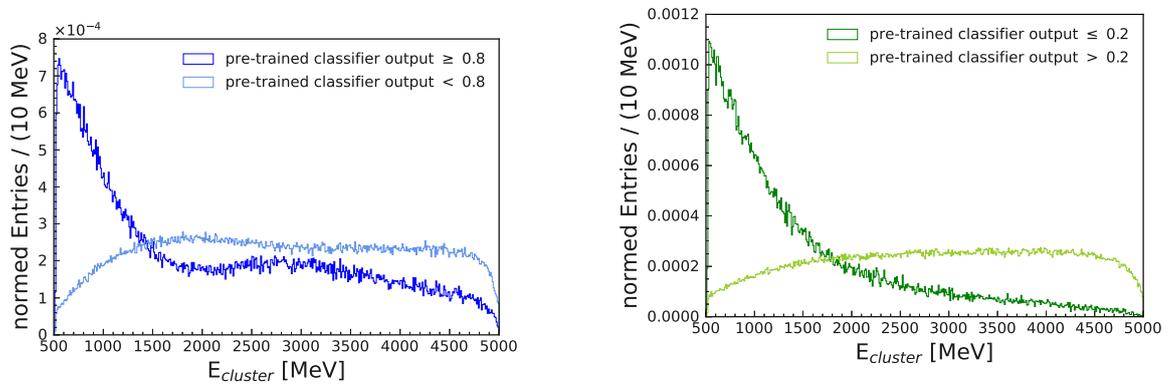


Figure 4.10: Left side are the reconstructed electrons and the right side are the reconstructed photons. Each plot compares unique (dark colored histogram) and ambiguous (light colored histogram) reconstructed particles. The ambiguous reconstructed particles are the reconstructed photons with a pre-trained classifier score higher than 0.2 and for ambiguous electron the score is lower than 0.8.

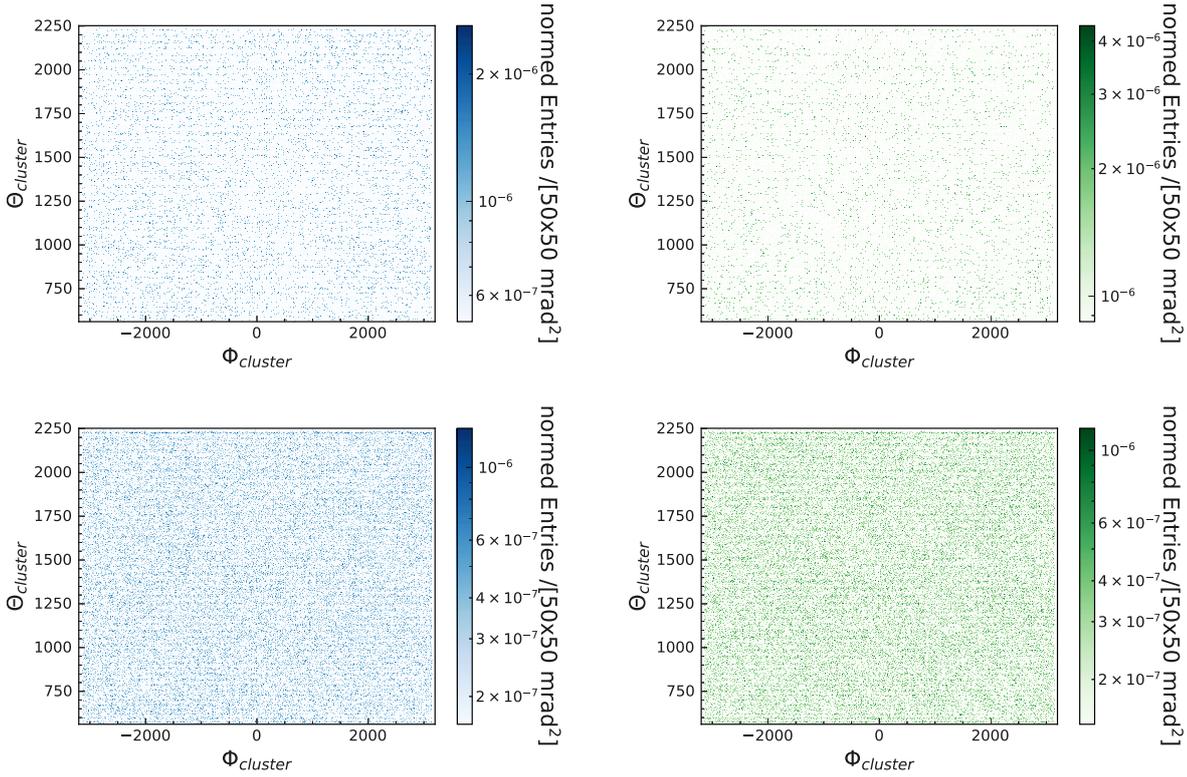


Figure 4.11: Left side are the reconstructed electrons and the right side are the reconstructed photons. First row are the unique reconstructed particles and the bottom row contains the ambiguous reconstructed particles. The ambiguous reconstructed particles are the reconstructed photons with a pre-trained classifier score higher than 0.2 and for ambiguous electron the score is lower than 0.8.

### 4.3 Pairwise metric

In the last section 4.2 a way to evaluate the effectiveness of the conversion was proposed and discussed. According to subsection 4.1.1, it is obvious that a conversion satisfying the pre-trained classifier will not be sufficient. The similarities between the input and the output of the two generators have to be measured.

The pairwise metric tries to evaluate how far the output differs from the input. The objection is to have a penalty on a conversion of a 4 GeV reconstructed particle into a 500 MeV GAN particle, even if the GAN particle would satisfy the pre-trained classifier in terms of particle transformation.

The main constraint on the output for both generators, is the use of cycle consistency within the CycleGAN (see figure 4.3). In addition, the cluster information of GAN particles is constrained by the cluster information of the respective reconstructed input particles. A reconstructed photon and reconstructed electron with the identical simulated variable (see table 2.1), will have the same cluster information, except for  $\Phi_{cluster}$ .

$E_{\text{cluster}}$  and  $\Theta_{\text{cluster}}$  of the GAN particle is therefore, the same as the  $E_{\text{cluster}}$  and  $\Theta_{\text{cluster}}$  of the respected reconstructed input particle. The difference in  $\Phi_{\text{cluster}}$  of a reconstructed input particle and the  $\Phi_{\text{cluster}}$  of the respected GAN particle is explained by the different electromagnetic charge and the setup of the magnetic field in Belle II. The combination of the magnetic field ( $\mathbf{B}$ ) and the different electromagnetic charges ( $q$ ) has to obey the Lorentz force

$$\mathbf{F}_{\text{Lorentz}} = q \mathbf{E} + q \mathbf{v} \times \mathbf{B}, \quad (4.5)$$

with  $\mathbf{E}$  as the electric field and  $\mathbf{v}$  as the velocity of the particle. Theoretically, the magnetic field in Belle II is such that it runs along the beam axis and acts exclusively on  $\Phi$ . The study of the altered  $\Phi_{\text{cluster}}$  due to the altered electromagnetic charge is described in the next section 4.3.1.

### 4.3.1 GAN $\Phi_{\text{cluster}}$

As mentioned above, both  $E_{\text{cluster}}$  and  $\Theta_{\text{cluster}}$  are not affected by the magnetic field. In figure 4.12, a comparison is drawn between MC information and cluster information of reconstructed particles. In the figure there is no difference between  $E_{\text{cluster}}$  and  $E_{\text{MC}}$ . The same goes for  $\Theta_{\text{cluster}}$  and  $\Theta_{\text{MC}}$ .

The only noticeable difference in the figure is the discrepancy between  $\Theta_{\text{cluster}}$  and  $\Theta_{\text{MC}}$  for an  $E_{\text{cluster}}$  smaller than 1 GeV. The discrepancies in this range are larger for reconstructed electrons than for reconstructed photons. This is to be expected since the low energetic reconstructed electrons are particularly prone to strike the crystals at an unfavorable angle. The bad angle have a negative effect on the  $\Theta$  reconstruction. This difference is overall still quite small and will be ignored for this thesis.

More interesting is the comparison of reconstructed photons with reconstructed electrons in terms of  $\Phi_{\text{cluster}} - \Phi_{\text{MC}}$ . Figure 4.13 shows the same plots as in Figure 4.12, only with  $\Phi_{\text{MC}}$  and  $\Phi_{\text{cluster}}$ , revealing the curved trajectory for the simulated electrons.

The  $\Phi_{\text{cluster}}$  of GAN particles must follow the distribution in the figure. For GAN photons this means that the inputted reconstructed electrons are straightened in  $\Phi$  and vice versa for inputted reconstructed photons the GAN electrons have to be diffracted.

This distribution is dependent on the energy of the particle and its electromagnetic charge (see equation 4.5). A new variable will be needed in order to calculate  $\Phi_{\text{cluster}}$  of a GAN particle.

Due to the different ( $\Phi_{\text{cluster}} - \Phi_{\text{MC}}$ ) distribution for the different reconstructed particles, a "theoretical  $\Phi_{\text{cluster}}$ " is needed. The "theoretical  $\Phi_{\text{cluster}}$ " is the  $\Phi_{\text{cluster}}$  after taking the changed electromagnetic charge and the MC information into account.

A "theoretical  $\Phi_{\text{cluster}}$ " variable was created and will be referred to as  $\Phi_{\text{helix}}$  in the following. A schematic depiction of  $\Phi_{\text{helix}}$  is shown in figure 4.14 for the GAN electron by a reconstructed photon. For the GAN photons  $\Phi_{\text{helix}}$  will be handled respectively. The validity of  $\Phi_{\text{helix}}$  is discussed below. The creation of the new variable fulfill the sole

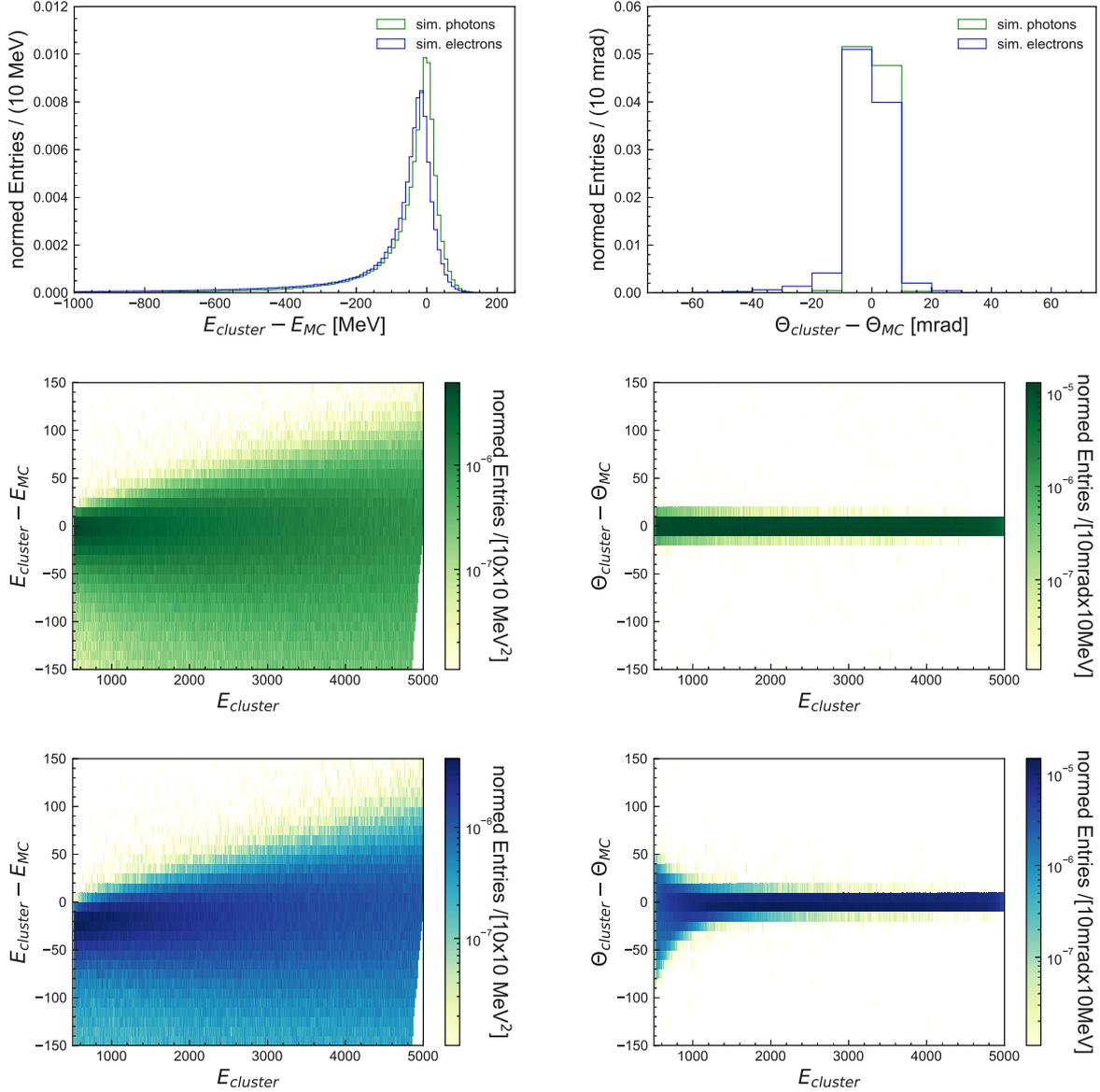


Figure 4.12: Comparison between reconstructed photons and reconstructed electrons regarding their cluster information  $E_{cluster}$  and  $\Theta_{cluster}$ . Left side are the differences in  $E_{cluster}$  and  $E_{MC}$ . Right side are the differences in  $\Theta_{cluster}$  and  $\Theta_{MC}$ .

purpose of constraining  $\Phi_{generated}$ . The distribution of  $(\Phi_{helix} - \Phi_{MC})$  has to mimic the distribution of  $(\Phi_{cluster} - \Phi_{MC})$  from the targeted particle.

A comparison will be done by matching the distribution of  $(\Phi_{helix} - \Phi_{MC})$  of reconstructed particle A with the distribution of  $(\Phi_{cluster} - \Phi_{MC})$  of reconstructed particle B. This comparison is shown in figure 4.15. As seen in this figure the distribution  $(\Phi_{helix} - \Phi_{MC})$  of the reconstructed photons resembles the distribution  $(\Phi_{cluster} - \Phi_{MC})$  of the recon-

#### 4 Cycle Generative Adversarial Network

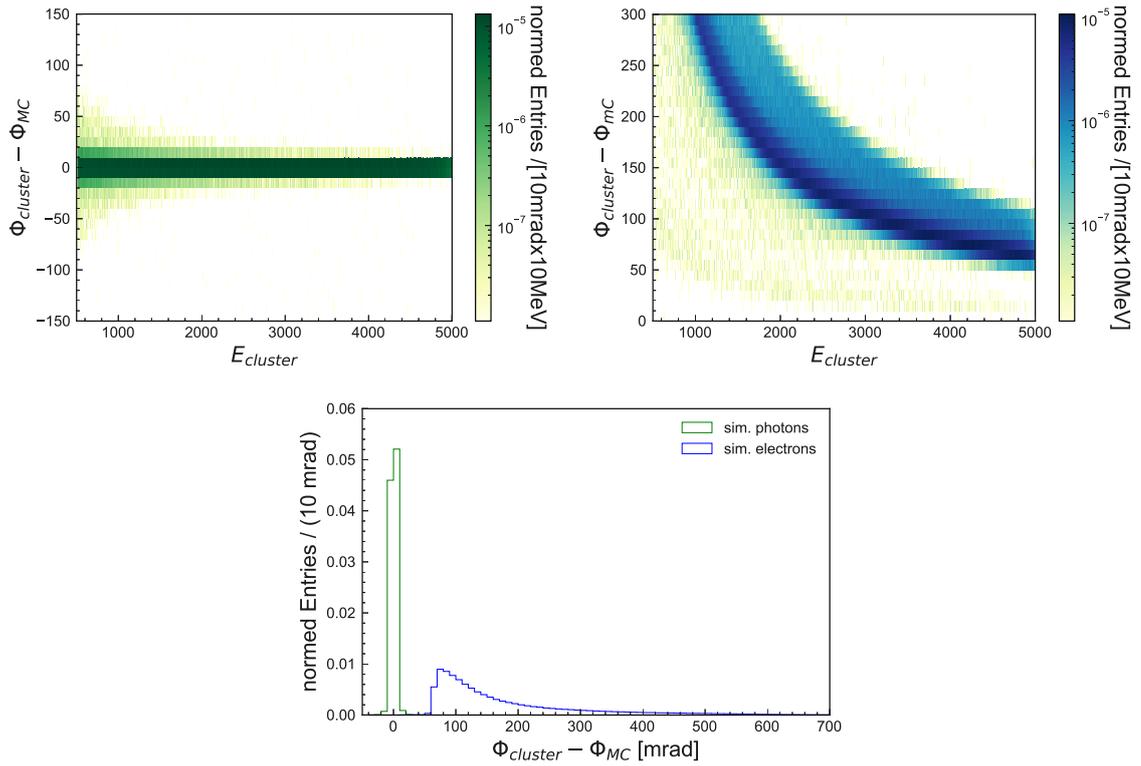


Figure 4.13: Comparison between reconstructed photons and reconstructed electrons regarding their  $\Phi_{cluster}$  and  $\Phi_{MC}$ .

