Development of a Graph Neural Network based Tagging Approach for Semi-Inclusive Tagging and Rest of Event Clean Up for the Belle II Experiment

Axel Heimeroth

Masterarbeit in Physik
angefertigt im Physikalischen Institut

vorgelegt der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität
Bonn

März 2021

I hereby declare that this thesis was formulated by myself and that no sources or tools other than those cited were used.

Bonn, ...
Date Signature

1. Gutachter: Prof. Dr. Florian Bernlochner
2. Gutachterin: Prof. Dr. Jochen Dingfelder

Acknowledgements

I would like to thank those who made this thesis possible. Prof. Bernlochner and Dr. William Sutcliffe for providing the topic and the supervision during the project. Prof. Dingfelder for taking the task of being the second corrector. Ilias Tsaklidis for the outstanding amount of help and creativity he provided me during the thesis in order to fix issues and try new things. Usually those were things I did not even had recognized as a problem or oppurtinity yet. Luka Vomberg, Ayako Braden and Ilias Tsaklidis for doing the proof-reading and providing me a lot of help on how to write a better thesis.

This thesis was written during rather unusual times and was made possible by the outstanding support of many, whom I want to thank for this not science related support.

Contents

CHAPTER 1

Introduction

Physics aims to describe nature with mathematical models. The most successful models at the moment are general relativity and the Standard Model (SM) of particle physics. General relativity is the description of gravity, interpreting it as an interaction of the curvature of space-time and matter. The SM describes the remaining three fundamental forces, namely the electromagnetic, weak and strong force. While the SM is very successful in general [1], some phenomena, like the neutrino masses or the CP violation [2], are beyond its current scope and need further investigation.
The different particle collider experiments around the world are designed to further investigate the SM. The Belle II experiment is one of them, located in Tsukuba, Japan. Belle II's main goals are to further investigate flavour physics and the CP violation. For the Belle II experiment electrons and their anti-particles, positrons, are collided at energies needed to produce $\Upsilon(4S)$ mesons. These mesons mostly decay into B meson pairs. For precise measurements of the decay products of these B mesons very capable hardware and analysis software are needed. At Belle II the efficient reconstruction, also called tagging, of tag B mesons is necessary, to fully reconstruct the desired signal B mesons. This is especially the case for hard to reconstruct signal B mesons, for example if neutrinos appear in the decay chain.
This thesis focuses on the development of software tools which shall improve the particle reconstruction at Belle II. The tools are based on a graph neural network, called the DSIT model, and aim to categorize the final state particles of a decay event in order to improve the tagging. This is done as a further development of an already existing semi-inclusive tagging method, the further development being named deep semi-inclusive tagging in this thesis. The main goal is to improve purity and efficiency of the tagging.

Later the DSIT model's capability to clean up the rest of event after the signal B meson was already reconstructed gets investigated.

The thesis starts with an overview of the Belle II experiment and related physics in chapter 2. Chapter 3 introduces the needed principles of deep learning and graph neural networks. In chapter 4 the Belle II tagging algorithm FEI and the deep semi-inclusive approach are explained and compared. A proof of concept for the deep semi-inclusive tagging method based on self simulated phase space events is shown in chapter 5. Then the deep semi-inclusive tagging method gets applied to official Monte-Carlo data in chapter 6. The neural network is trained to clean up the rest of event after the signal B meson reconstruction in chapter 7. A conclusion for the thesis is given in chapter 8.

CHAPTER 2

The Belle II Experiment and Physics Background

2.1 SuperKEKB

The SuperKEKB e^+e^- collider is located in Tsukuba, Japan. A graphical representation of the collider is shown in figure 2.1. SuperKEKB uses two beam pipes to accelerate bunches of electrons to 7 GeV and bunches of positrons to 4 GeV. The electron and positron bunches are then collided in the centre of the Belle II detector resulting in a centre of mass energy of $\sqrt{s} = 10.58\,\text{GeV}$. This is the invariant mass of the $\Upsilon(4S)$ meson, which decays for more than 96% [3] into charged or neutral B meson pairs. Therefore the SuperKEKB is often referred to as B factory.
The SuperKEKB collider reaches peak luminosities of $8 \times 10^{-35}\,\text{cm}^{-2}\text{s}^{-1}$ and produces around 10^{10} B meson pairs per year.

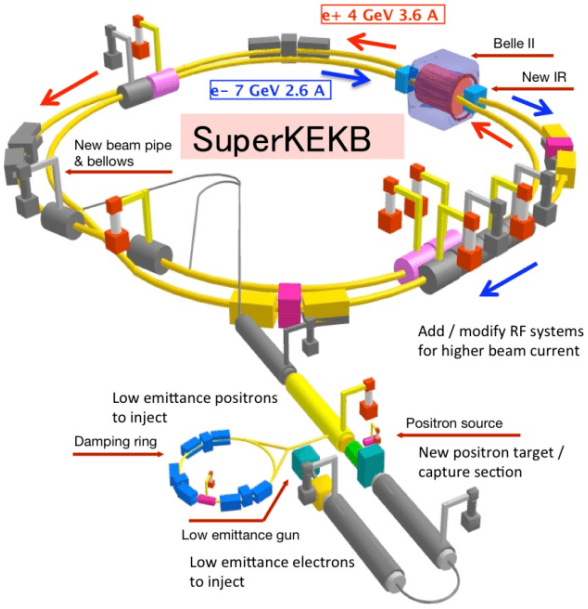


Figure 2.1: The SuperKEKB Collider [4]

2.2 Belle II Detector

The Belle II detector aims to measure the particles produced by the e^+e^- collision. The detector is planned to collect a total integrated luminosity of $50\,\text{ab}^{-1}$ in its lifetime.

