Karlsruhe Institute of Technology

ETP-KA/2022-02

Full Event Interpretation using Graph Neural Networks

Lea Reuter

Masterthesis

17th February 2022

Institute of Experimental Particle Physics (ETP)

Advisor: Prof. Dr. Torben Ferber
Coadvisor: Prof. Dr. Günter Quast

Editing time: 17th January 2021 – 17th February 2022

Karlsruhe Institute of Technology

ETP-KA/2022-02

Vollständige Ereignisinterpretation mit Graphneuronalen Netzen

Lea Reuter

Masterarbeit

17. Februar 2022

Institut für Experimentelle Teilchenphysik (ETP)

Referent:	Prof. Dr. Torben Ferber
Korreferent:	Prof. Dr. Günter Quast

Bearbeitungszeit: 17. Januar 2021 – 17. Februar 2022

This thesis has been accepted by the first reviewer of the master thesis.

Karlsruhe, 17th February 2022

.....

(**Prof. Dr. Torben Ferber**)

I declare that I have developed and written the enclosed thesis completely by myself, and have not used sources or means without declaration in the text.

Karlsruhe, 17th February 2022

.....

(**Lea Reuter**)

Contents

1. Introduction

The task of physics is to describe the phenomena of nature, and particle physics does so at the most elementary level. Currently, the most successful theory for this is the Standard Model (SM) of particle physics [1]. This theory has been confirmed and probed many times in recent years. Despite the success, there are discoveries that cannot be explained by the SM, like the observed matter-antimatter asymmetry [2], neutrino oscillations [3], or the nature of dark matter [4]. Collider experiments test the predictions and limits of the SM and search for new physics beyond the SM.

The Belle II experiment at the electron-positron collider SuperKEKB in Tsukuba, Japan, mainly operates at the $\Upsilon(4S)$ resonance, a bound state of b $\overline{\text{b}}$ quarks. The $\Upsilon(4S)$ resonance decays into pairs of B-mesons, which decay very rapidly in a large number of possible decay channels until finally, stable enough particles can reach the detector. By precisely measuring B-mesons and their respective decays, the Belle II experiment focuses on searches for dark matter, flavour physics, and CP-violation [5]. Precision measurements of rare B-meson decays including neutrinos are challenging, as the neutrinos are invisible for the detectors and thus information is missing. However, information on the tag-side B-meson of the $\Upsilon(4S) \to B\overline{B}$ event enables one to constrain signal-side B-meson decays of interest. This requires the correct reconstruction of the tag-side B-meson out of the finals state particles (FSPs).

Due to the large number of possible decay channels of B-mesons, there is a huge combinatorial space for the reconstruction, making an analytical solution infeasible. For this reason, machine learning algorithms are used. The current reconstruction algorithm used by Belle II is the `Full Event Interpretation` (`FEI`) [6], which uses multivariate methods in the form of Boosted Decision Trees. Although it has been successfully used in Belle II analyses [7–9], the FEI has several design constraints that limits its performance. First, it requires explicitly stated decay channels, limiting the branching fraction coverage [10]. Second, the stage-wise approach requires domain driven optimization in every step of the tree reconstruction, which is prone to error accumulation unlike any end-to-end solution.

In the previous work by I. Tsakidis [11], a novel neural network approach was proposed to reconstruct particle decays. The fundamental principle of this method is a compact representation of particle decays, enabling direct applications of deep learning to the problem using Graph Neural Networks. This method was tested on simple, idealized simulations of particle decays and a small subset of ground-truth Belle II simulations, not dealing with the reconstructed particle information.

This thesis expands on the previous work and explores if this representation can be applied to Belle II reconstructed particles, dealing with the experimental realities and large amounts of possible decay channels. In Chapter 2 the fundamental theory, as well as the experimental setup and the currently used `FEI` reconstruction algorithm are explained. Chapter 3 focuses on machine learning and tools used for this thesis. Chapter 4 describes the representation of the decay tree, followed by the neural network approach, network architecture, and the metrics to compare the existing `FEI` with this new approach. In Chapter 5, first studies on extending the number of different decay topologies with this method are performed on toy data. Based on these results, in Chapter 6 an approach to extend the representation to reconstructed Belle II simulated data is formulated and validated on specific sets of B-decays in Chapter 7. Finally, in Chapter 8 a first training on hadronic B^0-decays is performed and subsequently evaluated and compared to the `FEI` as a proof-of-concept. Chapter 9 concludes the results and gives an outlook for further studies needed to use this representation to develop a new Deep Learning Algorithm.

2. Belle II Experiment

This chapter gives a short introduction to the foundation of this work. Section 2.1 describes the Standard Model and motivation for Section 2.2, where B factories and the need for event reconstruction is outlined. In Section 2.3, the SuperKEKB collider and Belle II Detector are explained. Section 2.4 introduces the simulated samples for Belle II, used for the current reconstruction algorithm explained in Section 2.5.

2.1. Standard Model

The standard model of particle physics (SM) is currently the most successful theory to describe the physics of matter at the elementary level. Foundation of the SM are symmetry assumptions, since physics is supposed to be omnipresent independent of time. This results in conservation laws and interactions between the elementary particles. The following section is based on [1].

Elementary particles that make up matter are fermions. They are spin-$\frac{1}{2}$ particles that are further divided into six flavours of quarks and six flavours of leptons, which can be grouped into three generations, as seen in Figure 2.1.

The three fundamental forces of the SM are the weak, electromagnetic, and strong interaction, with their respective gauge bosons as force carriers. The strong interaction is described by quantum chromodynamics. The force carries are colour-charged gluons. Quarks also have a colour-charge. The strong interaction is subject to confinement which results in only colour-neutral particles being stable. It is energetically more favourable for colour-charged particles to create further particles, for example to form B-mesons out of a bound b $\overline{\text{b}}$-quarkonium pair. Electromagnetically charged particles interact through photons via the electromagnetic force. The weak interaction has the W and Z bosons as force carrier particles. It couples with the different quark flavours and enables a transition between the flavours of the particles, which is described as Flavour Physics. The transition probability is given by the Cabibbo-Kobayashi-Maskawa(CKM) matrix [13, 14]. The mass of the particles arises from the interaction with the scalar Higgs field. This was the last step in the development of the SM, fulfilling the symmetry requirements and making the SM coherent. The resulting Higgs boson was measured in 2012 [15].

The SM has been confirmed in collider experiments and is consistent with almost all experimental data. Nevertheless, there are also phenomena that cannot be described by

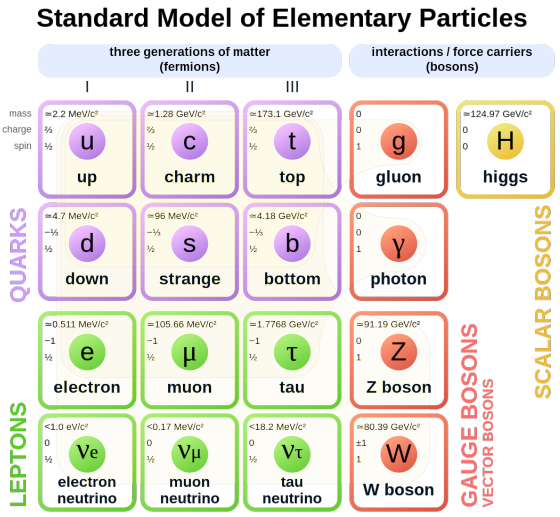


Figure 2.1.: Diagram of the SM, taken from [12]. The elementary particles are divided in elementary particles of matter, called fermions, and the force carriers, called bosons. For fermions, both quarks (purple) and leptons (green) are divided by their generations. The bosons are also divided in gauge bosons and the scalar Higgs boson. The mass (limits), charge and spin are noted for each particle according to [13].

the SM. The SM only includes three fundamental forces but does not explain gravity, the fourth fundamental force. Other observations that the SM does not describe are neutrino oscillations [3, 16], the observable CP-violation in the universe, and dark matter [4]. This shows that the current understanding is incomplete and that there is physics beyond the standard model which needs to be explored to broaden the current understanding of physics.

2.2. B Factories

B-factories operate an electron-positron collider at the $\Upsilon(4S)$-resonance, a bound b $\bar{\text{b}}$-quarkonium state. With an energy of 10.579 GeV, this state decays with a branching fraction of over 96% into B-$\overline{\text{B}}$-meson pairs. As there is only an energy difference of 20 MeV between the mass of the $\Upsilon(4S)$ and the two B-mesons are left, no other particles are produced. Regarding the center-of-mass of the $\Upsilon(4S)$-resonance, the B-$\overline{\text{B}}$-meson pairs are produced nearly at rest. Asymmetric beam-energies improve the resolution of the decay lengths, by Lorentz boosting the B-mesons to travel trackable distances before decaying. Together with the advantage of an electron-positron collider of avoiding pile-up, this results in a clean environment for precision measurements of the B-meson.

One of the original motivations for B factories was observing CP-violation predicted by the CKM matrix [17] in the B-meson system. The two B-Factory experiments Belle [18] and BaBar [19] measured the CP-violating phase ϕ_1 successfully in 2008, achieving a Nobel prize in physics for Kobayashi and Maskawa.

As previously mentioned, there are observations that are not explainable by the SM. To further probe the theoretical predictions of the SM for the rare decays with more precise measurements, and continue the achievements of the Belle experiment [20], the Belle II experiment was founded in 2008 and the collider was upgraded. With an even higher targeted luminosity 40 times higher than its predecessor, the Belle II experiment enables measurements of new physics, e.g. axion-like particles in dark matter or find divergences such as lepton flavor violation [5, 21].

B-tagging

Figure 2.2 shows an example for an initial collision of the electron e^- and the positron e^+.

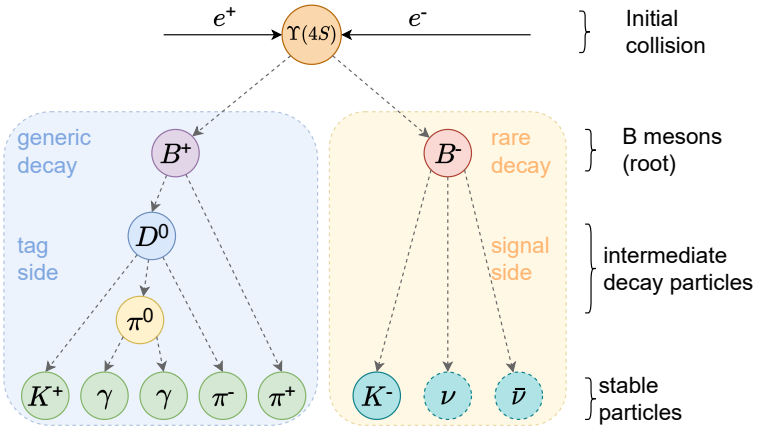


Figure 2.2.: Electron(e^-)-positron(e^+) collision resulting in a Υ(4S)-event that decays into two B-mesons. They are separated into the signal-(right) and tag-side(left) B-decay. The decay chains including the intermediate particles are shown until the stable particles are reached.

Measurements of rare signal decays that include invisible particles like neutrinos, e.g. $B^- \to K^- \nu\bar{\nu}$ in Figure 2.2 (right), are a challenging task as they leave no direct signature in the detector. Information of the full B decays is therefore missing. To constrain the information of the signal-side, the other B-meson, tag-side in Figure 2.2 (left), of the produced pair can be used [17]. The signal and tag-side B-meson can be combined to the full Υ(4S) event to maximize the information of the event and improve physics analyses. This is also called recoil B-meson reconstruction ([17] Chapter 7.4).

This is a challenging task, as B-mesons decay rapidly in large amounts of possible decay channels. Only stable enough particles, like for example e^-, μ^-, K^+, π^+, and γ, can be measured in the detector. These particles are referred to as final state particles (FSPs). There are three different approaches to B-tagging:

Inclusive reconstruction All particles that are left in the event are combined to reconstruct the B-meson, without consideration of any decay channels.

Exclusive reconstruction The goal is to explicitly reconstruct the B-meson, which is limited by the reconstruction efficiency of the stable particles. As there is a large number of

possible decay channels, the high multiplicity in the event decay makes it impossible to try out every combination of stable particles [6]. An analytical solution to event reconstruction is therefore intractable. Instead, machine learning algorithms are employed [6].

2.3. SuperKEKB and Belle II Detector

SuperKEKB is an electron-positron collider in Tsukuba, Japan, and is an upgrade of the previous collider to target a higher luminosity. It is operated at the $\Upsilon(4S)$-resonance with an asymmetric beam energy of 7 GeV and 4 GeV. This is achieved by two storage rings, the low-energy ring (LER) for the positrons at 4 GeV and the high-energy ring (HER) for the electrons at 7 GeV. A linear collider (linac) with a damping ring for positrons accelerates the respective particle bunches to be stored in these rings. The Belle II experiment is at the collision point. To detect as many decay products as possible, the detector is built with different detection approaches for the respective regions of the detector to optimize the coverage and detection rate. A schematic view of the detector is shown in Figure 2.3 and the sub-detectors are described based on [22]:

Vertex Detectors (VXD) The innermost detector is used for precise decay vertex reconstructing. As the beam pipe radius is only 10 mm, there is a need in the inner region for a large number of detection channels, to satisfy the high hit rates. This is achieved by the pixel detector (PXD). The outer region of the VXD uses a four-layer silicon strip detector (SVD).

Central Drift Chamber (CDC) The main track reconstruction detector of the BelleII experiment is composed of Helium-Methane gas filled wire chambers that enable fast tracking of the trails of gaseous ionization. Together with the VXD, the CDC is used to reconstruct the tracks of charged particles. Furthermore, it is used for particle identification (low energy particles that can not reach the next detector part) and as a trigger system.

Particle Identification (PID) To identify particles, the Time-of-Propagation counter (top) is used in the barrel region and the Aerogel Ring-Imaging Cherenkov Counter (ARICH) in the end-cap regions. The time and pattern information of the Cherenkov effects are used to discriminate between the different particles.

Electromagnetic Calorimeter (ECL) Scintillation crystals are used to measure the energy of photons and neutral particles, as well as identify electrons. The ECL together with the KLM sub-detector is also used to identify K_L^0 mesons.

Magnet A 1.5 T superconducting magnet is used to induce a magnetic field, enabling tracking measurements of the resulting curved trajectories.

K_L^0 and muon detector (KLM) This detector part is outside the solenoid magnet and consists of alternating detector and iron plates. Charged particles are detected by resistive plate chambers (RPC) in the out barrel region and scintillator strips for the inner barrel and end-caps.

The collected detector information is used to reconstruct the FSPs. Due to the resolution of the detector, the kinematic features as momentum and energy are smeared. Furthermore,

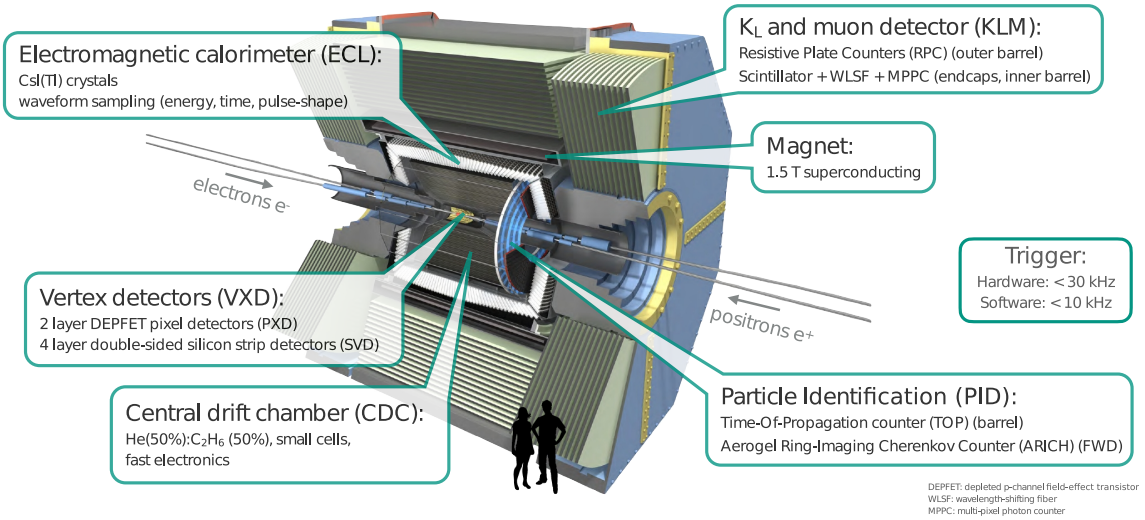


Figure 2.3.: Schematic overview of the Belle II detector at the collision point of the electron-positron collider, including the sub-detectors. Taken from [23]

particles can escape the detector either because they are not able to be detected or are outside of the detector acceptance. Besides detector efficiency and acceptance, there are other effects that also have to be taken into account when reconstructing the event:

Beam-induced background The drawback to the vertex reconstruction close to the interaction point (10 mm) and the higher luminosity of the Belle II experiment is that more beam-related background occurs. These particles are not part of the original e^--e^+ collision, instead they occur if other interactions are there at the beam. Examples for this are the synchrotron radiation occurring from the HER upstream and backscattering from HER downstream, bremsstrahlung and Coulomb scattering that change the beam momenta, resulting in potential shower particles or Touschek scattering as intra-bunch scattering from the LER or electron-positron pair conversion of photons. The full list of different beam-induced background effects is defined in [22].

Secondary particles Primary particles of the event generation are used as input for the detector simulation. Interactions with the detector can result in additional particles, for example photons in the TOP detector or in-flight decays of a K^0_S.

This is more challenging for the reconstruction as these background particles have to be taken into account. Additionally, not every electron-positron collision results in an $\Upsilon(4S)$-event. The highest cross section for non-$\Upsilon(4S)$ events is for $e^-e^+ \to e^-e^+$-events and quark-antiquark states [21]. This is an additional background that needs to be further suppressed when evaluating analyses on experimental data [24].

2.4. Belle II Simulation and Software Framework

Particle collisions do not occur deterministically. The Standard Model predictions can only be made about the statistical distributions of the various processes. Additionally, due to the systematic uncertainties of the detectors, not every particle can be detected correctly. For

these reasons, analyses are often developed on simulated data. Here, the analyst can take advantage of the true information and optimize the process and prevent biased analyses.

High energy physics (HEP) uses Monte-Carlo (MC) methods to simulate particle collisions. For Belle II, the $e^-e^+ \to \Upsilon(4S) \to B\overline{B}$ events are simulated by event generators, for example by `EventGEN` [25] and `PYTHIA8.2` [26]. Event generation includes the physical decay process up to the stable FSPs.

These generated events contain the primary particles, FSPs, and are used as input for the detector simulation. The detector simulation is performed by `Geant4`. The mentioned background effects in Section 2.3 are included in this step of the simulation. There is limited precision for these detector simulations regarding the exact hardware compositions and the multitude of physical processes that occur To improve these simulations, the mentioned effects in Section 2.3 are actively measured and updated in the simulations [27].

The Belle II Analysis Software Framework II (`basf2`) [28] is used to generate the simulation samples and is the software framework used for analyses.

2.5. Full Event Interpretation

The `Full Event Interpretation` (`FEI`) algorithm [6, 10] is the currently used event reconstruction algorithm for Belle II. The `FEI` is an exclusive reconstruction algorithm and can be further separated into hadronic and semileptonic reconstruction. Exclusive refers to explicitly reconstructing the B meson including intermediate particles. For the semileptonic reconstruction, B mesons are reconstructed for decays including a lepton and neutrino. Part of the information is missing as the neutrino is undetectable. For hadronic B-decays, in principle every FSP can be measured, and the full information can be used to constrain the decay. Hadronic tagging is limited by the low branching fraction of $\mathcal{O}(10^{-3})$ for a typical B decay and large combinatorics due to higher numbers of FSPs compared to semileptonic decays, which complicates the reconstruction. It depends on the analysis whether one wants to trade more information for significantly lower efficiency.

The B-meson reconstruction utilizes a hierarchical approach, displayed in Figure 2.4. In a six-stage approach, the intermediate particles are reconstructed step-by-step:

Stage 0 The final state particle candidates e^-, μ^-, K, π, (p, K_L^0) and γ are reconstructed from the collected detector information of the tracks, displaced vertices and neutral cluster entries in section 2.3.

Stage 1 J/ψ, (Λ) and π^0 candidates are reconstructed out of the FSPs of stage 0.

Stage 2 K_S^0 (and Σ^+) candidates are reconstructed including the previous stages.

Stage 3 D (and Λ_c^+) candidates are reconstructed

Stage 4 D^* candidates are reconstructed

Stage 5 B candidates are reconstructed

The `FEI` consists of a multivariate classification in the form of Boosted Decision Trees for each stage. The decay channels for Λ and Σ baryons, as well as the inclusion of p as FSPs

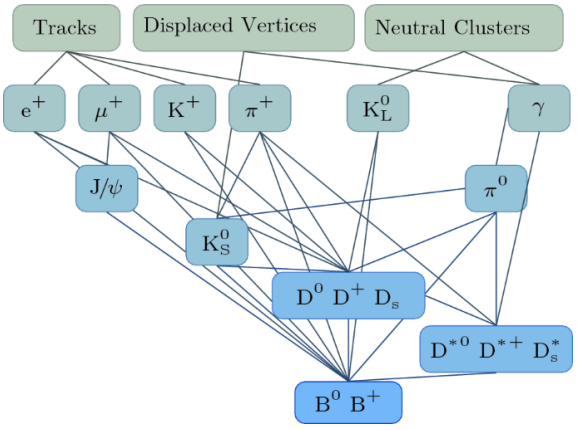


Figure 2.4.: Diagram of the six-stage approach of the FEI, taken from [6]. Starting with the detector information, for each stage the particle candidates are combined until the final stage reconstructs B-mesons candidates.

were added in later developments of the FEI. For the comparative dataset in this work, the FEI is trained without the baryons and without including K_L^0 due to the low efficiency.

Approximately 100 decay channels are explicitly stated, resulting in $\mathcal{O}(10\,000)$ distinct decay chains. Although this is a large number of possible decays, the total branching fraction is restricted to around 15 % [29]. Including detector uncertainties leading to imperfect particle reconstruction and even missing particles, the reconstruction efficiency of correctly reconstructed B decays is only 0.46 % for hadronic B^0 and 2.04 % for semileptonic B^0 [6] decays.

The `FEI` gives a list of possible B-meson candidates that the user can choose from. They are differentiated by the signal probability value that is assigned to each B candidate by the stage 5 classifier. This signal probability can also be used as a discriminator to gain a purer sample, but loses efficiency in the process. There are two different methods to train the FEI:

Generic FEI The `FEI` is trained on generic $\Upsilon(4S) \to B\overline{B}$ MC events without regarding a specific signal side. Therefore this `FEI` can be applied to all signal-side analyses.

Specific FEI The tag-side of a specific signal side is used to train the FEI. This is advantageous to the generic FEI, as the training is performed on signal-constrained background. Therefore it can improve the purity of the signal.

3. Machine Learning

Machine learning is used in a variety of HEP analyses [30, 31] and plays a prominent role in the search for new physics by analyzing large amounts of data generated by accelerator experiments. Often, signal and background events have similar kinematic properties, which makes it difficult to distinguish them. Instead of applying individual cuts on single analysis variables, like traditional cut based analyses do, machine learning enables one to recognize structures and patterns in the data by learning from examples.

Deep learning is inspired by the way that neurons process information in the human brain. Similarly, information gets combined and processed by the neurons of an artificial Neural Network to recognize patterns. Section 3.1 describes the basic building blocks of Neural Networks and the training process. Due to the increasing computational power for Graphic Processor Units (GPU), it is possible to train more complex models with end-to-end trainable approaches, which enables representation learning [32]. Section 3.2 focuses on tools used for deep learning approaches. Graph Neural Networks in particular are becoming more and more interesting, as they are suitable for evaluating non-Euclidean data structures relevant in particle reconstruction [33]. Section 3.3 gives insight to Graph Neural Networks.

3.1. Neural Networks

Artificial neurons are the building blocks of Neural Networks (NNs) and are based on Rosenblatt Perceptron [34], which is shown in Figure 3.1. The n inputs transmit the information to the neuron. The perceptron processes the information by multiplying each of the inputs x_i with weights w_{ij}, that determine the contribution of the respective input. To transmit the information to the next neuron, an activation function $f(x)$ is used to activate the neuron. They are non-linear functions, the simplest example for this is a step-function that can be used to set the neuron output either "ON" or "OFF". This activation value can be further shifted by a bias b, to activate dependent on the summed value, resulting in the neuron output:

$$x_j = f\left(\sum_i^n x_i w_{ij} + b_j\right). \tag{3.1}$$

These artificial neurons are the basic elements of the layers for a Multilayer Perceptron (MLP). An MLP is a basic neural network and consists of the input layer, hidden layers, and an output layer as shown in Figure 3.2. The input layer does not perform any calculations,

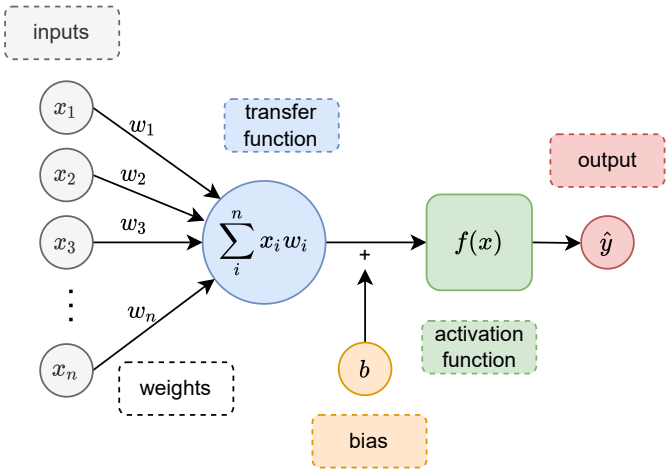


Figure 3.1.: Structure of an artificial neuron, the perceptron. Information is processed by multiplying the input values x_i with the weights w_i, and adding the bias b. Transformation with the activation function results in the output value of the neuron.

but defines the information used for the network, e.g. the experimental data, that the network is learning from. The successive layers feed the information into the next one in the forward direction, therefore these networks are often called feed-forward networks. The hidden and output layers are computational layers. For a fully-connected MLP, each neuron output is connected to all successive neurons in the next layer. The final layer of the network, the output layer, classifies the sample according to the user specified classes N. This last layer is a linear layer, where usually a softmax is applied to interpret the output as a probability:

$$\hat{y}_n = \frac{e^{x_j^l}}{\sum_i^N e^{x_i^l}}. \tag{3.2}$$

Supervised learning takes advantage of samples that are already classified correctly, thus called labeled samples. Propagating one sample through the network, the output of this sample $\hat{y}$ can be compared with the true label, called target, y. The weights w_{ij}, as well as the biases b, are learnable parameters a of the model, that determine the output of the network. The learnable parameters are adjusted by *backpropagation* [35] in the training steps. A loss function L is used to quantify the prediction quality. In this work, the cross entropy loss is used:

$$L(y, \hat{y}) = -\frac{1}{N} \sum_{n=1}^{N} \left(y_n \log \hat{y}_n + (1 - y_n) \log(1 - \hat{y}_n) \right). \tag{3.3}$$

If the model predictions are correct, then the loss function value is zero. The aim for the training is, that the loss function converges to zero, so minimizing the loss function. This is achieved by calculating the gradient descent with respect to the learnable parameters for the loss function value $\nabla L(\hat{y}, y)$. This is done recursively for each layer, starting with the

last layer l, to speed up the computation time. For each of the learnable parameters the respective gradient is weighted with the learning rate η and then subtracted:

$$a_i^{k+1} = a_i^k - \eta \frac{\partial L}{\partial a_i^k}. \tag{3.4}$$

The learning rate is defined between zero and one and determines the increment for the loss function minimization per iteration k. This forward and backward calculation is repeated for several iterations, called epochs until the lost function is below a certain value. The learning rate, as well as the number of hidden layers and nodes, are hyperparameters that have to be tuned by the user. They are not determined by the training process and require further optimization. If the learning rate is small, the training process needs a large number of epochs to converge, as the training is processing in small steps. Choosing a learning rate that is too high can yield to the network not converging, as too large values can make it impossible to reach a minimum. To improve this, the `adam` optimizer [36] can be used when calculating the updating step as an extension to gradient descent. Here the learning rate is adjusted according to the momentum as defined in [36] enabling better minimization.

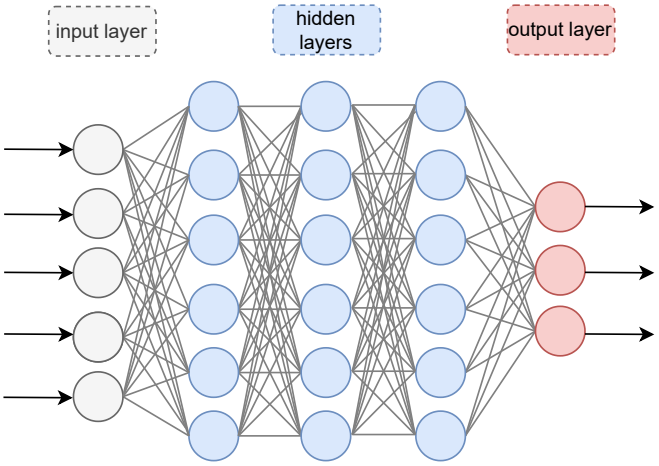


Figure 3.2.: Structure of a Multilayer Perceptron as a basic building block of Neural Networks. It consists of several nodes per layer. The input layer feeds the information to the hidden layers, that process the information and forwards it to the next layer. The output layer is the last layer that produces the final output values, in this case for multiclassification.

As most data includes some noise, these update steps are not calculated on every single available sample but bundled into batches. This averages the gradient over the batch but also results in the need to increase the learning rate to adapt for this. To further speed up the training, batch normalization can be used [37].

Often in HEP, there is a difference in class frequencies for the labeled samples. This issue is referred to as imbalanced classes and can result in the model only learning to predict frequent classes. To counteract this, the classes can be weighted according to their frequency when calculating the loss, so that minority classes contribute higher to the loss function.

This can be achieved either by calculating the class weights on the training sample or by using the dynamically scaled focal loss [38] based on cross entropy loss.

3.2. Deep Learning

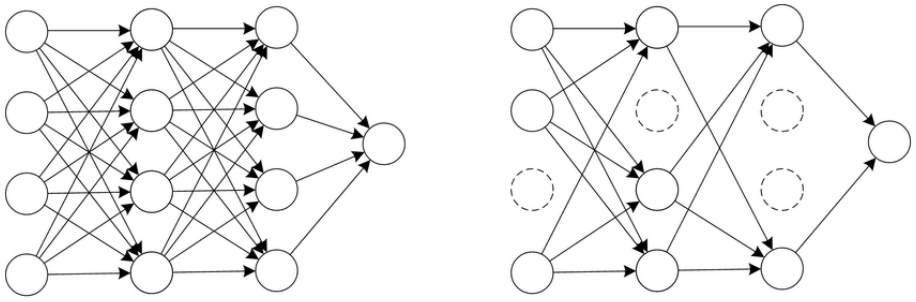
The higher computing capacities enabled even larger and more complex models. This enables the model to extract more information from the training samples and to learn representations of the data [32]. Deeper models consist of various MLPs with large feed-forward layer widths. To enable trainings for these deeper structures and large amounts of learnable parameters, the following effects have to be considered.

Overtraining

It can happen that the model is not learning a generalized, deeper representation of the training samples but instead only memorizing it. This is called overtraining.

To prevent the model from overtraining, the model performance is also evaluated on a validation dataset, that is statistically independent of the training dataset. This validation set is not used for the backpropagation but to monitor the overtraining, showing the ability of the model to generalize the data. If the performance on the training and validation dataset is diverging, the training is stopped, as this indicates that the model is starting to memorize the training dataset. This is referred to as early-stopping, as the training finished earlier than the stated number of epochs to prevent overtraining.

Another technique to prevent overtraining is to regularize the training by dropout [39]. Especially for fully-connected layers, to prevent the model from learning co-dependencies of the nodes, for each iteration in the training, there is a probability p that a node is "dropped out". The resulting model in the training step is with fewer nodes and edges as shown in Figure 3.3. Dropping a percentage of the model nodes forces the network to learn deeper representations and more robust patterns.