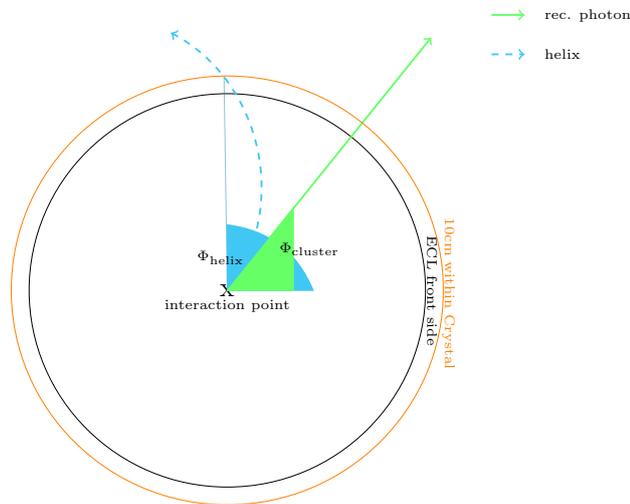


Figure 4.14: Schema of the  $\Phi_{helix}$  variable. The green straight is a reconstructed photon and the blue dashed would be the same reconstructed photon with the electromagnetic charge of an electron.

structed electrons. The same goes for the distribution ( $\Phi_{helix} - \Phi_{MC}$ ) of the reconstructed

electrons with the distribution  $(\Phi_{\text{cluster}} - \Phi_{\text{MC}})$  of the reconstructed photons.

The figure 4.16 summarizes it by mapping the centroid values for each bin and for all distributions to it. For each bin on the  $E_{\text{cluster}}$  axis in this plot, the median  $\Delta\Phi$  value was plotted. All  $(\Phi_{\text{helix}} - \Phi_{\text{MC}})$  distributions are almost identical to its  $(\Phi_{\text{cluster}} - \Phi_{\text{MC}})$  distribution. The only noteworthy difference between those distributions are that the  $(\Phi_{\text{cluster}} - \Phi_{\text{MC}})$  distributions are a bit wider. This was expected because  $\Phi_{\text{cluster}}$  itself has a limited resolution, while  $\Phi_{\text{helix}}$  variable is a theoretical value and does not concern with resolution problems. The resolution problem mainly stems from a low inefficiency in the energy deposition of the reconstructed particle energies. In the case of reconstructed electrons, minimal effects of bremsstrahlung amplify the resolution problem. Overall those small discrepancies are manageable and the distributions  $(\Phi_{\text{helix}} - \Phi_{\text{MC}})$  are satisfyingly close enough.

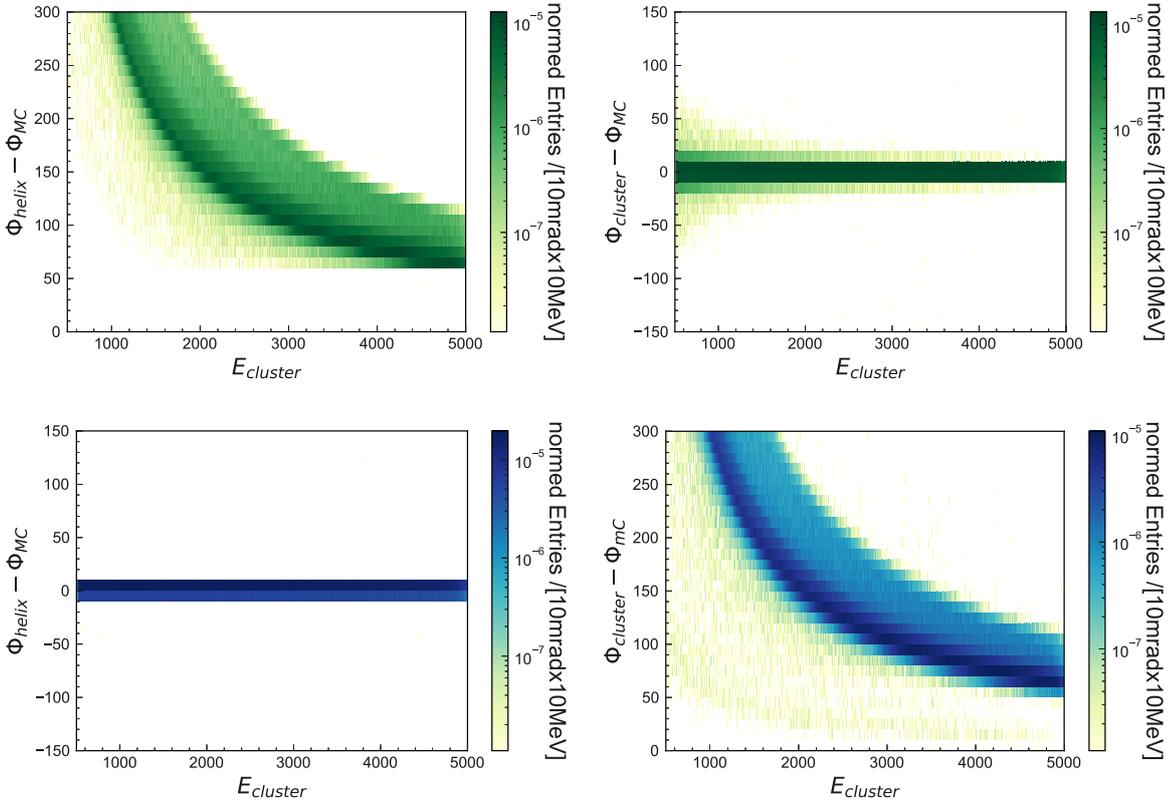


Figure 4.15: Comparison between  $(\Phi_{\text{helix}} - \Phi_{\text{MC}})$  (left side) and  $(\Phi_{\text{cluster}} - \Phi_{\text{MC}})$  (right side) in dependence to their  $E_{\text{cluster}}$ . Reconstructed photons are in the top row and the reconstructed electrons are in the bottom row.

### 4.3.2 Constraining term for generator

The previous subsection explains why you don't need a constraint on the cluster information for the GAN particles. The only mechanism to ensure the pairwise relation

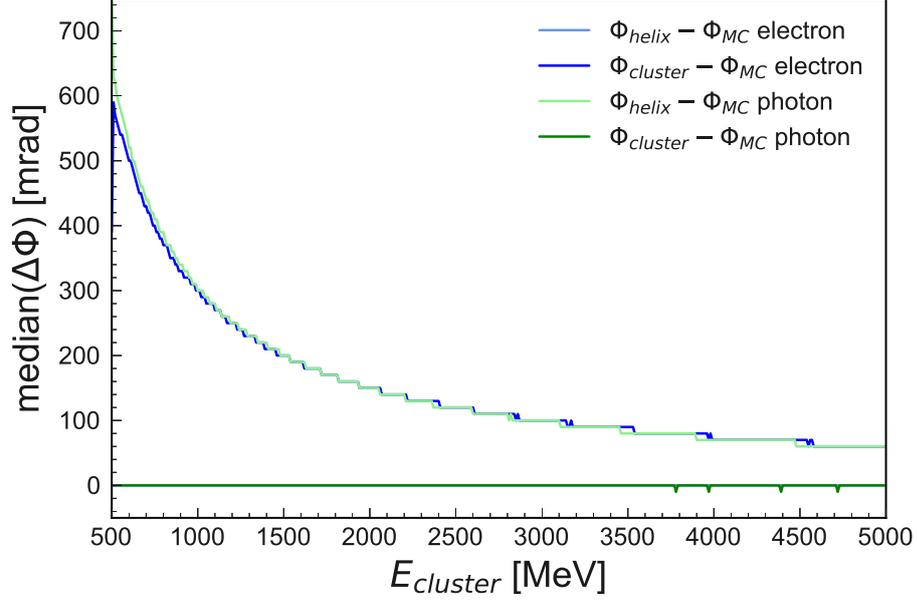


Figure 4.16: Plot shows direct comparison of median  $\Delta\Phi$  for  $(\Phi_{\text{cluster}} - \Phi_{\text{MC}})$  and  $(\Phi_{\text{helix}} - \Phi_{\text{MC}})$ .

is the cycle consistency term 4.4, which will be added into the loss for the generators in the CycleGAN 4.1. To guide the generator in the right direction, the loss term for the generator contains another term. The additional term penalizes a GAN image if the total sum of its values in MeV differs from the total sum of the input particle in MeV. Effectively this term calculates the difference between the E25 variable in MeV of the GAN particle with the E25 variable in MeV to the reconstructed particle. It is common practice to use the mean squared loss for the calculation of the difference. The additional loss term will look like this:

$$\text{Loss}_{\text{E25,A}} = \frac{1}{N} \sum_i \left( \sum_j^{25} 10^{G_A(x_i)_j} - \sum_j^{25} 10^{x_{i,\text{image},j}} \right)^2. \quad (4.6)$$

$N$  is the batch size and  $i$  iterates over all reconstructed particle ( $x_i$ ), while  $j$  iterates over all pixel in the image.  $G$  is the generator and  $x_{i,\text{image}}$  is only the 25 pixel values from the image part of the reconstructed input particle.  $x_{i,\text{image},j}$  is the  $j$ .th pixel of  $x_{i,\text{image}}$ .  $G(x_i)_j$  is the  $j$ .th pixel of the GAN image. This term does not constrain the shape of the shower, but it does constrain the total energy. Therefore, it is very unlikely that the generator will output an image with a total sum of 4 GeV from a reconstructed input particle with a total sum of 2 GeV.

### 4.3.3 Measuring closeness

The measuring of the similarity between reconstructed input particle and the GAN particle, will be done by comparing the shower variables. Even if the generator convert the image, the changes is expected to be minimal. Therefore, if the differences over all shower shapes are normal distributed around zero this is an indicator for a pairwise image-to-image conversion. The width of the normal distribution can be interpreted as uncertainty regarding the conversion by the generator.

The conversion by the CycleGAN is counted as a success if 2 things between reconstructed input particle and GAN particle are true. First, the pre-trained classifier value must change between input and output, and second, the difference must be normally distributed around zero across all shower shape variables.

## 4.4 Trainingparameters and models

The blueprint for a CycleGAN is shown in figure 4.2. This section will introduce the details of the CycleGAN used. Understanding the CycleGAN means understanding all the parts that make up the CycleGAN.

The parts are two discriminator neural networks and two generator neural networks. Both discriminator will be the same and both generators will be the same in their setup. This reduces the task of presenting the details of CycleGAN to presenting the critic model and the generator model in detail.

**Inputs and outputs** The inputs and outputs of the discriminators and generators will be discussed first. The objective of the discriminators are to distinguish between reconstructed particles and GAN particles. In the best case the generated particles will resembles the reconstructed particles. Ideally, the GAN particles are closer to the target class than the reconstructed input particles. This means that the discriminator must be at least as sophisticated as the pre-trained classifier, if not more so. Like the pretrained classifier the discriminator scores ( $D(x) \in \mathbb{R}$ ) the input ( $x$ ). The score of the input, which is the output of the discriminator, tries to minimize the BCELoss according to equation 4.2.

For each reconstructed particle the discriminator gets an image and the cluster information, exactly like the input for the pretrained classifier 4.4. The input for one discriminator can also be the combination of the output of one generator as an image and the cluster information for the GAN particle. In the section 4.3, the cluster information for a GAN particle was discussed.

The input for the generators will be the same as for the discriminator. One give the generator an image and the cluster information of a reconstructed particle and it tries to generate an image of the target class from it. As mentioned multiple times, the image between an reconstructed electron and reconstructed photon can be almost identical, so on the one hand the generator will not have to change much. On the other hand,

#### 4 Cycle Generative Adversarial Network

section 4.2 suggested the existence of a difference between reconstructed photons and reconstructed electrons even if the difference is subtle. The generators must recognize the subtle difference and transform the input accordingly. The value of the pixel of the given input image is not in MeV. The generator changes the log10 scaled values by adding for each pixel a value ( $\varepsilon$ ) between -1 and 1. This will influence the original energy value by a factor between 0.1 and 10, as shown in equation 4.7

$$\begin{aligned}\tilde{y} &= \tilde{x} + \varepsilon \\ y &= x \cdot 10^\varepsilon,\end{aligned}\tag{4.7}$$

whre  $x$  is the original value in MeV and  $y$  is the value after the generator conversion in MeV.  $\tilde{y}$  and  $\tilde{x}$  are the respected log10 scaled value of  $y$  and  $x$ . The energy difference ( $y-x$ ) is dependent on the  $\varepsilon$  which is given by the generator. The effect of  $\varepsilon$  in MeV scale is not linear as displayed in figure 4.17.

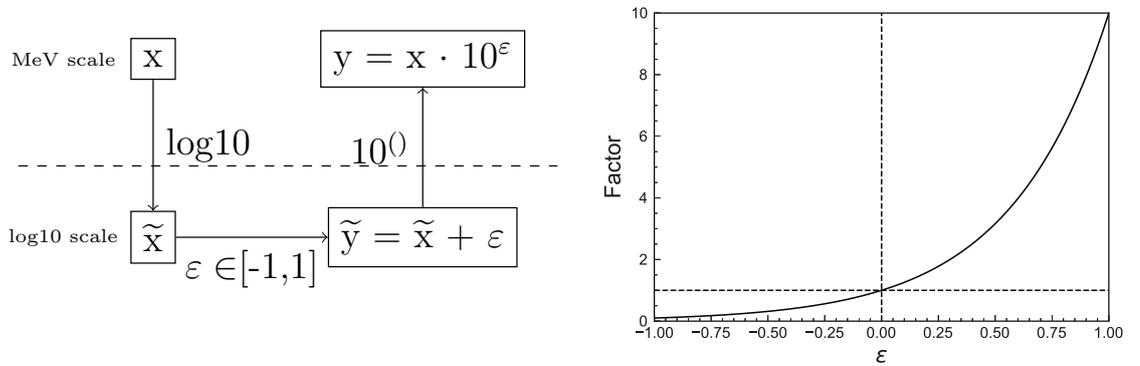


Figure 4.17: Flowchart of effective changes ( $\varepsilon$ ) done by the generator (left side).  $x$  is the original energy detected in the pixel. Factor to multiply pixel value in MeV dependent of generator produced  $\varepsilon \in [-1,1]$  (right plot).

**Loss function** The loss function for the discriminator is unchanged from equation 4.2, while the generator loss function get more help with additional constraints of equation 4.6. The cycle loss also has to be taken into account. The generator tries to minimize equation 4.4 additionally. So the total loss function to minimize can be written as

$$\begin{aligned}\text{Loss}_{\text{Total}} &= \text{L}_{\text{GAN}_{\text{electron}}} + \text{L}_{\text{GAN}_{\text{photon}}} \\ &+ \text{Loss}_{\text{E25,electron}} + \text{Loss}_{\text{E25,photon}} + \text{Loss}_{\text{cycle}}\end{aligned}\tag{4.8}$$

**The architecture** The architecture of the discriminator is the same as that of the resnet model used for the pre-trained classifier. The generators will have to process 28 values and it outputs a GAN image.

The architecture of the generator can be divided into two modules. The first module

is the preprocessing module and the second module is the generator main module. In the image generation the preprocessing step is used, which is described in paragraph basic\_meta model of subsection 4.2.1. As a reminder the preprocessing steps represent the cluster information ( $E_{\text{cluster}}$ ,  $\Theta_{\text{cluster}}$  and  $\Phi_{\text{cluster}}$ ) in a total of 8 neurons instead of three, along with 8 more neurons representing the border of the image as seen in figure 4.5.

The preprocessing module for the generator will do the preprocessing steps. The obtained 16 neurons will be concatenated with the 25 values of the image of the reconstructed input particle. The total 41 neurons will be passed to a layer with 16 neurons with the activation function LeakyReLU. This layer is the first and only hidden layer for the generator main module. Following the hidden layer the generator main module continues, with an output layer of 25 neurons. The activation function used here is Tanh. The output of this module will be called ( $\text{image}_{\text{alteration}}$ ). The purpose of the module is to alter the reconstructed input image, in order to fulfill its conversion.  $\text{Image}_{\text{alteration}}$  will be added to the reconstructed input image. Each 25 values of the output will alter one value of the  $\text{image}_{\text{input}}$ , in order to generate the GAN image =  $\text{image}_{\text{alteration}} + \text{image}_{\text{input}}$ . By using Tanh as an activation function, each value of the reconstructed input image, can increase or decrease by 1. The effect of  $\pm 1$  corresponds to either an increase or decrease by the factor of 10 in MeV scale for each pixel of the input image. The scale of the alteration seems reasonable, due to the already close images of reconstructed electrons and reconstructed photons.

**Training parameter** will start with an learning rate of 0.0001 and this learning rate will decay with an factor of 0.8 on the epochs [500,100,1250]. The initial run will be 1500 epochs long and the optimizer used for the discriminator and generators will be Adam.