The coordinate system within the Belle II detector is set that the z-axis is approximately along the electron beam. The angle θ is defined as the zenith angle in respect to the z-axis.

The trajectories, and also often their causative particles, of charged particles in high energy physics experiments as Belle II are referred to as tracks. The space coordinates of a particle creation are called vertex.

2.2.1 Sub Detectors

Belle II consists of multiple sub detectors which are placed in a cylindrical shape around the collision point. Figure 2.2 shows a cross section of the Belle II detector with its labeled components. The most important parts are explained here briefly, the *Belle II Technical Design Report* [5] explains everything in more detail.

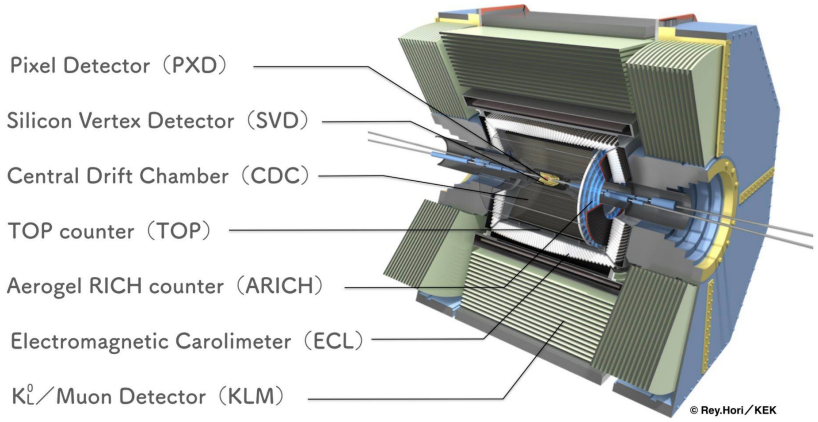


Figure 2.2: Cross section of the Belle 2 detector labeled with the names of its components [6]

Pixel Detector (PXD) and Silicon Vertex Detector (SVD)

The PXD and SVD are the two most inner layers of the Belle II sub detectors. They enable the high precision reconstruction of vertices.

Central Drift Chamber (CDC)

The CDC is a wire chamber filled with gas placed in a magnetic field. This allows to determine the momentum of charged tracks by measuring their curvature resulting from the magnetic field.

Time Of Propagation Counter (TOP) and Aerogel RICH Counter (ARICH)

The TOP and ARICH detectors are both Cherenkov detectors. The TOP detector is placed in the barrel region and the ARICH detector is placed in the end cap region of Belle II. Both detectors are

used for particle identification.

Electromagnetic Calorimeter (ECL)

The ECL consists of CsI(Tl) scintillation crystals and it is used among other things to identify electrons and to detect high energy photons.

K_L^0/Muon Detector (KLM)

The most outer layer of Belle II, the KLM, is designed to detect K_L^0 and μ. It consists of alternating iron plates and active detector elements.

2.3 Physics Processes in e^+e^- Collisions

Several processes take place in e^+e^- collisions, a non exhaustively overview is given here. The continuum events are non-resonant $e^+e^- \to q\overline{q}$ reactions. Then there are $e^+e^- \to e^+e^-$ reactions which have a more than 100 times higher cross section in Belle II than the actual wanted $\Upsilon(4S)$ production. Because the e^- and e^+ are collided in bunches, those $e^+e^- \to e^+e^-$ reactions can also take place simultaneously to a $\Upsilon(4S)$ event and therefore end up being detected in the detectors in the same event. Sources like these are what produces the background particles, which are not related to the investigated $\Upsilon(4S)$ decay event. The $\Upsilon(4S)$ ($b\overline{b}$) meson is produced in resonant $e^+e^- \to b\overline{b}$ reactions.

In comparison to hadronic collisions, e^+e^- collisions are relatively clean. This means the energy and momentum of the e^+e^- system is well known. Because the $\Upsilon(4S)$ mostly decays into B meson pairs, the number of $\Upsilon(4S)$ daughter particles is usually well known, too. The B mesons decay further and the particles which have a long enough life time to be detected in the detectors are referred to as final state particles (FSPs), regardless if they are stable particles like the e^- or not like the K and μ.