(a) Standard Neural Network (b) Network after Dropout

Figure 3.3.: Neural Network without dropout on the right and the reduced network after dropout on the left, showing the nodes that are dropped with dotted lines. Taken from [40].

Vanishing Gradients

Another problem when training deeper models is vanishing gradients. A saturating activation function compresses the node output in a compact range but is therefore only sensitive for changes around the mid-point. If the value of the saturating activation function differs too much from the mid-point, the derivatives [41] are small, thus the node is saturated. For the first layers of the model, calculating the gradient descendants by chain rule and multiplying these small derivatives can result in gradients close to zero. This results in the learnable parameters of these layers not being updated and remaining nearly untouched over several epochs. To prevent this from happening, different methods are used.

Therefore, different activation functions, based on rectified linear units (ReLUs) [42] are used, as the derivation is not saturated. Even further improvements can be achieved by using an exponential linear unit (ELU) activation function [43].

$$R(z) = \begin{cases} z, & z > 0 \\ \alpha(e^z - 1), & z \leq 0 \end{cases} \tag{3.5}$$

As it can achieve negative values the mean evens for zero, this speeds up the training process. This also improves the training as more stable and robust. Despite the more complex calculation, using the ELU activation function achieves faster convergence in fewer epochs [43].

Furthermore, initializing the weights correctly also reduces the risk of the learnable parameters not being adjusted due to saturated nodes [41]. Skip connections, that "skip" the gradient calculation for one layer, also prevent vanishing or exploding gradients [44]. By skipping layers, there is an alternate path for the gradient calculations for the earlier layers. The skip connections in this work are achieved by concatenating the features of the earlier layer with the later layers [45]. They enable the reusability of features and stabilize the convergence of the training. Based on these tools, the network structure used in this work is defined in section 4.2.

3.3. Graph Neural Networks

Deep learning enables to extract complex information and hidden structures out of data samples but is limited to Euclidean or grid-like data. Countless data, as clusters in social networks, molecules in biology, language, or particle decays are represented by graphs, that represent objects (the nodes) and the relations between these objects (edges). This is why Graph Neural Networks (GNNs) focus on directly applying these graph data structures to machine learning methods such as representation learning, taking into account both node and edge information [46]. The most successful GNN architecture for representation learning is transformer networks in language processing [47]. This is the baseline GNN that is used later in this work in Section 7.3 to compare the network of this thesis in Section 4.2 with a benchmark network.

Successful applications in particle physics for jet reconstruction [33, 48] imply the ability of GNNs to learn the kinematics of particles. Neural relation of inference [49] further shows the ability to learn and predict the interaction dynamic between particles. Nevertheless,

current work in these fields focuses on either classifying the nodes or the complete graph and not the labels of the edges and the actual graph structure. Chapter 4 therefore outlines the problem for particle decay reconstruction and introduces an approach to represent the actual graph structure to predict the relations, as well as the GNN model architecture used in this thesis.

4. Graph-based Full Event Interpretation

To learn the structure of the decay tree, a suitable representation of the structure is needed. A reconstruction algorithm has to deal with scenarios where the full decay tree is a-priori unknown and only information about stable, detected particles is available. The challenge is to predict the tree structure using only these available particles. Further challenges include the different multiplicities in event decays, and hence handling a variable number of particles. This results in an explosion of possible combinations for the detected particles and therefore needs an empirical approach to solve this challenge, as an analytical approach is intractable.

The currently used reconstruction algorithm, `FEI` (Section 2.5) is restricted by its branching fraction coverage. The FEI reconstructs B-mesons by explicitly stating the distinct decay processes, leading to over $\mathcal{O}(10\,000)$ reconstruction decay chains. This restricts the branching fraction coverage to approximately 15%. Furthermore, the reconstruction uses six distinct stages to reconstruct the B-meson step by step, starting with the final state particles and combining them to intermediate particles until potentially B-meson candidates are proposed. This hierarchical structure requires domain driven optimization in every step of the tree reconstruction. This is prone to error accumulation, as the later stages rely heavily on the performance of the previous ones.

To improve the B-meson reconstruction, an end-to-end trainable network, that can exploit the full branching fraction coverage, is proposed. Having an end-to-end trainable network offers better and easier optimization than the setup of the current reconstruction algorithm FEI. The goal of this thesis is to study and evaluate a novel graph neural network based approach for event reconstruction in a realistic experimental environment. The final objective is to compare the reconstruction of this approach with the currently used FEI algorithm. This thesis continues the previous work by I. Tsaklidis [11].

The approach to this problem of specifying the decay structure by the varying number of available particles is described in Section 4.1. This structural representation defines the training target of the graph neural network in this method. Section 4.2 outlines the network architecture and functionality to predict the decay tree structure. Finally, the metrics to evaluate the novel neural network approach to event reconstruction are explained and compared with the existing reconstruction algorithm FEI in Section 4.3.

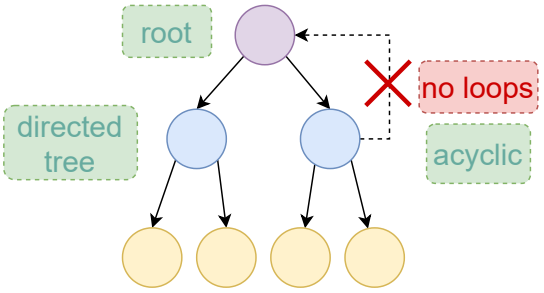


Figure 4.1.: An example graph including the requirements (in green) to correctly represent the structure of a particle decay according to physical laws.

4.1. Decay Tree Structure Representation

The **graFEI** (**gra**ph-based **F**ull **E**vent **I**nterpretation) approach was proposed and evaluated in [11]. The idea that a particle decay tree is naturally represented as a graph is the fundamental principle of the graFEI method. A graph is defined by objects (nodes $[v_1, v_2, ...v_n] \in \mathcal{V}$) and the connections between them (edges $[e_{1,2}, e_{2,3}, ...e_{i,j}] \in \mathcal{E}$). The structure of the graph is defined by the adjacency matrix $\mathbf{A}$ of dimension $n \times n$, where n is the number of nodes. Each entry for the adjacency matrix A_{ij} is equal to one if there is a connection between two nodes of $e_{ij} = (v_i, v_j)$, otherwise, it is zero. A set of features of the nodes and edges can form the respective feature matrix, that fully defines the graph together with the adjacency matrix. Nodes represent the particles in a decay process, and the edges correspond to the hierarchical decay relations. In the case of the decay tree the simplest set of node features can be the four momentum $p_i = (E_i, p_{\mathrm{x},i}, p_{\mathrm{y},i}, p_{\mathrm{z},i})$ of the respective particle v_i. In this work edge features are not considered, as they do not apply in this case.

Further requirements about the graph can be made when describing particle decays. Heavy particles produced in the collisions decay into lighter child particles. This means that the decay (tree) is directed from the heavy root particle (rooted). From a physical point of view, two nodes can only be connected by one edge and there can be no loops in the decay tree (acyclic). An example graph that meets these requirements is shown in Figure 4.1.

The experimental reality is that only a part of the tree can be identified, as heavy particles produced in the collision decay before reaching the detector. This results in decay chains where the B particles that were initially produced decay into daughter particles, like D^* or D, that can further decay into granddaughter particles like K^0_S and so on. Only e^-, μ^-, K^+, π^+, p, and γ particles are stable enough to reach the detector and be measured (Section 2.3). These measured particles are referred to as final state particles (FSPs). As a consequence, the complete set of nodes is unknown and the adjacency matrix is incomplete. In the case of event reconstruction, a different, equivalent representation is needed to define the decay tree structure. This solution takes advantage of the graph requirements stated above to create such a representation that corresponds to the problem. The adjacency matrix is encoded in the lowest common ancestor (LCA) matrix. The LCA matrix includes

all the information of the adjacency matrix and can therefore be used to determine the tree structure [50].

An example of an adjacency matrix encoded in the LCA matrix is shown in Figure 4.2 for the decay $\mathrm{B}^+ \to \overline{\mathrm{D}}^0(\to \mathrm{K}^+\pi^-\pi^0(\to \gamma\gamma))\pi^+$, with the B^+-meson as the root of this decay tree. The size of the LCA matrix $\mathbf{L}$ is defined by the number of terminal nodes, which in this case are the FSPs $n_{\mathrm{fsp}} \times n_{\mathrm{fsp}}$. The entries in the LCA matrix L_{ij} are defined by the lowest common ancestor the two nodes $v_{\mathrm{fsp},i}$ and $v_{\mathrm{fsp},j}$ share. In the example of Figure 4.2, the two photons γ have the common ancestors π^0, D^0 and B^+ in ascending order to the root node, with π^0 being the lowest. The LCA matrix encodes the tree structure in a condensed matrix, satisfying the experimental requirement of only having the final state particles available.

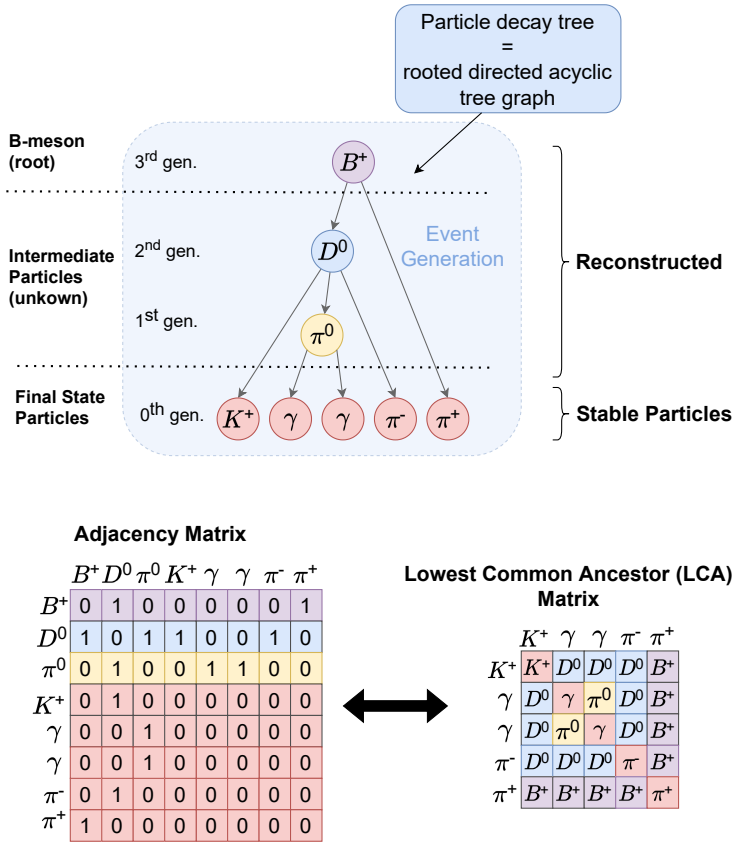


Figure 4.2.: Encoding of the decay tree as the Lowest Common Ancestor (LCA) Matrix for an example decay $\mathrm{B}^+ \to \overline{\mathrm{D}}^0(\to \mathrm{K}^+\pi^-\pi^0(\to \gamma\gamma))\pi^+$. The upper figure shows the decay tree from the root node B^+ to the final state particles. The adjacency matrix is shown on the bottom left and the Lowest Common Ancestor matrix on the bottom right. The colors refer to the particles, with red corresponding to the FSPs, yellow and blue the intermediate particles and violet the root particle. For the adjacency matrix, they note the particle type on the rows; for the LCA matrix they correspond to the particle over which the two FSPs are connected first.

The LCA matrix needs an appropriate representation for machine learning. Predicting each intermediate particle individually, by assigning each intermediate particle a distinct class, would lead to a high necessary number of classifications. This is due to the high number of different particles including excited states in the B-decays of the $\Upsilon(4S)$ events (D^*, D, D_s, D_s^*, J/ψ, ...). Both [11] and the studies in this thesis show that the predictive capacity of the network decreases with a high number (> 10) of distinct classes. For this reason, a simplified representation is used to further classify the intermediate particles. In this work, I used two different methods to assign classes to the LCA matrix.

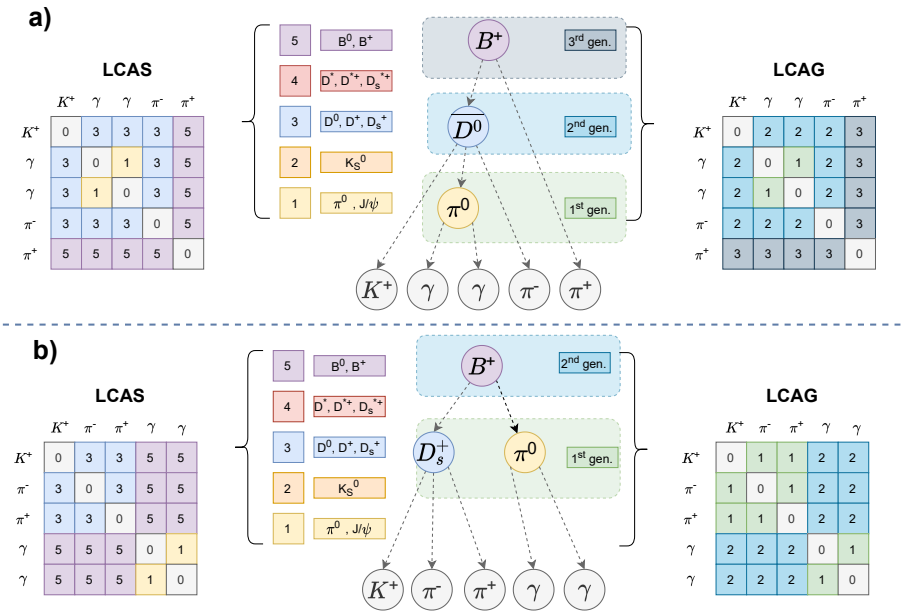


Figure 4.3.: Example on how to build the LCAS matrix (*left*) with the 5 respective stages and the LCAG matrix (*right*) using the generational view. Two different decays a) $B^+ \to \overline{D}^0(\to K^+\pi^-\pi^0(\to \gamma\gamma))\pi^+$ and b) $B^0 \to D_s^+(\to K^+\pi^-\pi^+)\pi^0(\to \gamma\gamma)$ with the same FSPs are shown to compare these representation methods.

LCAG (generational view)

The method used in [11] is the lowest common ancestor generation (LCAG) representation. Each intermediate ancestor particle is defined by its generation in the decay tree, meaning that with an entry of one the two row and column particles share a mother, and with an entry of two they share a grandmother. For the example decay, the LCAG encoding is shown in Figure 4.3 (right) for two different decays resulting in the same FSPs.

LCAS (stage view)

In a new approach, which is introduced in this work, particles are directly assigned to classes. This is motivated by the reconstruction stages of the FEI (Section 2.5) and the physics of the decay. Intermediate particles of the higher stages can only decay into lower stages or the FSPs. By assigning classes equivalent to the reconstruction stages of the FEI (Section 2.5), the directed, acyclic structure of the tree is ensured. Therefore, this approach is called the lowest common ancestor stage view (LCAS).

The definition of the stages, including the anti particles, is:

1 $\equiv \pi^0$, J/ψ

2 $\equiv K_S^0$

3 $\equiv D^0$, D^+, D_s^+

4 $\equiv D^{*0}$, D^{*+}, D_s^{*+}

5 $\equiv B^0$, B^+

An example for this is shown e.g. in Figure 4.3 (left). For intermediate particles not included in the list above, for example resonances, the particle is skipped in favor of a higher common ancestor that is included in the list above.

4.2. Graph Neural Network Approach

There are challenges to event reconstruction using NNs, that have to be accounted for when building reconstruction algorithms. The requirements that the reconstruction algorithm must meet are listed below.

Detected FSPs:
Only the detected particles can be used as FSPs to reconstruct the root node (B-meson).

Variable number of FSPs:
The number of FSPs varies greatly, as there is a large number of possible decay channels for a B-decay. Hence the reconstruction algorithm has to be able to handle a varying number of FSPs. Additionally, the method has to be invariant under the permutation of the final state particles, as the order of particles is unknown.

Unknown depth and structure of the tree:
The large number of decay channels leads to unknown intermediate particles and an unknown tree structure. This results in an explosion of possible combinations for decays including large numbers of FSPs and large numbers of possible intermediate particles.

The previous section Section 4.1 explained how the tree structure can be encoded only using the available FSPs. The full training cycle is shown in Figure 4.4. The goal now is to predict the edge labels. For this purpose, a fully connected graph is built from the FSPs. In this way, no prior assumption has to be made. Since the particles follow physical laws, the kinematic properties can be exploited to predict the edge labels. Using NNs enables one to make complex assumptions to predict the particle relationships. The input for this additional graph is therefore the feature matrix of the FSPs. Since a separate graph is built for each decay and each edge is predicted individually, this method is adaptable to any number of FSPs. Furthermore, the order of the particles is irrelevant for the prediction, since one can permute the LCA and the conversion to the adjacency matrix does not change [50]. The feature matrix can be used in a later step to determine the features of the intermediate particles, resulting in the reconstructed decay tree.

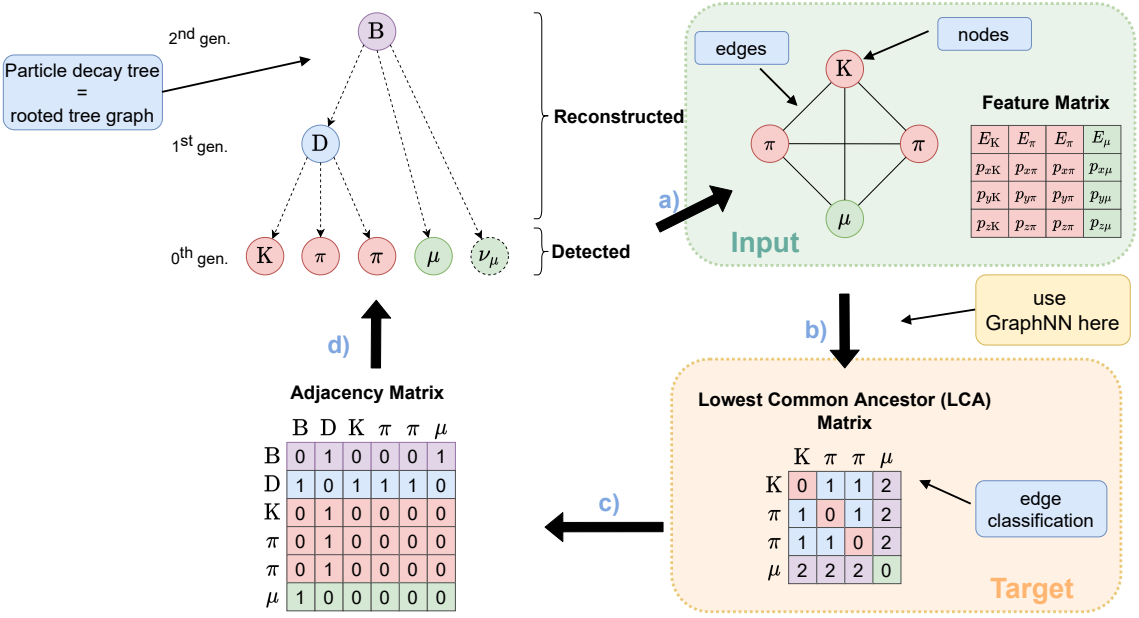


Figure 4.4.: Training cycle of the graFEI approach. The decay tree is shown in the top left. As only the detected final state particles are known, an additional graph is built out of these particles (a). This is a fully connected graph, including particle features, for example, the reconstructed four momenta of the particle. This serves as the input for the graph neural network (b). On the bottom right, the lowest common ancestor matrix is shown, which functions as the training target. Based upon this the adjacency Matrix (c), stating the structure of the tree, can be calculated. Finally, the decay tree can be reconstructed using the adjacency matrix and the feature matrix (d).

Training on a single training target enables the possibility to exploit the full simulated data coverage for the training, removing the necessity to explicitly state the decay channels, as this limits the covered branching ratio.

The network architecture is based on a modified GNN encoder described in the *Neural Relational Inference for Interaction Systems* [49]. The idea of the NRI model is to alternate between node and edge representation to create an optimal deeper representation of the features. The most important two layers are therefore the Node-to-Edge and Edge-to-Node layers.

An overview of the modified NRI model is shown in Figure 4.5. The layers that are used are a multilayer perceptron (MLP) consisting of 2 linear layers with an ELU activation function [51] and dropout [39] followed by a batch normalization layer [37] as defined in [49]. The model employs residual connections [44], also called skip connections, to avoid vanishing gradients (Chapter 3). Unlike the official NRI model, there is the possibility to switch more often between the edge and node representation for the modified NRI, by defining the number of blocks. The original NRI also only uses one MLP between each Node-to-Edge/Edge-to-Node layer, whereas here the option to add additional MLPs (**additional**

MLP layers) is given. The original model only uses one skip connection between the second and the final MLP, but to compensate for the larger network architecture in this modified NRI more skip connections are applied in each block. The original layer has a feedforward dimension of 256 for its MLPs. In this work, larger dimensions will be studied as well (**dimension feedforward**).

The steps in Equation (4.1) to Equation (4.4) is the message passing as defined in [49]. This can be done in parallel for all nodes and edges.

Embedding

First, the node features x_i are embedded into a different representation; the number of representations is defined by the user by setting the number of hidden nodes (width) in the feedforward layers. This new representation is from now on referred to as node representation v_i. This first node representation v_i^1 only depends on the features of one particle.

$$v_i^1 = f_{\text{emb}}(x_i). \tag{4.1}$$

Node-to-Edge

After the embedding, a new edge representation can be calculated. Through an edge $e_{(i,j)}$ the node representation of the two corresponding nodes v_i and v_j are concatenated and the information of these two nodes then gets combined by the following MLP block (MLP1).

$$e_{(i,j)}^1 = f_{\text{MLP1}}([v_i^1, v_j^1]). \tag{4.2}$$

Edge-to-Node

The next step is to transition back to the node representation. For node v_i all corresponding edge features get summed up, excluding the self-interaction and divided by the number of edges for this node. As this is a fully connected graph, this is the size of the other particles $N_v - 1$. This second node representation v_i^2 now uses information of the full graph. Another MLP block (MLP2) is used to map to the node representations.

$$v_i^2 = f_{\text{MLP2}}(\frac{1}{N_{v,fsp} - 1} \sum_{j \neq i} e_{(i,j)}^1). \tag{4.3}$$

Node-to-Edge

As the edge labels are going to be predicted, the following step is to then go back to the edge representation.

$$e_{(i,j)}^2 = f_{\text{MLP3}}([v_i^2, v_j^2]). \tag{4.4}$$

The steps in Equation (4.3) and Equation (4.4) can be repeated according to the **number of blocks** n that is defined, until the final edge presentation $e_{(i,j)}^n$ is reached. The last step is the output layer to predict the edges for the previously defined number of classes that are possible. Since for experimental realities there are events where not all detected FSPs actually belong to the primary particle decay (i.e. beam-induced background), an additional background class with the class value of zero is included in this thesis. The

number of classes would therefore be 6 for the LCAS matrix using the 5 stages and the background class, or the maximum number of generations in addition to the background class for the LCAG matrix.

For the LCA matrix displayed Figure 4.5, each edge contains two entries in the matrix with $L_{ij} = L_{ji}$, each corresponding to one direction of the edge. The prediction in the last output layer can differ, as the direction is maintained with the message passing steps. To match the LCA matrix, the prediction matrix $\mathbf{L}_{\text{out}-1}$ gets symmetrized by adding the transposed Matrix and dividing by 2:

$$\mathbf{L}_{\text{out}} = \frac{1}{2}(\mathbf{L}_{\text{out}-1} + \mathbf{L}_{\text{out}-1}^{\intercal}). \tag{4.5}$$

Equation (4.5) combines the directed edges of the previous steps.

The model is implemented in the machine learning library `PyTorch` [52]. As self-interaction of FSPs is not included in the model prediction, the diagonal is excluded when calculating the loss. To achieve this, the diagonal of the LCA matrix is set to a class value that is not being used in the class prediction. This is `PyTorch` specific and results in the diagonal of the LCA matrix being set to -1.

To be able to handle different decay multiplicities, the network must be able to handle a variable number of FSPs. Due to the implementation in `PyTorch`, the input tensors for a single batch have to have the same dimension, so matching the dimensions when building batches for the training is necessary. In [11], the number of FSPs is set to a maximum number of particles per decay. If a lower number of particles occurs in an event the resulting empty rows and columns get filled with -1. This is known as padding. The padded values are also ignored during the loss calculation due to masking with the ignored class of -1. This however influences the output of the GNN after the first embedding layers due to non-zero bias weights. Even though the input features are zero they can still contribute after the edge representation gets combined to the node representation.

The solution to this is to adjust the message passing tensors used in the Node-to-Edge and Edge-to-Node to only include the original number of final state particles. This way, empty dimensions get ignored when calculating and predicting the nodes and no longer contribute to the edge prediction in the output of the model.

By using this method, the network can still be trained with randomly shuffled batches and the computational requirement is lower when padding a batch only to the maximum number of leaves per batch. This is due to the reason that the edges scale quadratically with the number of nodes, therefore having more nodes (including the artificial padding nodes) greatly increases the number of edges.

To account for the imbalanced distribution of classes, the option to include class weights is examined in Section 7.3. To prevent the network from only predicting a majority class, the respective classes instead contribute to the loss function in the same order. In this work, cross entropy loss and focal loss get utilized for the training with the Adam optimizer [36]. The hyperparameters of the network architecture (number of blocks, dimension feedforward, additional MLP layers, and final MLPs) defined in this section are crucial to the performance of the network as they define the size of the network. The tuning of these is computationally

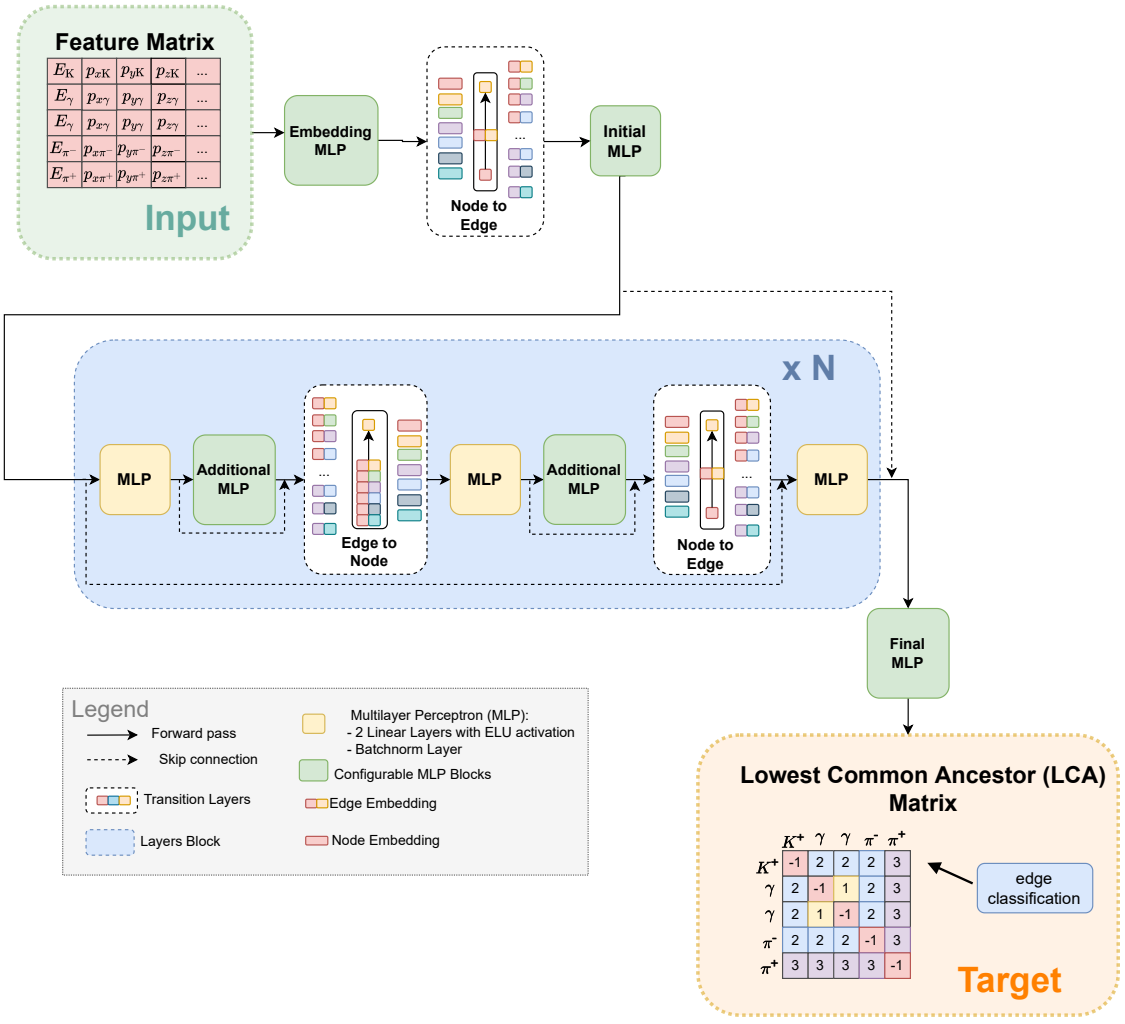


Figure 4.5.: The modified NRI encoder used in this work based on [49] and [11]. It shows the feature matrix of an example B-decay as input and the LCA matrix as the target of the model. The model architecture is displayed in the middle, where bold arrows denote the order of the different layers for the forward pass. The residual connections are shown as dotted lines. The blue block is a simplified display, as the layer's content of the blue block is appended according to the number N specified.

expensive, as the different network configurations have to be tested and their performance compared. For that reason, `Optuna` [53] is utilized in this work to optimize the configuration in the user-defined hyperparameter space. `Optuna` uses pruning algorithms to stop trainings early that perform worse than the previous optimization, as this saves computational time and resources.

4.3. Evaluation Metrics for the graFEI

For a comparison with the existing `FEI` algorithm (Section 2.5), similar metrics are used to benchmark the performance of the graFEI. A comparison between the metrics for the FEI and the graFEI is shown in Table 4.1.

For the `FEI` a decay counts as tagged if it is possible to reconstruct a B-meson. The equivalent for the graFEI is requiring that the predicted LCA matrix corresponds to a valid decay tree structure, called **valid tree**. This is done by demanding a rooted, directed, and acyclic tree, so that there are no loops in the tree structure when particles decay into a higher generation. The fraction of valid trees out of all samples is called the **valid tree efficiency**. In the case of a valid tree, the B-meson can be reconstructed by combining the kinematic information of the involved FSPs. It would also be possible for physics analyses to implement selections for the intermediate particles to increase the purity of their analysis similar to the `FEI`, but this is outside the scope of this thesis. Another metric that is used to monitor the training of the network is the **accuracy**. This metric is calculated independently from the other as it is not calculated per decay but per single particle edge. The accuracy is the number of correctly classified particle edges out of all possible edges. If all particle edges in a decay are predicted correctly, the accuracy for this decay is 100 % and the decay tree is correctly predicted. The fraction of the correctly predicted LCA matrices out of all is called the **perfect LCA**. Equivalent to the purity used in the `FEI` is the purity for graFEI, given by the fraction of the perfect LCA to the valid tree efficiency.

The only method to tag a B-decay is if the predicted LCA matrix is a valid tree. The `FEI` offers a signal probability to have the option of deciding between an analysis with low purity and high number of samples or high purity but a low number of possible samples. This is not yet implemented for the graFEI.

In Chapter 5, the approach described in this chapter is evaluated with the graFEI metrics for idealized simulations on a large number of possible decay channels, two orders higher than previous studies in [11]. Chapter 7 then studies the approach on simple B decays as subsets of the Belle II simulated samples. In Chapter 8, the `FEI` and the graFEI are compared in regards to these metrics on the Belle II simulated samples.

Table 4.1.: A direct comparison between the metrics used to evaluate the FEI and the new metrics used for the graFEI. Metrics in the same line are equivalent.