# 5 Results

In this section, the trained CycleGAN is presented. It will begin with the different developments of the various loss function terms (see equation 4.8). The different development will be discussed in detail later in the following section 5.1. The GAN particle and the cycle particle will also be compared to the reconstructed input particle.

After the following subsection there will be two sections to evaluate the results of the trained CycleGAN. There will be a section 5.2 and 5.3) dedicated to each evaluation metric described in section 4.2 and 4.3. This chapter concludes with section 5.4. This chapter is dedicated to the question whether the CycleGAN was a success or not.

## 5.1 Training evolution

Figure 5.1 shows the performance of the two discriminators over the epochs. The black line is again a benchmark for orientation. At the black line value, the discriminator cannot distinguish between reconstructed and GAN particles and outputs 0.5. For both discriminators, the loss value decreases over the epochs. The figure also indicates a better discrimination performance of the discriminator for photons.

Figure 5.2 describes the evolution of the discriminator on the GAN particles. The goal

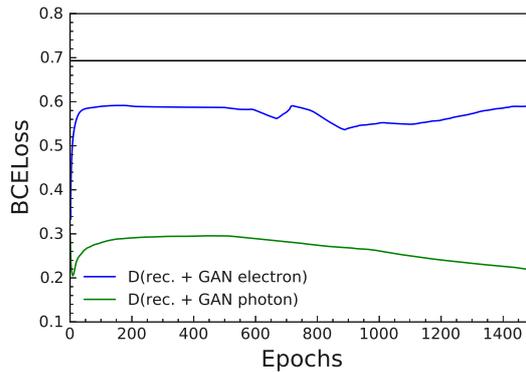


Figure 5.1: Binary-cross entropy loss of both discriminators for reconstructed and GAN particles during training.

of the generator is to produce images for the GAN particles that are recognized by the discriminator as reconstructed particles. For this figure, the black line is equivalent to

## 5 Results

saying that the generator produces images for the GAN particles that the discriminator cannot categorize as either reconstructed or GAN generated. If the line were to drop to zero, it would mean that the discriminator would categorize the GAN particle as a reconstructed particle. Ideally, the line in this figure would slope down to the black line and then oscillate around it. This scenario would suggest that the generator has gotten better, but also that the discriminator is finding ways to separate the GAN particles from the reconstructed particles. These new ways of separation by the discriminator give the generator new opportunities to improve. In this figure, the line first drops towards the black line, but does not reach it. The loss value increases for both discriminator afterwards, suggesting the discriminator outperform the generator.

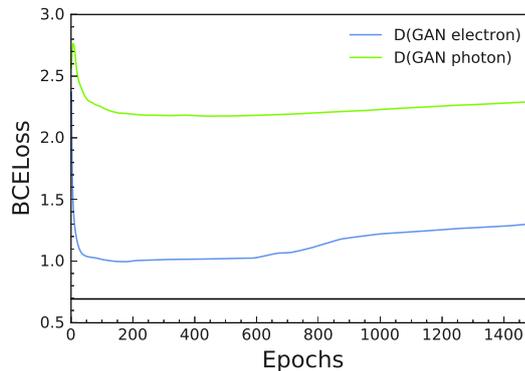


Figure 5.2: Binary-cross entropy loss of both discriminators for GAN particles during training.

For both figures the discriminators are getting better at distinguishing between reconstructed and GAN particles. there are many possibilities what this mean for the generator performance. One possibility would be that the generator performance got worse over the epochs. Another possibility would also be that the discriminators improved faster than the generators. To evaluate the performance of the generators over the epochs, the additional loss term for the generator is shown in figure 5.3.

The overall results of the CycleGAN is best shown in figure 5.4 and 5.5. In these figures, some random reconstructed particles were taken and the corresponding GAN particles and the corresponding cycle particles were shown. The cycle consistency term in the generator loss function constrains the generators by forcing them to produce cycle particles that match the reconstructed input particles as closely as possible. The  $Loss_{E25}$  term (see equation 4.6), will restrict that each generator output will not differs too much to the total energy of the reconstructed input image.

In general the images does not differ much in E25 and the cycle particle are quite close to the reconstructed input particle. At first sight there are no conspicuities in the figures 5.4 and 5.5. Figure 5.6 shows the change for each pixel in the image that the generator makes. The green distribution shows the values added to the image pixels of the original

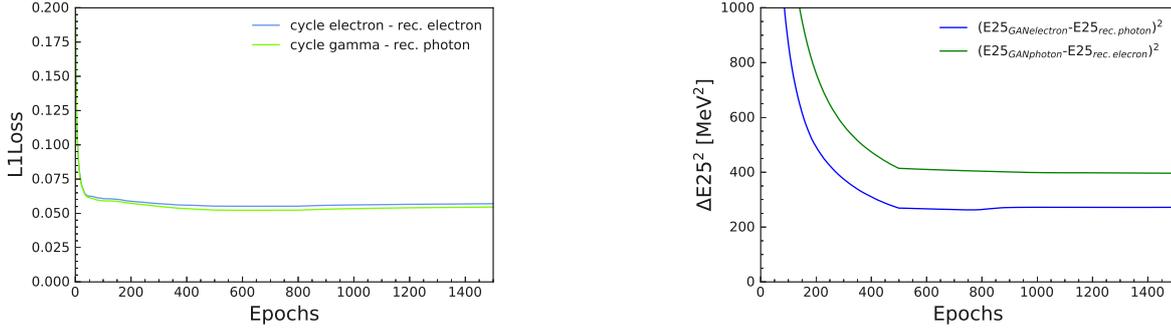


Figure 5.3: The cycle consistency loss over the epochs on the left side. The MSE loss of the total sum between GAN and reconstructed input particle image in  $\text{MeV}^2$  (right).

reconstructed input electron to obtain the image for the GAN photon. Respectively, the blue distribution shows the values added to the image pixels of the original reconstructed input photon to obtain the image for the GAN electron. The black line marks value zero. No changes are made to the pixel at the black line and this pixel neither gains nor loses energies. These changes are additive to the  $\log_{10}$  scaled energy. Meaning an additive change of 0.2 will increase the energy by a factor of  $10^{0.2} \approx 1.58$ . The changes can only be done between -1 and 1 restricting the effect on the energies by a factor between 0.1 and 10, as described in equation 4.7. The same figure is rescaled on the x-axis and describes the effective factorial changes between GAN and reconstructed input particle image in the appendix.

Figure 5.6 is a good way to understand both generators and their performance. Theoretically, the distributions for each pixel must lie on opposite sides of the black line. Otherwise, the cycle particles will not be mapped back to their original value. The pixel in the second row and first column is problematic because both distributions are mainly on the right side of the black line. This right shift is also in other pixels but it is not as extreme as for the pixel in the second row and first column.

The pixel in the third row and second column matches with what was expected. In this pixel the blue distribution is on the right side and the green distribution is on the left side. The green distribution is closer to the black line on the left side than the blue distribution, which leads to another problem. The different distances between the distribution and the black line on each side of the black line are expected due to the asymmetric derivative of  $f(x) = 10^x$  as seen in figure 4.17.

There is another layer to interpret the figure. For the whole image the total energy should stay more or less the same. In the generator loss function, an additional loss term  $\text{Loss}_{E_{25,A}}$  4.6 is implemented to ensure this. If a distribution for one pixel is on the right side of the black line, it must consequently be on the left side for another pixel. This will shift the energy of the reconstructed input particle image. Otherwise, this would mean that the generator adds or remove energy from the image. For the blue distribution, it seems to add more energy into the images for conversion. With the green

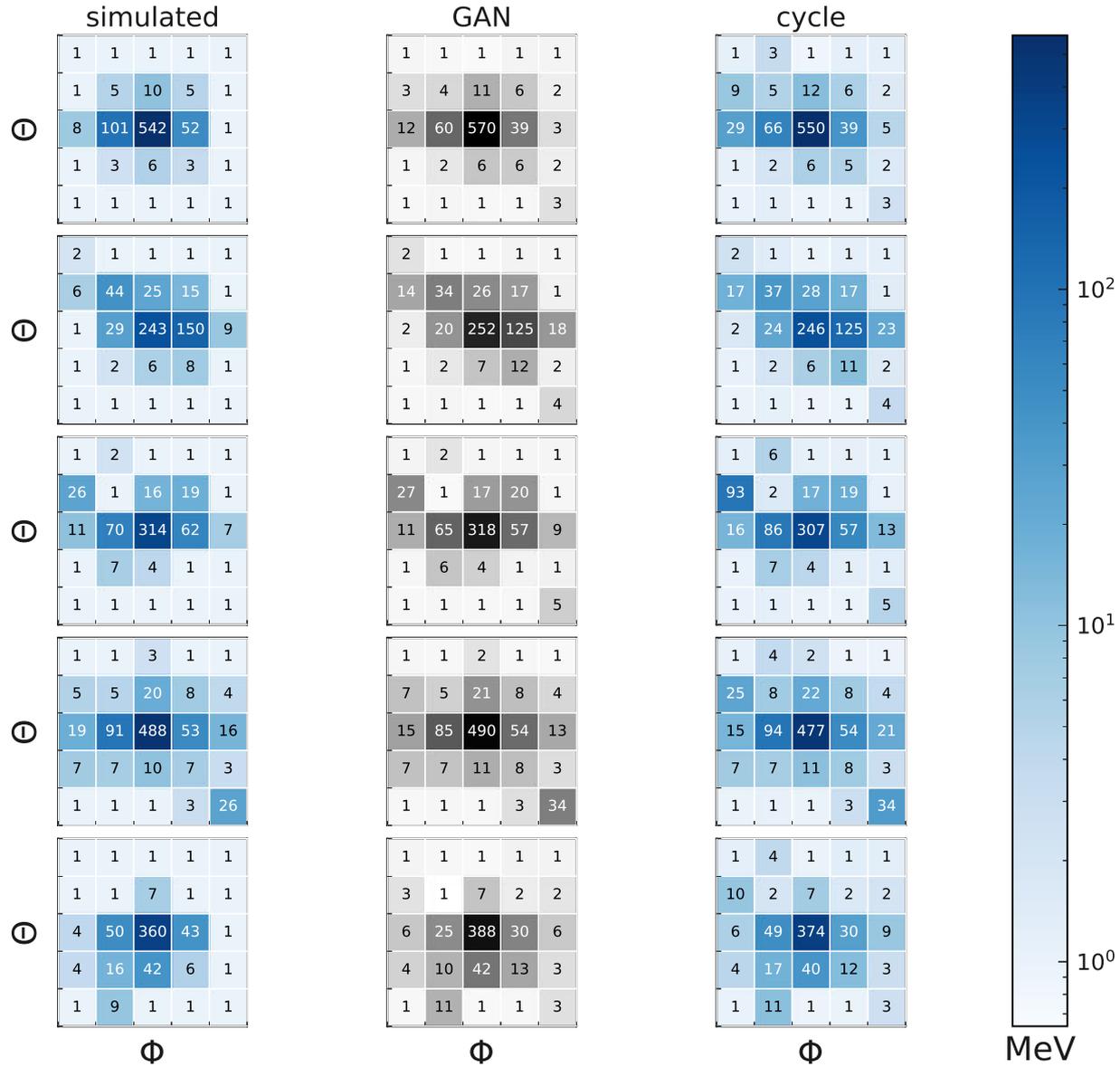


Figure 5.4: Images of reconstructed electrons [left column], corresponding GAN photons [center column] and corresponding cycle electrons [right column].

distribution, it seems to remove more energy from the image, but not to the same extent as the blue distribution adds. This leads to an artificial increase of the energy during the conversion. Such a case can be seen in figure 5.5 in the fourth row. In this row, the middle column added too much energy to the original image, but the right column could not compensate for this excess energy. This effect results in a bad cycle particle, which has more energy than the original reconstructed particle.

The task for this chapter is to understand the performance of the CycleGAN. In the following two sections, the evaluation metric described in the section 4.2 and 4.3 is used to test the conversion and take a closer look at its performance.

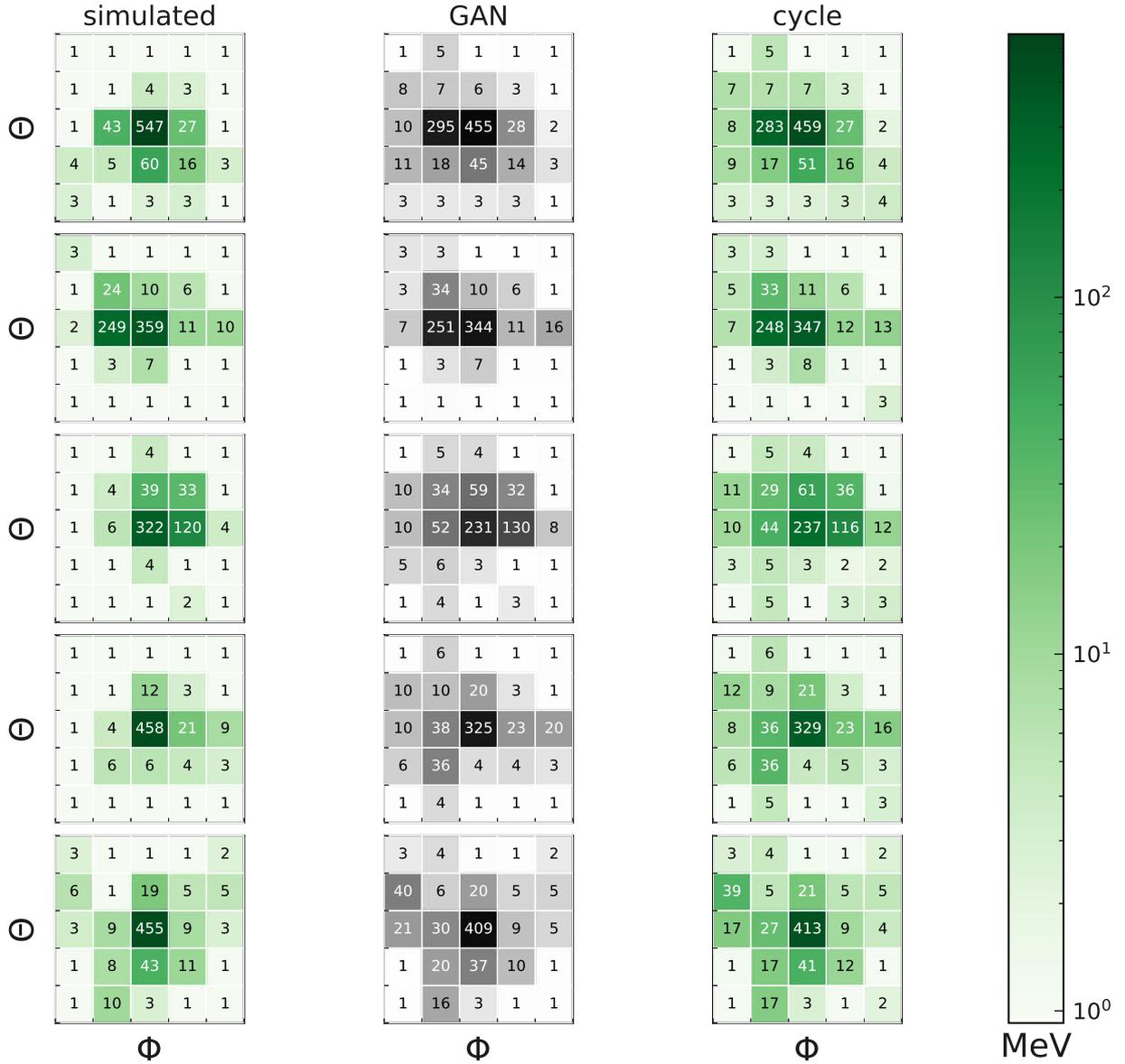


Figure 5.5: Images of reconstructed photons [left column], corresponding GAN electrons [center column] and corresponding cycle photons [right column].