2.4 Tag-Side Reconstruction

Due to the well known kinematics of the $\Upsilon(4S)$ meson in Belle II, even difficult to reconstruct B decays can be examined. The B with the decay of interest is called B_{sig} and the other one is called B_{tag}. Examples for hardly reconstructable decays are decays which involve at least one neutrino, because neutrinos are practically impossible to detect for Belle II. To fully reconstruct the B_{sig} decay the best possible reconstruction of the B_{tag} is necessary, because then the reconstructed B_{tag} and the known kinematics of the $\Upsilon(4S)$ can be used to restrict the B_{sig} meson's kinematics.

The B_{tag} reconstruction, also called tag-side reconstruction or tagging, can mostly be organized into three categories. Figure 2.3 gives an graphical overview for the three categories. Generally speaking the exclusive tagging usually has a higher precision than the inclusive tagging, but therefore has fewer events successfully reconstructed. The semi-inclusive tagging tries to combine the benefits of both other methods.

In Belle II jargon the term rest of event is used to define the final state particles which remain once a composite particle is reconstructed. Therefore a rest of event is always connected to a already reconstructed particle.

2.4.1 Exclusive Tagging

For the exclusive tagging the B_{tag} gets reconstructed explicitly. This means that every subsequent decay of the B_{tag} and its daughters needs to be reconstructed with a specific decay chain.

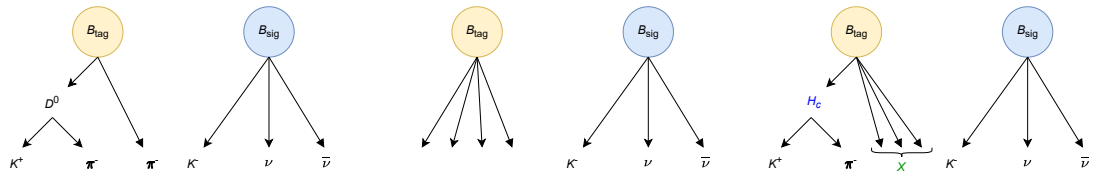
2.4.2 Inclusive Tagging

For the inclusive tagging the B_{sig} gets reconstructed first and then all remaining FSPs are used as B_{tag} descendants and combined to reconstruct the B_{tag}, with no explicit decay chains getting reconstructed.

2.4.3 Semi-Inclusive Tagging

For the semi-inclusive tagging the B_{sig} and the H_{c} get reconstructed first. The remaining FSPs are grouped into the X system. X and H_{c} are then combined to reconstruct the B_{tag}.

In general the H_{c} is a placeholder for any charmed hadron. For the scope of this thesis H_{c} means that the following hadrons or, if existing, their charge conjugated version were aimed to be reconstructed: D^{*0}, D^{*-}, D^{-}, D^{0}, D_{s}^{+}, Λ_c and J/ψ.



(a) Exclusive tagging (b) Inclusive tagging (c) Semi-inclusive tagging

Figure 2.3: Schematic overview of the three different tagging techniques

2.5 Physics Variables

Physics variables relevant for this thesis are listed here.

Particle Identification Variables (IDs)

Particle IDs get calculated based on the track's kinematics in a way that these IDs should help to decide which particle was measured in form of a charged track.

Vertex Information

The vertex information d_{r} is the transverse distance between the vertex of a particle and the e^+e^- collision point, and d_{z} is the z-axis distance between the vertex and the e^+e^- collision point.

ECL Cluster Variables

The ECL cluster variable `clusterReg` is the ECL region the cluster was measured and `clusterE9E21` is the "ratio of energies in inner 3x3 crystals [...] and 5x5 crystals around the central crystal without corners" [7].

Beam Constraint Mass M_{bc}

The beam constraint mass for the B_{tag} is defined as follows [8]

$$M_{\mathrm{bc}} = \sqrt{\left(\frac{\sqrt{s}}{2}\right)^2 - \left(\vec{p}^{\,\mathrm{CM}}_{B_{\mathrm{tag}}}\right)^2} \tag{2.1}$$

with $\sqrt{s}$ being the centre of mass energy and $\vec{p}^{\,\mathrm{CM}}_{B_{\mathrm{tag}}}$ the momentum vector of the B_{tag} in the centre of mass (CM) frame.
For a correctly reconstructed B meson M_{bc} is expected to be the B meson mass m_B.

Missing Mass Squared m^2_{miss}

The m^2_{miss} is defined for the reconstructed $\Upsilon(4S)$ meson as follows [9]

$$m^2_{\mathrm{miss}} = \left(\frac{\sqrt{s}}{2} - E^{\mathrm{CM}}_{B_{\mathrm{sig}}}\right)^2 - \left(\vec{p}^{\,\mathrm{CM}}_{B_{\mathrm{sig}}} + \vec{p}^{\,\mathrm{CM}}_{B_{\mathrm{tag}}}\right)^2 \tag{2.2}$$

with $\sqrt{s}$ being the centre of mass energy, $E^{\mathrm{CM}}_{B_{\mathrm{sig}}}$ the energy of the B_{sig} in the CM frame, $\vec{p}^{\,\mathrm{CM}}_{B_{\mathrm{sig}}}$ the momentum vector of the B_{sig} in the CM frame and $\vec{p}^{\,\mathrm{CM}}_{B_{\mathrm{tag}}}$ the momentum vector of the B_{tag} in the CM frame.
For a fully hadronic B_{tag} and a B_{sig} with one neutrino daughter particle like in a $D^*\ell\nu$ decay, the m^2_{miss} distribution is expected to peak at zero for correctly reconstructed events.