FEI	graFEI
tagging efficiency: fraction of reconstructed B-decays to all B-decays	**valid tree efficiency**: fraction of B-decays with a rooted, directed, acyclic, predicted tree to all B-decays
	accuracy: amount of particle edges that get classified correctly (independent of B-decays)
tag-side efficiency: fraction of **correctly** reconstructed B-decays to all decays	**perfect LCA**: fraction of B-decays with a **correctly** predicted LCA matrix
purity: fraction of correctly reconstructed decays out of all reconstructed decays	**purity**: fraction of perfect LCA out of all decays with valid trees

5. Extension of Studies on Previous Work

For the first part of this thesis, I extend previous studies [11] to analyze the behavior of the graFEI approach for larger datasets. This is used to formulate a training strategy for the $\Upsilon(4S)$-Belle simulated data, which contains a large number of decay channels and varying multiplicities in the event decays. I perform studies on a simulated phasespace dataset with 200 different decay topologies.

The first Section 5.1 describes the dataset. The following Section 5.2 investigates the behaviour of the training when combining a large number of different decays, for a variable number of particles. Lastly, Section 5.3 shows the results of the hyperparameter optimization with Optuna [53] on the Phasespace Dataset that is used for the later training of the Belle II simulated datasets.

5.1. Phasespace Dataset

To generate Monte Carlo n-body phasespace decays, which obey the laws of physics, the *Phasespace* [54] library is used. To study the graFEI performance a large generic dataset with a fixed root node is produced, mimicking the B-meson, and different decay topologies based on the previous work by Tsaklidis [11]. The decays are generated starting from the root node with a fixed mass of 100 (arbitrary unit). The child particles are selected at random between intermediate and final state particles (FSPs). The combined masses of the child particles have to be less than the mass of the root node. The intermediate particles have fixed masses of [90,80,70,50,25,20,10], while the masses of the FSPs are chosen from [1,2,3,5,12]. The number of children for each parent particle is fixed between two and five to mimic the nature of particle decays. Starting from the root node, the daughter particles are gradually generated according to the defined decay topology. This is continued until only FSPs remain. The decay tree structure and the masses of the intermediate and FSPs define the decay topology. The decay then is simulated according to the laws of physics, e.g. energy and momentum conservation. The simulated features of the FSPs are their respective four-momenta. The simulated energy and x-axis momentum are shown in Figure 5.1 for an example decay with a depth of three and six FSPs.

The dataset for the study of the graFEI approach consists of 200 decay topologies for the training and evaluation. Regarding B-decays for Belle II, the number of FSPs varies between two to 30 particles and there can be deep decay trees with high numbers of intermediate particles. The number of FSPs generated for one decay topology is between two and 16. The

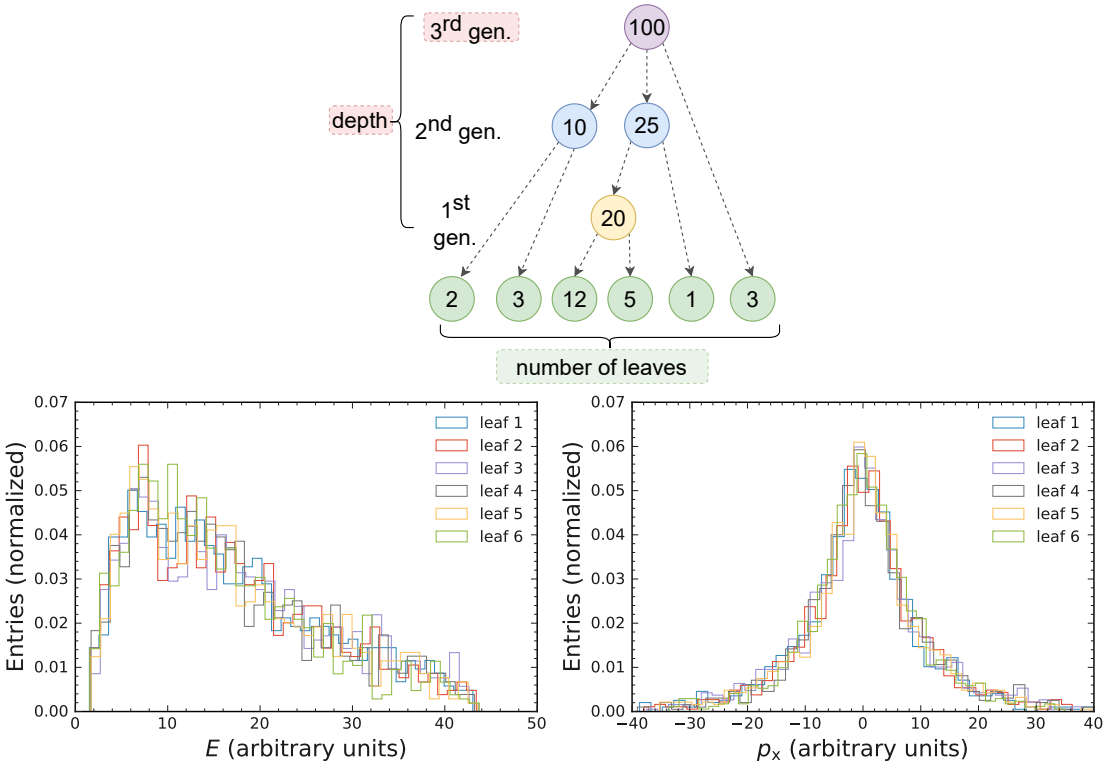


Figure 5.1.: Example decay at the top to show the energy and momentum distribution for the generated phasespace decays at the bottom. Starting from the root node with a fixed mass of 100, intermediate and final state particles are randomly selected until only FSPs remain. The decay then is simulated, yielding the energy and momentum distribution shown on the lower part. The energy and momentum are given in arbitrary units. This example is shown for one topology with its arbitrary particle mass at the top.

depth of the tree is between 1 and 7 generations until the root node is reached. The sample distribution of the phasespace dataset is shown in Figure 5.2. Leaves refer to the number of FSPs in the training sample and the depth of the LCA matrix is the maximum generation of the LCA matrix. As shown in Figure 5.1, the topologies are separated according to their maximum number of FSPs and the height of the root node. For each topology, the same number of samples is simulated, to represent an evenly distributed dataset.

The first training runs are performed with eight thousand samples per topology, resulting in 1.6 million training samples for the combined training. For the computationally expensive optimizations, the sample size was reduced to two thousand samples per topology. This also brings the experimental conditions of samples per decay closer to the ones expected of the Belle II dataset. This leads to a total training dataset size of 400 thousand samples. The test and validation datasets are generated the same way as the training dataset and are half the size of the training dataset.

I use this dataset to develop a first training strategy for the Belle II samples.

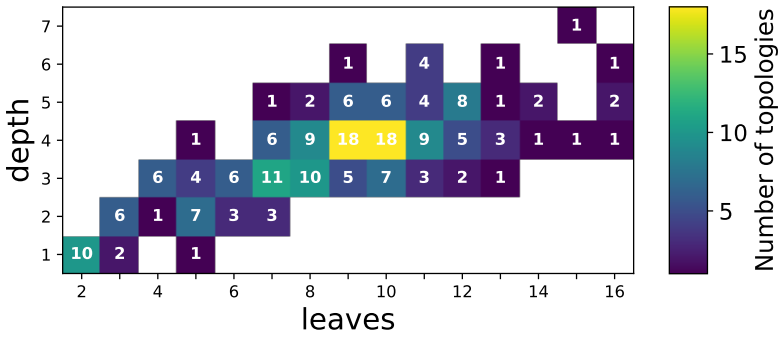


Figure 5.2.: The distribution of the different decay topologies for the phasespace dataset. A specific decay topology consists of fixed intermediate and FSPs as well as the defined tree structure. The depth is defined as the maximum number of generations between the final state particles and the root node. The number of leaves corresponds to the number of final state particles.

5.2. Studies on Phasespace Datasets

The first study in this thesis examines how the performance of the graFEI is affected by the number of possible topologies. There is a need to predict different decay trees with the same trained model, as only the FSPs of the $\Upsilon(4S)$ events of Belle II are detected. Therefore, different topologies with varying depth and number of FSPs are combined to test the approach.

Figure 5.3 shows the comparison between the training for each combination with the same depth and number of leaves (node) separately (left) and the full dataset simultaneously (right).

Single Node Training

I trained 43 models for all single node trainings, where different decay topologies are combined but the depth and number of leaves is the same. The model is successfully able to predict over 80% LCAG matrices up to nine leaves correctly. The model is therefore able to distinguish between these decay structures.

The higher the number of leaves and the depth, the more complicated it is to predict the LCAG matrix. This is due to the fact, that a larger number of combinations for the decays is possible. For these more complex decay trees, the predictive capacity of the model decreases steadily until it only is able to predict less than 10% of decay trees correctly. The accuracy is still higher than 77% for these decays. This shows the limitation of this method. The LCAG matrices are no longer predicted correctly for very complex decays. The total perfect LCAG score achieved is 76.6% over the full dataset and the 43 trained models.

Combined Training

For the combined training, only leaves up to five FSPs are well predicted with a perfect LCA score of over 80%. Further, the model is no longer able to predict decay trees with

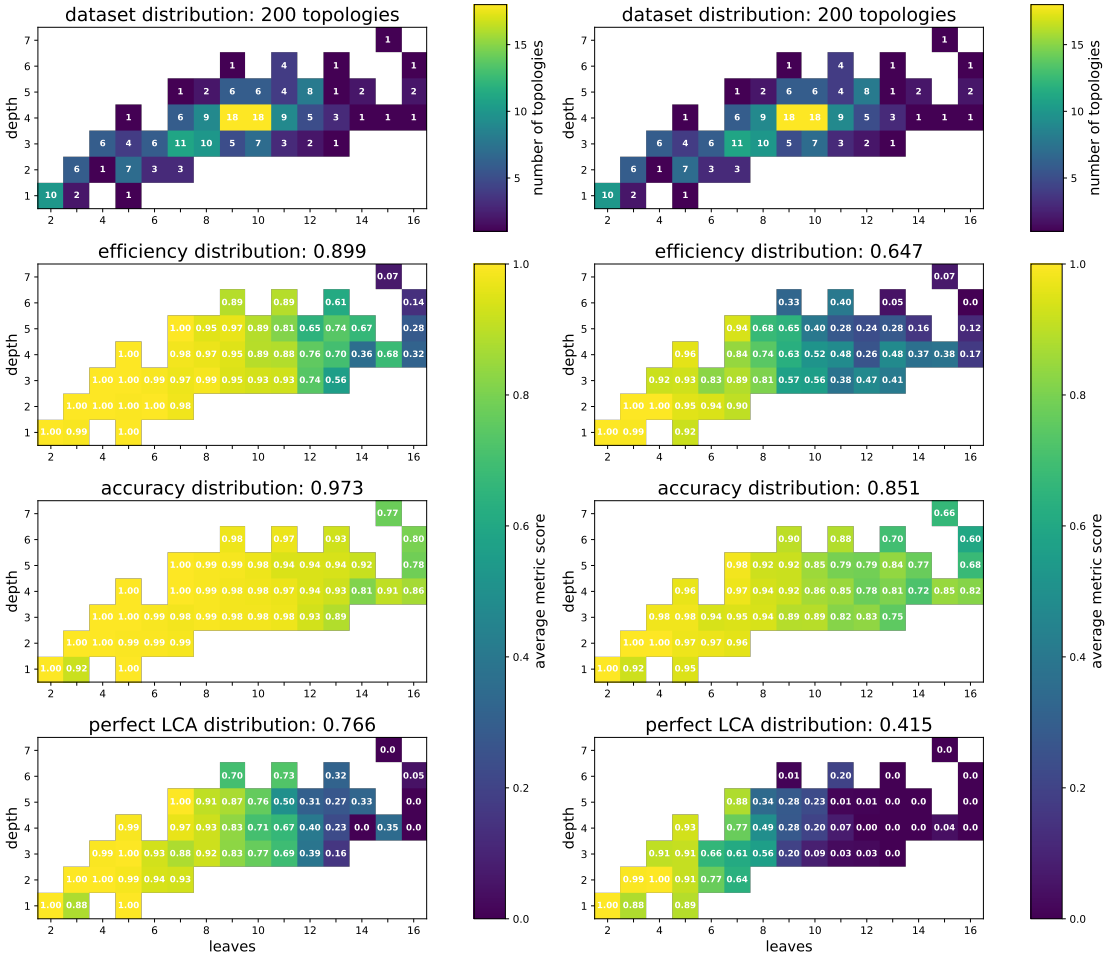


Figure 5.3.: Evaluation of the trained networks on the test phasespace dataset. On the left, the results are shown where each leaf and depth combination was trained separately and on the right, they are combined. The metric titles include the average metric value on the full dataset.

over twelve leaves, isntead the model learns to predict the easier decays. This is also shown in the accuracy, where the less complex decays achieve higher accuracies than the more complex ones.

The models for the single nodes achieve high perfect LCAG scores for up to eleven leaves. This is not the case when combining all training samples for the whole training process. Therefore, the idea is to identify potential patterns. In the following, I investigate the influence of the depth of the model on the performance of the reconstruction to find out if more complicated decays need deeper networks to increase the performance.

5.2.1. Ablation Node Study

Due to hardware constraints, the training samples are reduced to two thousand samples per topology for the following hyperparameter optimizations. At first, I trained on subsets with

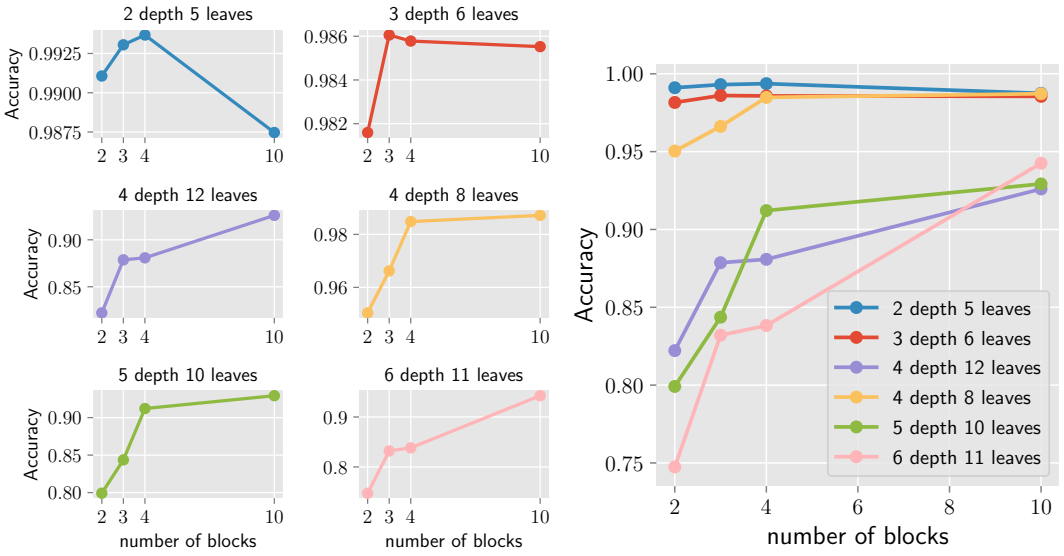


Figure 5.4.: Ablation study for the depth of the NRI model on varying topologies with a fixed size of FPSs (leaves) and fixed depth.

the same number of leaves and depths shown in Figure 5.4, and evaluate it on the model accuracy. These training runs are performed with the same hyperparameter selection, only varying the number of blocks. The mapped hyperparameter space covers the number of blocks of two, three, four, and ten. As the complexity of the decays rises, the depth of the network must be increased accordingly. There is a trend towards more blocks for decays with a higher number of leaves regarding the training runs with eight, ten, eleven, and twelve leaves. Nevertheless, no direct dependency to the number of leaves or depth can be observed, as the accuracy increases for all these training runs, so further studies are necessary.

5.2.2. Ablation Depth Study

To further investigate the trend of increasing the model complexity according to the decay complexity, I extend the search to combined decays with differing numbers of FSPs. For this, I combined all decays with the same depth and performed the same study described in the chapter before. The idea behind this ablation study is to find out whether a block is necessary for each additional level of ancestors. The algorithm could learn a step-wise reconstruction similar to the FEI. Thus, the first block would be responsible for the parent particles and the second block for the grandparent particle reconstruction. If this were the case, one could gradually add more blocks for the network if there are more extensive decays with a large depth and a high number of FSPs.

Issues with the training occur for these depth training runs, where the model is not able to learn anything. This is solved by scaling the features to a standardized range with the mean of zero and standard deviation of one. This ensures that the scaled features now contribute similarly to the loss, enabling the training to converge.

The results of this ablation study are shown in Figure 5.5 for the depths of two, four and five. The perfectLCAG score is shown for the training with the respective number of blocks. For each number of blocks, five trial models are trained. As seen in Figure 5.5, the results for the perfectLCAG score can vary for fixed hyperparameters. For a depth of four, there is a trend towards a higher number of blocks for the mean of the training runs. The performance of the model is similar to the results with four and six blocks. For a depth of five, the results are also similar for four to six blocks. There is a larger variance in these trainings. Overall these studies show no clear correlation between the depth of the predicted LCA matrix and the number of blocks and have high fluctuations. The model is performing best when the number of blocks is between four to six.

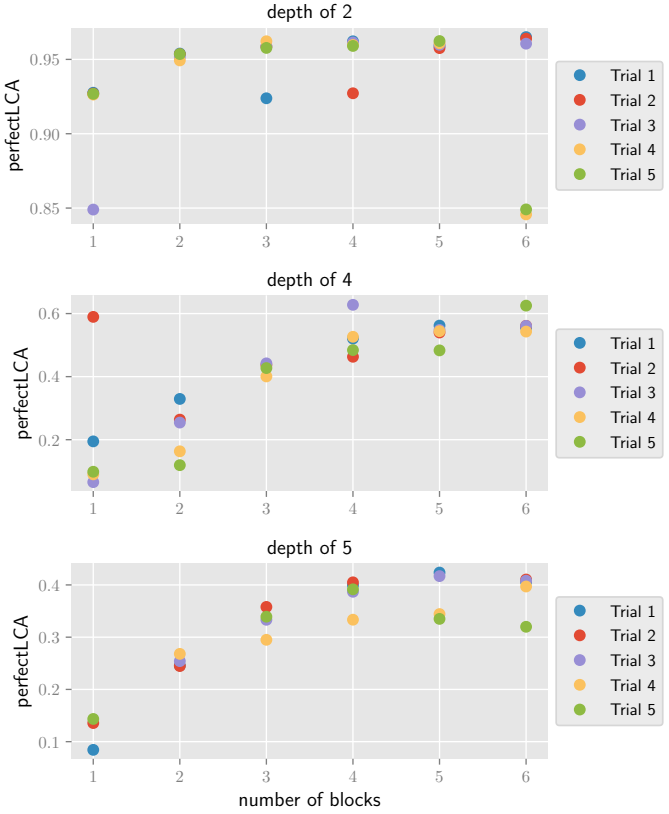


Figure 5.5.: Ablation study for the depth of the NRI model on varying topologies with a fixed tree depth.

5.3. Hyperparameter Optimization for Large Phasespaces

The previous studies in Section 5.2.2 show that the performance of the model improves for a low number of blocks, but converges and does not improve for a larger number of blocks. Furthermore, there is a high variance between the trials for one hyperparameter combination. This is indicating that the optimal number of blocks is not correlated to the decay depth.

To determine the optimal network architecture `Optuna` [53] is used for the hyperparameter optimization. The hyperparameter space of the `Optuna` optimization on the whole phasespace is reduced to a maximum number of four blocks. This is due to hardware constraints, as large models can no longer be computed on the GPU memory without implementing model parallelism. The training is computationally expensive and is therefore performed on the `HoreKA` [55] cluster. Regarding the model architecture in Section 4.2, the following parameters are tuned:

- additional MLP layers,
- feedforward layer widths,
- final MLP layers,
- initial MLP layers,
- loss function,
- number of blocks.

The result of this optimization is shown in Figure 5.6. The target metric for this optimization is the perfectLCAG. The biggest trend is for the larger NRI feedforward layer widths. The initial and final MLP layers have no influence on the predictive capacity. Both loss functions achieved similar results. The number of additional MLP layers in the block is also not decisive for the model performance.

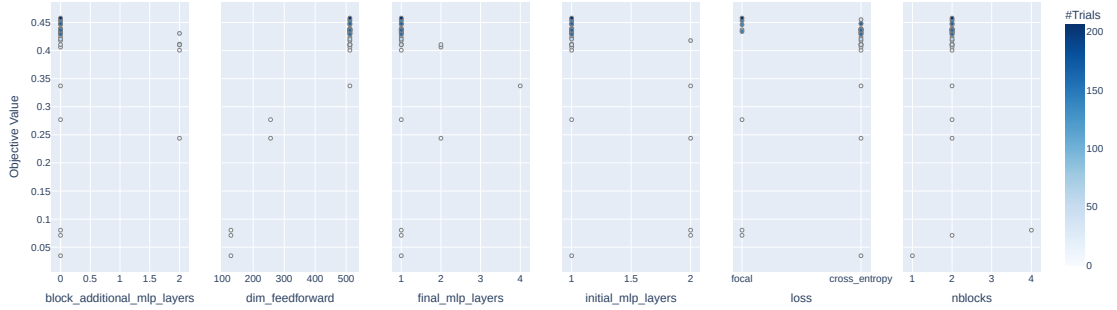


Figure 5.6.: `Optuna` Optimization of the NRI. The objective value corresponds to the perfect LCAG. The trial results are shown for the six model architecture hyperparameters.

The model performs best with two blocks. Comparing this with the previous results decays with larger depths or more complex decays require more complex models. This can be achieved by making the model even deeper with a higher number of blocks or wider by defining larger feedforward layer widths. This indicates that it is beneficial to transition five times between the edge and node representation (two blocks) and that the model needs this depth to learn generalization and not only memorize the data. It also shows that the model performance is not improving further when transitioning more often. In previous work [11] it was reasoned, that information loss occurs with a higher number of transitions. Results in Figure 5.5 on the other hand indicate that the model is neither gaining nor losing performance.

To confirm the biggest impact being the larger NRI feedforward widths, another optimization is shown in Figure 5.7. This is done with feedforward layer widths up to 2048. This hyperparameter optimization shows, that higher feedforward widths improve the model performance greatly, but also increase the possibility of overtraining.

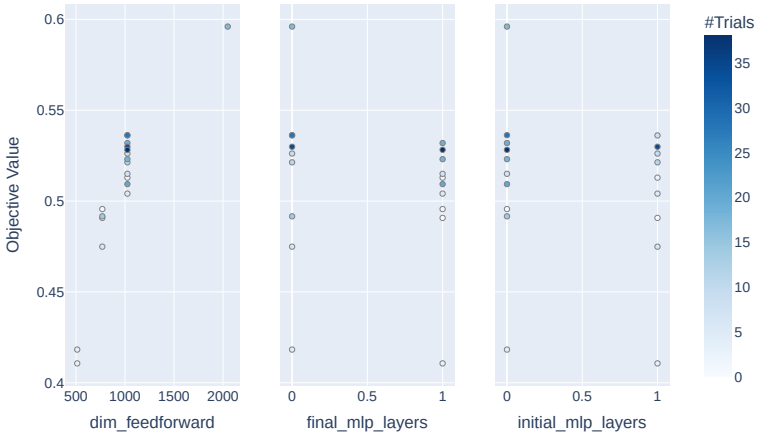


Figure 5.7.: `Optuna` Optimization of the NRI with larger NRI feedforward layer widths.

Using these optimized hyperparameters and a feedforward layer width of 512, due to hardware constraints, the training on the full phasespace dataset is repeated and shown in Figure 5.8. The perfect LCAG score for the full training achieves 61.1% compared to the previous 41.5%. The model is able to correctly predict over 70% LCAG matrices for up to eight leaves correctly and the predictions for the more complex decays improve.

This confirms that the graFEI approach can be used for large amounts of possible decays. These results are used as a baseline for the following studies in Chapter 7 and Chapter 8. Because these hyperaparameter searches are computationally expensive, they cannot be repeated for each study. To speed up the training, I use the smaller configuration when the training runs achieved similar results. For example, the additional MLP layers per block are set to zero as there is no indication that using a higher number would improve the results.

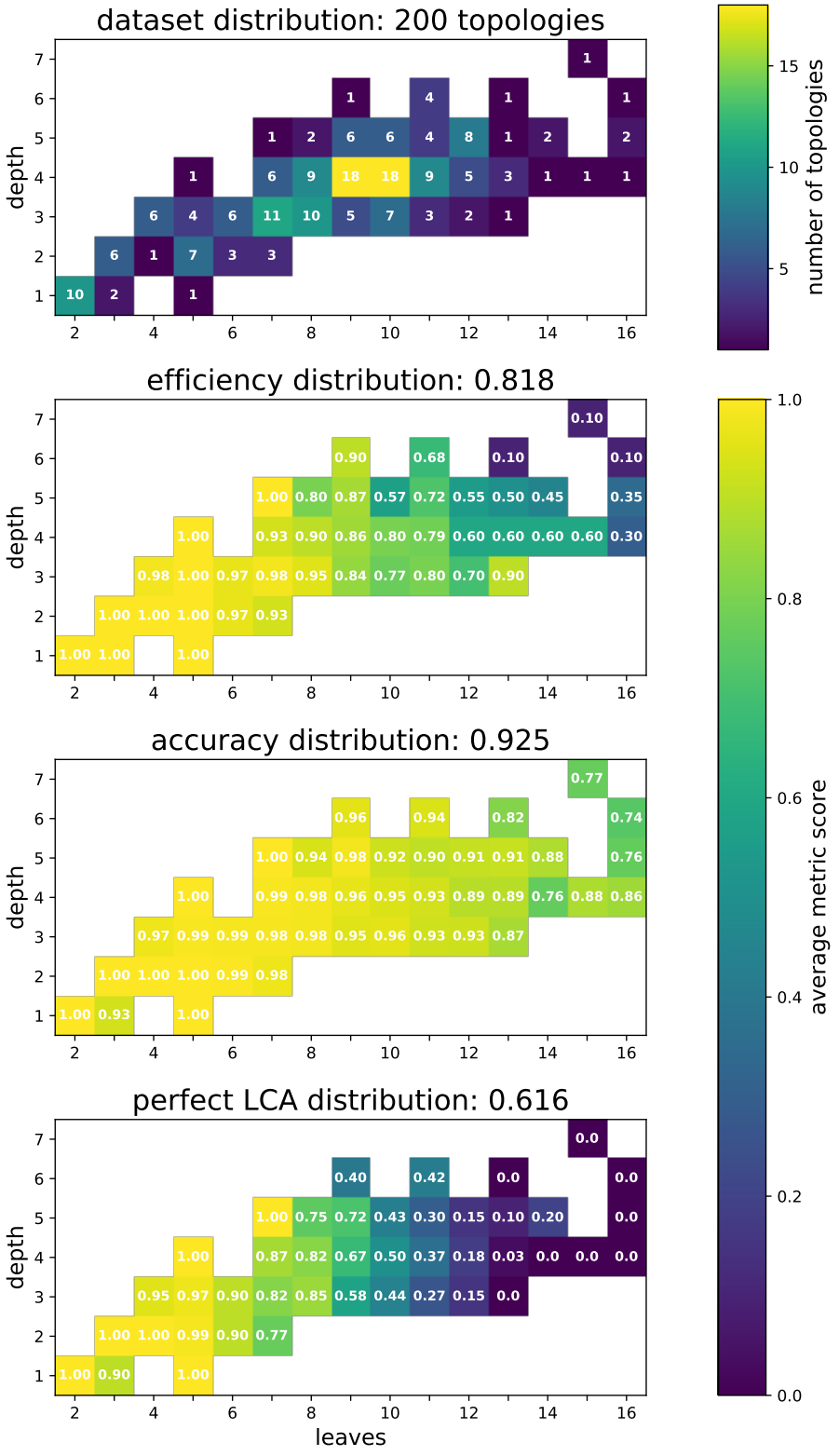


Figure 5.8.: Phasespace evaluated model that was trained with the optimized hyperparameters.

6. Belle II Simulated Training Datasets

This chapter shows how to adjust the graFEI approach for Belle II simulated events. In contrast to previous experiments in [11], this now includes the detector simulation in the Belle II MC samples, instead of only the event generation MC truth level information. In this work simulated Belle II $\Upsilon(4S)$ MC13a [28] samples of $\Upsilon(4S) \to B\overline{B}$ generic B-decays are used. The graFEI approach is applied to single B-decays, therefore each generic $\Upsilon(4S) \to B\overline{B}$ event is split into two B-decay training samples.

First, the FSPs have to be selected using the detector information. As described in Section 2.3, in many cases there are additional particles, for example beam background or split-offs, which are part of the event but do not belong to the B-meson decay.

Chapter 6 describes how I select FSPs including the detector simulation. The strategy is to build the LCA matrix on the event generation level, excluding the detector simulation. The selected FSPs are compared with the true FSPs of the event generation to build the training LCA matrix. In Section 6.2 I adjust the training target (LCA matrix) according to these selected FSPs. Furthermore, I introduce the input features used for training in Section 6.3. The last section of this chapter describes the full preprocessing and training process for the dataset to give insight on the hardware constraints in Section 6.4.

6.1. Belle II Final State Particles

To determine the FSPs per decay the information of the detector simulation level is used to reconstruct the final state particles. Charged particles are derived from reconstructed tracks, whereas photon candidates are reconstructed from the ECL clusters. The challenge is to remove falsely reconstructed particles as well as the additional particles not belonging to the decay while having a high reconstruction efficiency regarding particles belonging to the B-meson decay. To achieve this, the selection criteria are evaluated in this chapter. To get started, I apply the default recommendations within the collaboration shown in Table 6.1.

For the charged particles, only the highest particle identification probability (PID) out of $e^{\pm}$, $\mu^{\pm}$, $\pi^{\pm}$, $K^{\pm}$ or p determines the identity of each particle.

Further selection criteria to ensure a good reconstruction are to demand that the particle is within the CDC (Section 2.3) acceptance, requiring that the particle polar angle θ is within the range $17° < \theta < 150°$ and not outside the detector coverage (beam pipe). Another

Table 6.1.: Particle selections for the final state particles.

	particle selection criteria
charged particles $e^{\pm}, \mu^{\pm}, \pi^{\pm}, K^{\pm}, p$	$17° < \theta < 150°$ nCDChits > 20
photons γ	$17° < \theta < 150°$ $\Delta t_{\mathrm{ECL}} < 1 \cdot 10^6$ $\frac{E_1}{E_9} > 0.4$ or $E > 75\,\mathrm{MeV}$ Cluster forward/barrel $E > 50\,\mathrm{MeV}$, backward $E > 75\,\mathrm{MeV}$

requirement is that at least 20 registered hits in the CDC assigned to the particle track are measured. This improves the PID reconstruction so that 76.3 % of the charged particles are correctly predicted and assigned the correct mass hypothesis.

The selection criteria for the photons (γ) are also provided by the Belle II collaboration and defined in Table 6.1. Reconstructed photons are also required to be in the CDC detector acceptance. Another selection is that the error of the cluster timing Δt_{ECL} has to be smaller than 10^6 to remove failed waveform fits in the ECL. Furthermore, the ratio between the measured energy of the center crystal E_1 of the ECL and the 3x3 crystals surrounding this center crystal, E_9, has to be more than 0.4. This is because larger values are more likely to be generated by photons, whereas hadrons tend to produce smaller values. If that is not the case, then the energy E of the photon has to be at least 75 MeV. Lastly, energy cuts depending on the ECL region of the cluster are applied. If the particle is measured in the forward or barrel region, it has to have an energy of at least 50 MeV. For the backward region, the energy has to be at least 75 MeV. This is to remove low energy photons, as they are mostly created by background processes.

6.2. Building LCA Matrix for Reconstructed Data

Particle reconstruction is imperfect and particles can escape the detector undetected. Reconstructing the FSPs can result in their features being incorrect, e.g. through misidentifying particles, or features like the four-momentum getting smeared. Although selection criteria are applied when selecting the FSPs, there can still be reconstructed particles in the event that do not belong to the primary decay. These cases have to be accounted for when building the training target for the training on Belle II simulated data.

To build the training LCA matrix, the true LCA matrix is determined from the event generation for both B-meson decays in one $\Upsilon(4S) \to B\overline{B}$ event. Then the final state particles are reconstructed according to the selection criteria defined in the previous section. The next step is to then match the reconstructed FSPs to the true FSPs of the event generation.