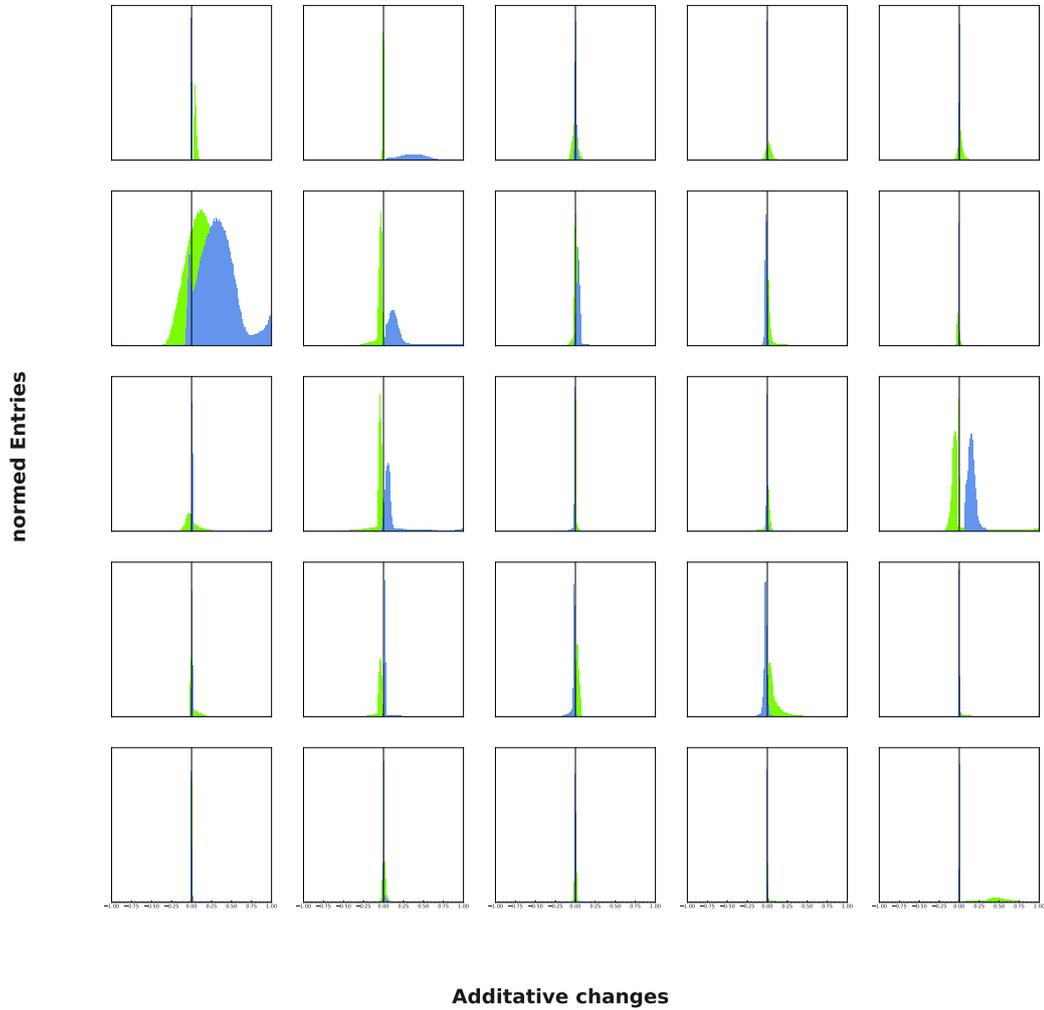


Figure 5.6: Image value difference for each pixel between GAN particle to its reconstructed input particle. The blue distribution calculate the difference between GAN electrons to reconstructed input photons. Respectively, the green distribution calculate the difference between GAN photons to reconstructed input electrons. The values of reconstructed and GAN particles within a pixel is still in  $\log_{10}(\text{MeV})$  scale. Those additive changes will be factorial changes between a factor of 0.1 and 10 in MeV. The black line marks the value zero, where no changes are made.

## 5.2 Pre-trained classifier evaluation

In this section the pre-trained classifier of section 4.2 will be used to evaluate the conversion success of both generators. The pre-trained classifier was trained to differentiate between reconstructed electrons from reconstructed photons. For each input the pre-trained classifier outputs a score between zero and one. A score close to 1 means that the input is categorized as a reconstructed electron, and if it is close to 0, the input is categorized as a reconstructed photon. In figure 5.7 the score distribution from the output of the generators is compared with their input.

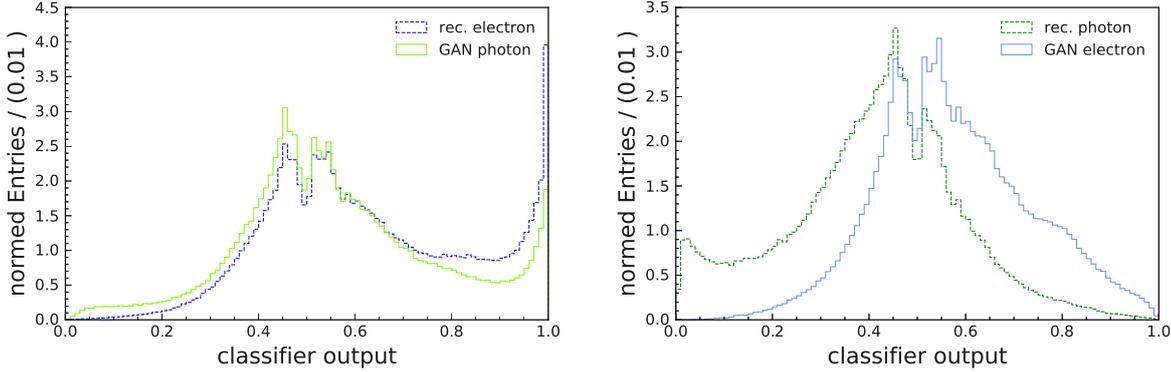


Figure 5.7: Comparison between reconstructed electron and GAN photon (left side), as well as the comparison between reconstructed photon and GAN electron (right side).

On both sides of the figure a shift into the correct direction can be observed for the output. The conversion of reconstructed photons into a GAN electrons seems to be more successful than the other way around. It is interesting which particles were easily converted and which particles were more problematic. For better understanding, in figure 5.8 the difference of the score in energybin has been plotted for the difference of the GAN photon score to their reconstructed input electron. Figure 5.9 plots the difference in score between GAN electrons and their reconstructed input photons.

In figure 5.9, no changes can be seen in the output of the pre-trained classifier. For the same conversion in figure 5.7 on the left side, a minimal shift in the distribution can be seen. In order to check the energy dependence for the minimal shift, figure 5.9 will be zoomed in the range of -0.1 to 0.1 (see figure 5.10). In the zoomed plot it is observable that the majority have a slight decrease in pre-trained classifier score, except for the  $E_{\text{cluster, rec. electrons}}$  range from [1,2] GeV. The general decrease is small and seems to be not dependent on  $E_{\text{cluster, rec. electrons}}$ .

In figure 5.8, the distribution seems to be generally above the black line. The pre-trained classifier score increases after applying the reconstructed photon to the generator of the CycleGAN. In figure 5.8, a strong difference between reconstructed input particle and GAN particle is observed for reconstructed input particles with less than 1 MeV  $E_{\text{cluster}}$ .

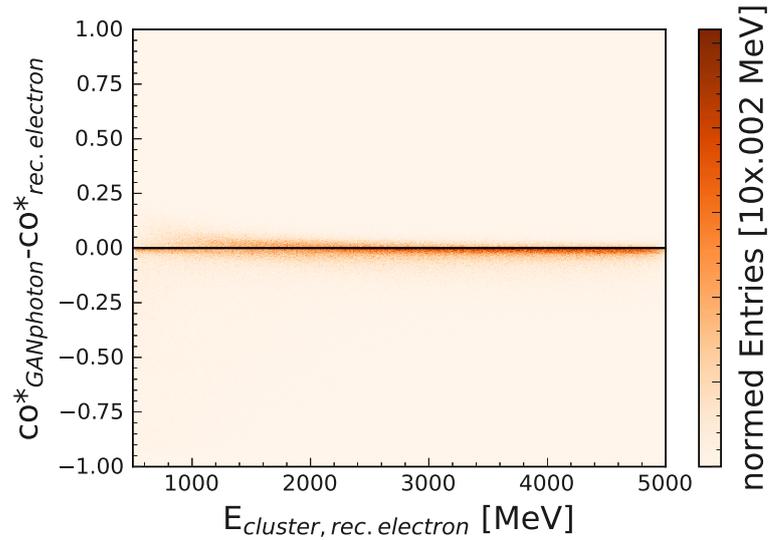


Figure 5.8: Comparison between reconstructed electron score and GAN photon score, given by the pre-trained classifier. The black line marks no changes in the score of the pre-trained classifier. Co\* is short for pre-trained classifier output.

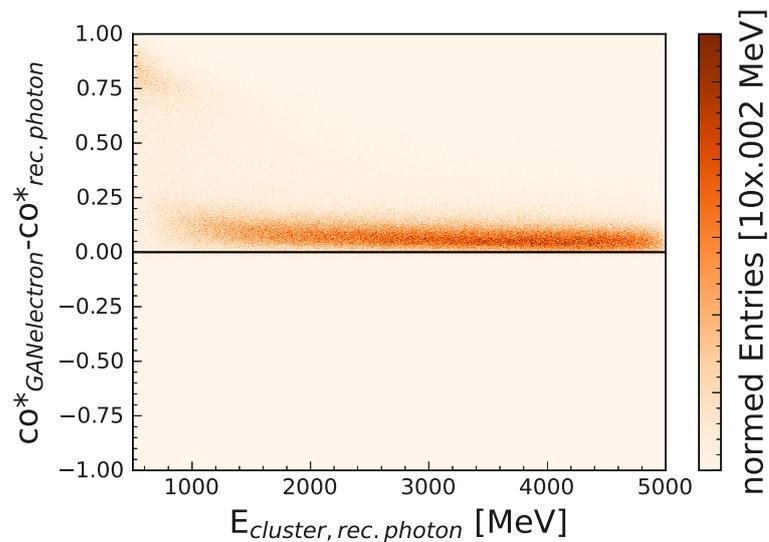


Figure 5.9: Comparison between reconstructed photon score and GAN electron score, given by the pre-trained classifier. The black line marks no changes in the score of the pre-trained classifier. Co\* is short for pre-trained classifier output.

Figure 5.11 replicates the left side of figure 5.7, with only the reconstructed photons showing a change in the pre-trained classifier score of 0.3 or more.

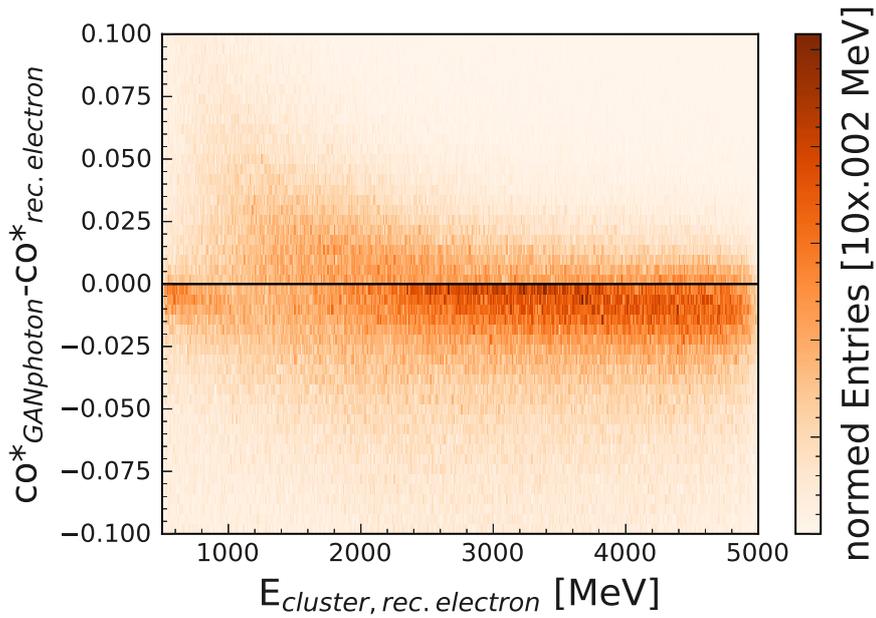


Figure 5.10: Comparison between reconstructed electron score and GAN photon score, given by the pre-trained classifier. Co\* is short for pre-trained classifier output.

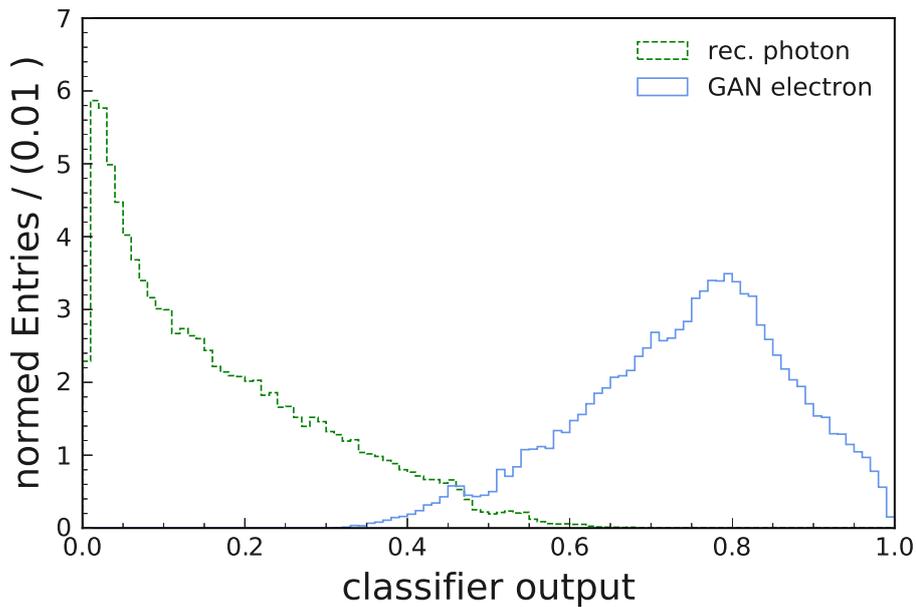


Figure 5.11: Comparison between reconstructed photon and GAN electron, which have a pre-trained classifier score difference of 0.3 and above.

In conclusion it was observed, that the conversion from reconstructed photon with an  $E_{\text{cluster}}$  smaller than 1 MeV to a GAN electron makes significant changes in the pre-trained classifier output. The conversion between a reconstructed electron to a GAN photon is almost not noticeable. The pre-trained classifier score get closer to zero for the GAN photons, but the change is only by a small margin.

### 5.3 Pairwise metric evaluation

This section investigates the difference between the shower shape of the generated particle and the shower shape of the corresponding reconstructed input particle.

The shower shape are listed in table 3.2. In figure 5.12, the difference between the reconstructed input electron image and the resulting GAN photon image for each shower shape variables is shown in a two dimensional plot. Accordingly, figure 5.13 shows the differences between reconstructed input photons and GAN electrons. From the previous results, better performance is expected in the direction from the reconstructed input photon to the GAN electron.

The difference in shower shape between input and output is expected to be larger for lower energy particles ( $<1\text{MeV}$ ) than for higher energy particles. The two dimensional plots are binning the difference of the shower shape in  $E_{\text{cluster}}$  of the reconstructed input particle. Instead of showing the absolute difference for each energy bin. All difference values are divided by the shower shape variable value of the respected reconstructed input particle.

Let  $x$  be the value of a the shower shape variable  $mv$  of the reconstructed input particle and  $y$  be the value of  $mv$  for the GAN particle. All the  $y$  axis in the plots in figure 5.12 and 5.13 are  $\frac{y-x}{x} = \frac{y}{x} - 1$ .  $\frac{y}{x} - 1$  gives the percentage changes between  $y$  and  $x$ .

In conclusion both figures displayed expected behavior. As mentioned before the images between reconstructed photons and reconstructed electrons are extremely close, especially for the higher energetic particles. Therefore, it is reassuring that all plots are centered around zero. If the distances between the distributions and the black lines in the figure were too great, that could mean that the generators does not produce the pairwise results. The maximum change is still within a five percent range, which is quite large but still reasonable.

The greater distance between the distributions in figure 5.13 and the black line compared to the distance between the distributions and the black line in figure 5.12 is also consistent with the result from the introduction of this chapter and section 5.2.

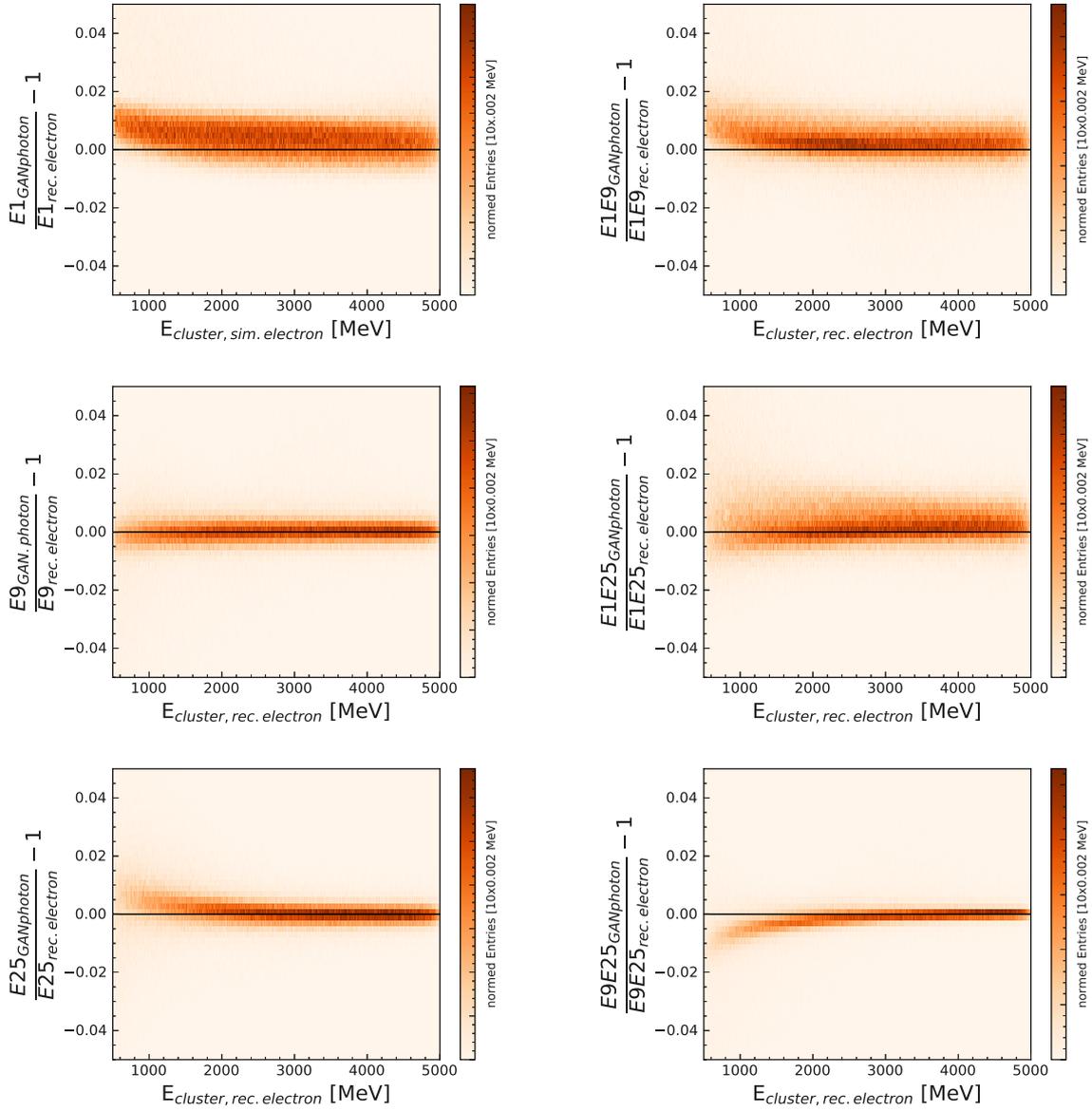


Figure 5.12: 2D shower shape difference (in percent) between reconstructed input electrons and GAN photons. Black line is at zero and benchmark no difference in respected shower shape.