Energy Difference ΔE

The energy difference ΔE for a B_{tag} meson is defined as follows [8]

$$\Delta E = E^{\mathrm{CM}}_{B_{\mathrm{tag}}} - \frac{\sqrt{s}}{2} \tag{2.3}$$

with $E^{\mathrm{CM}}_{B_{\mathrm{tag}}}$ being the energy of the B_{tag} in the CM frame and $\sqrt{s}$ being the centre of mass energy.
For a correctly reconstructed B meson ΔE is expected to be zero.

2.6 basf2 Reconstruction Software

`basf2`, short for Belle II Analysis Software Framework, is the official data analysis and simulation software of the Belle II experiment [10]. It gets used to process real detector raw data, to generate Monte-Carlo events and to reconstruct particle decays. For the latter the analyst creates a path. To

which one adds all wanted analysis steps, like filling lists with FSP candidates or reconstructing specific decays out of FSPs or other already reconstructed composite particles. Also `basf2` allows the analyst to create custom modules with new functionalities which then can be added to particle reconstructions.

2.6.1 Particle Categories based on basf2 Variables

Throughout this thesis it is often necessary to assign final state particles to categories. Those categories depend on from which ancestor particle the final state particle stems from. For simulated Monte-Carlo decay events this is checked with the `basf2` variables `genMotherPDG(i)` and `genMotherID(i)`, which store the PDG code and array index of the i-th mother particle. The array index gives every particle in an event a unique number, counting starts by the $\Upsilon(4S)$ meson getting a zero. If a final state particle has no related ancestor B meson it is considered to be background.

2.6.2 stdMostLikely Function

The `stdMostLikely` function of `basf2` decides which mass hypothesis to choose for charged particles based on the particle ID information. Also this function applies selections on the final state particles. These are `thetaInCDCAcceptance` == 1, `nCDCHits` > 20, $d_r < 0.5$ and $|d_z| < 2$. In section A.1 the first two variables are explained.

CHAPTER 3

Deep Learning and Graph Neural Networks

Deep learning is a branch of the wider field of machine learning based on artificial neural networks (NN) as these NNs are designed to mimic structures which are found in human or animal brains. Many of the features of NNs have analogous structures or features in the biological structure of brains. [11] [12]

3.1 Deep Learning Principles

Deep learning is based on the processing of input data with a NN in order to use latent patterns in data for tasks like classification, regression, data generation, etc. In the example of picture recognition the input data consists of the pixels of the picture and the output can be a classification of what is shown in the picture, for example a dog or a cat. This classification by the NN can be seen as its prediction. The inner structure of a NN consists of layers of neurons. The most simple layer is the feed forward layer, which will be used as an example to explain the main features of NNs. Each of a layer's elements is called a node. The single elements of input and output are also called input and output nodes.
Figure 3.1 shows an example of a NN with eight input nodes on the left, two subsequent feed forward layers with six nodes each and four output nodes on the right. The arrows represent the connections of the nodes and the direction of the data flow. One sees that every node of a layer is connected to all nodes of the subsequent and previous layer, which is the principle of feed forward layers. Inside of the feed forward layer nodes a calculation takes place. Every incoming value from a previous layer gets multiplied by a corresponding weight. The sum of these weighted inputs is then used as the argument of a non-linear activation function. This function's output is then the node's output which gets propagated to the next layer.
The general structure of the network, like which and how many layers are used, are critical for the performance of the network. Different network architectures are needed for different kinds of data. Features of the network like number of layers or the number of nodes per layer are referred to as hyper parameters.
If the structure of the network is considered to be the brain of the network, the already mentioned weights are its memories. In order to get the right weights for meaningful predictions a NN gets trained. This is done by feeding input data to the network for which the correct prediction or label is already known. Then the loss function can be calculated, which processes the network's outputs into a single real number. Usually the loss function's value gets lower for higher accuracies of the network's

predictions. The weights get adjusted with the back propagation in order to minimize the loss function [13]. This weight adjustment is done for subsets of the whole training data at once, called batches. The speed of these adjustments is regulated with a factor called the learning rate. Its purpose is to prevent too big fluctuations in the weights during the training process.

The training data gets processed in multiple cycles which are called training epochs. Usually the training data gets split into a training set and a validation set. The training set gets used to adjust the weights. The validation set is used to prevent over training, which occurs when the network starts learning the statistical fluctuations of the particular data set used for training. Therefore the performance worsens when data the network has not been trained on yet is presented to it. Randomly dropping some nodes out of the network during training is one of many other regularization techniques to reduce over training. The hyper parameter drop rate defines the fraction of nodes to ignore during training. Other regularization techniques are for example data augmentation, increasing the training data by creating copies with slight deviations, or early stopping, stopping the training once the loss of the validation data increases.