The detector reconstruction effects are categorized as follows:

Missing

Invisible particles like neutrinos cannot be measured by the detector. Furthermore,

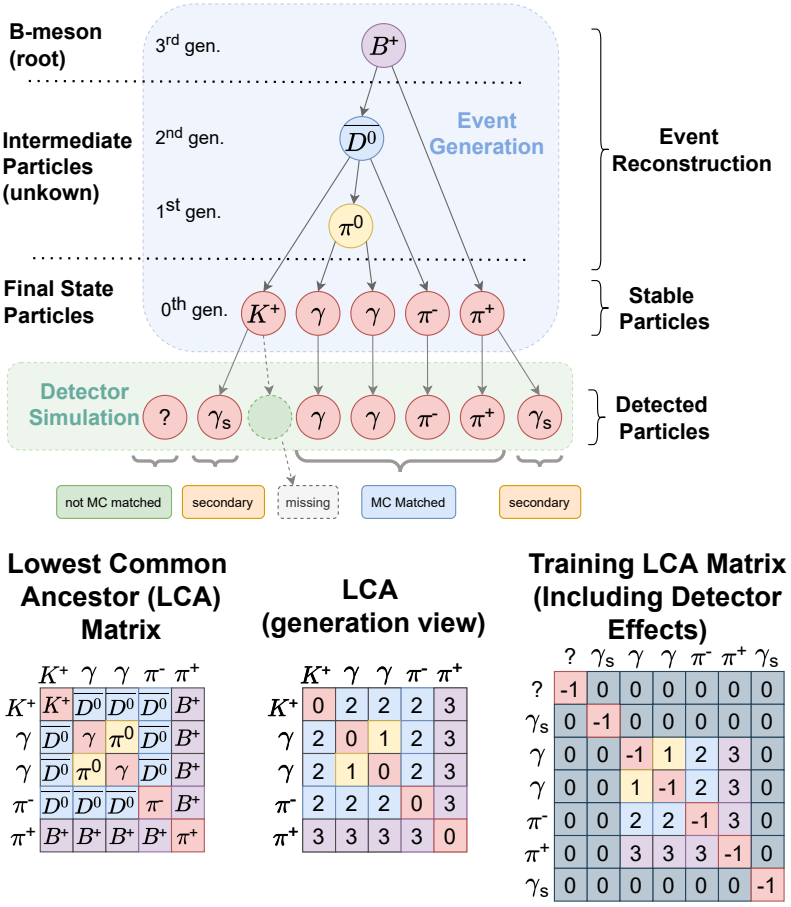


Figure 6.1.: An example of building the training target LCA matrix is shown. On the top, the two stages of the event simulation are shown with the event generation followed by the detector simulation. The LCA is generated out of the FSP of the event generation and encoded in the LCAG matrix. To adjust for the experimental realities of the detector simulation, the LCAG matrix is further modified. LCAG matrix rows of missing particles are deleted. Additional particles that are part of the detector simulation event but do not belong to the B-meson decay are added as a background class with the value of zero, as not MC matched or secondary particles. This results in the training LCAG matrix shown on the bottom right being the final training target. The diagonal entries are assigned to -1 to be later on ignored in the training, as particle self interactions are excluded in the training.

FSPs can escape the detector due to the detector efficiency or do not match the reconstruction criteria. Therefore, they cannot be reconstructed and are referred to as missing particles. To adjust the training LCA matrix for this possibility, the missing particle is deleted from the LCA matrix. In the example of Figure 6.1 the kaon K^+ is missing, therefore the respective row and column are deleted for the training LCA matrix and only the four remaining true final state particles are contributing. For the

generic B-decays with the selection criteria defined in the previous selections, there are on average 4.12 particles missing per hadronic B-decay and 3.04 particles are missing for semileptonic decays. This is shown in Table 6.2. The percentage of decays where all final state particles were reconstructed and no particles are missing is only 1.6 % for hadronic decays and 4.9 % for semileptonic decays.

Duplicates

It can happen that a particle is reconstructed two or more times and one true FSP is assigned to more than one particle. This means there are duplicate particles that have to be considered. With the event selections, duplicates occur in 29.2 % of hadronic decays and 25.0 % of semileptonic decays. To include duplicates, the particle row and column of the true LCA matrix can be copied according to the number of duplicate particles (in this example up to 7).

Secondaries

The particles from the event generation that build the true LCA matrix are called the primary particles. They are part of the physics generator. The secondary particles are generated by the simulation when particles interact with the detector. Therefore they can be assigned to their corresponding B-decay when MC-matched. In the example decay in Figure 6.1 there are two additional photons that are secondary particles. Because they do not belong to the original decay tree; particle relations with secondary particles are labeled as 0. This means the ignored index on the diagonal has to be changed to another number for `PyTorch`, and is from now on set to **-1**. Secondary particles are present in 75.7 % of hadronic B-decays and 60.1 % of semileptonic B-decays with an average of 1.9 and 1.2 particles per decay, respectively.

Unmatched

If the MC matching does not find a candidate for a reconstructed FSP, this particle is *unmatched*. It is not possible to assign this particle to a true particle of the event generation. This also means that it is not possible to assign the particle to one of the two B-decays in the $\Upsilon(4S)$ event. Unmatched particles are present in 91.7 % of all B-decays and therefore cannot be ignored. $\Upsilon(4S)$-events have on average three unmatched particles. The solution is for one B-decay to randomly select a subset of unmatched particles and assign them to this B-decay. For the signal decay $B \to \nu\bar{\nu}$ this is not necessary as everything in the event belongs to the tag-side B-decay. Comparing the distribution for a B-decay on B-generic to the tag-side of $B \to \nu\bar{\nu}$ shows that the distribution is similar and this is a sufficient approach to deal with unmatched particles. This is shown in Figure 6.2 for the example decay $\mathrm{B}^0 \to \mathrm{D}^{*-}(\to \overline{\mathrm{D}}^0(\to \mathrm{K}^+\pi^-\pi^0)\pi^-)\pi^+\pi^+\pi^-\pi^0$. The distribution is similar when randomly adding unmatched particles as shown on the left as it does for the true distribution on the right. The particle connection to the other particles for unmatched particles is also labeled as zero as well as for the secondaries. An example is shown in Figure 6.1 for the unmatched particle.

The training LCA matrix now consists of two parts, the adjusted true LCA matrix, referred to as primary LCA matrix from now on, and the entries for the background particles. Furthermore, events are skipped as shown in Table 6.2 if the event does not pass the event cuts of the FEI or if less than two distinct primary particles got reconstructed, as in this

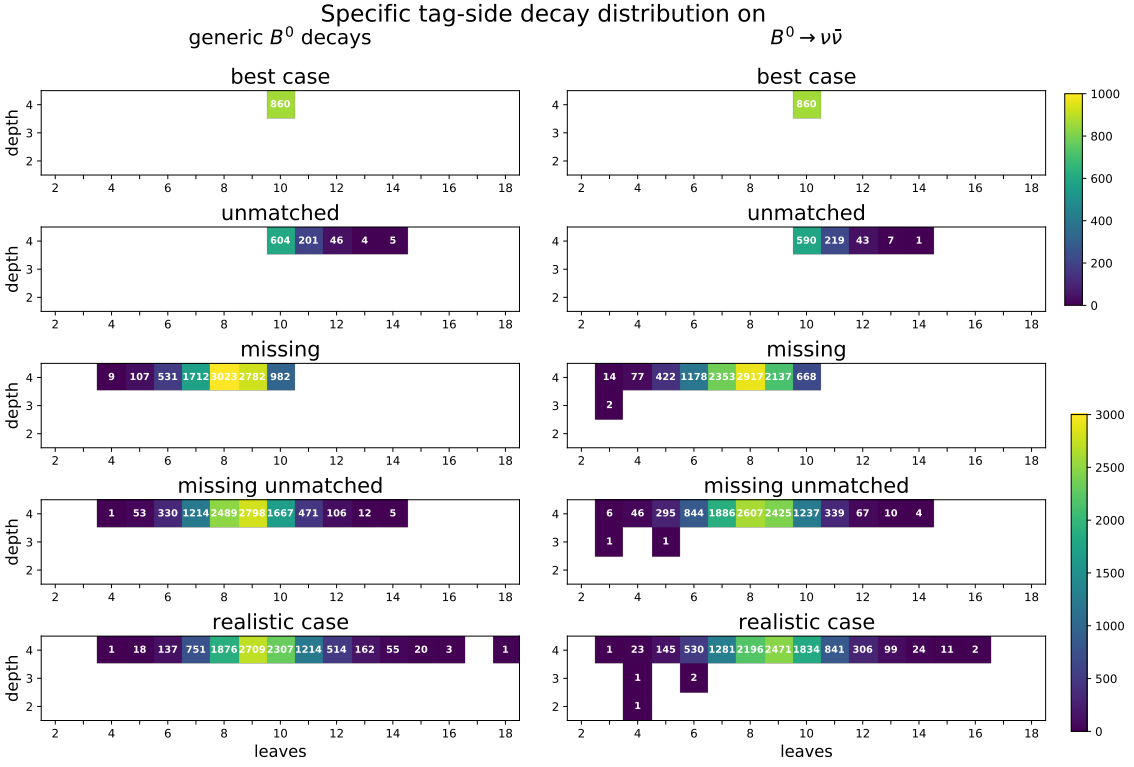


Figure 6.2.: The distribution of the training LCA matrix is divided by the number of FSPs per decay, named leaves and the maximum generation in the training LCA matrix referred to as depth. Unmatched particles can not be assigned to one of the two B-meson decays of the $\Upsilon(4S)$ event and are therefore distributed randomly. To verify this approach, a comparison for the same tag-side decay $B^0_{D^*}$ on generic $\Upsilon(4S) \to B^0_{D^*}B^0$ decays and $\Upsilon(4S) \to B^0_{D^*}$ $B \to \nu\bar{\nu}$ is shown.

case there is nothing to reconstruct. These event cuts require at least three particles to be reconstructed in the CDC acceptance, with a transverse momentum of $p_t > 100\,\text{MeV}$ close to the interaction point ($|z| < 2$ and $|d| < 0.5$). Furthermore, at least three entries in the ECL within the CDC acceptance and energy $E > 100\,\text{MeV}$ are required. The visible energy of the center-of-mass system should be at least 4 GeV and the energy stored in the ECL between 2 GeV and 7 GeV. These selections are applied to remove events where not enough FSPs are reconstructed to be able to predict a B-decay tree. Despite these selections, there are still B-decays with less than two primary reconstructed particles. These also get removed from the training samples as they represent unreconstructable decays.

Table 6.2.: The particle distribution for the generic B-decays was calculated on 200 000 events, expecting 400 000 B-decays, including the selection criteria defined in Table 6.1.

	hadronic	semileptonic
Number of B-Decays	200731	186072
Skipped B-Decays (<2 primaries)	256	1201
Average number of primary FSPs (min/max)	8.93 (2, 27)	6.76 (2, 24)
Average LCA length (min/max)	13.05 (2, 35)	9.81 (2, 33)
B-Decays with secondary FSPs	151914 (75.68%)	111904 (60.14%)
Average secondaries (min/max)	1.89 (0, 15)	1.24 (0, 15)
B-Decays with duplicate FSPs	58520 (29.15%)	46538 (25.01%)
Avg duplicate FSPs (min/max)	0.36 (0, 7)	0.3 (0, 7)
B-Decays with duplicate and secondary FSPs	166842 (83.12%)	129736 (69.72%)
B-Decays with missing FSPs	196293 (97.79%)	174236 (93.64%)
Avg missing FSPs (min/max)	4.12 (0, 18)	3.04 (0, 19)
B-Decays with perfectly reconstructed FSPs	3232 (1.61%)	9082 (4.88%)
Events with unmatched FSPs	177415 (91.73%)	
Average unmatched (min/max)	3.04 (1, 14)	

6.3. Input Features for Training

To learn the physics of the decay, the features of the used particles are relevant for the training. In the previous work [11], only the four momentum and the charge were used for the training. In this thesis additional features to describe the particles are used for the input. They are inspired by the features used by the FEI. Additionally, features are used to compensate for the vertex fitting used by the FEI for each particle stage.

Previous studies in Section 5.2.2 showed, that it is sometimes necessary to scale the input features to a standardized range for the model to learn and achieve a decreasing loss function. If the inputs are already Gaussian distributed, they are fit and scaled to a standard deviation $\sigma = 1$ and mean $\mu = 0$ using the distribution of the respective feature on the training dataset. As some of the input features of the Belle II dataset are exponentially distributed, at first the power transformation (Equation (6.1)) is applied followed by the standard transformation (Equation (6.2)).

$$\text{power}(x) = \text{sgn}(x) \cdot |x|^{\lambda}. \tag{6.1}$$

$$\text{linear}(x) = \frac{x - \mu}{\sigma}. \tag{6.2}$$

These are the features that are used in the training on the Belle II simulated data:

Particle identification probabilities (PID)

The particle identification probabilities are used when reconstructing the FSPs, as the particle is reconstructed using the mass hypothesis with the highest respective PID. For the FEI they are used in the first stage to reconstruct the FSPs. The PIDs are calculated by using the information from all available detectors, excluding the silicon vertex detector. The respective PID is given by the fraction of the likelihood of the corresponding particle x to all particle likelihoods as shown in Equation (6.3). These PID are used as input because even though 76.2% of all charged particles get identified correctly, 23.8% of particles are misidentified and use the wrong mass hypothesis and energy. Therefore, by including the PID, the network could learn and adjust for misidentifications.

$$\text{PID}_\text{x} = \frac{\mathcal{L}_x}{(\mathcal{L}_\mu + \mathcal{L}_e + \mathcal{L}_\pi + \mathcal{L}_K + \mathcal{L}_p + \mathcal{L}_d)}. \tag{6.3}$$

Kinematics

These are the particle kinematics. Instead of using the four-momentum with the x- and y-axis p_x, p_y, the full momentum p, the transverse momentum p_t and the momentum along the beam axis p_z are used. As these three variables are exponentially distributed, to achieve a normal distribution they are power transformed with $\lambda = 0.5$. Additionally, p_t and p then get scaled linearly on top of the power transformation. Furthermore, the PDG code of the particle as well as the corresponding mass hypothesis is used. These two input features are reliant on correct identification. The particle mass is also used to calculate the energy, making these three variables dependent on the correct PID. All other variables are not influenced by misidentification.

Vertex Information

The vertex information that is used here is the distance of the vertex to the impact point of the $\Upsilon(4\text{S})$ event collision. d_r describes the transverse distance and d_z the distance for the z-axis in respect to the interaction point. When scaled these two variables are power transformed with $\lambda = 0.25$.

ECL cluster variables

These attributes are only available if there are entries in the ECL cluster. Input features that are included are the region of the ECL cluster entry, tokenized by their class. Additionally, the number of hits in the ECL cluster and the timing of the ECL cluster are studied as inputs for the model. Furthermore, the ratio of the energy in the inner 3x3 crystals to the outer 5x5 crystals is added as a variable, as these values tend to be higher for photons and smaller for hadrons.

Basic decay information

An additional variable that is implemented and evaluated for the graph neural network is the number of FSPs for the whole B-decay. This is the same for each particle in the respective B-decays and is used as an input for the GNN to gain information over the size of the new fully connected graph.

6.4. Training Workflow

The full training workflow is shown in the flowchart in Figure 6.3. The starting point is the MC simulated events produced by the Belle II collaboration [28]. For both B-decays in each $\Upsilon(4S)$ event, the LCA matrix is built out of the primary particles of the event generation MC-truth level. This is achieved by recursively iterating over all MC-truth particles until there are no more primary daughter particles. The intermediate primary particles are saved and either classified by the LCAS or LCAG (Section 6.2) representation method. On the reconstructed level, for each event, the particles are reconstructed according to the selections in Table 6.1. Additional information is saved to later identify to which of the two B-decays they belong and if they are primary or secondary particles. The features for each particle in the event get saved. This is done on the Grid provided by Belle II [28], so therefore the MC datasets do not have to be downloaded locally, speeding up the process. Each file takes to 45 minutes to process. Furthermore, the processing on the Grid can be done in parallel and reduces the file size to 10 % of the original MC file size. For B-generic, this means that 51 million $\Upsilon(4S) \to \overline{B}^0 B^0$-events that take 500 GB of disk space only use 50 GB of disk space after the preprocessing. This space will be further reduced when the training LCA matrices are directly built on the Grid. For the experiments in this thesis, the option to modify the training LCA matrix is included to analyze the different levels of detector effects. These preprocessed files can then be downloaded to train on a GPU machine located at the ETP or on the Throughput Optimized Analysis System (`TOpAS`) at KIT which provides GPUs [56, 57].

The Belle II Grid software provided outputs as `ROOT` [58] files. A problem occurs when loading large amounts of training samples into memory. The occupied memory can reach values of over 300 GB for 35 million training samples. The FEI was trained on 180 million events, which would lead to over 300 million training samples. This results in an uncommonly high amount of memory usage in HEP computing with the default memory management where the complete dataset is loaded into memory. The alternative is to lazy-load the training samples when a batch is needed to reduce the memory usage. This can be done with `uproot` [59] using the lazy-load method[1]. A faster method entails converting the files to `hdf5` [60] files, saving the location of the training samples in the file[2], and only loading them when accessed by the dataloader for a training step. However, this requires a converting step, which requires additional computing resources. Another option would be to shuffle the data only per file for the `ROOT` files, one would then load only one active file at a time into memory and build the batches to potentially speed up the training.

Trainings on small datasets and small models, which means low numbers of trainable parameters, are performed on `NVIDIA GTX TITAN X 12GB` GPUs at the ETP. Trainings with large models require a large amount of GPU memory for calculations, which exceeds the 12 GB available. The same applies for large numbers of FSPs l and high batch sizes b, as the input tensors of the model scale quadratically with the FSPs number according to the batch size $b \times l \times l$. These more demanding trainings are performed on `NVIDIA V100`, `NVIDIA V100s` and `NVIDIA A100 40GB` GPUs provided by TOpAS.

[1] https://uproot.readthedocs.io/en/latest/uproot.behaviors.TBranch.lazy.html
[2] https://docs.h5py.org/en/stable/high/group.html

Due to the previously mentioned memory constraint, trainings for the final dataset are performed on `NVIDIA A100 40GB` GPUs on the KIT Hochleistungsrechner Karlsruhe (`HoreKA`) [55], one of the 15 most powerful supercomputers in Europe 2021. This is done to speed up the trainings by loading the full dataset into 512GB RAM provided by the respective node.

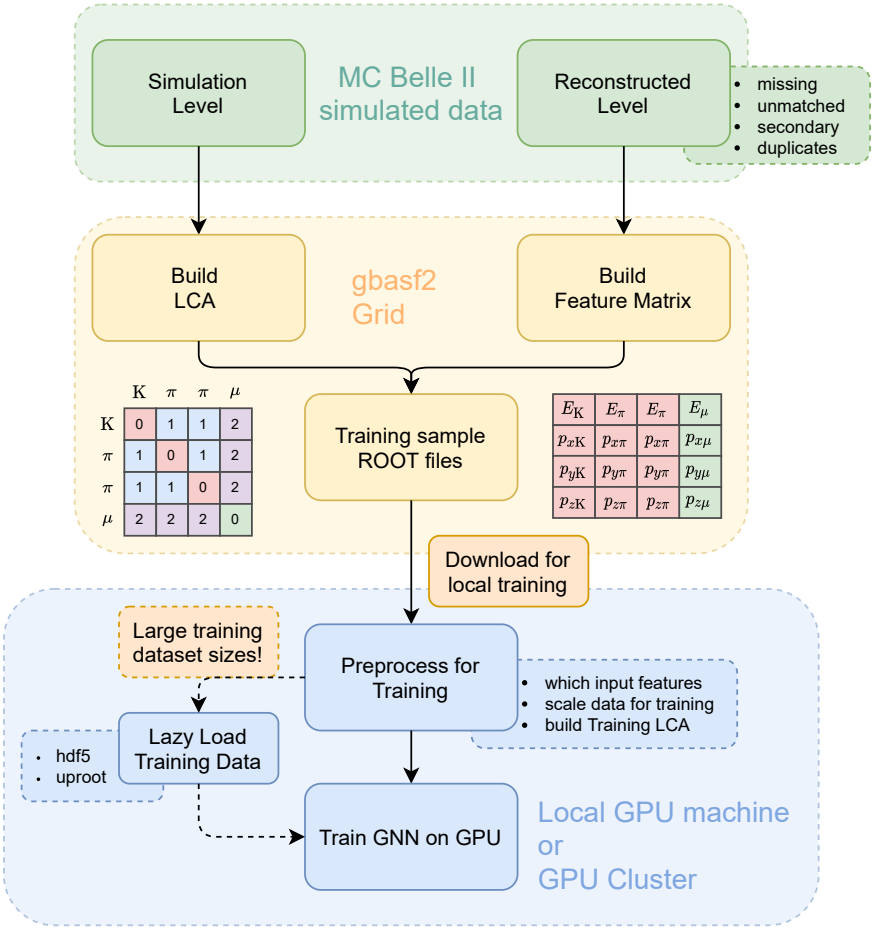


Figure 6.3.: Flowchart for the full training process including the data processing. Starting with the Belle II simulated samples provided by [28], the LCA matrix and input feature matrix are build in parallel on the Belle II Grid. They are downloaded to train on the local GPU machine or GPU cluster. The training target LCA matrix is built in the training step to enable studies regarding the input features and the detector effects. This step would be incorporated into the previous step when applying the graFEI method to speed up the training process further.

7. Studies on Belle II Simulated Data

Section 5.1 verified the graFEI approach on a large phasespace for a simple MC generated dataset. The next step is to show particle decay tree reconstruction on simulated Belle II B-decay samples using the graFEI approach. As described in Section 2.3, in many cases there are additional particles, for example beam background or split-offs, which are part of the event but which do not belong to the B-meson decay. This makes it difficult to determine the true FSPs of the event. Chapter 6 describes my approach to adjusting the training for reconstructed simulated data. In this chapter, I examine how the various effects of detector simulation affect the graFEI performance.

The first study in Section 7.1 analyzes one simple B-decay to show a proof of concept for this method. Here the different reconstruction effects defined in Section 6.2 are further investigated and compared to each other. The results of the previous Chapter 5 show, that when combining decays, the model is learning to predict the easier decays despite the distribution of the training samples. To study this behaviour further, I combine a small set of different decays with different decay topologies. This is to analyze the predictive capacity regarding the different decays in Section 7.2. Based on these two studies the FSP selection is adjusted in the last Section 7.3.

7.1. Training on Single Reconstructed Decay

To train on simulated reconstructed B-decays the first step is to investigate if the model is able to handle the adjustments to the training LCA matrix mentioned in the previous Section 6.2. The different effects to match the experimental setup are missing particles, duplicated particles, secondaries and unmatched particles. Additionally, it has to be studied if the model is able to learn with the information of the reconstructed FSPs. As the particle reconstruction is imperfect, the features of the particles can differ from the event generation and introduce uncertainties and resolution effects. For example, the resolution for the Belle II tracking detectors for the momentum is smeared by 0.1% [22]. This impacts conservation laws, as e.g. the four-momentum of particles are not adding up to the root B-meson. This is another obstacle the model has to compensate for when learning to predict the physics behind particle decays. To verify the model predictive capacity in regards to these experimental effects, I start with reconstructing a simple example decay $B^0 \to D^{*-}(\to \overline{D}^0(\to K^+\pi^-)\pi^-)\pi^+$.

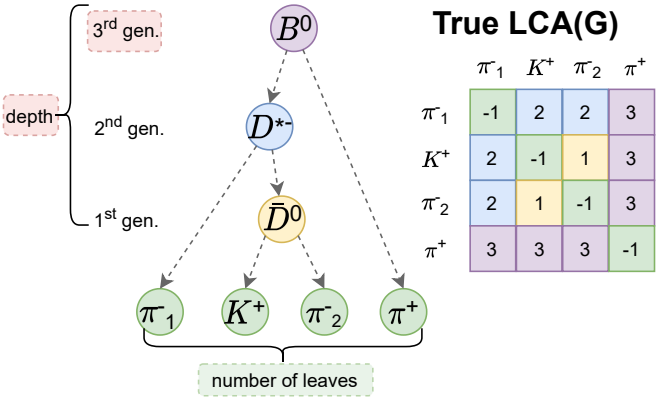


Figure 7.1.: The decay-tree structure (left) and the true LCAG matrix (right) for the specific decay $B^0 \to D^{*-}(\to \overline{D}^0(\to K^+\pi^-)\pi^-)\pi^+$ is shown with the root B-meson at depth three and four FSPs.

7.1.1. Single decay simulated reconstructed Datasets

Figure 7.1 shows this example decay of $B^0 \to D^{*-}(\to \overline{D}^0(\to K^+\pi^-)\pi^-)\pi^+$ at the left with the corresponding true LCAG matrix at the right. This decay consists of four primary FSPs with the B-meson at the third generation at the event generation level. Reconstructing the decay inflates and blurs the number of FSPs that belong to this decay.

In the example of Figure 7.2, if the π^+ is missing, the primary training LCA matrix gets reduced to three rows and columns with a maximum depth of two. If two pions π^+ and π_1^- are missing, then the depth gets even further reduced to one.

At the bottom of Figure 7.2 the distribution of the training samples is shown, sorted by the number of leaves and the maximum depth in the training LCAG matrix, for 20 thousand training samples each. The dataset to validate and test the training consists of 10 thousand samples each, so half the size of the training dataset. The following cases are considered in this study:

best case
: This describes the best-case scenario, where every FSP could be reconstructed, therefore no missing particles are included. Furthermore, no B-decays with duplicates are included. Unmatched or secondary particles are ignored when building the training LCA matrix.

duplicates
: Describes the best-case scenario, but B-decays with duplicates are included as expected for the distribution. In this case 6% of decays include duplicate particles, and as shown in Figure 7.2, the number of duplicate particles can go up to four in this example decay. Compared to the full generic B-decay phasespace dataset, 32.0% (Table 6.2) of decays include duplicates.

missing
: Based on the best-case scenario, decays with missing particles are included. In this

case, the primary LCAG matrix is incomplete. Over 50% of decays have missing particles in this example decay, blurring the distribution. For the full dataset, this number is even larger, since with the current selections only 2% of all B-decays have all FSPs reconstructed (Section 6.2). Furthermore, the average of missing particles with 0.7 is also larger for the full generic B-decay phasespace with 3.2 missing particles on average. Therefore it is crucial to verify that the model is able to learn LCA matrices even with missing particles.

unmatched
Starting from the best-case scenario, unmatched particles are included. This first introduces the new class for background particles of zero.

secondaries
Based on the best-case scenario, secondary particles are included with the background class.

missing, unmatched, secondary
Here, the three mentioned reconstruction effects are included. B-decays including duplicates are not included here. This inflates the distribution greatly with up to 17 leaves. The most dense region of the dataset is at a depth of three with 5 leaves, so shifted from the true distribution. This shows how challenging this task of predicting the particle decay trees is.

realistic case
The realistic-case scenario describes the distribution that is expected when looking at data, so all four reconstruction effects are included. Compared to the previous dataset, this now gets inflated even further.

7.1.2. Single Decay Evaluation

Each of the single decay datasets shown in Figure 7.2 is used to train a model with the same hyperparameters and input features for a fair comparison between them. The performance on the validation set is shown in Figure 7.3 on the left. The accuracy is shown at the top and the perfect LCAG is shown at the bottom. Table 7.1 shows the results. The best-case scenario achieves an accuracy and perfect LCAG score of 99.6%, showing that the network is robust including the resolution effects. The model is also able to adapt to each individual reconstruction effect as the predictive capacity is over 99% for the perfect LCAG, when only including one of the detector effects consisting of missing, secondary, unmatched or duplicates.

Most importantly, the model is able to reconstruct decays including missing particles as stated in the previous Section 7.1.1. Although the primary LCAG matrix is incomplete, the model is able to predict nearly every LCAG matrix of the validation dataset correctly. This means the model is robust against missing particles and the training learns to predict LCAG matrices even with missing kinematic information.

The model is also able to correctly separate unmatched (secondary) particles from the primary LCAG matrix for 99.3% (99.5%) of decays in the validation dataset.

The performance drops when looking at decays where particles are missing and the two background effects of unmatched and secondary particles are included (Table 7.1). Although

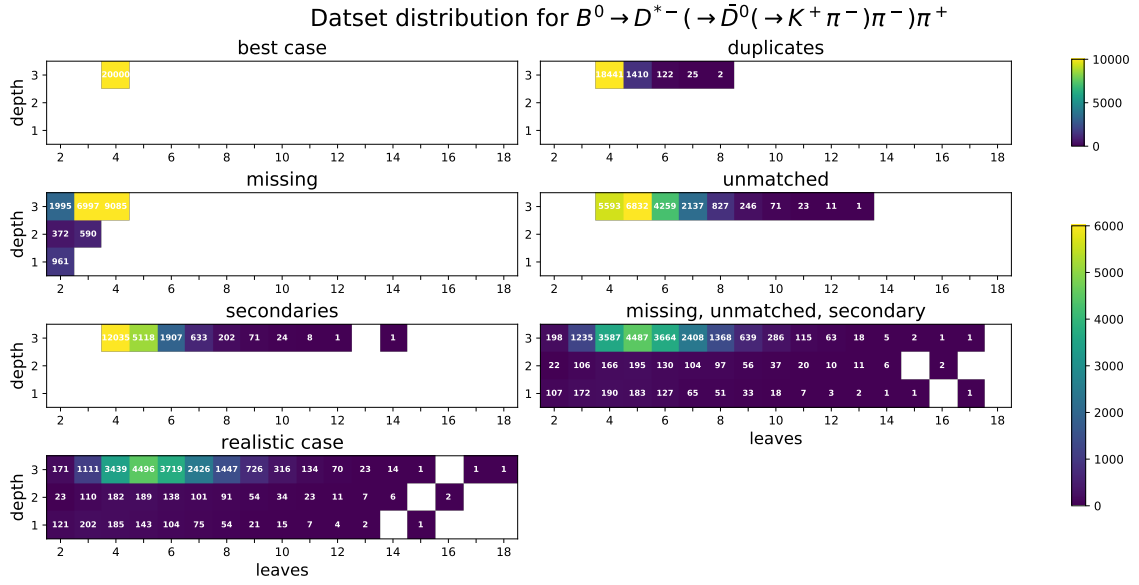


Figure 7.2.: The distributions for the Training LCAG matrices are shown with the different reconstruction effects, calculated on 20 000 samples in each case. For the best-case scenario, all FSPs were reconstructed and no additional reconstruction effects were included, the realistic case is where every effect is included.

Table 7.1.: The accuracy and perfect LCAG of the model of Figure 7.3 (left) evaluated on the validation set for the datasets defined in Section 7.1.1 for each effect of the experimental setup. The performance drops when including all these effects.

	perfect LCAG (%)	Accuracy (%)
best-case	99.6	99.8
duplicates	99.3	99.7
missing	99.3	99.7
unmatched	99.3	99.8
secondary	99.5	99.8
missing, unmatched, secondary	90.8	98.5
realistic-case	84.5	97.4

the accuracy is still at 98.5% in this case, the predictive capacity drops to 90.8%, which is 9% less compared to the other trainings. This shows that a very high accuracy is needed for the network to predict particle decays, and also shows that the accuracy is an imperfect measure of the network's performance. Looking at the distribution in Figure 7.2 shows that this case is more demanding than the previous ones.

Additionally, including duplicates for the realistic case causes the predictive capacity of the model to drop to 84.5%. The confusion matrix for this realistic-case training is shown in Figure 7.4 for the test dataset. The confusion matrix shows the distribution of the classes.