## 5 Results

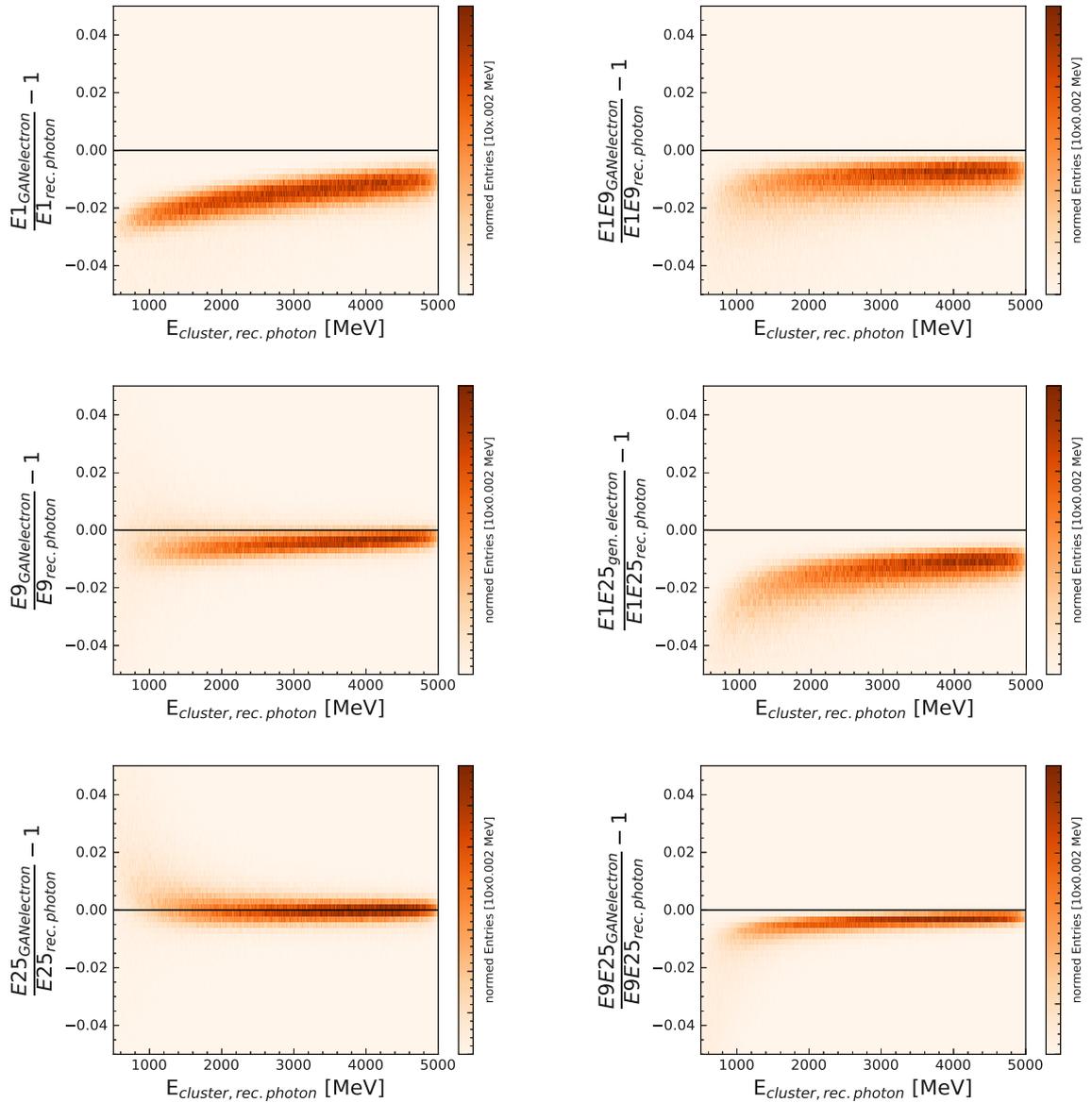


Figure 5.13: 2D shower shape difference (in percent) between reconstructed input photons and GAN electrons. Black line is at zero and benchmark no difference in respected shower shape.

## 5.4 Conclusion and summary

The goal for this thesis is to understand issues and test the applicability of CycleGAN. In terms of performance, the last three sections have revealed a lot of weak points. Starting with the difference in performance between converting reconstructed electrons to GAN photons and converting reconstructed photons to GAN electrons.

The pre-trained classifier has observed a positive change towards the distribution of the target class for both directions. In both plots of Figure 5.7, the distributions for the GAN particles at least shift towards the center. For the conversion direction from reconstructed photons to GAN electrons, these changes are quite pronounced for some, as can be seen in figure 5.9 and 5.11. The changes in the other direction are present but not pronounced or slightly wrong for a specific range of  $E_{\text{cluster}}$ . Figure 5.10 shows the changes in the pre-trained classifier output over the  $E_{\text{cluster}}$  of the reconstructed input electrons. The distribution is close to zero (the black line), but the centroid for each  $E_{\text{cluster}}$  bin is slightly below the black line. This leads to a small change in the figure 5.7 (upper diagram). Those results suggesting that the generators in the CycleGAN do something right in there conversion. The generators may not yet be performant, but the development is already promising.

As described in the 4.3 section, the goal is not only to obtain a conversion between reconstructed particle and GAN particle, but the reconstructed particle and its corresponding GAN particle should additionally be in a pairwise relationship. The pairwise relationship is tested by looking at the changes for all shower shapes. For a pairwise relationship, the values for all shower shapes are expected to change only minimally. In figures 5.12 and 5.13 it can be seen that the difference between the shower shape values of the reconstructed input particle and the shower shape values of the GAN particle is close to zero. The plots in figure 5.12 are almost all at zero. The plots in Figure 5.13 are different, but still close to zero. Consequently, the GAN and reconstructed input particles are similar. This speaks for a successful conversion by the generators. Both generators were able to output images close to the input images, which is the minimum requirement for a pairwise relationship between reconstructed input particle and GAN particle. This closeness or the pairwise relationship between input and output of the images can be observed in figure 5.4 and 5.5.

The conversion is going in the right direction and the pairwise relationship between reconstructed input particle and GAN particle can be seen. According to subsection 4.1.3, this trained CycleGAN can be considered a success. It is not performant and there are some issues with the CycleGAN, but with respect to the two metrics described in section 4.2 and 4.3, the CycleGAN does not fail. There is a lot of potential to address certain issues and improve the CycleGAN. An indication that the CycleGAN has not reached its full potential is given by the figures 5.1 and 5.2. From both figures it can be seen that the discriminator outperforms the generator. The curve in 5.1 decreases while 5.2 increases. This does not necessarily mean that the generator is getting worse, but rather that the generator is maintaining its performance while the discriminator is getting better. There are several factors that could complicate generator improvement. Figure 5.3 shows that the generators for these additional loss terms stagnate quite early.

## 5 Results

The total loss for the generator is the sum of the values of both plots from Figure 5.3 and from Figure 5.2. From the comparison of the y-axis of these three diagrams, it is quite easy to understand that the  $\text{Loss}_{E25,A}$  (see equation 4.6) dominates the total loss of the generator. The high values for the  $\text{Loss}_{E25,A}$  is not observed in the figure of images 5.4 and 5.5.  $\text{Loss}_{E25,A}$  is prone to outliers. If there is an outlier with a  $\Delta E25$  of 1500 MeV, the square is fatal and affects the total  $\text{loss}_{E25,A}$ , even if most  $\Delta E25$  are below 1 MeV. One can counteract this by using the absolute difference or some other function to represent  $\Delta E25$  in the loss functions. The problem with the outlier is then still present, but the effect would not be so drastic.

Figure 5.6 gives a good insight into the work of the generators. This figure is described in detail in the section 5.1. There were a couple of problems in this figure. One of them was that the blue and green distributions must theoretically be on opposite sides of the black line for the cycle particle to be correct in each pixel. Another problem was that the distance between the distributions and the black line must be at least similar. If this is not given, the cycle particle over- or undercorrects the reconversion. In the same sense, a distribution must be on the right side of the black line for some pixels and on the left side for other pixels. Otherwise, the generators have not shifted energy, but rather generated energy (if the distribution is exclusively on the right side) or lost energy.

The generators make additive changes to the reconstructed input image, but these additive changes have a factorial effect on the images. Effectively, the generators do not decide how much energy in MeV a pixel gains or loses. The generators decide by which factor a pixel is multiplied. The log10 scaling was done to amplify the small values at the tail of the energy distributions, but this could make it difficult for the generator to learn. Naively, the best conversion would take energy from one pixel and distribute it to other pixels. In the current setup, the generator reduces the energy from one pixel by a factor and then multiplies it by another factor at another pixel. Factoring the pixel values does not control the redistribution of energy within the image and  $\text{Loss}_{\text{cycle}}$  and  $\text{Loss}_{E25,A}$  is not enough to counteract the phenomena. Removing the log10 scale can lead to sparsity issues due to the dominant center pixel where most of the energy is detected and reconstructed. The next chapter will discuss possible improvement to increase the performance of the CycleGAN. Also, the connection to the overall motivation will be drawn.

# 6 Outlook

In this final chapter, the future of this project is discussed. In the first section 6.1, the problems in the current setup are addressed. The details are explained and possible solutions are suggested. The second section talks about the general motivation and what the next steps could be.

## 6.1 Problems

The CycleGAN result was not a failure, but Chapter 5 still showed weaknesses in current performance. These weaknesses need to be discussed. Solutions to these problems are proposed. A paragraph is devoted to each problem.

**Bias** The physics of the electromagnetic shower shape between electrons and photons is identical. In the crystal, an electron emits photons due to bremsstrahlung and these photons convert into an electron and positron pair. The combination of bremsstrahlung and pair production forms the electromagnetic shower. Physically, the generation of the electromagnetic shower by electrons and photons differs only in whether it starts first with bremsstrahlung or with pair production. Another point that could make a difference between electronic and photonic shower shapes is the curvature of the electrons. The curvature of the electrons may cause the electrons to enter the crystal at a steeper angle. A steeper entry angle changes the shower shape, because the energy distribution is more likely to be spread over multiple crystals. This effect is more dominant for low energy particles and negligible for high energy particles. This could explain the better performance of the pre-trained classifier for the low energy particles.

The overall motivation is interested in converting real detected electronic shower shapes into photonic shower shapes to provide a calibration sample for low energy photons. The neural networks in CycleGAN and the pre-trained classifier are trained on a training set of reconstructed particles with particle energy from 500 MeV to 5 GeV. As mentioned earlier, the differences in shower shape between electrons and photons occur predominantly for low energy particles. The majority of the training set is not low energetic. This bulk can cause a bias and complicate the task of converting images. A simple solution would be to filter for reconstructed particles with particle energy less than 1 GeV. Filtering the training set would remove the bulk and unnecessary bias and still work for the overall motivation.

**Log10 scale** The typical image of a reconstructed particle had most of its energy deposited in the center crystal of the image. Sparsity and edges are common problems for GANs, especially when one pixel is dominant. The small dimension of the images in the training set is more of a liability. To counteract these problems and for one enhance the tails of the shower shapes, as well as diminish the center pixel, the log10 scale was used. Each pixel was raised by one MeV and then for each pixel the log10 was taken. This scaled down the image. Images 5.4 and 5.5 are also in log10 scale, otherwise only the center pixel would be colored and the other pixel would not be noticeable. At that time the log10 seem to be a natural choice.

The introduction of log10 scaled images introduced other issues. The perfect conversion would shift energy from one pixel to another without introducing or removing energy. The conversion from low energy electrons shower shape to low energy photons shower shape should shift energy from the tail distribution to the center in order to have a more radial and centered shower shape. The perfect conversion takes energy from one pixel and redistribute the energy in other pixels.

The additive changes by the generator, as seen in Figure 5.6, do not shift the energy in the same way. The addition are done on the log10 of the energy values. Let  $x$  be the energy values and  $\tilde{x} = \log_{10}(x)$  the log10 of  $x$ . Then the change caused by the generator is  $\varepsilon$  and the new value is  $\tilde{y} = \tilde{x} + \varepsilon$ . By rescaling to the normal energy scale, we factorize the original value  $x$ . The new value is  $y = x \cdot 10^\varepsilon$ . Redistributing energy from one pixel to another now becomes more difficult. Instead of taking  $\Delta x = y - x$  amount of energy and redistributing it to other pixels, the generator reduces the energy from one pixel by multiplying its value by a factor of  $< 1$ . The redistribution of  $\Delta x$  to other pixels must be done by factors of the target pixels. Instead of adding the amount  $\Delta x$  to the target pixel, the generator must find the correct factor  $10^\varepsilon$  by which to multiply the target value. This task is much more complex because the redistribution of energy is now strongly dependent on the value of the target pixel. Moreover, a pixel can be affected by many other pixels. Therefore, the factor  $10^\varepsilon$  depends on the value of the target pixel and on the  $10^\varepsilon$  factors of the pixels from which the energies originate. The complexity increases if the training set has a large energy spectrum, as in this thesis.

Another problem with the introduction of the log10 scale arises, when considering the derivative of log10. The derivative is not symmetric and this leads to different effects between positive and negative  $\varepsilon$ . Figure 4.17 illustrates this asymmetry. Ratio  $\frac{10^\varepsilon - 1}{10^{-\varepsilon} - 1} \in [1, 10]$  is only one for  $\varepsilon = 0$ . For every other  $\varepsilon$  this ratio is not one and  $\varepsilon$  will result in a bigger  $\Delta x$  than  $-\varepsilon$ . The generator only has influence on  $\varepsilon$ , but  $\varepsilon$  is highly complex and dependent on few things which can vary between images. In theory, each pixel can influence another pixel, which further influences each  $\varepsilon$ .

The ideal would be a scaling solution that on the one hand adopts the objective of logarithmic scaling, but on the other hand changes each pixel additively after rescaling.

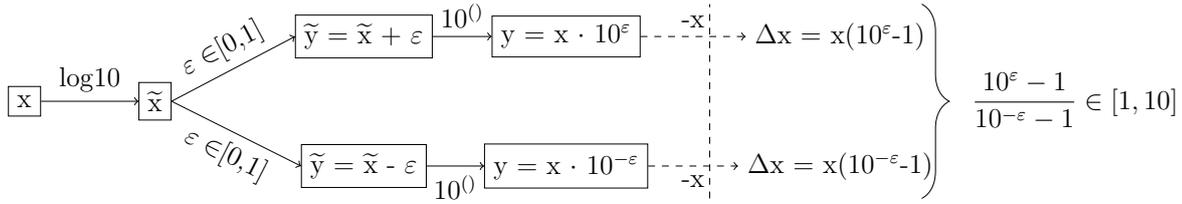


Figure 6.1: Flowchart of effective changes done by the generator.  $x$  is the original energy detected in the pixel,  $\tilde{x} = \log_{10}(x)$ .  $\varepsilon$  is the changes the generator does on the pixel to create  $\tilde{y} = \tilde{x} + \varepsilon$ .  $\frac{10^\varepsilon - 1}{10^{-\varepsilon} - 1} \in [1, 10]$  returns the ratio between enhancing and diminishing energy.