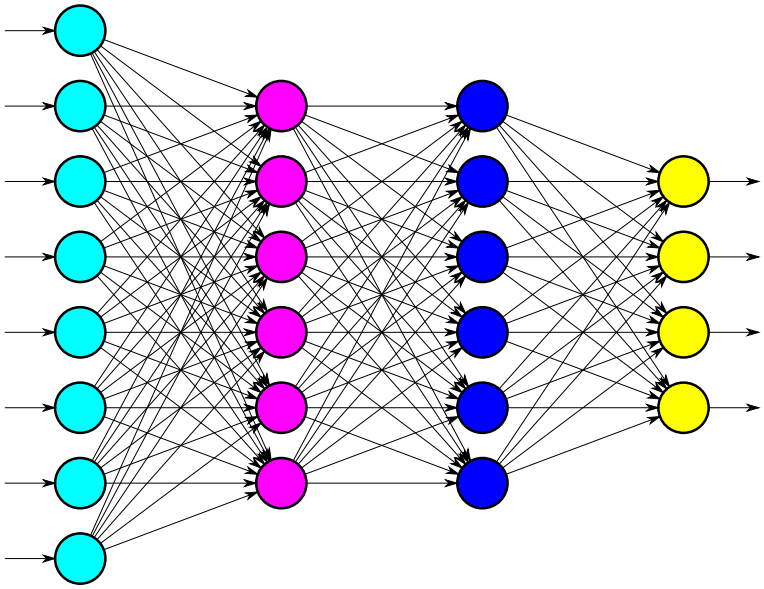


Figure 3.1: A neural network with eight input nodes on the left, two subsequent feed forward layers with six nodes each and four output nodes on the right. The arrows represent the connections of the nodes and the data flow. [14]

3.2 Graph Neural Networks

Graph NNs use graph representations of their input data. In graph representation the nodes represent the objects which are of interest and the edges represent the relations between the objects. The graph that represents the relations between the interconnected entities can be designed to be fully connected, meaning all nodes are connected to each other, or in any other wanted layout.
For example in high energy physics the nodes of the graph can be used to represent particles and the edges are then representing the relations between the particles. See [15] for a more detailed overview

of graph NNs in high energy physics.

3.3 NRI Model

The NN used in this thesis is based on the NRI model provided by ***N**eural **R**elational **I**nference for Interacting Systems* (NRI) [16]. The NRI model is a graph auto encoder. The principle of an auto encoder is to transform the input data into a representation of a smaller dimension and then using this representation to recreate the input data. For the usage in this thesis only the dimension reduction part of the auto encoder is needed. The main features of the NRI model are the Node2Edge and Edge2Node layers. These layers transform the node based representation of the data to an edge based representation and vice versa. See [17] for an similar application of the NRI model than used in this thesis and a more extensive explanation of the Node2Edge and Edge2Node layers. The code implementation of the NRI model used as a basis for this thesis also stems from there. The used implementation is based on the `pytorch` framework [18].

The NRI model consists of an initial MultiLayer Perceptron (MLP), then a Node2Edge layer, followed by a second MLP and then multiple blocks. A MLP consists of two feed forward layers. A block contains a Edge2Node layer, followed by a MLP, an Node2Edge layer and another MLP. The last layer of the NRI model is a feed forward layer with the output dimension set by the user.

CHAPTER 4

FEI and Deep Semi-Inclusive Tagging

4.1 FEI Tag-side Reconstruction

The Full Event Interpretation (FEI) is an particle reconstruction algorithm, designed to be used for the Belle II experiment. It uses boosted decision trees to hierarchically reconstruct the decay chains of B mesons. Each reconstructed particle in the decay chain gets assigned a probability called `SignalProbability` which should indicate how likely it is that the reconstruction was correct. More than 100 decay channels have been hard coded into the FEI. This leads to more than 10^4 B meson decay chains that can be reconstructed. Figure 4.1 shows a schematic overview of the hierarchically B meson reconstruction of the FEI. The reconstruction starts with the detector data from which the final state particles are reconstructed. Subsequently the composite particles and finally the B meson get reconstructed. See [19] for more details on the FEI.

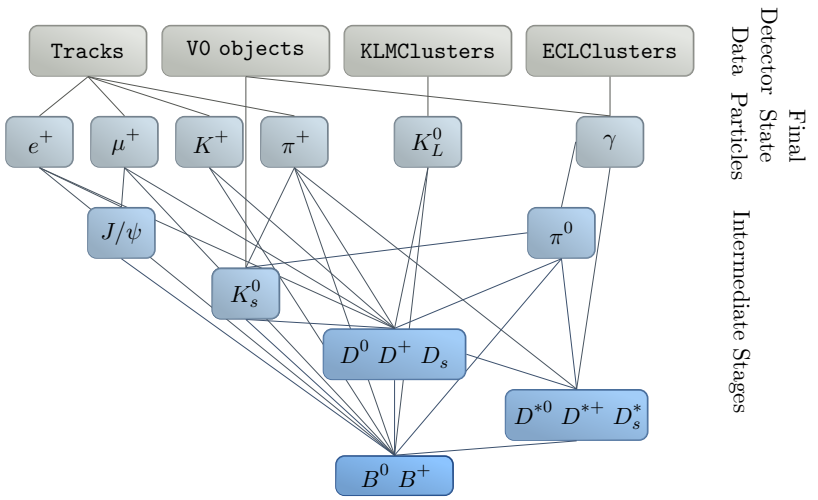


Figure 4.1: Schematic overview of the hierarchically B meson reconstruction of the FEI [19]