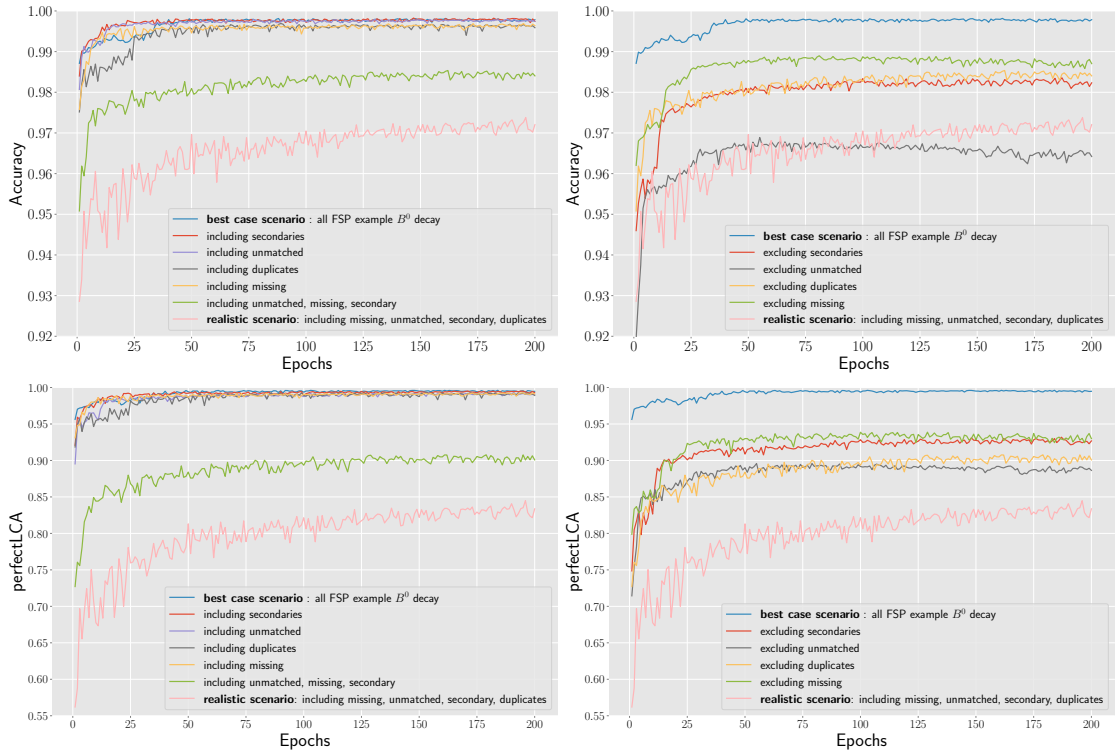


Figure 7.3.: Training on a single decay $B^0 \to D^{*-}(\to \overline{D}^0(\to K^+\pi^-)\pi^-)\pi^+$, comparison between smearing. The training was done on 20 000 samples each. Top shows the accuracy and the bottom shows the perfect LCAG. Left shows the validation performance on the datasets in section 7.1.1. Next to the best-case and realistic scenario, for each reconstruction effect a single training is done. On the right the validation of the training excluding each of these reconstruction effects is shown. This is to compare and evaluate the performance of these reconstruction effects as an ablation study.

Table 7.2.: Ablation study for the realistic-case scenario evaluated on perfect LCAG and Accuracy. Starting from the realistic case, for each training one effect of the experimental setup is excluded to evaluate the importance of this effect on the model performance.

	perfect LCAG (%)	Accuracy (%)
best-case	99.6	99.8
excluding duplicates	90.8	98.5
excluding missing	93.9	98.9
excluding unmatched	89.6	96.9
excluding secondary	93.2	98.4
realistic-case	84.5	97.4

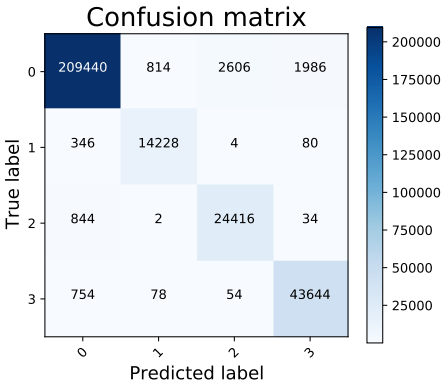


Figure 7.4.: Confusion matrix for the realistic case training of the decay $B^0 \rightarrow D^{*-}(\rightarrow \overline{D}^0(\rightarrow K^+\pi^-)\pi^-)\pi^+$. The labels one to three describe the edge labels for the primary LCAG matrix. The dominating class label is the background class of zero describing particles not belonging to the true B-decay.

The background class dominates the other classes. For the primary LCAG matrices, which consists of the true LCAG matrices, only 2.7% get falsely predicted. In comparison, the training LCAG matrices are 15.5% falsely predicted. The largest contributor to the loss of accuracy is the falsely predicted unconnected particle pairs. An unconnected particle pair is an edge where at least one of the particles is a background particle. As the background particles (unmatched and secondary) do not belong to the true particle decay, their edges are assigned a class label of zero. 71.1% of the false particle-pair predictions occur when an unconnected-edge gets assigned to the decay, as shown in the first row of Figure 7.4.

The primary particle-edges that get falsely predicted are falsely assigned to the background class, making up 25.3% of total false particle-pair predictions.

This shows that it is more difficult for the model to adapt to these background particles and correctly predict their respective edges. The background particles are additional particles occurring by primary particles interacting with the detector or beam background. Therefore they are harder to distinguish, as they do not belong to the original B-decay.

On the right of Figure 7.3 the results of the trainings excluding the respective reconstruction level are shown. Comparing these trainings results in Table 7.2, the impact of the reconstruction effects can be determined in this ablation study. For this, models are trained where, starting from the realistic case, each reconstruction effect is excluded for duplicates, missing, unmatched, and secondary particles. The perfect LCAG score for the realistic case is 84.5%. The smallest improvement is for unmatched particles, where the perfect LCAG score only improves by 5.1 percentage points. The model is therefore able to compensate for unmatched particles the easiest. Next are duplicate particles with 6.3 percentage points; although to note that the number of events including duplicates is higher regarding the full phasespace of generic B-decays (Section 7.1.1). Missing and Secondary particles are equally hard for the model to predict. These are both the reconstruction effects that correlate with the B-decay. Secondaries appear when primary particles interact with the detector. Missing particles result in an imperfect LCAG matrix.

Figure 7.3 also shows again that the accuracy is an imperfect measurement. The model trained on excluding unmatched achieves an accuracy of only 96.9% compared to the realistic-case model with 97.4%. Nevertheless, the perfect LCAG score is 5.1 percentage points higher for the model excluding unmatched.

Nevertheless, this first study on the simple decay shows that the model is able to correctly predict 85.4% of decays for the realistic case. It gives insight to the behaviour of the model for the reconstruction levels and verifies this approach.

7.2. Training on Mix of Selected Decays

The results in the previous section show that the model is able to predict LCAG matrices correctly for samples including all reconstruction effects described in Section 6.2. I therefore extend the study to show that the model is able to learn and predict different decays with varying FSPs and decay multiplicities. I train both on the best-case scenario and the realistic-case scenario. As in Section 7.1, the training on the best-case scenario verifies that the model is able to correctly predict decays if the feature resolution is smeared. The training on realistic-case scenario shows the behaviour of the model including the reconstruction effects of Section 6.2.

7.2.1. Mix of Selected Decays Dataset

Six distinct hadronic B^0-decays are chosen. They are decays selected out of decay channels the FEI is trained on. I selected decay channels with a high branching fraction to secure sufficient training samples for the least preprocessing expense. The decay channels are described in Table 7.3 including the motivation behind choosing the respective decay. This section focuses on hadronic decays; studies and results on selected semileptonic decay channels are shown in Appendix B.

The size of each of the decays is chosen according to the expected branching ratio, therefore the sample size per decay varies. This is to match the expected physics as the goal is to learn by example of the full MC simulated data coverage.

The decays selected from of 50 million $\Upsilon(4S) \to B^0\overline{B}^0$-events This results in 13 036 samples for training and 4398 for testing and validation for the best-case scenario. For the realistic scenario, there are 93 131 training samples and 31 390 samples for validation and testing. The percentage of each decay to the full dataset is shown in Table 7.4. Compared to the simple decay in Section 7.1.1, each reconstruction effect has a higher contribution for the realistic-case. Here I include events with photons. They are reconstructed by the ECL and are more often missing particles, as their energy can be lower than the selection criteria require. Furthermore, more secondary particles are selected and they occur in more decays. This is expected, as there are now more primary FSPs on average that can interact with the detector. Therefore, the reconstruction is expected to perform worse on these more complicated decays.

7.2.2. Training Evaluation

A combined training was performed on the mix of the selected decays for the best-case scenario as well as the realistic scenario. The results are shown in Table 7.4 for the test dataset.

Table 7.3.: Selected FEI channels for hadronic B^0-meson decays. Every π^0 decays into two photons as shown in the second decay. All decay channels have a high branching fraction in the MC files, enabling high statistics.

Decay channel	FSPs	Depth	Motivation
$B^0 \to D^-(\to K^+\pi^-\pi^-)\pi^+$	4	2	simple decay, one 2-body decay followed by one 3-body decay
$B^0 \to D^-(\to K^0_S(\to \pi^+\pi^-)\pi^-\pi^0(\to \gamma\gamma))\pi^+$	6	3	similar to above, particles in the 3-body decay decay further, includes photons
$B^0 \to D^-(\to K^+\pi^-\pi^-)\pi^+\pi^+\pi^-$	6	2	same number of FSPs as the decay above, shallower decay tree
$B^0 \to \overline{D}^0(\to K^+\pi^-\pi^0)\pi^+\pi^-$	6	3	different decay topology as the second decay for the same number of FPSs and depth of the tree
$B^0 \to D^{*-}(\to \overline{D}^0(\to K^+\pi^-\pi^0)\pi^-)\pi^+$	6	4	cascading decay with low number of FSPs and high depth
$B^0 \to D^{*-}(\to \overline{D}^0(\to K^+\pi^-\pi^0)\pi^-)\pi^+\pi^+\pi^-\pi^0$	10	4	high number of FSPs, complicated tree structure

Best-Case

The best-case scenario achieved a perfect LCAG rate for all combined decays of 76.8%. Three of the decays achieve a perfect LCAG score of over 90%.

Furthermore, the model is also able to learn to differentiate between the two decays with the same depth and number of FSPs but different decay topologies. For the most complicated decay $B^0 \to D^{*-}(\to \overline{D}^0(\to K^+\pi^-\pi^0)\pi^-)\pi^+\pi^+\pi^-\pi^0$ with ten FSPs and a depth of four, the predictive capacity is only 26.7%. Training this decay individually achieves a perfect LCAG score of 41.3% in comparison. This is also observed in Section 5.1, where the model is learning to predict easier decays first. Training individually has a higher performance for complicated decays as when combined with easier decays.

The model is able to predict a valid tree for 96.8% of the best-case dataset. This leads to a high purity of 78.0% for all decays. This example shows that the network is able to predict decay structures of different sizes of FSPs.

Realistic-Case

For the realistic case, the overall perfect LCAG score drops to 12.5%. Figure 7.5 shows the distributions of the dataset and the metrics. For the full dataset the accuracy is at 70%. The distribution shows that the perfect LCAG is higher for the decays with an easier structure. Easier decays are decays with a smaller depth and fewer leaves. The more complicated the decays get, the worse the perfect LCAG score gets. This is independent of the dataset

Table 7.4.: Results of the training on the combined data set consisting of the decays defined in table 7.3, as well as the evaluation on each individual decay channel used. On the top is the best case scenario where all final state particles are reconstructed and no additional particles are included, realistic case is further down including unmatched and secondary particles as well as events with missing particles.

	Decays	Size (%)	perfect LCAG (%)	accuracy (%)	valid-tree efficiency (%)
	Full Dataset	100	76.8	91.0	96.8
best case	$B^0 \to D^-(\to K^+\pi^-\pi^-)\pi^+$	30.3	95.4	97.2	98.8
best case	$B^0 \to D^-(\to K^0_S(\to \pi^+\pi^-)\pi^-\pi^0(\to \gamma\gamma))\pi^+$	8.7	93.3	98.4	99.0
best case	$B^0 \to D^-(\to K^+\pi^-\pi^-)\pi^+\pi^+\pi^-$	15.0	70.0	93.0	95.3
best case	$B^0 \to \overline{D}^0(\to K^+\pi^-\pi^0)\pi^+\pi^-$	9.1	74.0	91.9	97.5
best case	$B^0 \to D^{*-}(\to \overline{D}^0(\to K^+\pi^-\pi^0)\pi^-)\pi^+$	19.4	92.1	97.6	96.6
best case	$B^0 \to D^{*-}(\to \overline{D}^0(\to K^+\pi^-\pi^0)\pi^-)\pi^+\pi^+\pi^-\pi^0$	17.5	26.7	84.9	93.5
	Full Dataset	100	12.5	70.0	55.9
realistic case	$B^0 \to D^-(\to K^+\pi^-\pi^-)\pi^+$	8.4	61.9	88.2	88.2
realistic case	$B^0 \to D^-(\to K^0_S(\to \pi^+\pi^-)\pi^-\pi^0(\to \gamma\gamma))\pi^+$	4.8	27.6	77.8	75.4
realistic case	$B^0 \to D^-(\to K^+\pi^-\pi^-)\pi^+\pi^+\pi^-$	7.4	28.8	80.1	71.6
realistic case	$B^0 \to \overline{D}^0(\to K^+\pi^-\pi^0)\pi^+\pi^-$	4.4	15.5	69.0	74.8
realistic case	$B^0 \to D^{*-}(\to \overline{D}^0(\to K^+\pi^-\pi^0)\pi^-)\pi^+$	10.7	25.1	77.3	72.9
realistic case	$B^0 \to D^{*-}(\to \overline{D}^0(\to K^+\pi^-\pi^0)\pi^-)\pi^+\pi^+\pi^-\pi^0$	64.3	1.5	67.9	45.3

distribution. For example, 64.3% of the dataset consists of the most complicated decay $B^0 \to D^{*-}(\to \overline{D}^0(\to K^+\pi^-\pi^0)\pi^-)\pi^+\pi^+\pi^-\pi^0$. The perfect LCAG score for this decay is nevertheless only 1.5%. This is also shown for the distribution of the accuracy. The easier decays have a higher accuracy. For the realistic case, the model predicts a valid tree for over half of the decays, resulting in a purity of 22.4%. The valid-tree efficiency is also biased towards easier LCAS matrices.

At this point, I showed that the model is able to correctly predict LCAG matrices for the best-case scenario for a mix of decays. But it struggles with more complicated decays and including the reconstruction effects.

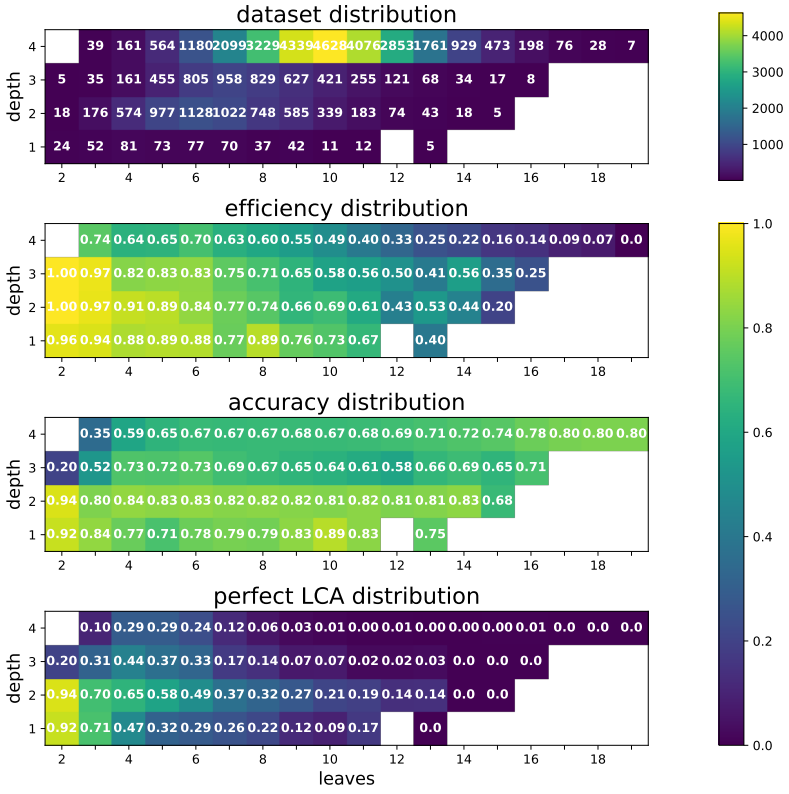


Figure 7.5.: Evaluation on the test dataset for the mix of selected decays. The distribution is divided by the number of leaves (FSPs) and depth of the training LCAG matrix (max generation in the training LCA matrix). The distribution is shown for the dataset distribution and the valid-tree efficiency, accuracy and perfect LCAG metrics.

7.3. Update Particle selections

To summarize the previous studies, the graFEI approach can be applied to large phasespaces Section 5.1. In section 7.1 I show, that the model is able to learn to predict decay tree structures including resolution and reconstruction effects. The model achieves a high perfect LCAG score for each individual reconstruction effect. When adding all these effects at once in the realistic case, the performance of the model drops. This is also observed in section 7.2, where also the limitations of this approach is shown: the model is learning to predict the easier decay.

I performed additional trainings on a transformer model [47] (Section 3.3) to create a baseline. For this, I started with the best-case scenario and added decays including missing particles. The perfect LCAG of the NRI model (Section 4.2) achieved a score of 43.2% compared to the 31.8% of the transformer model. This validates the use of the NRI model as the current best model for this task.

Therefore I first improve the FSPs selection. Section 7.1 shows that missing particles and secondary particles had the most impact on the results for this simple decay. The trainings on Section 7.2 confirms this for the realistic case. Compared to the best-case scenario, the performance drops by 60 percentage points. Using the dataset of Section 7.2.1 also shows that adding duplicates nearly halves the performance of the model. The perfect LCAG drops from 22.3% to 12.5%. The goal is to improve the selections so that fewer background particles are included. Furthermore, fewer events should contain duplicate particles, while not losing primary particles with these new selections. To confirm that the FSPs selection improves the training I show a comparison to the previous results.

7.3.1. Investigate Final State Particles

The new versus previous selection criteria are shown in Table 7.5. To ensure that the selected FSPs belong to the B-decay, an additional requirement is added regarding the interaction. For charged particles, the transverse distance of the particle track with respect to the interaction point d_r has to be less than 0.5 cm. Furthermore, the absolute distance on the z-axis of the track to the interaction point d_z has to be less than 2 cm. This improves the selection for all primary charged particles from 86% to 96%. The number of unmatched and secondary particles is reduced to 3% from 14%.

The only exception is K_S^0 with a relatively long lifetime of $10 \cdot 10^{-10}$ s, which results in a secondary decay vertex. The largest hadronic branching fraction of K_S^0 is $K_S^0 \to \pi^-\pi^+$ at 69.2% [13], which includes two charged particles. K_S^0 could therefore be removed by the vertex selections, although they do belong to the original B-decay. It would be possible to account for them by adding K_S^0 as additional FSPs, and then excluding them from these selection criteria. This is out of the scope of this work but warrants further investigation.

Although the selection for nCDChits to be larger than 20 hits improves the identification rate for FSPs, it also removes a lot of true FSPs with low energy. Only 30% of the particles with less than 20 hits in the CDC are correctly identified. But the previous studies in Section 7.1 show that having a lot of missing particles affects the prediction capacity greatly. Therefore, this selection is removed to evaluate if the model is able to compensate for this misidentification with the PID input variables defined in Section 6.3.

Table 7.5.: List of the final state particles and the corresponding selections, with the first selection and the updated selection on the right.

	previous selection criteria	new selection criteria
charged particles $e^{\pm}, \mu^{\pm}, \pi^{\pm}, K^{\pm}, p$	$17° < \theta < 150°$ nCDChits > 20	$17° < \theta < 150°$ $d_r < 0.5\,\text{cm}$ $\lvert d_z \rvert < 2\,\text{cm}$
photons γ	$17° < \theta < 150°$ $\Delta t_{\text{ECL}} < 1 \cdot 10^6\,\text{ns}$ $\frac{E_1}{E_9} > 0.4$ or E > 0.075 Cluster forward/barrel $E >$ 50 MeV, backward $E >$ 75 MeV	$17° < \theta < 150°$ $\Delta t_{\text{ECL}} < 1 \cdot 10^6\,\text{ns}$ $\frac{E_1}{E_9} > 0.4$ or E > 0.075 Cluster forward/barrel $E >$ 50 MeV, backward $E >$ 75 MeV $\lvert t_{\text{ECL}} \rvert <$200 ns $\lvert d_{\text{ECL,T}} \rvert > 40$ or $E > 0.4\,\text{GeV}$

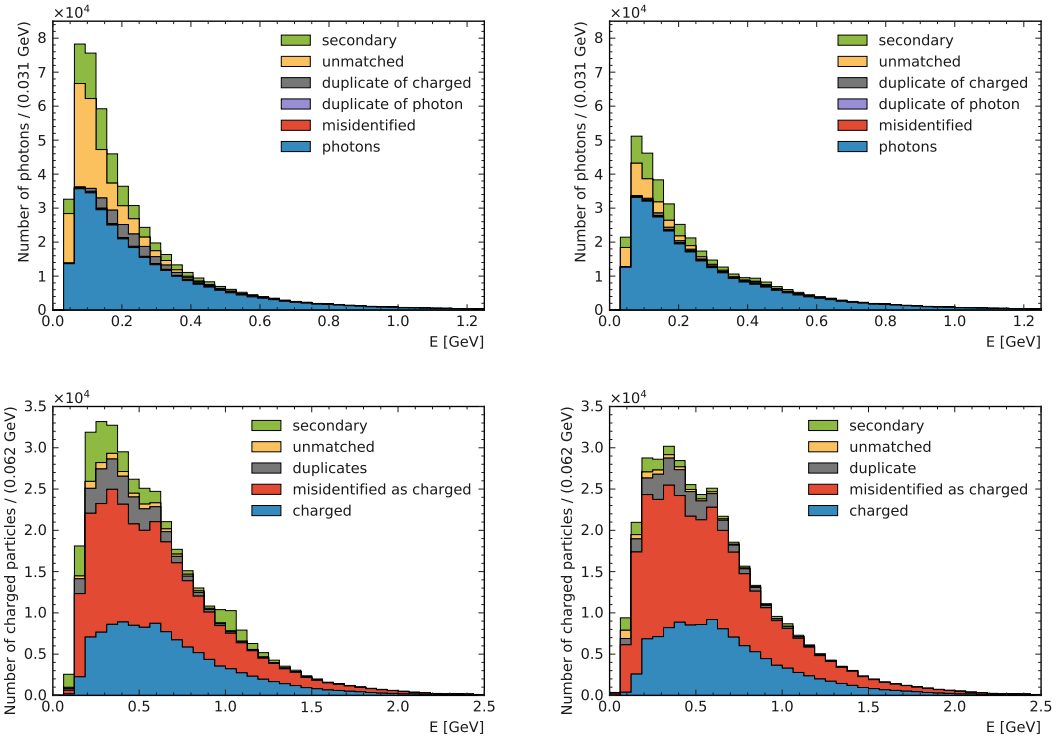


Figure 7.6.: Energy distribution for reconstructed final state particles with previous selection criteria (left) and updated selection criteria (right). Charged particles misidentify only as other charged particles.

Investigating the duplicates reveals that 40% of the duplicates are charged particles that

get additionally reconstructed as a photon. Other duplicates are charged particles that get reconstructed additional times but with different directions of the four-momenta. Therefore there are two additional selections applied when selecting photons.

The distance between the ECL cluster and nearest track $\left|d_{\mathrm{ECL,T}}\right|$ should be at least 40 cm or the deposited energy E should be greater than 0.4 GeV. To remove photons not belonging to the event an additional selection is applied, by requiring that the cluster timing $|t_{\mathrm{ECL}}|$ is less than 200 ns. If calculated correctly, $|t_{\mathrm{ECL}}|$ should be zero. This ensures that charged particles are not additionally reconstructed as photons.

A comparison of the FSPs for the updated selections is shown here Figure 7.6 for their respective energy distribution. Comparing the previous selections on the left with the resulting ones on the right show that the photon selections removed a large portion of secondary and unmatched particles as well as nearly all duplicate photons. For the charged particles, overall, more misidentified particles are included for lower energies, but since these are also true FSPs, the performance could improve. There are also fewer secondary and unmatched particles. The updated numbers are shown in Table 7.6. Overall the number of unmatched particles get reduced from 3 in 92% of decays to 1.7 and only occur in 60% of events. Secondary particles also are reduced, for hadronic decays they only occur in half of the events with an average of 0.97 secondary particles compared to the previous 1.9. Duplicates are only in 7% of hadronic decays compared to the 30% before. Despite reducing these background effects, the number of missing particles went up by 1% from 4.12 to 4.18 particles. B-decays were all FSPs where reconstructed correctly went up from 1.6% to 2.3% for hadronic decays.

Table 7.6.: The particle distribution for the generic B-decays calculated on 200,000 events, expecting 400,000 B-decays, including the updated selection criteria defined in Table 7.5

	hadronic	semileptonic
Number of B-Decays	200785	186552
Skipped B-Decays (<2 primaries)	202	721
Average FSPs (min/max)	8.94 (2, 26)	6.78 (2, 25)
Average reconstructed FSPs (min/max)	8.87 (2, 26)	6.71 (2, 25)
Avg LCA len. (min/max)	13.05 (2, 35)	9.79 (2, 33)
B-Decays with secondary FSPs	105475 (52.53%)	71503 (38.33%)
Avg secondaries (min/max)	0.97 (0, 12)	0.62 (0, 15)
B-Decays with duplicate FSPs	14002 (6.97%)	13251 (7.1%)
Avg duplicate FSPs (min/max)	0.07 (0, 5)	0.08 (0, 4)
B-Decays with duplicate and secondary FSPs	112405 (55.98%)	79792 (42.77%)
B-Decays with missing FSPs	195889 (97.56%)	172969 (92.72%)
Avg missing FSPs (min/max)	4.18 (0, 18)	3.08 (0, 19)
B-Decays with perfectly reconstructed FSPs	4566 (2.27%)	12634 (6.77%)
Events with unmatched FSPs	115844 (59.82%)	
Average unmatched (min/max)	1.7 (1, 12)	

7.3.2. Comparison on Performance

Using these new selections, I repeat the previous studies to compare the performance with the updated datasets. I use the same model hyperparameter configuration for these repeated trainings. The first results are on the single decay training of Section 7.1. The performance on the validation set improved by 7 percentage points to 93.1% total for perfect LCAG.

Regarding the mix of hadronic decays, I used the same 50 million $\Upsilon(4S) \to B^0\overline{B}^0$ events for the training, validation and test datasets. The perfect LCAG score doubled with these new, updated selections, as shown in Table 7.7. The model trained on samples using the updated selections is able to predict 14.9% of decay trees correctly for the most complicated decay $B^0 \to D^{*-}(\to \overline{D}^0(\to K^+\pi^-\pi^0)\pi^-)\pi^+\pi^+\pi^-\pi^0$ in contrast to the previous result of 1.5%. As expected for the decay including the K^0_S, the number of missing particles for this specific decay went up with on average 1.29 missing particles per decay to 2.02 average missing particles. Nevertheless, the model is able to predict 61.7% of decay trees correctly. This is an overall improvement despite having more missing particles.

Furthermore, I explored the impact of the class weights on the performance. For both trainings, including class weights impaired the predictive capacity of the model. This occurs despite the background class of zero dominating the edges for the previous selections (Section 7.1.2). In Table 7.7 the primary scores are shown for the trainings in parentheses. These are calculated ignoring the background classes and only regard the original decay tree. For the previous selections the primary perfect LCAG score improved 1.0 percentage point by adding class weights, but the overall perfect LCAG score worsened by 6.7 percentage points. For the updated results the (primary) perfect LCAG improved 4.4 (4.8) percentage points, when not including the class weights. The purity also improves when removing the class weights. The number of decays that the model predicts a valid tree for increases less than the number of decays where the tree is predicted correctly. For both selections, the purity without class weights is around 43%, although it is slightly better for the previous selections. This impact on the purity is not large enough to compensate for the overall improvement regarding the perfect LCAG with these updated selections. For the training on the full MC simulated samples class weights, I am therefore no longer including class weights to improve the performance of the model. The promising results of this section warrant the proceeding of this method to the full generic Belle II simulated datasets in Chapter 8.

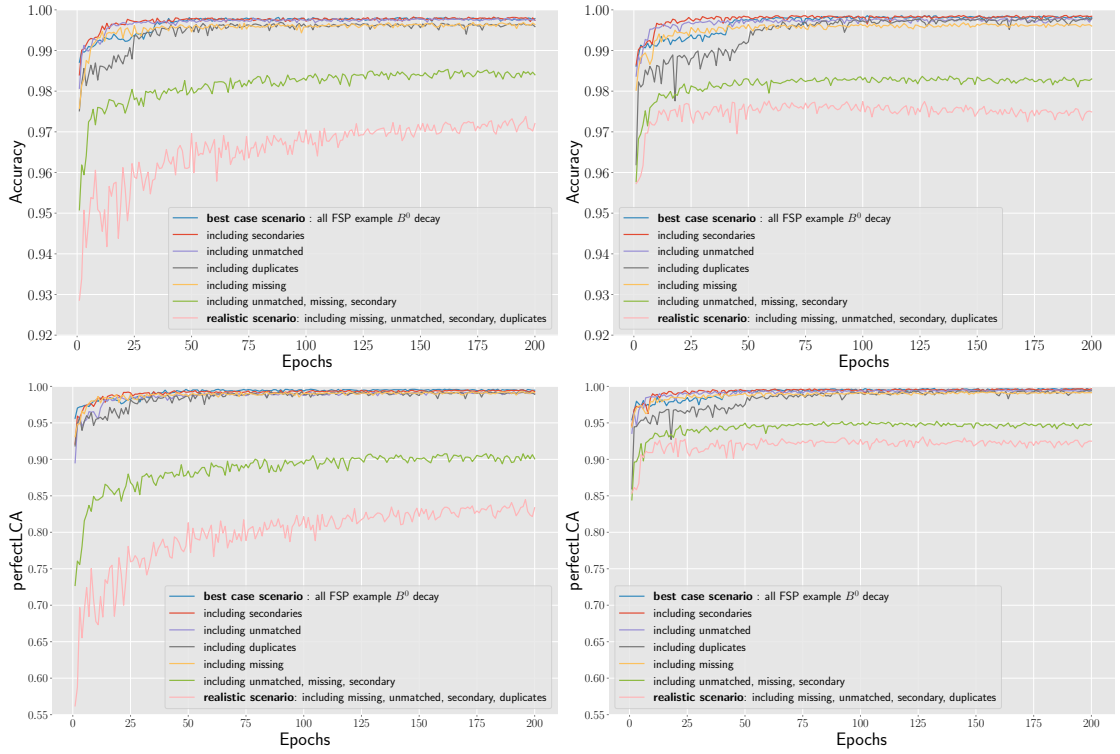


Figure 7.7.: Updated single decay, old cuts on the left and new cuts on the right

Table 7.7.: Results on the mix of hardonic decays for the best case, missing and realistic scenario with and without class weights added. On the left are the results on the dataset with the previously used selection criteria, on the right on the dataset with the new ones.

	previous selections			updated selections		
	perfect (primary) LCA(G) (%)	accuracy (primary) (%)	valid-tree efficiency (%)	perfect (primary) LCA(G) (%)	accuracy (primary) (%)	valid-tree efficiency (%)
best case	76.7	91.0	97.3	72.4	90.8	96.5
incl missing	37.7	82.2	88.5	43.1	85.3	90.5
realistic case	12.5 (27.2)	70.0 (73.7)	55.9	25.6 (32.7)	74.7 (77.5)	66.5
realistic w/o CW	19.2 (26.3)	79.9 (71.0)	44.2	30.0 (37.5)	78.7 (81.2)	70.8

8. B-Decay Reconstruction on Full Simulated Belle II Dataset using graFEI

Using the training strategies developed in the previous chapters, I extend the training to the full generic hadronic simulated B^0-decay dataset. These trainings are performed to explore whether the LCA representations can be applied as a machine learning training target on datasets covering the large number of decay channels at the Belle II experiment. The goal is for the model to learn the LCA representation and correctly predict it for single hadronic B^0-decays. Applying such a method to a physics analysis would first require the reconstruction of the signal-side B-meson. Predicting the LCA matrix with the remaining particles in the event then allows for the reconstruction of the tag-side B-meson to constrain the signal-side B-meson. I evaluate the trained models on $\Upsilon(4S) \to B^0\overline{B}^0$ events, where one of the B^0-mesons decays in two neutrinos $B^0 \to \nu\bar{\nu}$. As neutrinos pass the detector undetected, every event consists of detected particles originating from the tag-side B-decay. This provides an unmodified *monogeneric* dataset to test this approach. Furthermore, I study the two representations for the LCA matrix described in Section 4.1, namely the generational approach (LCAG) and the staged approach similar to the `FEI` (LCAS).