**Optimization and different neural networks** As described in the previous paragraph, the generator was set to solve a “simple task“ in a very complicated and high-dimensional way. This work highlights the challenges and framework for using a CycleGAN, but does not optimize it. As seen for the pre-trained classifier, performance was not optimal for the majority of the training set. The model for the pre-trained classifier can be optimized with deeper resnet blocks or more hidden layers. There is also the possibility to use CNNs to give more emphasis to the neighboring pixels. The preprocessing steps can also be further improved by adding more layers and or varying the number of representation neurons for the edge and cluster information. This model optimization is not limited to the pre-trained classifier and will also apply for the discriminators of the CycleGAN. The number of neurons, hidden layers etc. for all neural networks in this thesis was not optimized yet and the hyperparameters can be studied. Table 6.1 gives an overview of all hyperparameters to be optimized. Besides hyperparameter optimization, one can also try other neural networks for the generator or discriminator. For the generators, autoencoders seem to be very suitable and are worth trying.

**Assymetrie in complexity** The generators and the discriminators must be adapted to their objectives. In the chapter Results 5 the performances of the two generators are very different. The simulated photon to GAN electron generator produces images that are different from the original reconstructed input photon image, while the reconstructed electron to GAN photon generator is almost an identity operator. This difference performance needs to be addressed. Both generators are identical in their setup and architecture. There is no difference in the calculation of the loss terms or the generation of images. The only difference is the input and the target of the two generators. If the complexity of the conversion is not symmetrical in both directions, this will lead to an asymmetry in the performance.

The CycleGAN is a bidirectional image-to-image convertor. In the paper [38] the authors addressed the problematic of an assymetrie between the complexity of the conversion in both direction. If the conversion from domain A to domain B is much more complex than the conversion from domain B to domain A the respected generator needs to adapt to the complexity. In order to calculate the asymmetry [39] one has to ask how

Table 6.1: An overview over all parameters one can change to optimize the CycleGAN.

Discriminitor and pre-trained classifier
number of neurons
number of layers
dropout value
modify preprocessing steps
switch loss function from BCE to MSE or other
switch optimizer to RMSprop or other
learning rate
activation function
Generator
number of neurons
number of layers
activation function
dropout
switch loss function for $Loss_{E25}$
weight different terms in generator loss function
optimizer
learning rate
CycleGAN
change trainings set and scale differently
train discriminator n times more often than generator
decay of learning rate
epochs
batch size

much different are the information between domain A and B, by starting to calculate the average image entropy of those domains. Let  $P_{ij} = \frac{f(i,j)}{N^2}$ , with  $f(i,j)$  being the frequency of the tuple of pixel value  $i$  and neighbor pixel value  $j$  and  $N^2 = N \times N$  the size of the 2D image. The entropy per image can be calculated as  $Ent_{image} = -\sum_j \sum_i P_{ij} \cdot \log(P_{ij})$ . The average 2D image entropy of given domain X will calculate the average of all images within that domain.  $Ent_{avg.images\ of\ X} = \frac{1}{M} \sum_m Ent_m$  is the equation for the average entropy of given domain X, with M being the total number of images within the domain. The complexity of two domains A and B can be described by the ratio in equation 6.1

$$Ent_{ratio} = \frac{\max(Ent_{avg.images\ of\ A}, Ent_{avg.images\ of\ B})}{\min(Ent_{avg.images\ of\ A}, Ent_{avg.images\ of\ B})}. \quad (6.1)$$

If the ratio is one, there is no difference in the amount of information a domain contains. Intuitively, if the ratio is  $1 + \lambda$ , this will mean that there is at least a  $\lambda \cdot 100\%$  difference between the two domains. In paper [38] the authors define the task of the CycleGAN as

significantly asymmetric, if the ratio exceeds 1.5.

If one applies the equation 6.1 to this thesis, with A the images of reconstructed photons and B the images of reconstructed electrons, one will get  $\approx 2.3$ . This asymmetry seems not to be negligible, and this is shown in the results. The paper suggests to adopt the generators according to the complexity of the task. For this thesis, that would mean to build a more sophisticated model for the generator from reconstructed electrons to GAN photons.

## 6.2 Future steps

The first step for the project is to address the problems listed in the previous section. This will optimize the current setup of the CycleGAN. The bias problem is quite easy to test in the current setup. The CycleGAN needs to be trained on a subset of the trainings set in order to see if the described problem has any effect on the performance. For this hypothesis the CycleGAN was retrained with all reconstructed particles from the subset with an  $E_{MC}$  below or equal to 1 MeV. The figure 6.2 is the equivalent figure to figure 5.7, except this is the performance of the CycleGAN trained on the subset of the trainings set. 5.9.

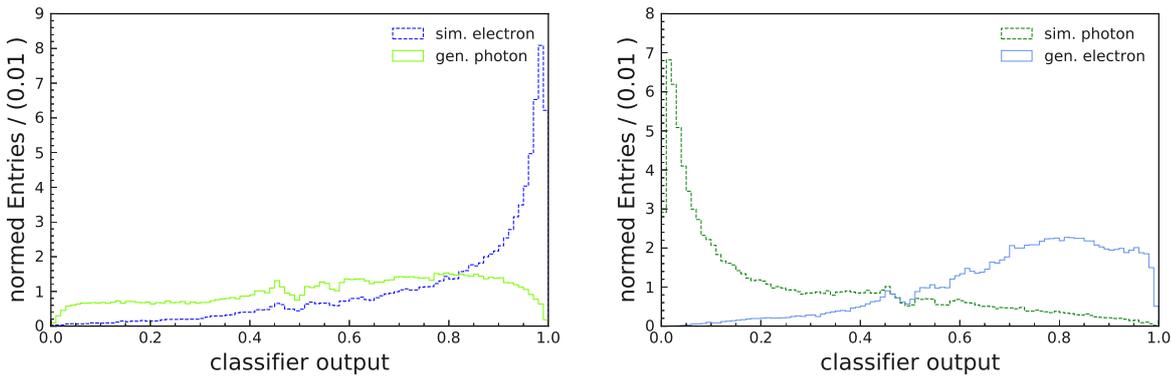


Figure 6.2: Comparison between reconstructed electron and GAN photon (left side), as well as the comparison between reconstructed photon and GAN electron (right side).

In comparison figure 6.2 showed significant improvements in the generation of GAN electrons from reconstructed photons (left side). The peak at pre-trained classifier output one is completely vanished and the whole distribution has redistributed over the whole axis. Figure 6.3 supports this observation. The plot shows more clearly a decrease in pre-trained classifier output along the x-axis. Compare to the distribution in figure 5.8 the reduction of the pre-trained classifier output after the conversion by the CycleGAN is obviously greater. Figure 6.4 seem to perform similar to the previous CycleGAN. In comparison to figure 5.9 the new CycleGAN might be a bit worse for the given energy

range, but in general the performance is not bad. The distribution in figure 6.4 is almost exclusively above the black line, which is the wanted behaviour.

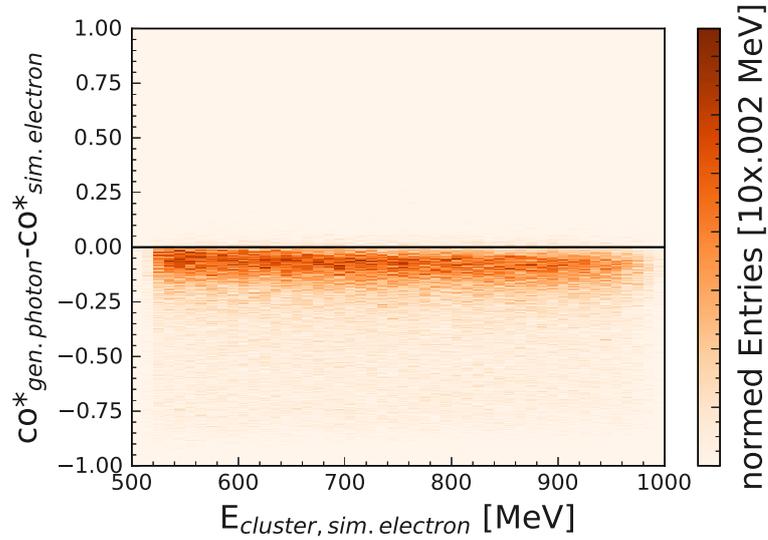


Figure 6.3: Comparison between reconstructed electron score and GAN photon score, given by the pre-trained classifier. The black line marks no changes in the score of the pre-trained classifier. Co\* is short for pre-trained classifier output.

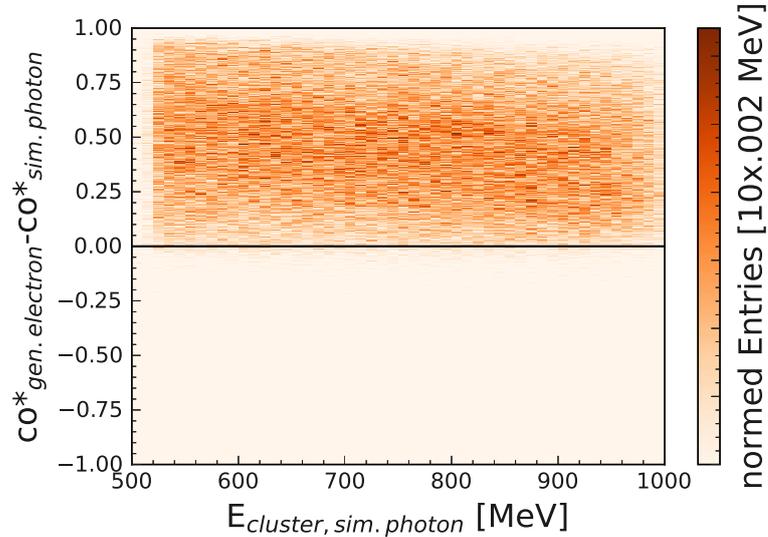


Figure 6.4: Comparison between reconstructed photon score and GAN electron score, given by the pre-trained classifier. The black line marks no changes in the score of the pre-trained classifier. Co\* is short for pre-trained classifier output.

Figure 6.5 meets more adequately the expectations as figure 5.6. The expectations are

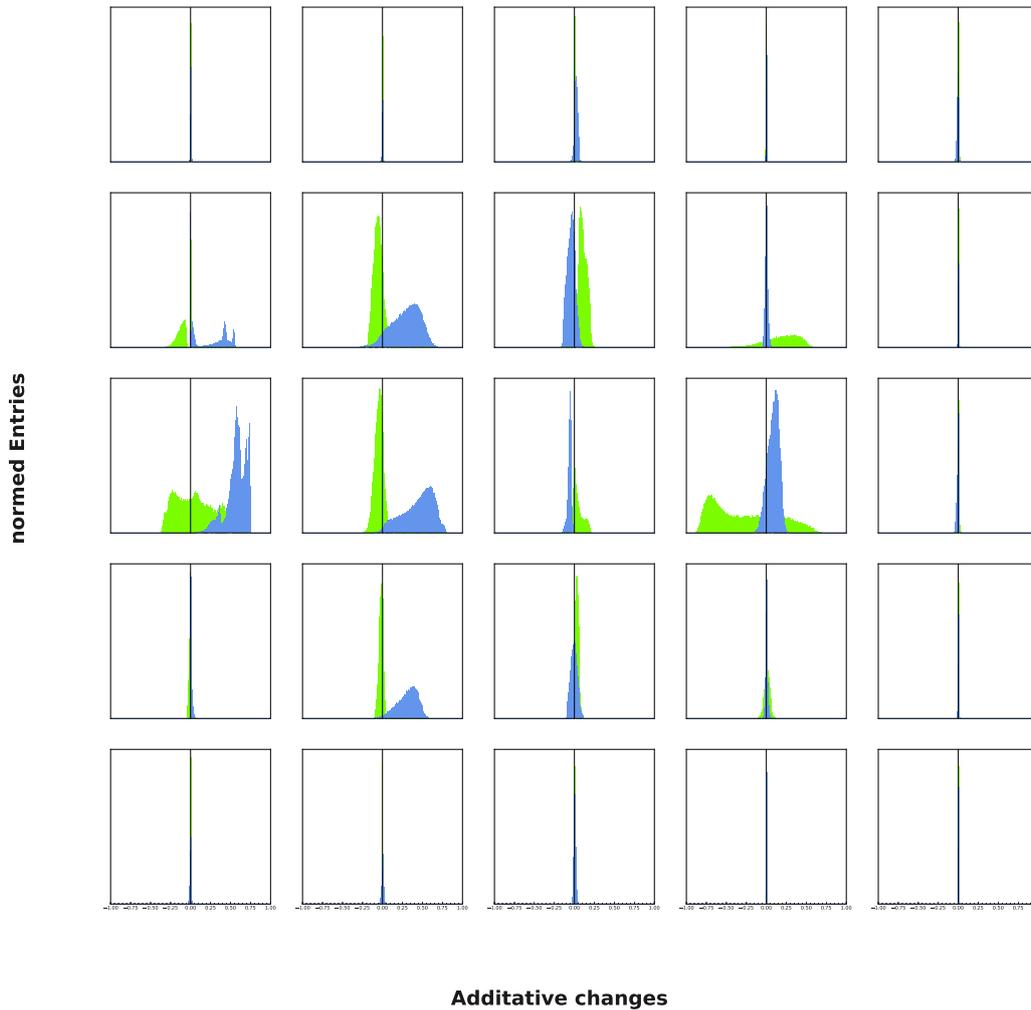


Figure 6.5: Image value difference for each pixel between GAN particle to its reconstructed input particle. The blue distribution calculate the difference between GAN electrons to reconstructed input photons. Respectively, the green distribution calculate the difference between GAN photons to reconstructed input electrons. The values of reconstructed and GAN particles within a pixel is still in  $\log_{10}(\text{MeV})$  scale. Those additive changes will be factorial changes between a factor of 0.1 and 10 in MeV. The black line marks the value zero, where no changes are made.

described in section 5.1.

**Generating cluster information** The next steps to achieve a shower shape conversion from detected electron to GAN photon is to expand the task of the generator. In the current setup the generator only generates the GAN image, but in the future it shall also generate the cluster information for the GAN shower shape. Currently, the cluster information for the GAN image is calculated analytically. The analytical calculated cluster information is based on MC variables of the reconstructed input particle, especially  $\Phi_{\text{helix}}$ . The MC information will not be available for the detected electrons anymore. Therefore, it is important to have something to generate the cluster information for the GAN particle instead. There are many ways to provide the cluster information for the GAN particle. One way would be to train a cluster model, which essentially learns the lorentz equation 4.5. It can take the cluster information from the reconstructed input particle and transform this into the cluster information of the GAN particle, with the analytical cluster information as a target. This will be essentially a prediction task for the cluster model. In reality, there is a relation between image and cluster information and one can also input the GAN image to the cluster model. The problem of adding the GAN image in the cluster model would only introduce noise. The target does not change with the introduction of the GAN image, which makes it unnecessary. The solution to generate cluster information is not to target the analytical calculated cluster information of the GAN particle, but to target the correct center crystal. Each crystal has an ID and the combination of image and crystal ID of the local maximum can be used as an input for the basf2 reconstruction package, which will return the cluster information. In this case the cluster model have to output a crystal ID, which will be as close to the analytical calculated cluster information as possible. This approach is desirable, because this does not expect the cluster model to get the perfect cluster information for the GAN images, but rather it is trained to have the resolution of one crystal ID.

The difficulty of the cluster model lies in the technical implementation. The basf2 reconstruction package needs the crystal ID as a long type. By converting the output from float type to long type will break the computational graph in pytorch. This break of the computational graph is fatal, because this would mean that there is no possibility to do the backward step, which is responsible to calculate the essential gradients for the optimization step. If this could get resolved, there is a second issue preventing this thesis to implement this approach. The second issue lies within the basf2 reconstruction package. The basf2 reconstruction package uses a look up table to get the coordinates for the given crystal ID and its neighbor and calculate from there the position of the reconstruction object. The principle of the look up table is not compatible with a computational graph. There is no derivatives for the location of the crystals.

A good goal for the next thesis building on this work, is to optimize the current setup, as discussed in the previous section, and to find a work around for the generation of cluster information.

**Applicability test** If the next thesis has found a work around for the issues regarding the generation of cluster information, the next logical step would be to try and prove the assumptions the whole project is building on. In subsection 4.1.1 two assumptions were

made. Firstly, the mapping between reconstructed particles is the same as the mapping between detected particles and secondly, the adoption of background. The second assumption by adding noise to the input and check, if the noise can be extracted from the output. In this stage, it is important to legitimize those assumptions.

The applicability of the whole project is defined by the explanation why those assumptions are legitimate. All the work beforehand is to check, whether the project will be possible or not.

**Expand to particles** The final thesis on this project will expand a working shower shape conversion. Until this point, the only information was taken from the ECL, but the electrons are also measured in the CDC. It is also of interests to generate whole particles and its complete detection in the Belle II, instead of generating the shower shape of the hypothetical particle. At this stage, one has to ensure that a GAN track in CDC also points correctly at a GAN shower shape. The track needs the correct curvature and for this, it has to estimate the correct particle energy and the correct electromagnetic charge.