4.2 Deep Semi-Inclusive Tagging Model

The Model

The NRI model as described in section 3.3 was altered for the application used in this thesis. The resulting model was named the **DSIT model** (**D**eep **S**emi-**I**nclusive **T**agging **model**). The alternation consisted in changing the output of the model from an edge representation of the data to a node or particle based representation. This was achieved by adding an Edge2Node layer before the final feed forward layer.

The DSIT model is used to categorize final state particles of an $\Upsilon(4S)$ decay event. For a number of particles m and a number of features per particle n the input has the dimension $m \times n$ and the output has the dimension m. Figure 4.2 represents the flow of input data through the network to the output data. For categorization into k categories the DSIT model produces an integer prediction of each particle's category in the range $[0, k-1]$. The m particles are organized into a fully connected graph. Each node of the graph, representing a particle, is a vector of dimension n which holds the input features for this specific particle.

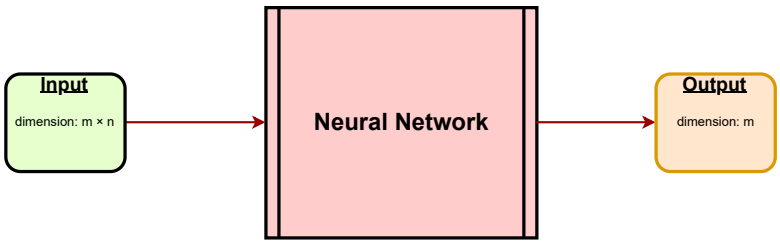


Figure 4.2: Schematic Overview of the input data flow through the DSIT model and its resulting output. m is the number of particles represented in the input, n is the number of features provided per particle.

Hyper Parameters

The hyper parameters of the model are n_{hid}, the number of nodes per layer, n_{blocks}, the number of blocks in the model, the drop rate, the initial learning rate, which was set to 0.001 and the batch size. During the training of the DSIT model the learning rate was adjusted automatically using the `ReduceLROnPlateau` functionality from `pytorch`. This functionality reduces the learning rate if the loss was not reduced for several epochs. The used loss function is the cross entropy loss function.

Metrics

To monitor the performance of the DSIT model, several metrics were defined. Most important is the **accuracy**, which is the fraction of correctly predicted particles based on all particles. Then there are the category-wise accuracies, which are the respective fractions of correctly predicted particles per category. The event based metrics measure the fraction of events in categories of how many prediction errors the DSIT model made per event. The best possible outcome, namely zero prediction errors, is called **perfect** (separation). One error per event is called **1E**, 2 errors per event **2E**, three errors per event **3E** and four or more errors per event **≥ 4E**.

Separation Cases

Depending on the supposed usage different separation cases are created for the DSIT model. The separation case defines which categories are to be predicted by the DSIT model. For each separation case all final state particles which are not supposed to be predicted by the DSIT model, get deleted. This means the DSIT model only gets presented the particles which should get predicted by it.

Deep Semi-Inclusive Tagging

In this deep approach we want to improve the semi-inclusive tagging that was described in section 2.4. The H_c gets reconstructed by the FEI and the B_{sig} and X system are meant to be reconstructed based on the DSIT model's prediction. For this the DSIT model is used with the B_{sig}- X - background separation case.

4.3 Comparison FEI and Deep Semi-Inclusive Tagging Model

As mentioned the FEI is using hard coded decay channels. This restricts the amount of B decays the FEI can reconstruct. In comparison the DSIT model has no hard coded restrictions for the decay channels it can possibly learn.

Also the general structure of the FEI's boosted decision trees can potentially decrease its flexibility. Early, wrong decisions during the hierarchically designed reconstruction process are influencing the complete subsequent reconstruction. The DSIT model on the other hand does not work hierarchically.

A potential down side of the DSIT model could be that there is only one prediction for each final state particle of an event. If errors are in these predictions, the complete reconstruction is flawed. The severity of this flaw is then influenced by how important the falsely predicted particle was. For example falsely predicting a low momentum particle is less bad for the reconstruction than falsely predicting a high momentum particle. The predictions can also only produce a limited number of $\Upsilon(4S)$ candidates, while using the FEI in general produces more candidates which can be compared to one another.

CHAPTER 5

Deep Semi-Inclusive Tagging Toy Model

The goal of this chapter is to show a proof of concept. The goal was to create simple, easy adjustable test data, in order to prove the concept of the deep semi-inclusive tagging based on the DSIT model.