Section 8.1 focuses on the training of the models. First, I investigate the nominal best-case reconstruction scenario. This consists of B-decays with only primary FSPs, excluding background particles such as unmatched and secondary particles and events with duplicates. This provides a controlled environment to study the dependence of the model on the number of possible decay channels and the uneven distribution of training samples per decay channel. Next, I study the performance of the model on the realistic reconstruction scenario, including all the aforementioned reconstruction effects (Section 6.2).

The trained models of the realistic scenario are applied to the full test dataset including semileptonic B^0-decays in Section 8.2. The semileptonic B^0-decays are unknown decay-tree structures. The model was never exposed to these decay-tree structures, neither in training nor validation processes. This is to investigate how the model handles unknown LCA matrix structures that follow the same physical laws and whether the model is able to adapt to these decays. I then evaluate the beam-constrained mass on hadronic B^0-decays predicted by the model.

In Section 8.3 I compare the graFEI approach with the current reconstruction algorithm, `FEI`, on the same test dataset, using the metrics established in Section 4.3.

Finally, I evaluate the model on generic $\Upsilon(4S) \to B^0\overline{B}^0$ events in Section 8.4, in order to gain an insight into the predictions of the model on double-generic background events.

8.1. Training graFEI on Hadronic B-Decays

8.1.1. Nominal Best-Case Scenario

At first, I trained a model for each LCA matrix representation defined in Section 4.1 on the hadronic nominal best-case scenario. These representations are the FEI-inspired stage-wise LCAS representation and the LCAG generational approach. The sample size for this training is 1.8 million samples for training and 400 thousand samples for validation. The best-case scenario consists of decays where all FSPs are reconstructed, excluding duplicates. Unmatched and secondary background particles are ignored when building the training LCA matrix. The number of different decay topologies for the Belle II dataset is three orders of magnitude larger than in the previous phasespace dataset. Furthermore, the numbers of training samples per topology vary, as shown in Chapter 7. Therefore, this is expected to be a more challenging training dataset than the phasespace dataset in Section 5.1 for the model to learn. The `Optuna` hyperparameter search for the nominal best-case scenario in Appendix A confirmed the results of Section 5.3; the perfect LCA score increases with larger NRI forward layer widths. Therefore both the LCAS-model and LCAG-model are trained with the exact same set of hyperparameters. Due to hardware constraints, the size of NRI feedforward layer widths is restricted to 512. The same training dataset and input features are used to compare the two representations. The test dataset for the best-case scenario consists of 170 thousand hadronic B-decay samples.

LCAG Generational View

Training with the LCAG representation achieves an overall perfect LCAG score of 22.1% on the validation set and 21.6% on the test dataset, which can be attributed to statistical fluctuation due to the smaller size of samples. The total accuracy is 60.4% (61.0%) on the test (validation) dataset. Training the LCAG-model with the new selections discussed in Section 7.3 improved the training from 11.4% to 22.1% on the validation set, thereby doubling the predictive capacity.

The model is predicting a valid tree for 62.5% of samples, which results in a purity of 34.56%. Figure 8.1 shows the distributions of these metrics. As in the previous studies, the model learns to predict the simple decays as seen from the perfect LCAG and accuracy distribution. The model is able to correctly predict 50% of decay tree structures up until six leaves. For more than 14 FSPs the model is no longer able to correctly predict any tree structures. Note that the `FEI` combines at most 14 FSPs to reconstruct a B-meson.

LCAS Stage View

The LCAS representation uses the `FEI` stages to encode the LCA matrix. Since the particles are divided into groups, this is a very different approach to the LCAG matrix. As with the `FEI`, resonances are skipped, resulting in some missing information about the decay tree. Compared to the LCAG matrix, it is potentially more difficult to identify missing particles

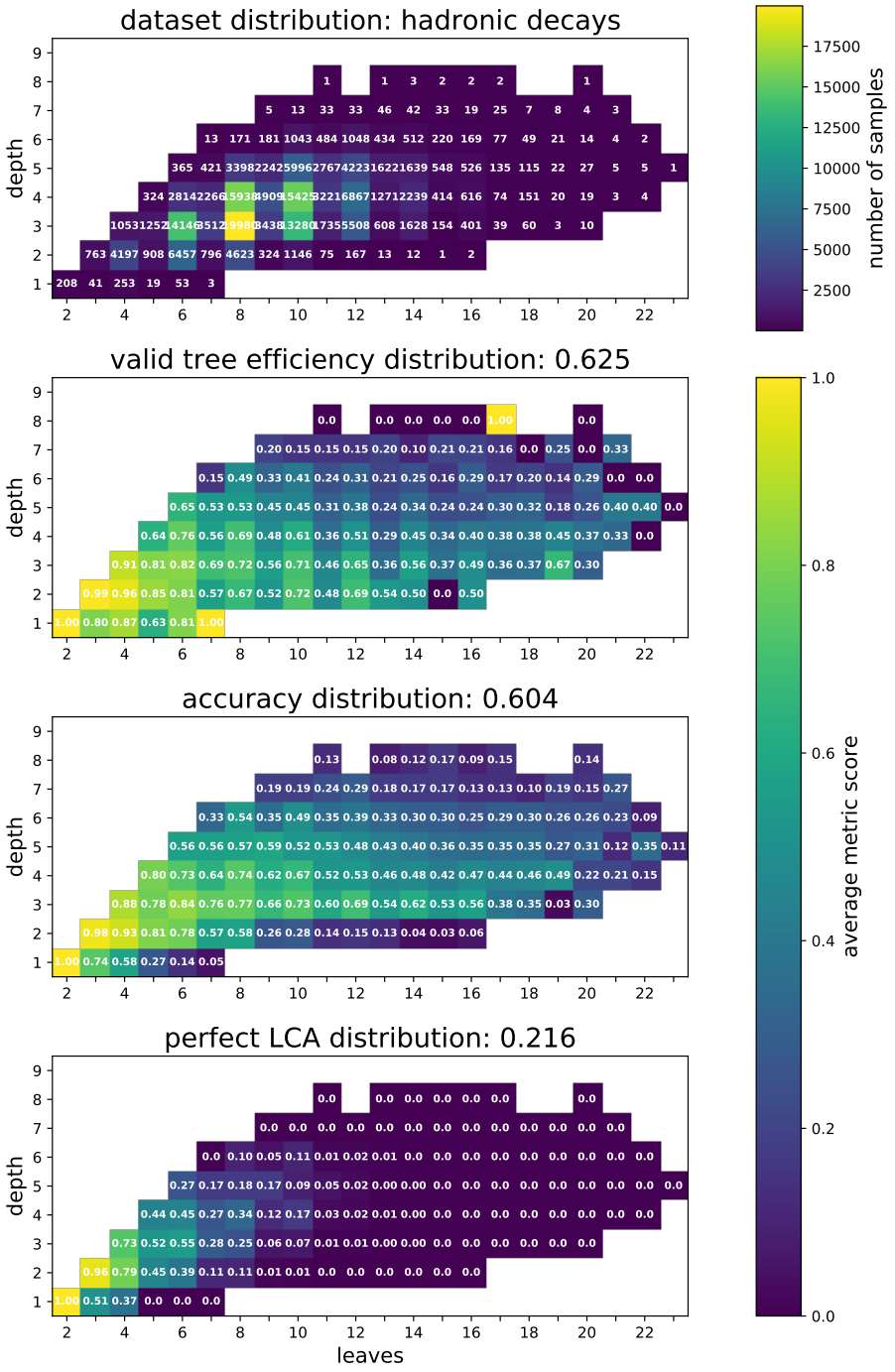


Figure 8.1.: Evaluation of the generic hadronic B-decay trained model on the test dataset, for the best-case scenario only including the full set of selected primary FSPs. From top to bottom are the dataset distribution, efficiency, accuracy, and perfect LCAG score split by the number of leaves and depth of the decay tree. The metric titles include the average metric value on the whole test dataset.

in the LCA matrix. The advantage, however, is that the classes are directly correlated with the intermediate particles, allowing analysts to include requirements on the masses of the intermediate particles to improve the analysis.

The nominal best-case includes only events where all primary FSPs were reconstructed. For the LCAS representation, the distribution here is only ranked according to the number of leaves in Figure 8.2, as the depth is always fixed to five due to the LCAS matrix stages in the best-case. The perfect LCAS score is 48.7%[1] on the test dataset compared to the 51.1% on the validation dataset, showing slight overtraining. Compared to the LCAG matrix, using the LCAS matrix as a representation improves the performance of the model by a factor of 2.25. This is consistent with the discussion in chapter 4, where it was stated Chapter 4 that the more classes there are, the harder the prediction is. The LCAS representation is using five classes instead of the maximum of eight used by the LCAG representation.

The model predicts a valid tree for 93.6% of the dataset, with a purity of 52% of these trees being correctly predicted. The purity therefore improved compared to the LCAG-model, but the model is also predicting a valid tree for nearly every sample in the test dataset.

Again, the model is learning to predict the easier decays with fewer FSPs. Another observation is that the model is learning to predict even numbers of FSPs better than uneven ones for FSPs between 5 and 22. Decays with an uneven number of FSPs include photons, as the charge has to add up to zero for B^0-mesons. Photon features have limited information, as only the undirected ECL cluster information of the detector (Section 2.3) is available. This complicates the model learning by conservation laws and could explain the worse performance for decays with an uneven number of FSPs for less than ten FSPs. For more than ten FSPs, the average number of primary photons in the event increases mainly linearly. Here, the higher contribution of decays with an even number of FSPs to the dataset could explain the better performance. Furthermore, the model is not predicting LCAS matrices with more than 20 FSPs. Even though only 1% of decays with 15 to 20 FSPs are correctly predicted, this is still 6 FSPs more than the LCAG-model or `FEI` are able to reconstruct.

8.1.2. Realistic Scenario

With these promising results on the best-case scenario trainings, the training is extended to the realistic scenario. To compare the results with the generic `FEI` (Section 2.5), I train the graFEI models on the same set of monogeneric hadronic B^0-decays. The training targets are defined as the ones in Section 6.2 and include decays with missing particles, as well as unmatched and secondaries.

The final training on the realistic hadronic simulated samples is carried out on `HoreKA`, on a node using two Intel Xeon Platinum 8368 CPUs, 512 GB of RAM to load the whole dataset, and four NVIDIA-A100 40GB to train. With 37 million hadronic B^0-decay training samples and 10 million validation samples, this training is time and memory consuming. A full epoch with the full dataset takes close to five hours to evaluate the training process. Therefore the training is updated step-wise on subsets of the full dataset. Each training

[1] The new selections also improved the training by 25 percentage points total, doubling the predictive capacity.

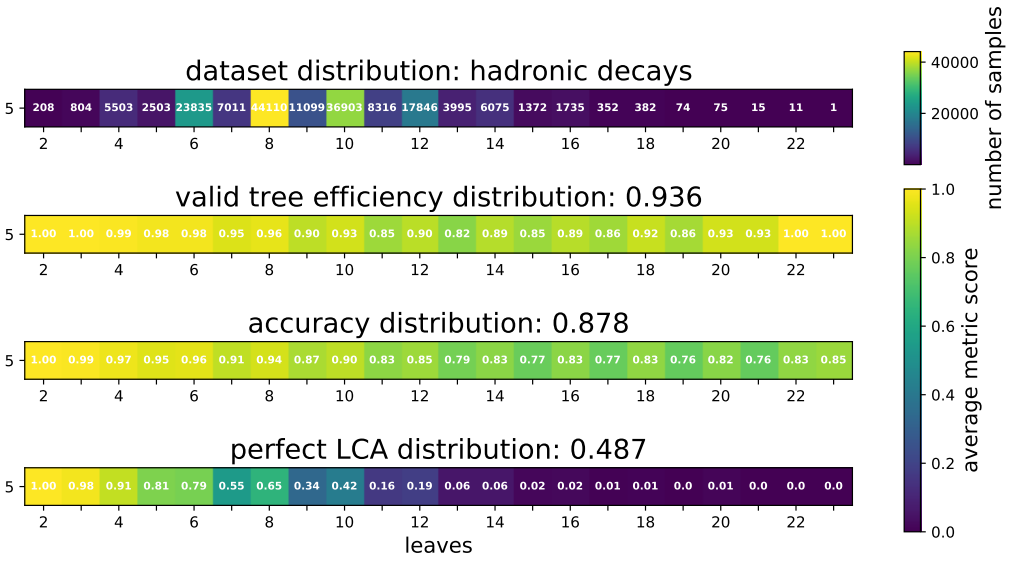


Figure 8.2.: Evaluation of the best-case hadronic trained model on the hadronic test dataset, showing the dataset distribution at the top and the efficiency, accuracy, and perfect LCAS score split by the number of leaves. The metric titles include the average metric value on the whole test dataset.

step consists of 30 000 batches with 128 samples per batch, which takes around 35 minutes. The model converges after 100 epochs, resulting in 58 hours total training time on `HoreKA`. For comparison, the `FEI` is trained on 113 million $\Upsilon(4S) \to B\overline{B}$ events. Previous studies on the phasespace dataset (Chapter 5) showed that more data improved the perfect LCAG score of the model. This is also the case for the best-case scenario, for the LCAG-model: the predictive capacity for the model improved from 9.7% when trained on 300 thousand training samples to 21.6% when trained on a dataset with 6 times the sample size in Section 8.1.1. Therefore training on larger numbers of samples would likely improve the following results even further.

These models are evaluated on a subset of 1 million hadronic B^0-decays in this section.

LCAG Generational View

The perfect LCAG score on decays of the hadronic test dataset is 1.8% with a purity of 6.1%. The model is mostly predicting easier decay trees as shown in Figure 8.3. This could also be because these decays are more likely to provide a distinct kinematic pattern. Especially compared to the background particles, for easier decays high energy primary particles are expected, making it easier to distinguish these decays. 1% of the decays in the most dense region of the dataset are predicted correctly (depth of four with eighth to twelve leaves), showing the limitations of this approach.

Previous trainings showed, that the perfect LCAG score decreased for the more complicated decays when combined with easier decays, compared to training with similarly complex decay trees. To improve the perfect LCAG score for the more complex decays, one could

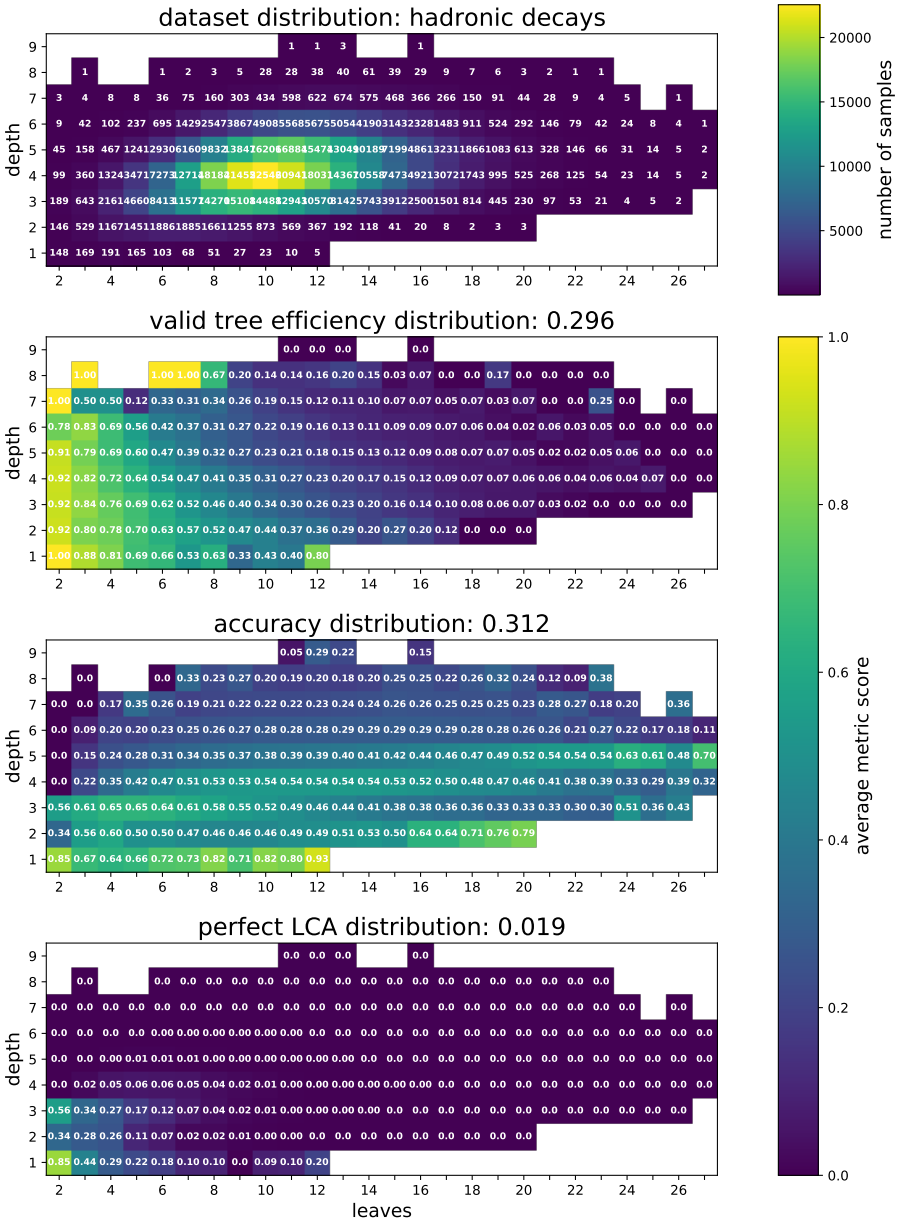


Figure 8.3.: Evaluation of the monogeneric hadronic trained LCAG-model on the test dataset, for the realistic case. From top to bottom are the dataset distribution, efficiency, accuracy, and perfect LCAG score split by the number of leaves and depth of the decay tree.

limit the dataset to the most dense region described above. Furthermore limiting the dataset to a maximum number of FSPs as the `FEI` does, could also improve the overall perfect LCAG score and speed up the training tremendously (Section 6.4), while still covering a larger branching fraction than the `FEI`. This approach could be used in analysis-specific trainings. Further investigation about feature scaling could improve the model performance for the

Belle II simulated datasets. Another approach is to optimize the input feature selection. For example, including more information for photons could improve the performance of the model.

LCAS Stage View

The prediction of the LCAS-model is better than the LCAG-model, now 3.7 times the perfect LCA score compared to the LCAG-model. The purity is also better; the model is predicting a valid tree for 51.7% of the test B-decays, 13.0% of these predicted trees are correct. The accuracy is 40% higher than the LCAG-model training and overall better distributed across the dataset. The most dense region here cannot be directly compared with the LCAG-model evaluation, but there is also the trend that easier decays are predicted correctly as shown in Figure 8.4. This distribution also shows the difference between the two

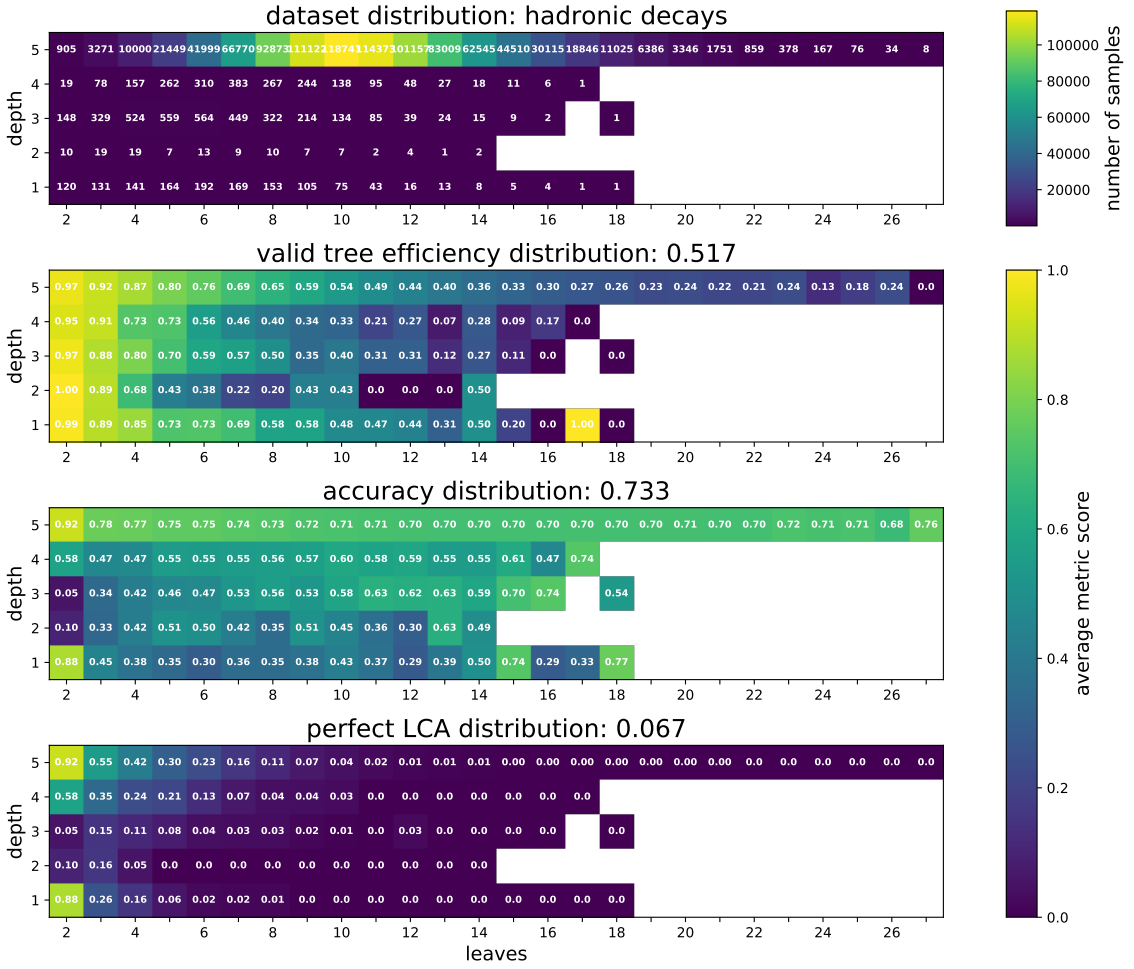


Figure 8.4.: Evaluation of the monogeneric hadronic trained LCAS-model on the test dataset, for the realistic case. From top to bottom are the dataset distribution, efficiency, accuracy, and perfect LCAS score split by the number of leaves and depth of the decay tree.

LCA representations: The LCAG matrix shows the B-meson at the maximum generation, whereas the LCAS matrix always has the B-meson root at the depth of five. As missing particles are included here, the maximum entry in the LCAS matrix can be smaller than five. If, for example, both pions of the $K^0_S \to \pi^+\pi^-$ in the decay $B^0 \to J/\psi K^0_S$ were not detected or excluded by the selection criteria, this would result in the LCAS matrix with a depth of 1.

8.2. Applying and Evaluating graFEI

The full monogeneric test dataset consists of 12 million $\Upsilon(4S) \to B^0\overline{B}^0$ events, where one B^0-meson decays in two neutrinos $B^0 \to \nu\bar{\nu}$. In contrast to the previous evaluation in Section 8.1.2 on a subset of hadronic B^0-decays, the trained models are now applied to this full tag-side test dataset including semileptonic decays. For each event, the predicted LCA matrix is stored together with the information of whether this is a valid tree. Then, in an additional step, the B-meson can be reconstructed out of the selected FSPs using this predicted LCA matrix.

8.2.1. Beam-constrained Mass

The Physics of the B Factories [17] (Chapter 7.1) defines the beam constrained mass M_{bc} to distinguish between background and signal events for hadronic decays[2]. This established procedure is also used in [6], where the currently used reconstruction algorithm `FEI` was compared with the predecessor. I use the beam constrained mass M_{bc} to evaluate the performance of the model in addition to the graFEI metrics (Section 4.3). This confirms the model is learning to predict B-mesons that are relevant for analyses.

The beam constrained mass is calculated according to [17] by using the reconstructed 3D-momentum of the predicted B-meson, $\boldsymbol{p}_{\mathrm{B}}^{\mathrm{CMS}}$, and half of the center-of-mass energy of the collision beam, $E_{\mathrm{beam}}^{\mathrm{CMS}}$:

$$M_{\mathrm{bc}} = \sqrt{(E_{\mathrm{beam}}^{\mathrm{CMS}})^2/c^4 - (\boldsymbol{p}_{\mathrm{B}}^{\mathrm{CMS}})^2/c^2}. \tag{8.1}$$

The 3D-momentum for the predicted B-meson is calculated by adding up the 3D-momenta for the predicted primary FSPs. If the model is predicting a valid tree for an event, the predicted LCA matrix is used to select the predicted primary FSPs. The 3D-momenta of these predicted primary FSPs are used to calculate the beam constrained mass for the event. The predicted background leaves are not included in this reconstruction. Unlike the fully inclusive b-tagging method (section 2.3), only the particles that are predicted to belong to the decay are included here.

Using the beam energy $E_{\mathrm{beam}}^{\mathrm{CMS}}$ is more beneficial than using the energy of the predicted B-meson, as the B-meson energy is calculated including the mass hypothesis of the FSPs. The misidentification rate is 19.8% for all primary FSPs due to the current selection of the FSPs. They are currently only chosen by their highest PID without taking further constraints into account. This choice is made by design in Section 7.3, as the idea is for the

[2] ΔE requires further knowledge of the selected FSPs, that is out of the scope of this thesis. Here, the focus is on exploring if the model is able to learn the decay-tree representation.

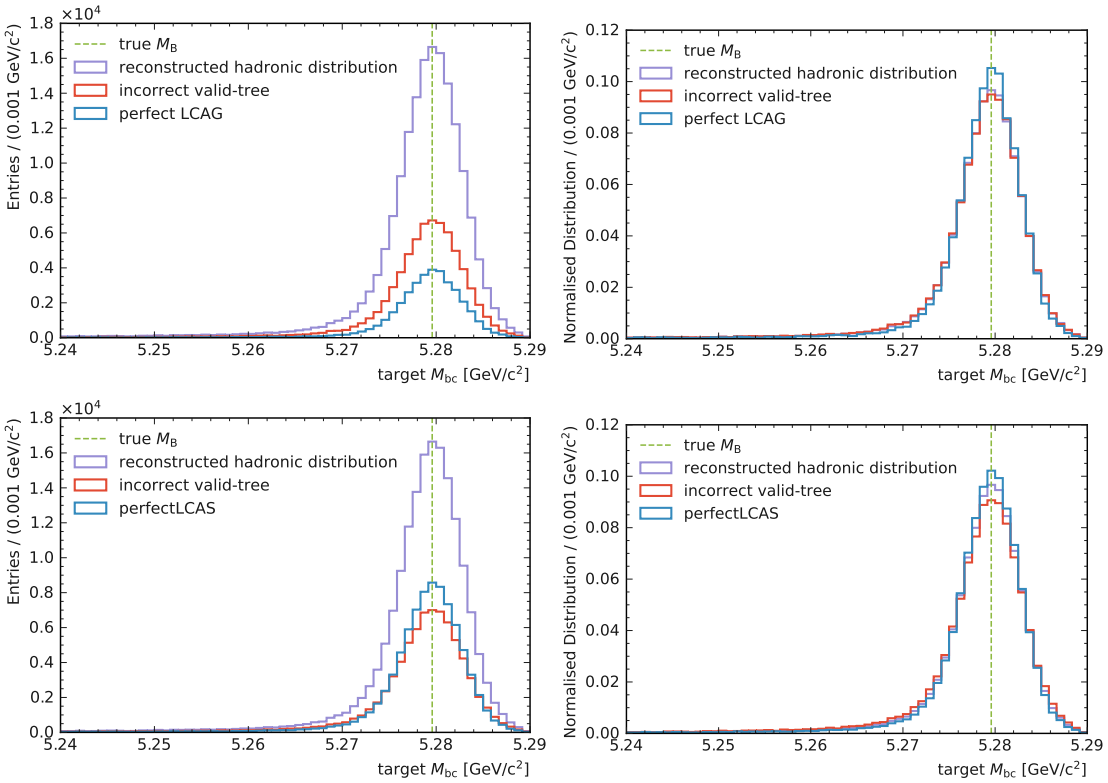


Figure 8.5.: Baseline reconstructed beam-constrained mass M_{bc} for the hadronic nominal best-case test dataset on LCAG (top) and LCAS (bottom) with the normalized distribution on the right.

model to compensate for this by including the PIDs as input features. An additional step that reconstructs FSPs before the LCA matrix prediction, is needed for developing a full reconstruction algorithm. An example of this is using stage zero of the FEI (Section 2.5), which reconstructs the FSPs.

To provide a baseline, the beam constrained mass for the true distribution, as well as correctly and incorrectly predicted decay trees of the nominal best-case scenario (Section 8.1.1), are shown in Figure 8.5 for the LCAG-model (top) and the LCAS-model (bottom).

These are the decays where all FSPs are reconstructed and nothing else is included in the event; it is the optimal reconstructed distribution. One can see the broadening of the mass peak around the true B-meson mass due to the detector resolution effects. The model is learning to predict decays that are closer to the actual B-meson mass. To determine if this is an effect of the dataset, I investigate the normalized distribution for M_{bc} for decays with an increasing number of FSPs in Figure 8.6. Figure 8.6 shows that for increasing numbers of FSPs, the M_{bc} peak around the true B meson mass broadens further. As mentioned in Section 8.1.1, this could be due to the fact that the average number of photons increases with an increasing number of FSPs. Photon resolution is limited as there is no directional information for the ECL cluster. This could explain the broader peak when calculating the M_{bc}. Since the model is learning to predict easier decay trees as shown

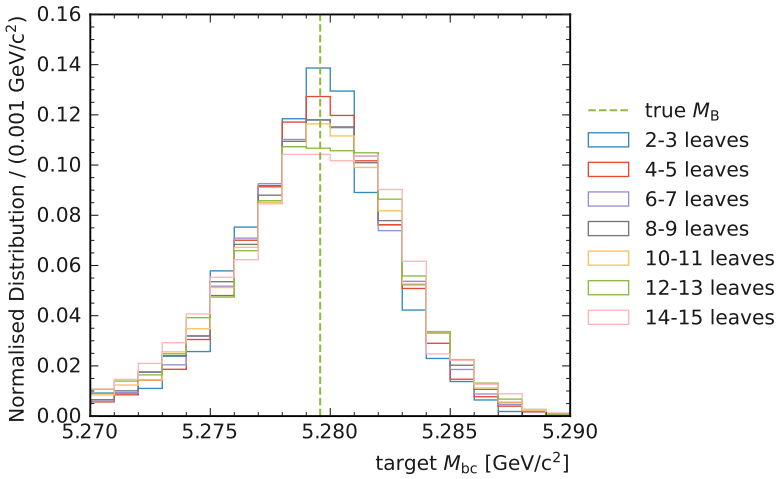


Figure 8.6.: Beam-constrained mass M_{bc} for the nominal best-case target distribution divided by the number of FSP leaves per LCA.

in Section 8.1.1, there is a higher peak expected for M_{bc} as observed in Figure 8.5. The best-case LCAS-model is able to predict particle decay trees with up to twelve FSPs with at least 16% perfect LCAS score. Figure 8.7 shows the distribution for up to and more than twelve FSPs, where the higher peak for the lower number of FSPs can be observed. Therefore, this effect is attributed to the dataset. Future studies are needed to investigate if specific decay channels or certain physics processes, e.g. excluding photons, are favored by the model learning.

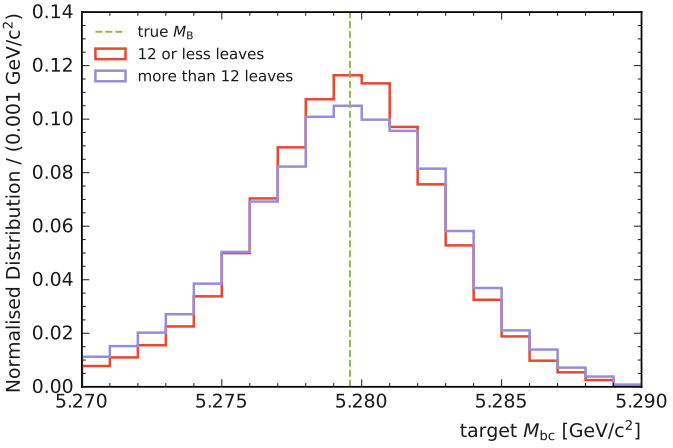


Figure 8.7.: Beam-constrained mass M_{bc} for the nominal best-case target distribution for twelve or fewer FSPs in purple and more than twelve leaves in red.