The expansion can also try to include the end caps of the ECL. Until this stage, only the barrel of the ECL was used. The endcap are especially tricky, because the crystals are differently shaped and organized as in the barrel. The inclusion of the end cap can also be a thesis of itself.



# Appendix

## Overview: Simulation in Electromagnetic Calorimeter

## 6 Outlook

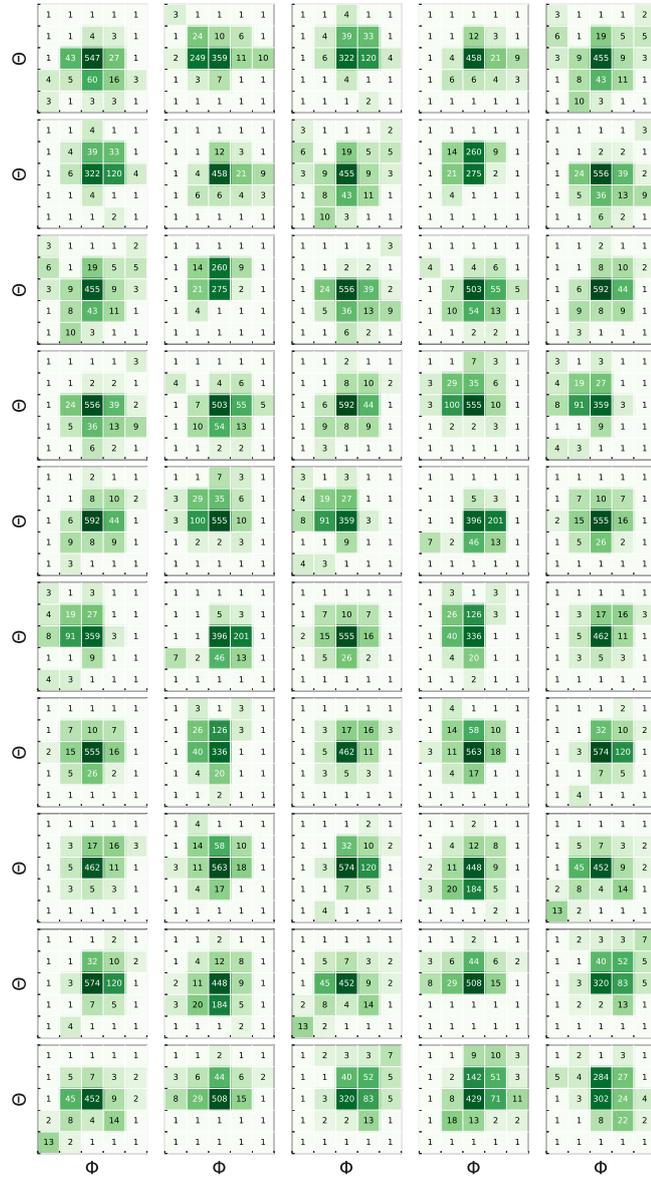


Figure 6.6: 50 random reconstructed photonic shower shape images in log10 scale.

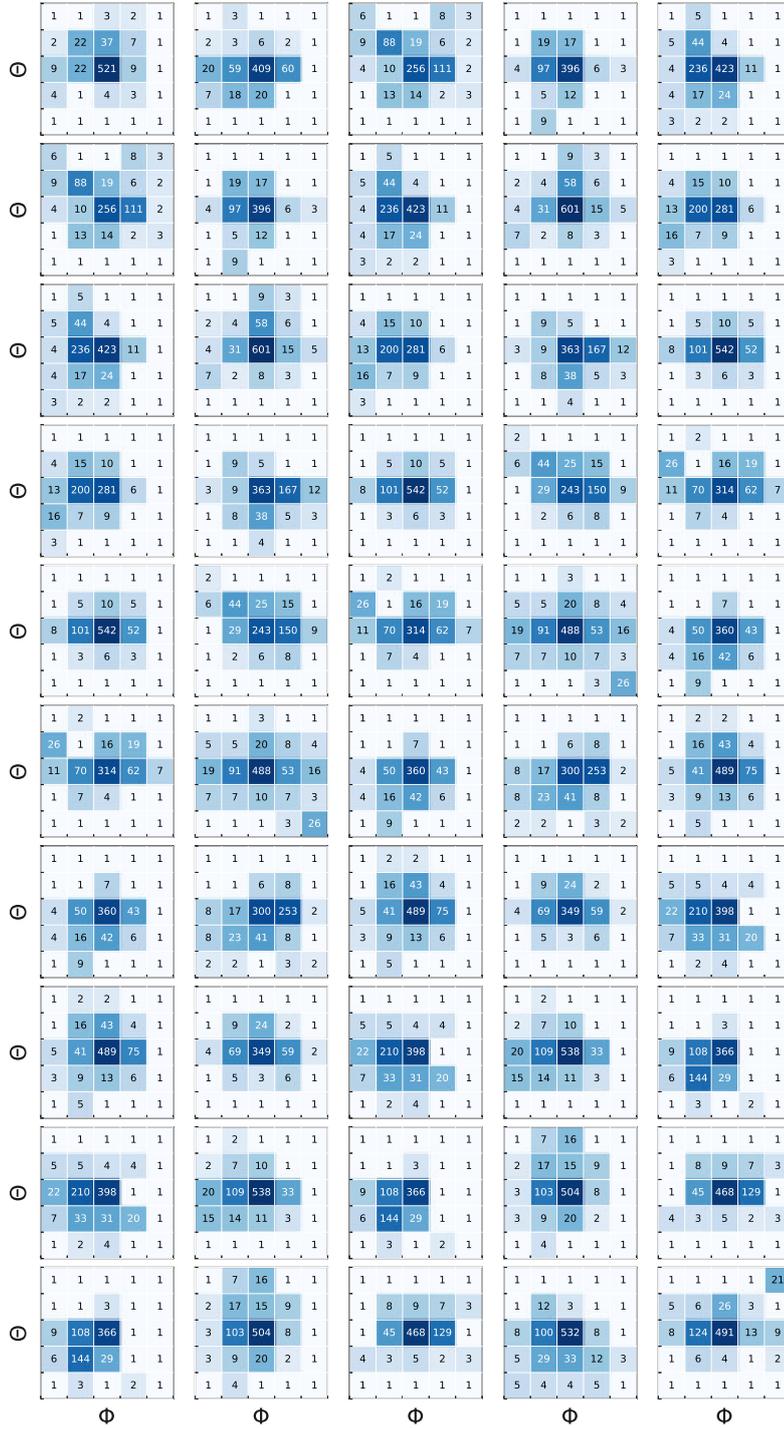


Figure 6.7: 50 random reconstructed electronic shower shape images in log10 scale.

## 6 Outlook

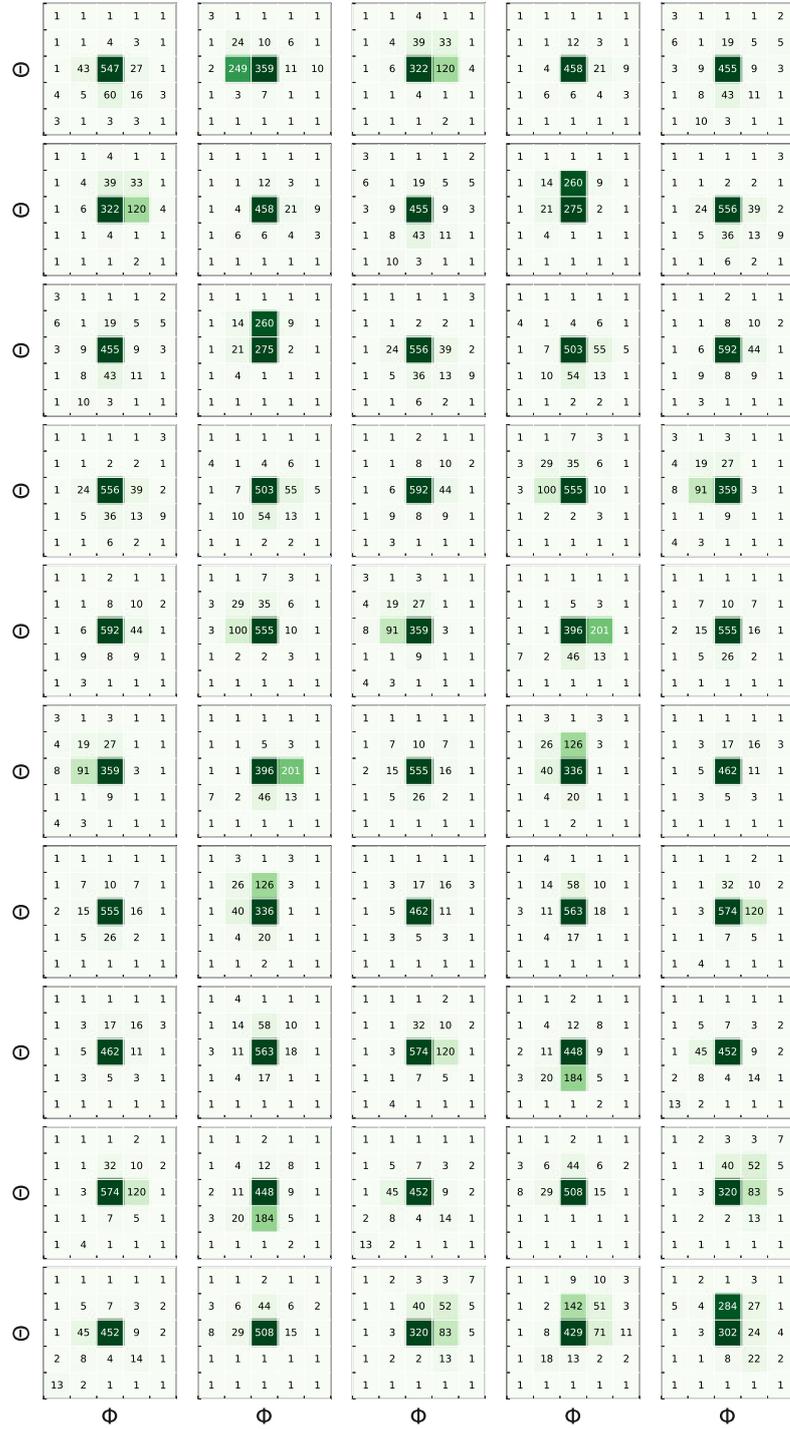


Figure 6.8: 50 random reconstructed photonic shower shape images in MeV scale.

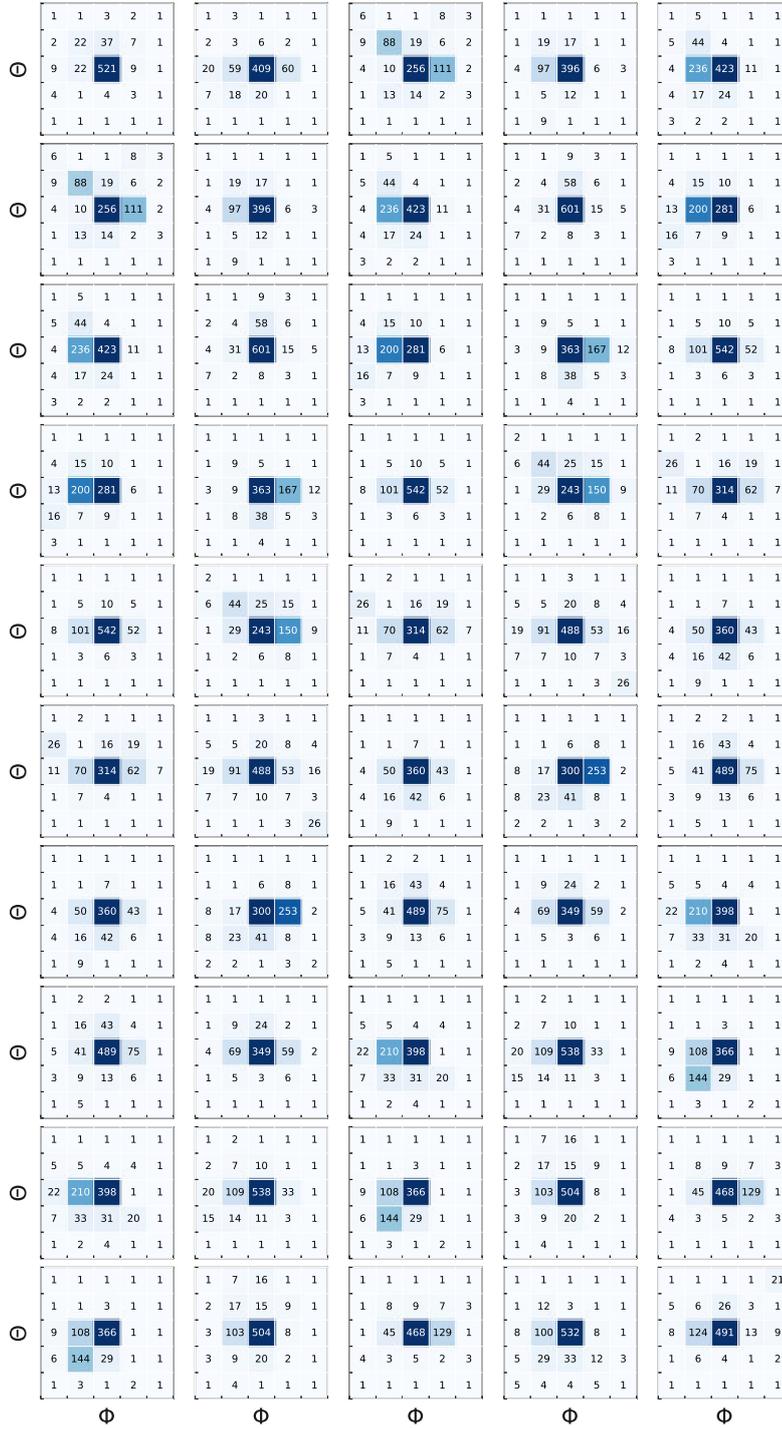


Figure 6.9: 50 random reconstructed electronic shower shape images in MeV scale.

## Results

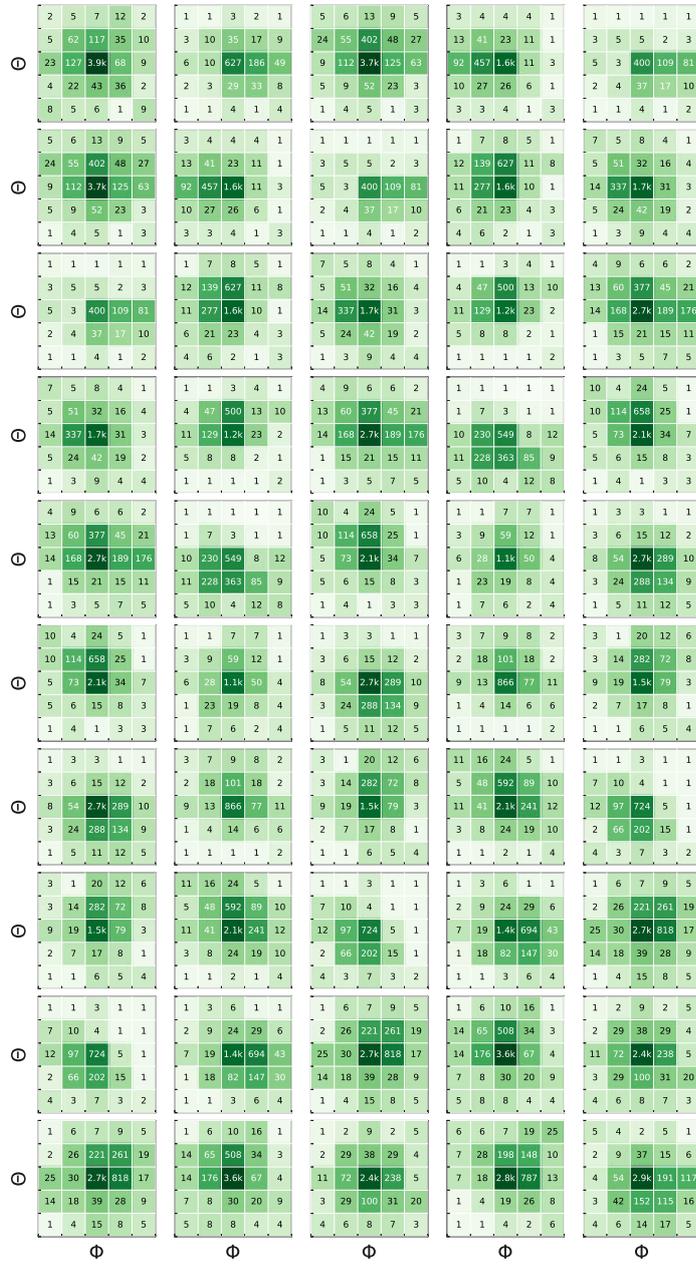


Figure 6.10: 50 random GAN photonic shower shape images in log10 scale.