5.1 Training Data Creation

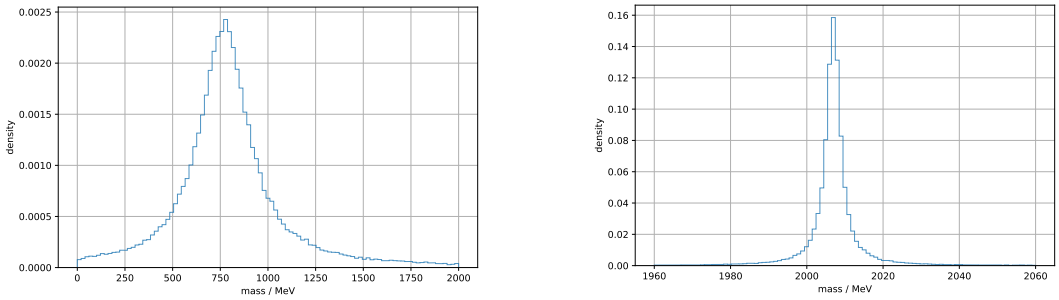
The particle decay events used for training data creation were simulated with the python package `PhaseSpace` [20]. For this two sets of six different decay modes each of the $\Upsilon(4S)$ were simulated. The used decay modes are shown in figure A.1 and A.2. For each of the six modes $15 \cdot 10^3$ events were simulated resulting in $90 \cdot 10^3$ events total. These events got split into 90% training data and 10% validation data.

5.1.1 Simulated Data Attributes

In order to simulate a more realistic experimental scenario, the complexity of the data was gradually increased. During this process we evaluated the performance of the DSIT model, to identify potential bottlenecks in the method.

For the ρ and D^{*0} mesons mass distributions were simulated instead of fixed mass values. The used mass distributions are shown in figure 5.1 and were calculated using a Breit-Wigner distribution and parameters from [3].

Pions with $p < 100\,\text{MeV}$ and photons with $p < 50\,\text{MeV}$ were dropped from the data set, to simulate background suppression selections. 5% of all final state particles were excluded from the data set randomly. Background particles were added randomly on an event basis. These background particles were then randomly picked from the daughter particles of a separately simulated 3-body decay of a hypothetical particle with mass $m = 1\,500\,\text{MeV}$ which decayed into a photon, a pion and a kaon. The number of extra particles was decided based on drawing the amount of background particles to add per event from a binomial distribution with $p = 0.5$ and $n = 3$. The momentum values for all charged final state particles were smeared with gaussian noise were the standard deviation was 2% of the original value. The momentum values of the photons were smeared in dependance of their respective momenta. The values for the momentum dependant smearing got taken from [8] in order to simulate a Belle II realistic scenario.



(a) ρ meson

(b) D^{*0} meson

Figure 5.1: ρ and D^{*0} meson mass distributions used for the toy data simulated with `PhaseSpace`

The $\Upsilon(4S)$ was boosted into the lab frame accordingly to the boost of the asymmetric e^+e^- collision in Belle II. All final state particles which did not fit in the θ range of $12.4^\circ - 155.1^\circ$ were dropped from the data set to match the detection acceptance of the ECL detector.

5.1.2 Separation Cases

Three different separation cases were simulated: X - B_{sig}- background, X - H_{c}- background and X - H_{c}- B_{sig}- background. For the B_{sig}- H_{c}- X - background separation the modes in figure A.1 were used, for the other two separation cases the modes in figure A.2 were used.

5.1.3 Simulated Features per Particle

The already described simulation steps provide the 4-momenta for all particles which are supposed to be given as input to the DSIT model. In addition to that the charge and a simulated hadron ID were added to each particle. The charge was assigned according to the respectively used decay modes (figure A.1 and A.2) and was simulated without any uncertainty. Two hadron IDs were provided, one for the pion ID and one for the kaon ID. The hadron IDs were simulated with a exponential distribution with range $[0, 1]$. The used scale factor for the distribution was 0.05. If a particle was for example a kaon, the pion ID was drawn from a exponential distribution having its maximum at zero, while the distribution for the kaon ID had its maximum at one. For photons and pions the distributions were switched accordingly.

See figure 5.2 for per mode momentum spectra of kaons, pions and photons (no background particles) on B_{sig}- X - background separation data as example.