8.2.2. LCAG Generational View

The LCAG-model is applied to the full dataset and evaluated in regard to the full 12 million events in Table 8.1. Observing the results in Table 8.1, for 3% of the decays in the test

dataset a perfect LCAG is predicted. The total number of hadronic decays with a perfectly predicted LCAG matrix is 1.15%.

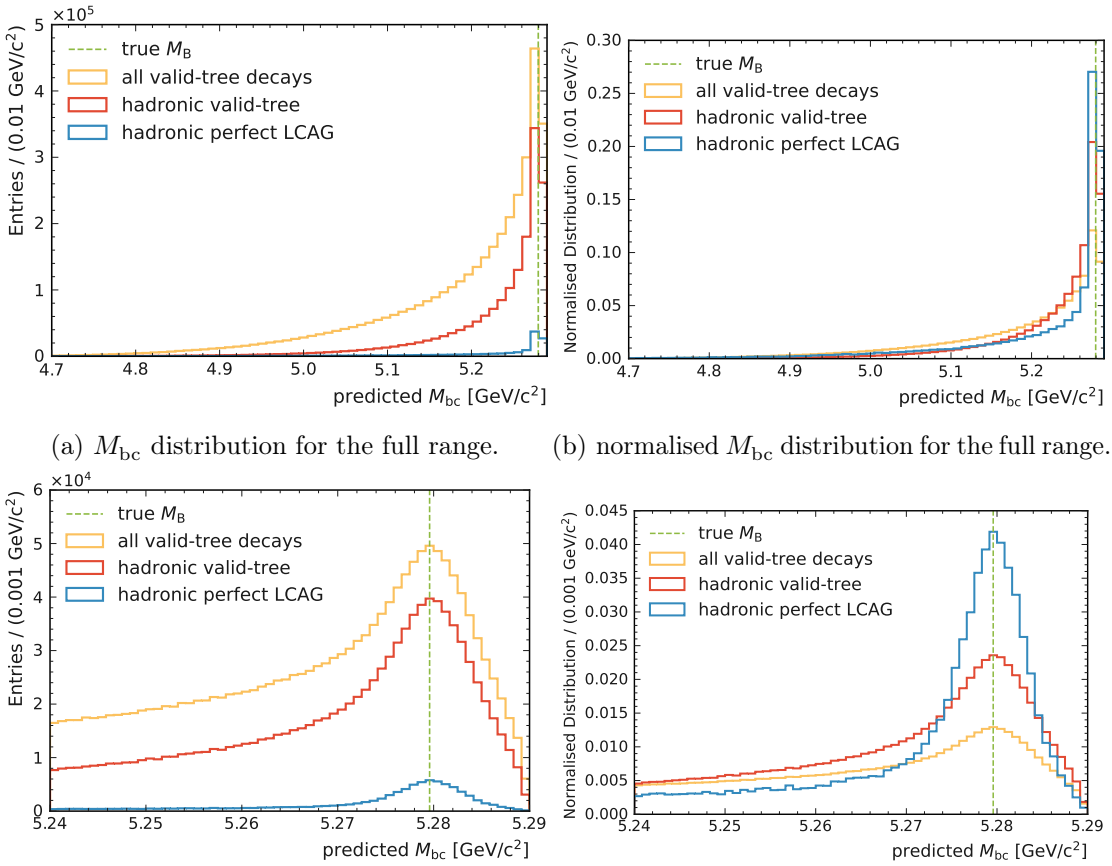
Table 8.1.: Evaluation for the realistic case, evaluated on a 12 million monogeneric events test dataset. The percentage values are calculated in regards to the 12 million events. The model is trained on generic hadronic B^0 decays.

	perfect LCAG in %	valid-tree efficiency in %	perfect purity in %
hadronic decays	**1.15**	**31.98**	**3.59**
semileptonic decays	1.85	31.98	5.79
all decays	3.00	31.98	9.38

As the model was trained on generic hadronic B-decay tree structures, semileptonic decay trees are unknown to the model. The model is able to predict the LCAG matrix perfectly in 1.85% of semileptonic decay events. This provides evidence for the hypothesis that the model is capable of learning the physical conservation laws, which has to be tested in future studies. Although hadronic and semileptonic decays are two different analyses channels, they share the same fundamental physical basis. Conservation of mass, kinematics, and charge number applies in both channels for the particle relations that the model is predicting. Semileptonic decays are easier to predict for the model as they have less (primary) FSPs on average with 8.05 (6.78) as hadronic decays, which have 10.85 (8.94), which explains the larger perfect LCAG score (Table 7.5).

The model is predicting a valid tree in 31.98% of events, resulting in a purity of 9.38%. This purity can be improved in future studies, for example by demanding that no single leptons are part of the predicted primary FSPs for hadronic decays or exploiting the knowledge of the decay tree structure. The distribution for the reconstructed beam-constrained mass, calculated with the primary predicted LCA matrix for each event, is shown in Figure 8.8 for the windows of $4.7\,\mathrm{GeV/c^2} < M_{\mathrm{bc}} < 5.3\,\mathrm{GeV/c^2}$ and $5.24\,\mathrm{GeV/c^2} < M_{\mathrm{bc}} < 5.29\,\mathrm{GeV/c^2}$. On the right, the normalized shapes are shown. The distribution peaks at the actual B^0-meson mass, validating that the method can be used to reconstruct B^0 mesons. In Figure 8.9, the reconstructed target M_{bc}, calculated with the true primary FSPs is shown. The red line corresponds to the shape of the hadronic decays that were also trained on, and the yellow line is the reconstructed target M_{bc} of the full test dataset including semileptonic decays. The grey line shows the distribution for the semileptonic decays. The M_{bc} distribution for hadronic decays has also entries for lower values, as missing particles are included. One can see the higher peak for the hadronic decays with a perfect LCAG. Around the true B^0-meson mass the hadronic decays make up the largest part of the valid-tree decays, which is expected.

Figure 8.10 shows the reconstructed target M_{bc}, calculated from the true primary FSPs, versus the predicted M_{bc}. The most dense area is at the expected true mass of the B^0-meson at $5.2797\,\mathrm{GeV/c^2}$ and only a small number of samples have high differences between the predicted and the target beam-constrained mass. This is due to the fact that 6.28%, so nearly half of the 14.04% of hadronic decays with a valid tree, have correctly-tagged FSPs. In this case, although the LCAG matrix is predicted incorrectly, it consists of only the



(a) M_{bc} distribution for the full range.

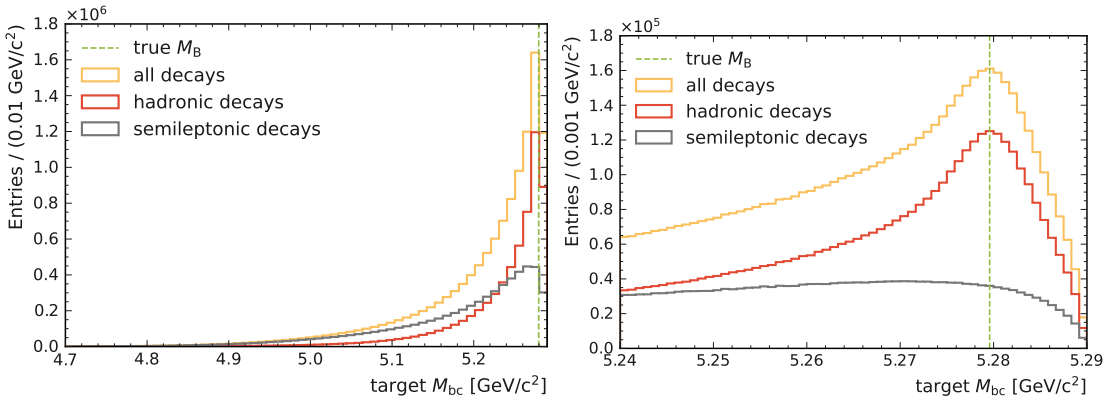
(b) normalised M_{bc} distribution for the full range.

(c) M_{bc} zoomed to window $5.24\,\mathrm{GeV/c^2} < M_{\mathrm{bc}} < 5.29\,\mathrm{GeV/c^2}$.

(d) Normalised M_{bc} zoomed to window $5.24\,\mathrm{GeV/c^2} < M_{\mathrm{bc}} < 5.29\,\mathrm{GeV/c^2}$.

Figure 8.8.: These figures show the reconstructed beam-constrained mass M_{bc} for B-decays of the test dataset for the LCAG graFEI. For each event with a valid tree the B-meson is reconstructed and the M_{bc} calculated. The yellow line shows the distribution for all decays with a valid tree, the subset of hadronic decays is shown in red and the subset of perfect hadronic predicted LCAG matrices is shown in blue. The top shows the full distribution, while the bottom is zoomed to the relevant M_{bc} region for hadronic decays close to the true mass of the B-meson (green). The normalized distributions are shown on the right.

primary FSPs and the additional background leaves are separated correctly. As I reconstruct B-mesons by adding the four-momenta of the predicted primary FSPs, the momentum $\boldsymbol{p}_{\mathrm{B}}^{\mathrm{CMS}}$ of the B-meson is reconstructed correctly in this case.



(a) M_{bc} distribution for the full range.

(b) M_{bc} zoomed to window $5.24\,\mathrm{GeV/c^2} < M_{\mathrm{bc}} < 5.29\,\mathrm{GeV/c^2}$.

Figure 8.9.: These figures show the full test dataset distribution of the beam-constrained mass M_{bc} for the reconstructed target. The distribution is not-normalized and shown for hadronic (red) and semileptonic (grey) decays zoomed to the relevant window.

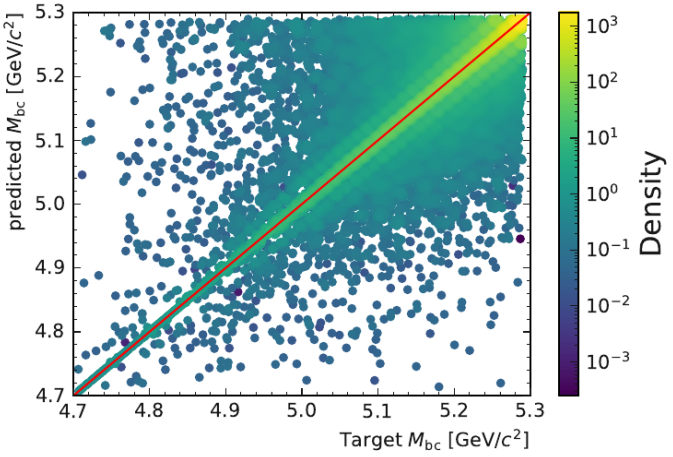


Figure 8.10.: The distribution of the predicted M_{bc} to the target M_{bc} for all hadronic decays with a valid tree predicted by the LCAG graFEI.

8.2.3. LCAS Stage View

Next is the evaluation for the LCAS-model in Table 8.2. The number of perfectly predicted LCAS matrices for hadronic (all) decays is 3.17% (7.78%) out of the full dataset, so 2.8 times higher than the LCAG-model.

The valid-tree efficiency is also higher with 53.72% compared to the previous 31.98%, which improves the purity for hadronic decays to 5.89%. The model is predicting a valid tree for nearly half the decays. Again, the model is able to correctly predict the LCAS matrix for semileptonic decays, in this case for 4.61% of the events with a purity of 8.58%. The distribution of the beam-constrained mass is shown in Figure 8.12 for the window of

Table 8.2.: Evaluation for the realistic case for the LCAS, evaluated on $B^0 \to \nu\bar{\nu}$ and trained on generic hadronic B^0 decays.

	perfect LCAS in %	valid-tree efficiency in %	perfect purity in %
hadronic decays	**3.17**	**53.72**	**5.89**
semileptonic decays	4.61	53.72	8.58
all decays	7.78	53.72	14.48

$4.7\,\mathrm{GeV/c^2} < M_{\mathrm{bc}} < 5.3\,\mathrm{GeV/c^2}$ and $5.24\,\mathrm{GeV/c^2} < M_{\mathrm{bc}} < 5.29\,\mathrm{GeV/c^2}$. Compared to the previous results for the LCAG-model, the model predicts a perfect LCAS for more events than the LCAG-model. Comparing the shape of events with a perfect LCAS to the shape with a perfect LCAG, the peak at the true B-meson mass has a broader width. The LCAG-model learns to reconstruct B-mesons closer to the actual B-meson mass. Looking at the true distribution of the M_{bc} of the test dataset in Figure 8.9

Figure 8.11 compares the reconstructed predicted and target beam-constrained, where mass the highest density is again at the true B^0-meson mass and on the diagonal. As before, this is because for hadronic decays the correctly-tagged FSPs make up 49.9%.

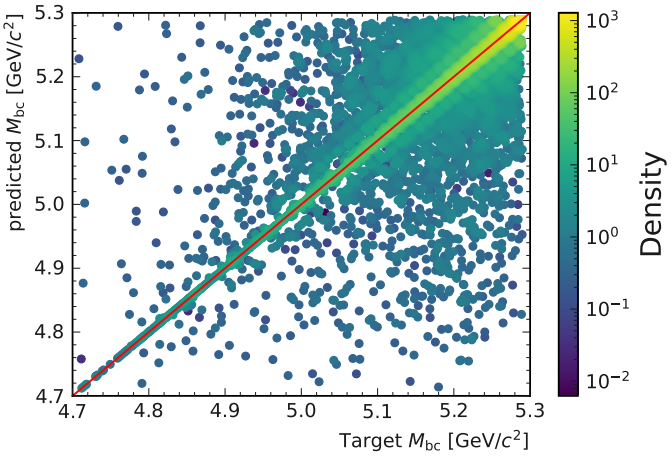
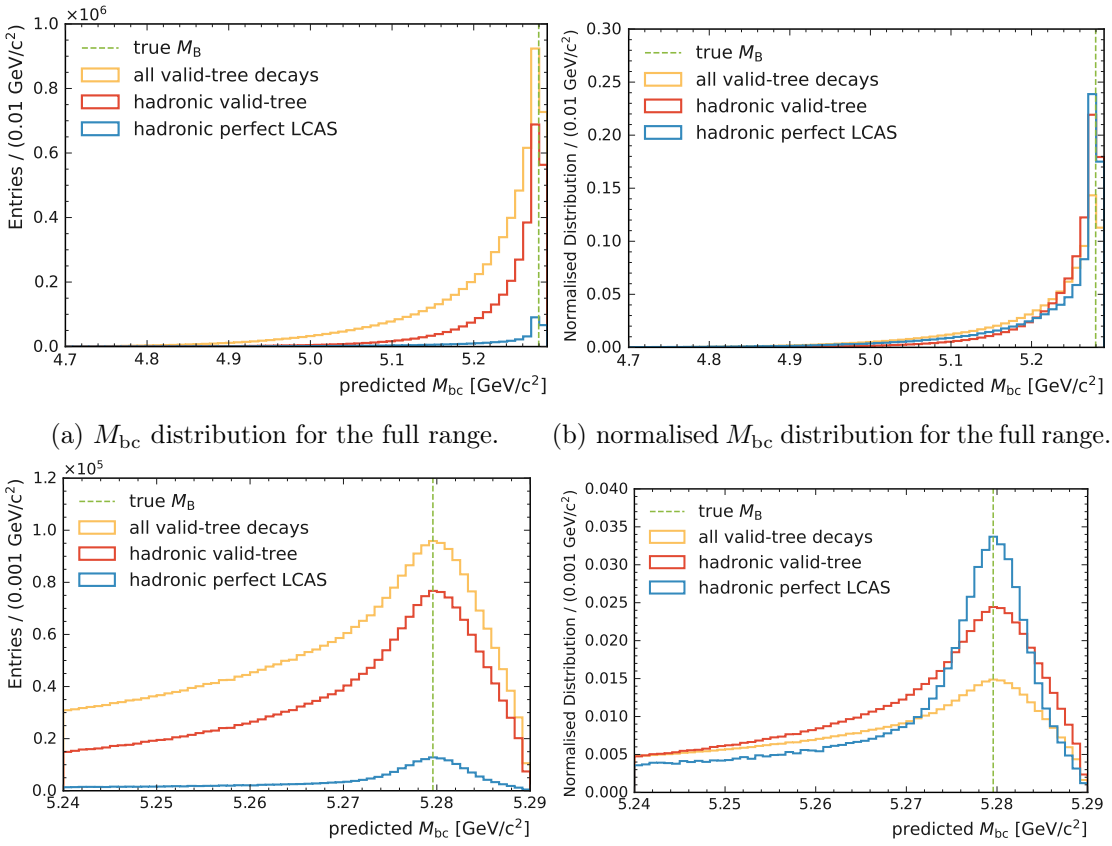


Figure 8.11.: The scattered distribution of the predicted M_{bc} to the target M_{bc} for all hadronic decays with a valid tree predicted by the LCAS graFEI.



(a) M_{bc} distribution for the full range.

(b) normalised M_{bc} distribution for the full range.

(c) M_{bc} zoomed to window $5.24\,\mathrm{GeV/c^2} < M_{\mathrm{bc}} < 5.29\,\mathrm{GeV/c^2}$.

(d) Normalised M_{bc} zoomed to window $5.24\,\mathrm{GeV/c^2} < M_{\mathrm{bc}} < 5.29\,\mathrm{GeV/c^2}$.

Figure 8.12.: These figures show the reconstructed beam-constrained mass M_{bc} for B-decays of the test dataset for the LCAS graFEI. For each event with a valid tree the B-meson is reconstructed and the M_{bc} calculated. The yellow line shows the distribution for all decays with a valid tree, the subset of hadronic decays is shown in red and the subset of perfect hadronic predicted LCAS matrices is shown in blue. The top shows the full distribution, while the bottom is zoomed to the relevant M_{bc} region for hadronic decays close to the true mass of the B-meson (green). The normalized distributions are shown on the right.

8.3. Comparison between graFEI and FEI

There is a conceptual difference between the `FEI` and the graFEI. The goal for graFEI is to predict decay tree structures in the form of the respective LCA matrix including missing particles. The hadronic `FEI` suggests B-meson candidates for the respective decay channel out of the detector information including all particles.

To put the performance into perspective, I compare the graFEI approach to the reconstruction used in the `FEI` algorithm. The hadronic `FEI` is applied to the same test dataset as the two graFEI models in the previous Section 8.2. There is no selection on signal probability, and for each event, the B^0-meson candidate with the highest signal probability is selected.

There are no further selections applied to the FEI apart from the default channel selections.

The comparison for the metrics of section 4.3 is shown again in Table 8.3 for better readability. The relevant metric for the FEI performance is the tag-side efficiency, defined by the fraction of correctly predicted B-mesons out of all events. In the following section, the tag-side efficiency is compared to the perfect LCAS/G. As previously mentioned, the FEI and graFEI are different concepts. To allow a first comparison, I constrain M_{bc} further to the window of $5.27\,\mathrm{GeV/c^2} < M_{\mathrm{bc}} < 5.29\,\mathrm{GeV/c^2}$. The assumption is that only low-energy particles are missing in this area for the graFEI LCAS/G approach, making it relevant for an analysis.

Table 8.3.: A direct comparison between the metrics used to evaluate the FEI and the new metrics used for the graFEI.

FEI	graFEI
tagging efficiency: fraction of reconstructed B-decays to all B-decays	**valid tree efficiency**: fraction of B-decays with a rooted, directed, acyclic, predicted tree to all B-decays
tag-side efficiency: fraction of **correctly** reconstructed B-decays to all decays	**perfect LCA**: fraction of B-decays with a **correctly** predicted LCA matrix
purity: fraction of correctly reconstructed decays out of all reconstructed decays	**purity**: fraction of perfect LCA out of all decays with valid trees

Results

The evaluated metrics are specified in Table 8.4. The FEI is only able to predict 0.41% of hadronic B^0 decays correctly. Using the LCAG graFEI instead, 0.54% of decays are predicted correctly in this window. The use of LCAS graFEI for decay tree predictions improves this even further with 1.32% of total decays having a perfect LCA predicted. This is 3.2 times the FEI tag-side efficiency.

Table 8.4.: Evaluated metrics for hadronic decays for the window of $5.27\,\mathrm{GeV/c^2} < M_{\mathrm{bc}} < 5.29\,\mathrm{GeV/c^2}$ for mono-generic B^0-decays on the 12 million samples test dataset.

FEI metrics	FEI in %	graFEI LCAS in %	graFEI LCAG in %	graFEI metrics
tagging efficiency	1.87	13.96	6.89	valid-tree efficiency
tag-side efficiency	**0.41**	**1.32**	**0.54**	**perfectLCA**
purity	21.92	9.46	7.84	perfectLCA purity

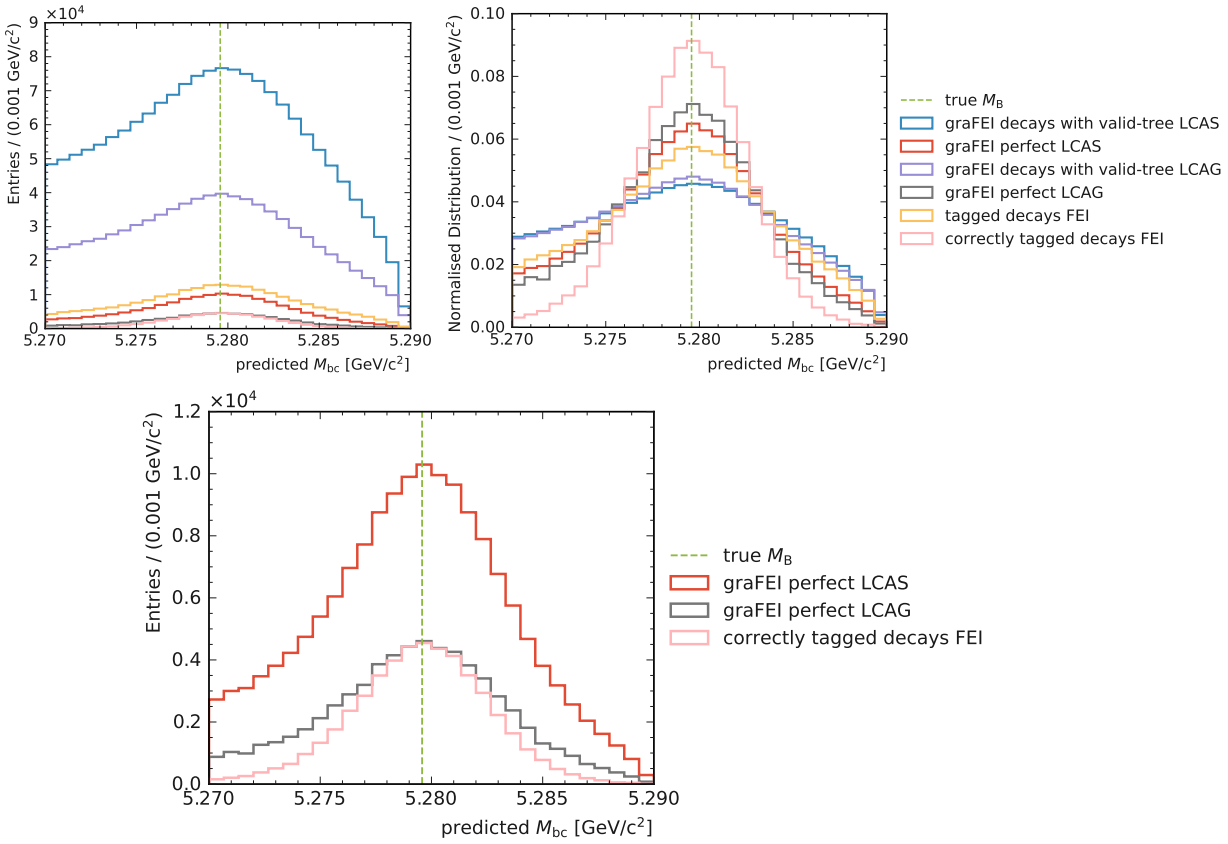


Figure 8.13.: M_{bc} distribution on the test dataset for the window of $5.27\,\mathrm{GeV/c^2} < M_{\mathrm{bc}} < 5.29\,\mathrm{GeV/c^2}$. Top right shows the distributions for correcly predicted and tagged mono-generic events for the `FEI` and the two graFEI models. Top left shows the shapes of these distributions normalised. Bottom shows only correctly reconstructed events to compare the total numbers.

M_{bc} Distributions

The beam-constrained mass M_{bc} distribution is shown in Figure 8.13 for the two different graFEI models and the FEI. Figure 8.14 shows the stacked distributions of the decays with a predicted valid tree. The graFEI does predict more decays, but as a trade-off, the purity is worse. The purity for the `FEI` is twice as high as the LCAS graFEI. The purity can be further improved using the signal probability, the probability calculated by the multivariate classifier (Chapter 3) of the `FEI` for each B-meson candidate [10]. Similar classification methods could be employed to enhance the purity of the graFEI approach in future work.

One simple example to improve the purity by introducing simple heuristics is shown in Figure 8.15 for the LCAS graFEI. Here only LCAS matrix predictions with up to nine FSPs are included. Another requirement is that no background particles have to be included in the predicted particles. This improves the purity to 20.7%, which is similar to the `FEI` result. As a trade-off, the perfect LCAS value is reduced to 0.66%. This is still predicting 61.0% more hadronic decays correctly than the FEI. This can be further optimized in regards to

the respective analysis.

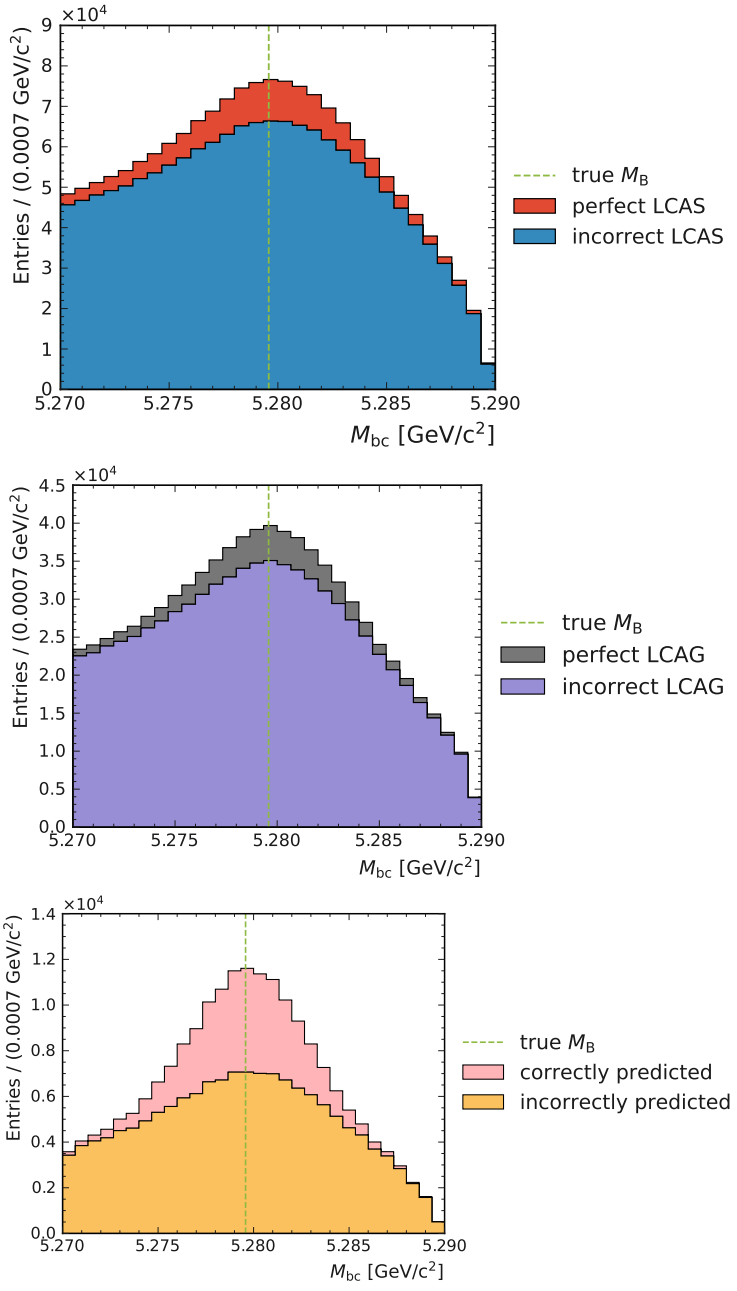


Figure 8.14.: Stacked distributions for tagged events, split into correctly and incorrectly predicted decay structures. Top shows the graFEI-LCAS, middle the graFEI-LCAG and bottom the `FEI` distribution for M_{bc}.

Discussion

This thesis serves as a means to demonstrate that the graFEI approach can be used as a training target for reconstructed decays, even for large numbers of possible decays, without

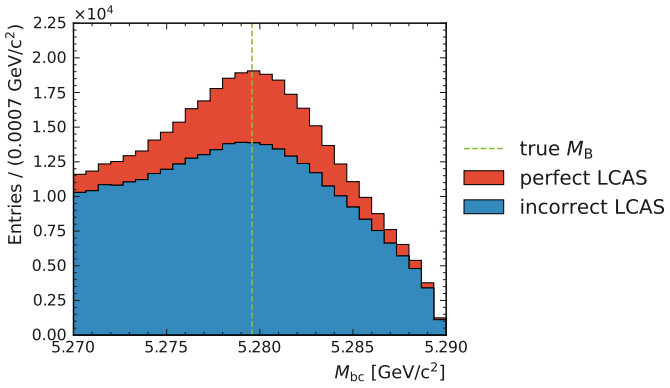


Figure 8.15.: Mono-generic decays with a predicted valid-tree stacked as perfect LCAS and incorrect LCAS. The purity is improved by demanding no background-FSPs in the event (class label of zero). Additionally only decays up to 9 leaves are included. This improves the purity to 20% with a total perfect-LCA score of 0.66%.

the need to explicitly state the decay channels. The focus is to show that the model is able to learn the decay-tree representation, and the ability to handle missing particles. Having a suitable representation enables analysts to train their own model as desired. For example, one could further specify the training targets according to the required task, thereby training on hadronic decays without missing charged particles. This is to ensure all information is available for the tag-side, so more conclusions can be drawn about the signal-side. As this is a trade-off between purity and efficiency, it depends heavily on the analysis. This has to be taken into account for the comparison in this section for the conceptual different approaches of the FEI and graFEI.

The LCAS and LCAG matrix representations are also very different approaches. The LCAS matrix encodes the tree structure according to the intermediate particle groups. Analysts could use kinematic variables similar to the partial reconstruction techniques ([17], Chapter 7.3), as the physical properties of the intermediate particles can be identified more easily according to the respective class. For example for semileptonic decays with one missing neutrino, the angle $\cos\theta_{\mathrm{BD}_\ell}$ ([17], Chapter 7.2) between the direction of the reconstructed D-system and the nominal, inferred B-meson is evaluated. This metric is also interesting for hadronic decays with the LCAG method. This representation enables hadronic decays to be predicted including missing particles, which enables analysts to extend their analyses to these decays with an appropriate discriminating variable similar to the partial B-meson reconstruction in [17] Chapter 7.3. The LCAG matrix contains more information of the decay tree structure, as each intermediate particle generation is included. The different approaches that gain different insights for the decay tree structure offer analysts more possibilities to optimize their analyses depending on the desired result with an easily applicable end-to-end trainable target. As there are deviations between simulation and experimental data, scaling factors have to be determined to use the graFEI approach for a reconstruction algorithm in future work.

8.4. Double-Generic Mixed Background Decays

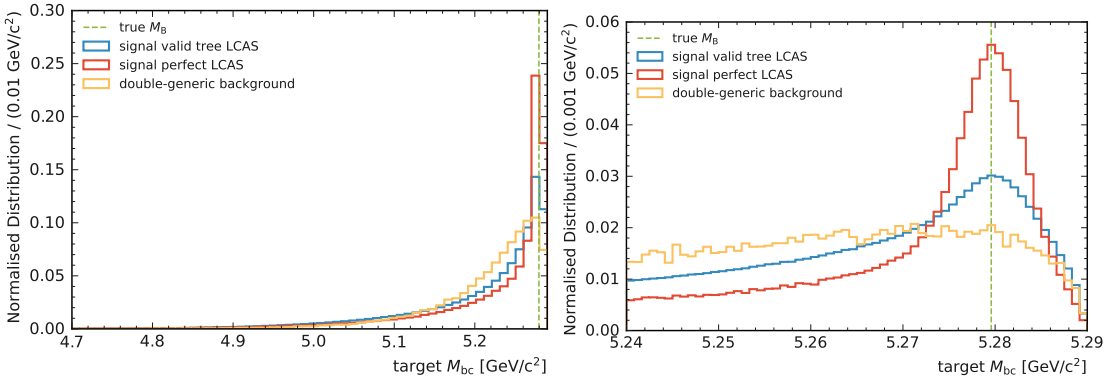
The background events for the signal-side $B^0 \to \nu\bar{\nu}$ are generic $\Upsilon(4S) \to B^0\overline{B}^0$ events, which in the following is referred to as double-generic background. It consists of events including B-decays, which means that in principle a B^0-meson can be reconstructed correctly, making it hard to distinguish from signal events. Using 1 million double-generic background events, the performance of the models is evaluated.