## 6 Outlook

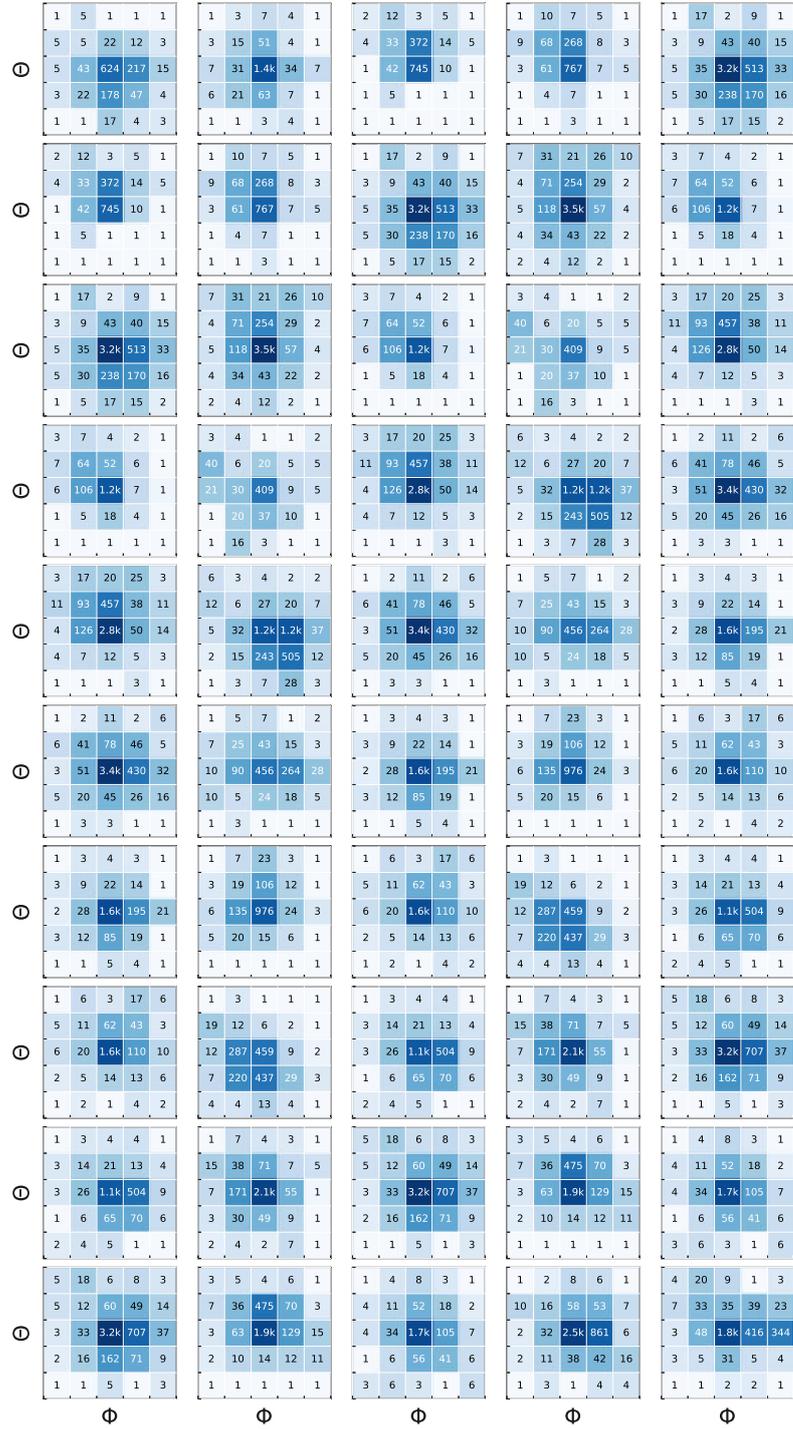


Figure 6.11: 50 random GAN electronic shower shape images in log10 scale.

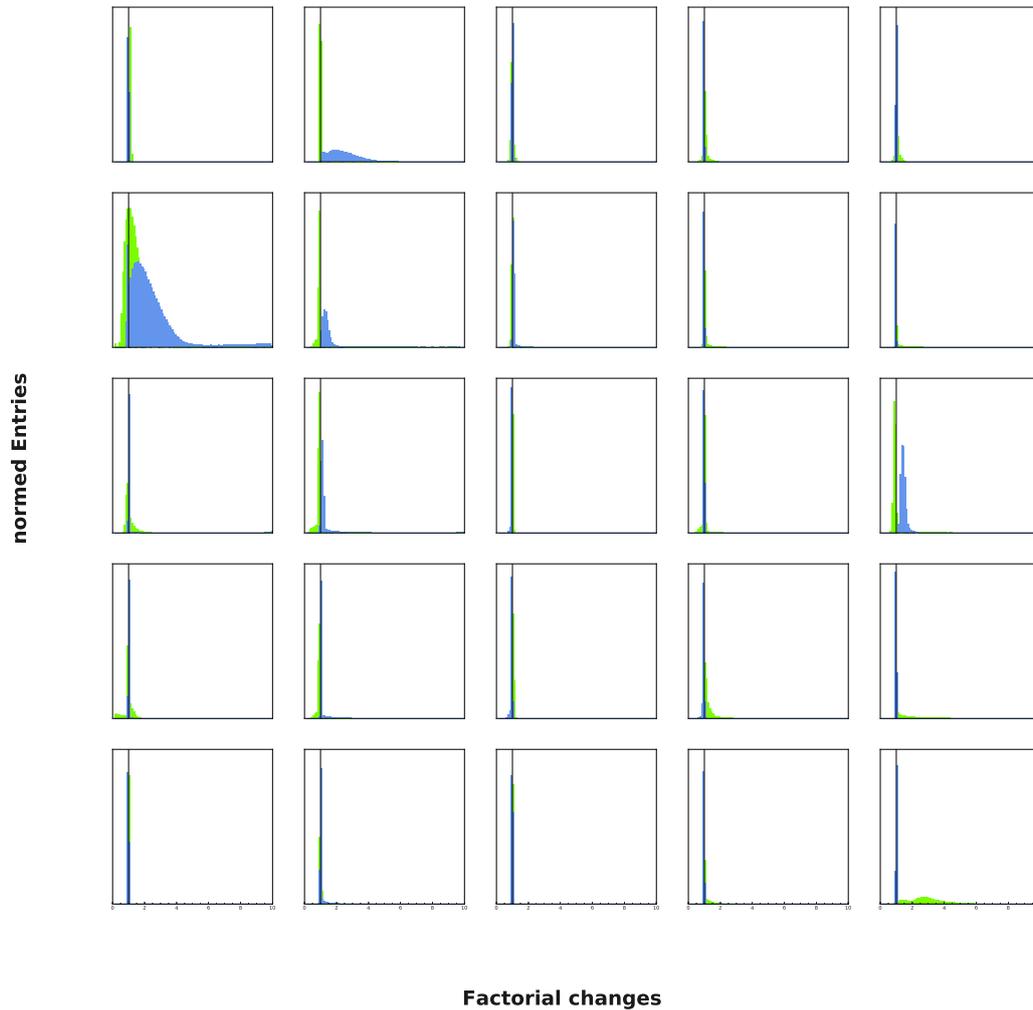


Figure 6.12: Image value difference for each pixel between GAN particle to its reconstructed input particle. The blue distribution calculate the difference between GAN electrons to reconstructed input photons. Respectively, the green distribution calculate the difference between GAN photons to reconstructed input electrons. The x-axis gives the factorial change between pixel value of GAN images and reconstructed input particle image. The black line marks the value one, where no changes are made.

## Outlook

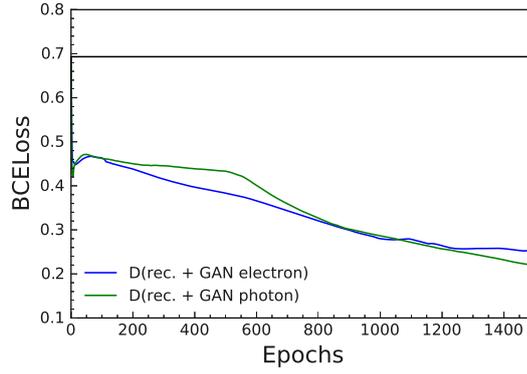


Figure 6.13: CycleGAN only trained on simulated particles with  $E_{MC} = 1$  GeV. Binary-cross entropy loss of both discriminators for reconstructed and GAN particles during training.

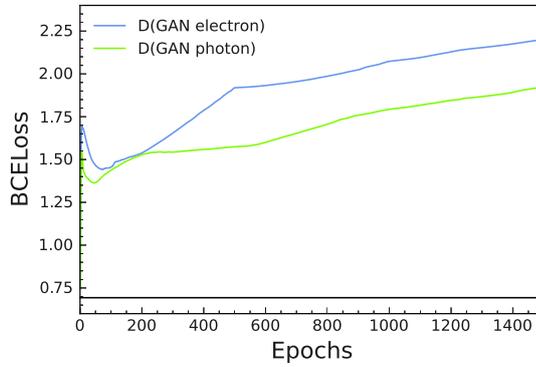


Figure 6.14: CycleGAN only trained on simulated particles with  $E_{MC} = 1$  GeV. Binary-cross entropy loss of both discriminators for GAN particles during training.

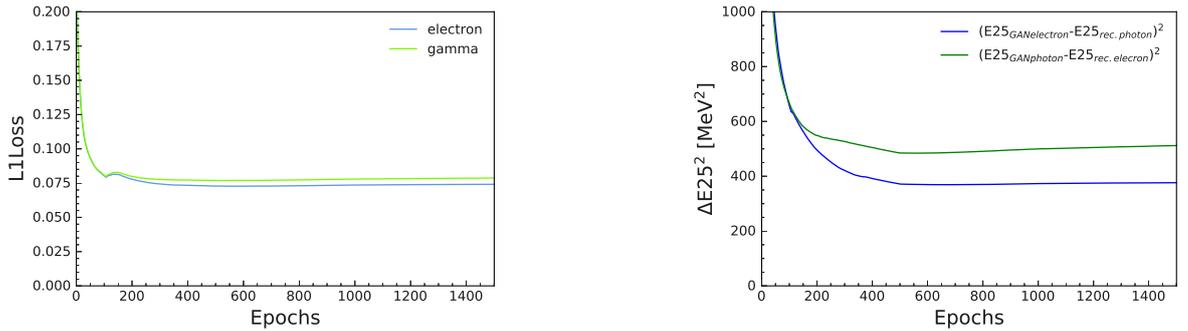


Figure 6.15: CycleGAN only trained on simulated particles with  $E_{MC} = 1$  GeV. The cycle consistency loss over the epochs on the left side. The MSE loss of the total sum between GAN and reconstructed input particle image in  $\text{MeV}^2$  (right).

# Bibliography

- [1] Steven Weinberg. A model of leptons. *Phys. Rev. Lett.*, 19:1264–1266, Nov 1967.
- [2] Sheldon L. Glashow. Partial-symmetries of weak interactions. *Nuclear Physics*, 22(4):579–588, 1961.
- [3] A. Salam and J.C. Ward. Electromagnetic and weak interactions. *Physics Letters*, 13(2):168–171, 1964.
- [4] F. Englert and R. Brout. Broken symmetry and the mass of gauge vector mesons. *Phys. Rev. Lett.*, 13:321–323, Aug 1964.
- [5] Peter W. Higgs. Broken symmetries and the masses of gauge bosons. *Phys. Rev. Lett.*, 13:508–509, Oct 1964.
- [6] G. S. Guralnik, C. R. Hagen, and T. W. B. Kibble. Global conservation laws and massless particles. *Phys. Rev. Lett.*, 13:585–587, Nov 1964.
- [7] Wikimedia Commons. File:standard model of elementary particles.svg — wikimedia commons, the free media repository, 2020. [Online; accessed 22-December-2020].
- [8] Particle Data Group. Review of Particle Physics\*. *Progress of Theoretical and Experimental Physics*, 2020(8), 08 2020. 083C01.
- [9] V.C. Rubin, Jr. Ford, W.K., and N. Thonnard. Rotational properties of 21 SC galaxies with a large range of luminosities and radii, from NGC 4605 (R=4kpc) to UGC 2885 (R=122kpc). *The Astrophysical Journal*, 238:471–487, jun 1980.
- [10] Mrittunjoy Guha Majumdar. Cosmological special relativity: Fundamentals and applications. *Physics Education*, 12 2013.
- [11] P. A. R. Ade and et al. Planck2015 results. *Astronomy and Astrophysics*, 594, Sep 2016.
- [12] R. Essig, J. A. Jaros, and et al. Dark sectors and new, light, weakly-coupled particles, 2013.
- [13] G. W. Bennett, Bousquet, and et al. Final report of the e821 muon anomalous magnetic moment measurement at bnl. *Phys. Rev. D*, 73:072003, Apr 2006.
- [14] Matthew J. Dolan, Torben Ferber, Christopher Hearty, and et al. Revised constraints and belle ii sensitivity for visible and invisible axion-like particles. *Journal of High Energy Physics*, 2017(12), Dec 2017.

## Bibliography

- [15] F. Abudinén et al. (Belle II Collaboration). Search for axionlike particles produced in  $e^+e^-$  collisions at belle ii. *Phys. Rev. Lett.*, 125:161806, Oct 2020.
- [16] F. Abudinen et al. Search for axionlike particles produced in  $e^+e^-$  collisions at belle ii. *Phys. Rev. Lett.*, 125:161806, Oct 2020.
- [17] M. Aaboud, G. Aad, B. Abbott, and et al. Measurement of the photon identification efficiencies with the atlas detector using lhcb run 2 data collected in 2015 and 2016. *The European Physical Journal C*, 79(3), Mar 2019.
- [18] Michela Paganini, Luke de Oliveira, and Benjamin Nachman. CaloGAN: Simulating 3D high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks. *Physical Review D*, 97(1):014021, January 2018.
- [19] Martin Erdmann, Jonas Glombitza, and Thorben Quast. Precise simulation of electromagnetic calorimeter showers using a wasserstein generative adversarial network. *Computing and Software for Big Science*, 3, 01 2019.
- [20] Aishik Ghosh. Deep generative models for fast shower simulation in ATLAS. Technical Report ATL-SOFT-PROC-2019-007, CERN, Geneva, Jun 2019.
- [21] Erik Buhmann, Sascha Diefenbacher, Engin Eren, and et al. Getting high: High fidelity simulation of high granularity calorimeters with high speed, 2021.
- [22] Russ T et al. Synthesis of ct images from digital body phantoms using cyclegan. *Int J Comput Assist Radiol Surg.*, 2019.
- [23] Hamza AB. Zunair H. Synthesis of covid-19 chest x rays using unpaired image-to-image translation. *Soc Netw Anal Min.*, 2021.
- [24] T. Abe et al. Belle II Technical Design Report. Technical report, KEK, 2010.
- [25] E Kou, P Urquijo, W Altmannshofer, and et al. The belle ii physics book. *Progress of Theoretical and Experimental Physics*, 2019(12), Dec 2019.
- [26] Delahaye jean pierre. Lepton colliders at the energy and luminosity frontiers: Linear colliders & superb factories. *Journal of Physics: Conference Series*, 110:012009, 06 2008.
- [27] K. Nishimura. The time-of-propagation counter for belleii. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 639(1):177–180, May 2011.
- [28] R. Pestotnik, I. Adachi, and R. Dolenec et al. The aerogel ring imaging cherenkov system at the belle ii spectrometer. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 876:265 – 268, 2017. The 9th international workshop on Ring Imaging Cherenkov Detectors (RICH2016).
- [29] K Miyabayashi, V Aulchenko, B G Cheon, A S Kuzmin, D Matvienko, I Nakamura, V Shebalin, B Shwartz, Y Unno, Y Usov, A Vinokurova, and V Zhulanov.

- Upgrade of the belle II electromagnetic calorimeter. *Journal of Instrumentation*, 9(09):P09011–P09011, sep 2014.
- [30] MS Windows NT kernel description. [https://pdg.lbl.gov/2009/AtomicNuclearProperties/HTML\\_PAGES/141.html](https://pdg.lbl.gov/2009/AtomicNuclearProperties/HTML_PAGES/141.html). Accessed: 2021-02-02.
- [31] Cheon Byunggu. Performance of the belle ii calorimeter trigger system at the superkekb phase 3 run, 2 2020.
- [32] Claudia Cecchi. The upgrade of the belleii forward calorimeter, 2 2016.
- [33] Andreas Moll. The software framework of the belle ii experiment. *Journal of Physics: Conference Series*, 331:032024, 12 2011.
- [34] J. Allison, K. Amako, J. Apostolakis, and et. all. Geant4 developments and applications. *IEEE Transactions on Nuclear Science*, 53(1):270–278, 2006.
- [35] Torben Ferber. *Electromagnetic calorimeter reconstruction in belle ii*, 2019. Available at [https://indico.cern.ch/event/708041/contributions/3269704/attachments/1809200/2954059/2019\\_03\\_11\\_acat\\_ferber\\_final.pdf](https://indico.cern.ch/event/708041/contributions/3269704/attachments/1809200/2954059/2019_03_11_acat_ferber_final.pdf).
- [36] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks, 2017.
- [37] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.
- [38] Hao Dou, Chen Chen, and Xiyuan Hu et al. Asymmetric cyclegan for image-to-image translations with uneven complexities. *Neurocomputing*, 415:114–122, 2020.
- [39] C. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:623–656, 1948.