M_{bc} Distribution

The full distribution is shown in Figure 8.16 for the two previously defined M_{bc}-windows. As there is no signal probability implemented yet, there is no method to distinguish between signal and background except simple heuristics. It can be observed that the distribution for double-generic events using this mono-generic trained model has a broad peak. This peak is around $5.27\,\mathrm{GeV/c^2}$, close to the true B^0-meson mass. As seen in Figure 8.16 at the bottom, the distribution is flat for the smaller mass window, compared to the mono-generic signal peak.

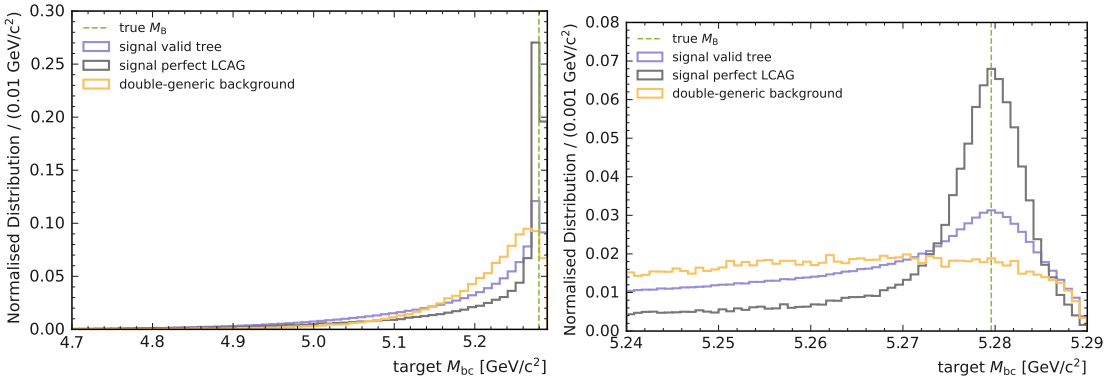
Options to Suppress Double-Generic Background

The results are shown in Table 8.5. The LCAS- and LCAG-models predict around 1% of background decays for the relevant beam-constrained mass window of $5.24\,\mathrm{GeV/c^2} < M_{\mathrm{bc}} < 5.29\,\mathrm{GeV/c^2}$. This is also due to the fact that the model is not able to predict valid trees for larger number of FSPs, and the average value for these background events is 19.79 compared to the 9.5 for the mono-generic signal events. One option to distinguish between signal and background without using the signal probability is the rest-of-event. In regards to events in the clean environment at Belle II, in principle, only the particles that originally belong to the two B-mesons of the $\Upsilon(4S)$-event should be detected. The signal B-meson is extracted previously by choosing the signal decay $B^0 \to \nu\bar{\nu}$. Each decay with a predicted valid tree results in a reconstructed B-meson of the tag-side. One can require that no additional charged particles should be left in the event after this. This is the completeness constraint in [6]. As stated before in Chapter 2, this does not happen as there are secondary and beam-induced additional particles, resulting in a loss of correctly predicted decays. Nevertheless, this is a strong separator between the signal and background if a higher purity is needed. A similar constraint for the LCA representations would be to require that the predicted LCA matrix only consists of predicted primary FSPs, and that no FSPs have predicted class labels of zero, resulting in no secondaries or unmatched particles. This can also be applied to only charged particles, as the particles belonging to the decay are clear with the LCA(S/G) matrix. Requiring this, only $6.8 \cdot 10^{-3}\%$ ($1.2 \cdot 10^{-3}\%$) of the double-generic background events have a predicted valid-tree LCAS (LCAG) matrix, thus separating strongly, while still maintaining a perfect LCAS (perfect LCAG) score of 0.77% (0.31%). For comparison, doing the same on the same background events for the `FEI`, the tag-side efficiency is 0.41% with a tagging-efficiency on double-generic mixed background of 0.12%.



(a) Normalised M_{bc} distribution LCAS.

(b) normalised M_{bc} distribution window $5.24\,\mathrm{GeV/c^2} < M_{\mathrm{bc}} < 5.29\,\mathrm{GeV/c^2}$.



(c) Normalised M_{bc} distribution LCAG.

(d) Normalised M_{bc} distribution LCAG. $5.24\,\mathrm{GeV/c^2} < M_{\mathrm{bc}} < 5.29\,\mathrm{GeV/c^2}$.

Figure 8.16.: Beam-constrained mass M_{bc} normalised distribution on the evaluated graFEI on mono-generic decays as signal and generic $\Upsilon(4S)$-events with two-decays as background with the LCAS-model on the (a) and (b) and LCAG-model (c) and (d) without any background suppression selections applied.

Table 8.5.: Values for tagged events on 1 million simulated double-generic mixed samples. Equivalent for the `FEI` is 0.41% tag-side efficiency on signal and tagging-efficiency on double-generic mixed background of 0.12%.

	background double-generic decays		single decays			
metrics in %	LCAS	LCAG	LCAS		LCAG	
	valid-tree	valid-tree	perfect	valid-tree	perfect	valid-tree
all	6.30	5.50	3.17	53.72	1.15	31.98
Mbc constrained	1.15	0.89	1.32	13.96	0.54	6.89
Mbc constrained and no class-label 0 leaves, motivated by `FEI`- no-Rest-of-Event	0.0068	0.0012	0.77	6.24	0.31	2.31

9. Conclusion and Outlook

The Belle II experiment is a B-factory operating mainly at the $\Upsilon(4S)$ resonance. The expected large dataset will enable precise measurements of rare decays to probe the Standard Model and searches for new physics. B-tagging is essential for many key measurements, as information of the tag-side B-meson of the $\Upsilon(4S) \to B\overline{B}$ event enables one to constrain rare signal decays which contain neutrinos in the final state. This is challenging due to the large number of possible decay channels and large combinatorics, and further limited by the detector efficiency and acceptance. The current reconstruction algorithm, the `Full Event Interpretation` (`FEI`), suffers from a low tag-side efficiency of 0.46% for hadronic decays and 2.04% for semileptonic decays (Section 2.5). Since rare decays are investigated to search for new physics and probe the Standard Model, Belle II analyses are often dominated by statistical uncertainties when evaluated with the `FEI`. A new end-to-end trainable graFEI approach was developed in previous work [11], where the entire decay tree structure is encoded into a single matrix (Section 4.1) of the final state particles (FSPs). The goal of this thesis was to investigate whether this approach can be scaled up to Belle II simulated data, dealing with the experimental realities and large amounts of possible decay channels. This enables the utilization of the full Belle II simulated sample coverage and offers the ability to handle missing particles. The decay tree structure in this work is encoded in the generational view LCAG matrix and the `FEI`-motivated stage-wise LCAS matrix, which is used as training target for the graph-based approach (graFEI).

To achieve this goal, the graFEI model was modified to allow for the different number of FSPs (Section 4.2). Tests were performed initially on a simple, ideally simulated phasespace dataset with 200 different decay topologies (Chapter 5). Based on this dataset, a hyperparameter search was performed to investigate the model setup (Section 5.3) and formulate a training strategy for the studies on the simulated Belle II dataset. The graFEI approach achieved promising results of 61.1% perfect LCAG on the phasespace dataset in Chapter 5, encouraging the application to the Belle II dataset. The next step was to adapt this approach to the Belle II simulated samples to include the experimental realities and select the FSPs out of the detected particles (Chapter 6). This adaptation was first tested on a single decay. Studies on this decay confirmed the successful addition of the background class. Furthermore, the influence of the detector effects in Section 7.1 were determined. Subsequently, the approach was evaluated for six different Belle II decay channels (Section 7.2) to investigate the limitations of the approach in more detail. Based on these results, the FSPs selections were reworked (Section 7.3), improving the model performance from 19.2% to 30% perfect LCAG on the mix of six different decays.

Finally, trainings were performed on the entire Belle II simulated dataset in Chapter 8 for the LCAS and LCAG matrix representations. The models were evaluated on the tag-side of the signal-side $B^0 \to \nu\bar{\nu}$, including the unknown semileptonic decay tree structures, achieving 3% perfect LCAG and 7.78% perfect LCAS (Section 8.2). To put the performance into perspective, the comparison with the `FEI` was performed on hadronic decays for the window of $5.27\,\mathrm{GeV/c^2} < M_{\mathrm{bc}} < 5.29\,\mathrm{GeV/c^2}$ (Section 8.3). Results of this first training show that the perfect LCAS score achieves 1.32% and the perfect LCAG achieves 0.54%, in contrast to the `FEI` tag-side efficiency of 0.41%.

This work demonstrates that the representation can be applied to Belle II simulated samples, and used as a training target when developing new deep learning reconstruction algorithms. The method enables one to handle missing particles, which can be used to extend the efficiency of the algorithm even further. It also showed the clear limitations of this approach: the model is only learning easier LCA matrices when combining decay channels of different complexities together. It is especially relevant to further study the performance on the decay channels that this approach is able to predict, e.g. targeting the reconstruction rate for D and D^* mesons. The results of this work were presented at the 20th International Workshop on Advanced Computing and Analysis Techniques in Physics Research [61] and a paper has been submitted to the 39th International Conference on Machine Learning. Proposed future research directions based on the studies of this thesis include:

Improving final state particle selection Since a large part of this work has focused on scaling the approach up to the complexity of the full dataset, the FSP selection optimization was out of scope. Optimal FSP reconstruction requires careful calibration with real data. The results in Section 7.3 showed that changes to the FSP selection impact the model performance greatly. One approach for this could be to employ stage-zero of the `FEI`, which is responsible for FSPs reconstruction that precedes the `FEI` reconstruction stages.

Implementing a signal probability To distinguish between background and signal, future works would greatly benefit from implementing a signal probability. There is a multitude of possible ways to implement this. One approach is training a deep ensemble [62]. There were previous successful implementations of deep ensembles by [63] for continuum suppression with predictive uncertainties at the Belle II experiment. Another approach could be combining the output for each edge to construct a signal score, although further calibration would be needed for the use as a probability.

Input Feature Optimization To ensure a fair comparison to the current reconstruction algorithm, `FEI`, similar input features were used. The `FEI` is BDT-based, therefore useful features are likely to differ when using a Neural Network. Further investigation and ablation studies to determine the importance of these features have the potential to improve the performance of the model and give insight into pattern learning.

Particle decay reconstruction from detected FSPs is highly relevant in searches for new physics in Belle II, but suffers from the low efficiency of the current reconstruction algorithm `FEI`. This work shows that a graph-based approach increases the efficiency from 0.41% to 1.32%, paving the way for the development of an end-to-end trainable, graph-based reconstruction algorithm to improve future analyses.

Bibliography

[1] B. Povh, K. Rith, C. Scholz, F. Zetsche, and W. Rodejohann, *Particles and Nuclei, An Introduction to the Physical Concepts*. Springer, 7 ed., 2015.

[2] L. Canetti, M. Drewes, and M. Shaposhnikov, "Matter and Antimatter in the Universe," *New J. Phys.* **14** (2012) 095012, arXiv:1204.4186 [hep-ph].

[3] J. Y. Kim, "Discovery of Neutrino Oscillations in the Super-Kamiokande Experiment," *Phys. High Technol.* **24** no. 11, (2015) 8–17.

[4] P. A. R. Ade, N. Aghanim, M. Arnaud, M. Ashdown, J. Aumont, C. Baccigalupi, A. J. Banday, R. B. Barreiro, J. G. Bartlett, N. Bartolo, E. Battaner, R. Battye, and et. al., "Planck2015 results," *Astronomy I& Astrophysics* **594** (Sep, 2016) A13. http://dx.doi.org/10.1051/0004-6361/201525830.

[5] B. Wang, "Searches for New Physics at the Belle II Experiment," in *Meeting of the APS Division of Particles and Fields*. 11, 2015. arXiv:1511.00373 [hep-ex].

[6] T. Keck *et al.*, "The Full Event Interpretation: An Exclusive Tagging Algorithm for the Belle II Experiment," *Comput. Softw. Big Sci.* **3** no. 1, (2019) 6, arXiv:1807.08680 [hep-ex].

[7] F. U. Bernlochner, M. F. Sevilla, D. J. Robinson, and G. Wormser, "Semitauonic b-hadron decays: A lepton flavor universality laboratory," *Rev. Mod. Phys.* **94** no. 1, (2022) 015003, arXiv:2101.08326 [hep-ex].

[8] **Belle-II**, R. Cheaib, "Towards first results on $|V_{ub}|$ and $|V_{cb}|$ with the Belle II experiment," *PoS* **ICHEP2020** (2021) 382.

[9] **Belle**, M. Gelb *et al.*, "Search for the rare decay of $B^+ \to \ell^+ \nu_\ell \gamma$ with improved hadronic tagging," *Phys. Rev. D* **98** no. 11, (2018) 112016, arXiv:1810.12976 [hep-ex].

[10] T. Keck, *Machine learning algorithms for the Belle II experiment and their validation on Belle data*. PhD thesis, KIT, Karlsruhe, 2017.

[11] I. Tsaklidis, P. Goldenzweig, I. Ripp-Baudot, J. Kahn, and G. Dujany, *Demonstrating learned particle decay reconstruction using Graph Neural Networks at BelleII*. PhD thesis, Strasbourg, Universitè de Strasbourg, Karlsruhe, Strasbourg, 2020.

[12] Wikipedia, the free encyclopedia, "Standard model of elementary particles," 2019. https://en.wikipedia.org/wiki/File:

Standard_Model_of_Elementary_Particles.svg. [Online; accessed February 7, 2022].

[13] **Particle Data Group**, P. Zyla *et al.*, "Review of Particle Physics," *PTEP* **2020** no. 8, (2020) 083C01.

[14] M. Kobayashi and T. Maskawa, "CP-Violation in the Renormalizable Theory of Weak Interaction," *Progress of Theoretical Physics* **49** no. 2, (02, 1973) 652–657, https://academic.oup.com/ptp/article-pdf/49/2/652/5257692/49-2-652.pdf. https://doi.org/10.1143/PTP.49.652.

[15] **ATLAS**, G. Aad *et al.*, "Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC," *Phys. Lett. B* **716** (2012) 1–29, arXiv:1207.7214 [hep-ex].

[16] **Super-Kamiokande**, S. Fukuda *et al.*, "Constraints on neutrino oscillations using 1258 days of Super-Kamiokande solar neutrino data," *Phys. Rev. Lett.* **86** (2001) 5656–5660, arXiv:hep-ex/0103033.

[17] **BaBar, Belle**, A. J. Bevan *et al.*, "The Physics of the B Factories," *Eur. Phys. J. C* **74** (2014) 3026, arXiv:1406.6311 [hep-ex].

[18] **Belle**, K. Abe *et al.*, "Observation of large CP violation in the neutral B meson system," *Phys. Rev. Lett.* **87** (2001) 091802, arXiv:hep-ex/0107061.

[19] **BaBar**, B. Aubert *et al.*, "Observation of CP violation in the B^0 meson system," *Phys. Rev. Lett.* **87** (2001) 091801, arXiv:hep-ex/0107013.

[20] **Belle**, J. Brodzicka *et al.*, "Physics Achievements from the Belle Experiment," *PTEP* **2012** (2012) 04D001, arXiv:1212.5342 [hep-ex].

[21] **Belle-II**, W. Altmannshofer *et al.*, "The Belle II Physics Book," *PTEP* **2019** no. 12, (2019) 123C01, arXiv:1808.10567 [hep-ex]. [Erratum: PTEP 2020, 029201 (2020)].

[22] **Belle-II**, T. Abe *et al.*, "Belle II Technical Design Report," arXiv:1011.0352 [physics.ins-det].

[23] D. Matvienko, "The belle ii experiment: status and physics program," *EPJ Web of Conferences* **191** (01, 2018) 02010.

[24] D. Weyland, "Continuum Suppression with Deep Learning techniques for the Belle II Experiment," Master's thesis, Karlsruhe Institute of Technology (KIT), 2017.

[25] A. Ryd, D. Lange, N. Kuznetsova, S. Versille, M. Rotondo, D. P. Kirkby, F. K. Wuerthwein, and A. Ishikawa, "EvtGen: A Monte Carlo Generator for B-Physics,".

[26] T. Sjöstrand, S. Ask, J. R. Christiansen, R. Corke, N. Desai, P. Ilten, S. Mrenna, S. Prestel, C. O. Rasmussen, and P. Z. Skands, "An introduction to PYTHIA 8.2," *Comput. Phys. Commun.* **191** (2015) 159–177, arXiv:1410.3012 [hep-ph].

[27] Z. J. Liptak *et al.*, "Measurements of Beam Backgrounds in SuperKEKB Phase 2," arXiv:2112.14537 [physics.ins-det].

[28] T. Kuhr, C. Pulvermacher, M. Ritter, T. Hauth, and N. Braun, "The belle ii core software," *Computing and Software for Big Science* **3** no. 1, (Nov, 2018) . http://dx.doi.org/10.1007/s41781-018-0017-9.

[29] T. Keck, "The full event interpretation for belle ii," Master's thesis, Karlsruhe Institute of Technology (KIT), 2014.

[30] K. Albertsson *et al.*, "Machine Learning in High Energy Physics Community White Paper," *J. Phys. Conf. Ser.* **1085** no. 2, (2018) 022008, arXiv:1807.02876 [physics.comp-ph].

[31] A. J. Larkoski, I. Moult, and B. Nachman, "Jet Substructure at the Large Hadron Collider: A Review of Recent Advances in Theory and Machine Learning," *Phys. Rept.* **841** (2020) 1–63, arXiv:1709.04464 [hep-ph].

[32] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," 2014.

[33] X. Ju, S. Farrell, P. Calafiura, D. Murnane, Prabhat, L. Gray, T. Klijnsma, K. Pedro, G. Cerati, J. Kowalkowski, G. Perdue, P. Spentzouris, N. Tran, J.-R. Vlimant, A. Zlokapa, J. Pata, M. Spiropulu, S. An, A. Aurisano, J. Hewes, A. Tsaris, K. Terao, and T. Usher, "Graph neural networks for particle reconstruction in high energy physics detectors," 2020.

[34] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain.," *Psychological review* **65 6** (1958) 386–408.

[35] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," **323** no. 6088, (Oct., 1986) 533–536.

[36] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations* (12, 2014) .

[37] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR* **abs/1502.03167** (2015) , 1502.03167. http://arxiv.org/abs/1502.03167.

[38] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," 2018.

[39] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research* **15** (06, 2014) 1929–1958.

[40] A. ben khalifa and H. Frigui, "Multiple instance fuzzy inference neural networks,".

[41] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Journal of Machine Learning Research - Proceedings Track* **9** (01, 2010) 249–256.

[42] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, G. Gordon, D. Dunson, and M. Dudík, eds., vol. 15 of *Proceedings of*

Machine Learning Research, pp. 315–323. PMLR, Fort Lauderdale, FL, USA, 11–13 apr, 2011. https://proceedings.mlr.press/v15/glorot11a.html.

[43] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," 2016.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," arXiv:1512.03385 [cs.CV].

[45] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," 2018.

[46] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond euclidean data," *IEEE Signal Processing Magazine* **34** no. 4, (Jul, 2017) 18–42. http://dx.doi.org/10.1109/MSP.2017.2693418.

[47] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[48] H. Qu and L. Gouskos, "ParticleNet: Jet Tagging via Particle Clouds," *Phys. Rev. D* **101** no. 5, (2020) 056019, arXiv:1902.08570 [hep-ph].

[49] T. Kipf, E. Fetaya, K.-C. Wang, M. Welling, and R. Zemel, "Neural Relational Inference for Interacting Systems," in *Proceedings of the 35th International Conference on Machine Learning (ICML '18)*, vol. 80 of *Proceedings of Machine Learning Research*, pp. 2688–2697. PMLR, 2018. http://proceedings.mlr.press/v80/kipf18a.html.

[50] M. Bender, M. Farach-Colton, G. Pemmasani, S. Skiena, and P. Sumazin, "Lowest common ancestors in trees and directed acyclic graphs," *Journal of Algorithms* **57** no. 2, (Nov., 2005) 75–94.

[51] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," 2018.

[52] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds., vol. 32. Curran Associates, Inc., 2019. https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

[53] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2019.

[54] A. P. Navarro and J. Eschle, "phasespace: n-body phase space generation in Python," *Journal of Open Source Software* **4** no. 42, (2019) 1570.

[55] "High performance computing system "hochleistungsrechner karlsruhe" (horeka) at kit." https://www.nhr.kit.edu/userdocs/. Online; accessed Februrary 14, 2022.

[56] R. F. von Cube, M. Giffels, C. Heidecker, G. Quast, M. B. Sauter, and M. J. Schnepf, "Federation of compute resources available to the german CMS community," *Journal of Physics: Conference Series* **1525** no. 1, (Apr, 2020) 012055. https://doi.org/10.1088/1742-6596/1525/1/012055.

[57] Caspart, René, Fischer, Max, Giffels, Manuel, von Cube, Ralf Florian, Heidecker, Christoph, Kuehn, Eileen, Quast, Günter, Heiss, Andreas, and Petzold, Andreas, "Setup and commissioning of a high-throughput analysis cluster," *EPJ Web Conf.* **245** (2020) 07007. https://doi.org/10.1051/epjconf/202024507007.

[58] R. Brun, F. Rademakers, P. Canal, A. Naumann, O. Couet, L. Moneta, V. Vassilev, S. Linev, D. Piparo, G. GANIS, B. Bellenot, E. Guiraud, G. Amadio, wverkerke, P. Mato, TimurP, M. Tadel, wlav, E. Tejedor, J. Blomer, A. Gheata, S. Hageboeck, S. Roiser, marsupial, S. Wunsch, O. Shadura, A. Bose, C. Cristescu, X. Valls, and R. Isemann, "root-project/root: v6.18/02," Aug., 2019. https://doi.org/10.5281/zenodo.3895860.

[59] J. Pivarski, P. Das, C. Burr, D. Smirnov, M. Feickert, T. Gal, L. Kreczko, N. Smith, N. Biederbeck, O. Shadura, M. Proffitt, benkrikler, H. Dembinski, H. Schreiner, J. Rembser, M. R., C. Gu, J. Rübenach, M. Peresano, and R. Turra, "scikit-hep/uproot: 3.12.0," July, 2020. https://doi.org/10.5281/zenodo.3952728.

[60] The HDF Group, "Hierarchical Data Format, version 5," 1997-NNNN. https://www.hdfgroup.org/HDF5/.

[61] L. Reuter, J. Kahn, I. Tsaklidis, O. Taubert, M. Götz, G. Dujany, T. Boeck, A. Thaller, T. Ferber, and P. Goldenzweig, "Particle decay tree reconstruction with graph neural networks," Poster presented at 20th International Workshop on Advanced Computing and Analysis Techniques in Physics Research, 2021. https://indico.cern.ch/event/855454/contributions/4598451/.

[62] S. Fort, H. Hu, and B. Lakshminarayanan, "Deep ensembles: A loss landscape perspective," 2020.

[63] L. Sowa, "Deep Continuum Suppression with Predictive Uncertainties at the Belle II Experiment," Master's thesis, Karlsruhe Institute of Technology (KIT), 2021.

A. Training Hyperparameters

Table A.1 shows the hyperparameters for the training runs in this thesis. The hyperparameter search performed with `Optuna` on best-case Belle II simulated dataset is shown in Figure A.1, that confirm the results of Section 5.3. The following parameters are tuned:

- additional MLP layers (aMLP) [1,2,3,4],
- feedforward layer widths (dimFF) [128, 256, 512],
- final MLP layers (fMLP) [1,2,3],
- initial MLP layers (iMLP) [1,2,3],
- loss function (loss) [cross entropy loss, focal loss [38]],
- number of blocks (nblocks) [2,3,4,5,6].

The training runs including the cross entropy loss are more robust and achieve better scores, therefore the cross entropy loss is used in this thesis. Figure A.1 shows the contour plots for each parameter combination in the order above, where the objective value referrers to the accuracy.

Table A.1.: Training hyperparameters for the NRI (Section 4.2), optimized by `Optuna`. The tuned hyperparameters are the feedforward layer width (dimFF), the number of blocks (nblocks), the number of additional MLPs (aMLP), the number of initial MLPs (iMLP), the number of embedding dimensions (emb), the dropout (do), the batch size (bsize) and the learning rate (lr).

Datset	hyperparameters								
	dimFF	nblocks	aMLP	iMLP	fMLP	emb	do	bsize	lr
single decay	512	2	0	1	1	-	0.3	64	0.001
Mix of six decays	1024	2	0	1	1	-	0.5	64	0.0001
best-case scanario	512	1	0	1	1	3	0.3	128	0.001
realistic scenario	512	1	0	1	1	3	0.3	128	0.001

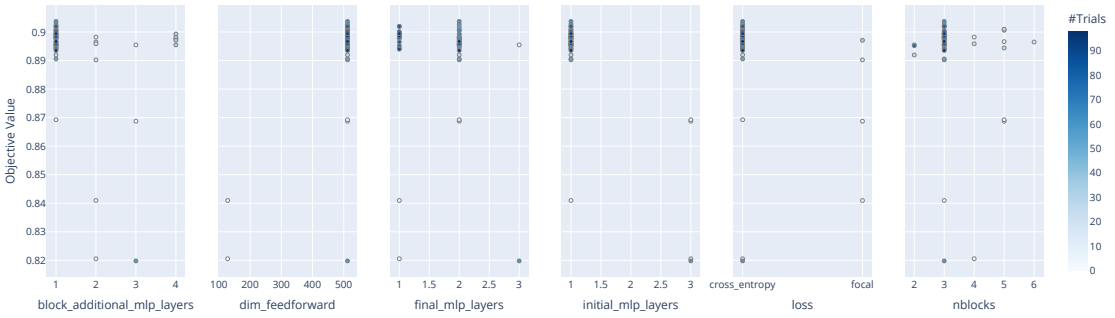


Figure A.1.: Hyperparameter Optimization of `Optuna` for the NRI. The objective value corresponds to the accuracy.

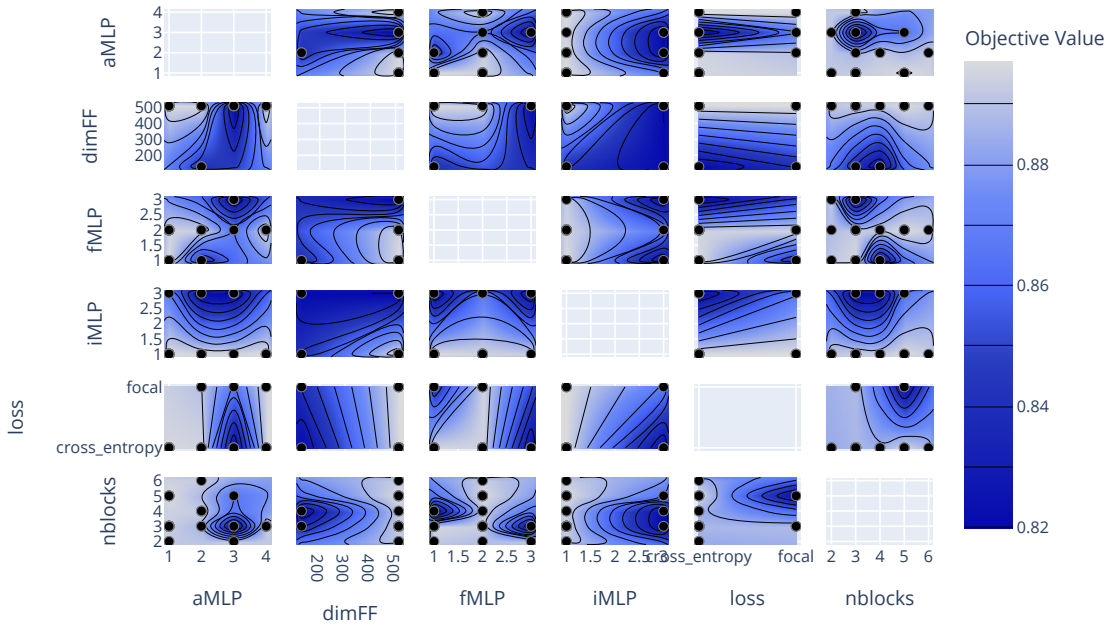


Figure A.2.: Contour plots for the hyperparameter search using `Optuna` for the NRI. The accuracy is given by the shade of blue, according to the color bar.

B. Extra Results on Mix of Selected Decays

B.1. Semileptonic Decays

A Training on a mix of selected decays is also performed on events including neutrinos for charged B^+ mesons. The selected decays are shown in Table B.1 and the results of the best-case training on the test dataset are shown in Table B.2. The trained model achieves a perfect LCAG score of 87.3%. The same behaviour as for the hadronic mix of selected decays is observed: with an increasing number of FSPs per decay, the perfect LCAG score decreases.

Table B.1.: Decay channels of the FEI channels for the charged B^+ meson decays including neutrinos. All π^0 decay into 2 photons as shown in the first decay.

Decay channel	FSPs	Depth	Motivation
$B^+ \to \overline{D}^0(\to K^+\pi^-\pi^0(\to \gamma\gamma))\ell^+\nu_\ell$	5	3	simple decay including 2 photons that
$B^+ \to \overline{D}^0(\to K^+\pi^-\pi^0\pi^0)\ell^+\nu_\ell$	7	3	decay including two π^0, hard to distinguish γ
$B^+ \to D^*(\to \overline{D}^0 \to K^+\pi^-\pi^0)\pi^0)\ell^+\nu_\ell$	7	4	deep decay tree, with high number of FSPs
$B^+ \to \overline{D}^0(\to K^+\ell^-\bar{\nu}_\ell)\ell^+\nu_\ell$	3	2	decay including two neutrinos (different to Belle II semileptonic definition)
$B^+ \to \overline{D}^0(\to K^+\ell^-\bar{\nu}_\ell)\pi^+\pi^+\pi^-$	5	2	shallow decay tree with higher number of FSPs

Table B.2.: Results of the Training on the combined dataset consisting out of the decays defined in table B.1, as well as the evaluation on each individual decay channel used.

Decay	Size (%)	Perfect LCAG (%)	Accuracy (%)	Efficiency (%)
Full Dataset	100	87.3	95.4	92.3
$B^+ \to \overline{D}^0(\to K^+\pi^-\pi^0(\to \gamma\gamma))\ell^+\nu_\ell$	28.2	96.7	98.2	98.8
$B^+ \to \overline{D}^0(\to K^+\pi^-\pi^0\pi^0)\ell^+\nu_\ell$	2.9	81.1	97.1	98.1
$B^+ \to D^*(\to \overline{D}^0 \to K^+\pi^-\pi^0)\pi^0)\ell^+\nu_\ell$	21.6	61.5	91.5	67.6
$B^+ \to \overline{D}^0(\to K^+\ell^-\bar{\nu}_\ell)\ell^+\nu_\ell$	8.8	93.7	98.4	99.3
$B^+ \to \overline{D}^0(\to K^+\ell^-\bar{\nu}_\ell)\pi^+\pi^+\pi^-$	38.5	93.8	98.4	99.3

B.2. Comparison with Transformer Model

For the baseline GNN comparison of the NRI to the Transformer model [47], the following hyperparameters are used for the Transformer:

- No. attentions: 1
- No. attention heads: 16
- embedding dimension: 256
- feedforward layer width: 512
- final MLP layers: 1
- loss: cross entropy

The Transformer model for the realistic case achieves a perfect LCAG of 14.0%, compared to the NRI model with a perfect LCAG of 30.0% excluding class weights on the updated selections for the mix of hadronic decays Section 7.3. This shows, that the performance of the NRI model is significantly better than the Transformer model.