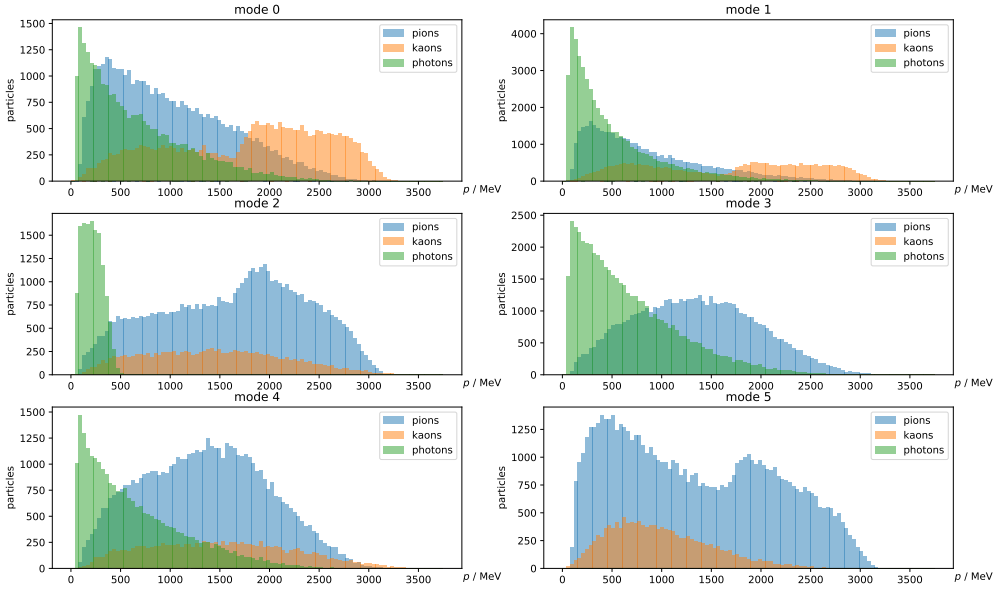


Figure 5.2: Momentum spectra of pions, kaons and photons per $\Upsilon(4S)$ decay mode for B_{sig}- X - background separation (no background particles).

5.2 Results

5.2.1 Hyper Parameters

Different hyper parameters and their combinations were tested until the best combination was found. n_{hid} and n_{blocks} proved to be the most crucial hyper parameters, who already affected the DSIT model's performance significant if only small changes were made. The best values found were setting n_{hid} to 1024, n_{blocks} to 8. In contrast the performance was only mildly affected by changes of drop rate and batch size. For this reason the drop rate was set to 0.3 and the batch size to 128.

5.2.2 DSIT Model Performance

As one can see in table 5.1 the performance of the DSIT model on the toy data was high for all three tested separation cases. For the X - H_{c}- background and the X - B_{sig}- background separation case the performance is better than for the X - H_{c}- B_{sig}- background separation, which makes sense because for the latter an additional category had to be predicted by the model.

Figure 5.3, 5.4 and 5.5 show the DSIT model training progress over the training epochs for the X - H_{c}- B_{sig}- background separation case. Figure 5.3 shows the cross entropy loss for the course of the DSIT model training for the training and validation data set separately. The deviation of training set and validation set loss from roughly epoch 20 onwards shows that overtraining occurs. But because the validation set loss does not increase notably the overtraining can be tolerated here. Figure 5.4

separation case	**perfect separation**	**accuracy**
X - H_c- background	93.4%	98.4%
X - B_{sig}- background	91.9%	98.3%
X - H_c- B_{sig}- background	81.8%	97.4%

Table 5.1: Results of the DSIT model for different separation cases on the validation data.

shows the fraction of number of prediction errors per event, normalized to the overall accuracy over training epochs on the validation data set. This shows that the events that are not predicted perfectly mostly only have small amounts of prediction errors per event, namely one or two. Figure 5.5 shows the category wise accuracies and the overall accuracy over training epochs on the validation data set. The categories X, H_c and B_{sig} show similar accuracies while the background accuracy is the lowest. All three figures show that the most learning success by the DSIT model is gained during the first 20 epochs and afterwards only small improvements are still achieved.

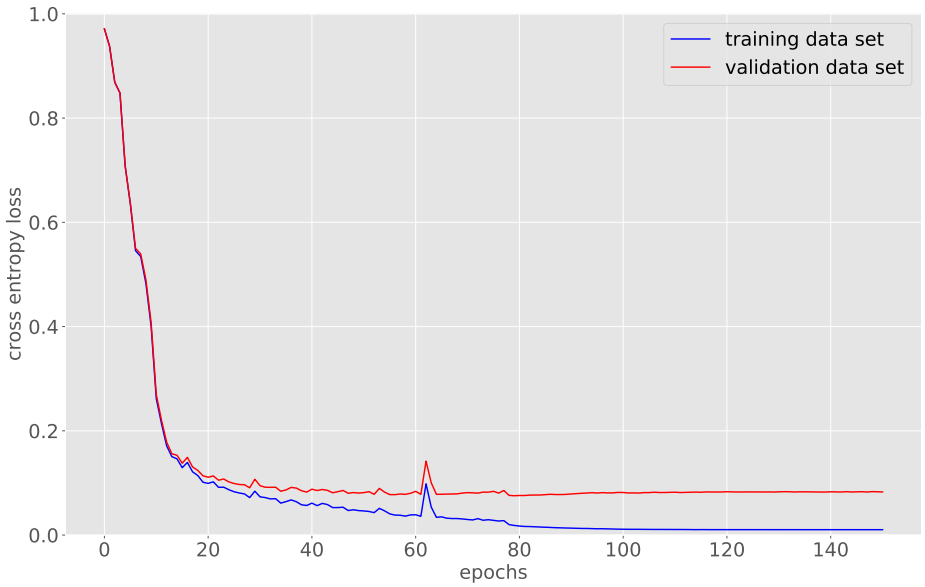


Figure 5.3: Cross entropy loss for DSIT model training for the X - H_c- B_{sig}- background separation case, over training epochs on training and validation data sets.

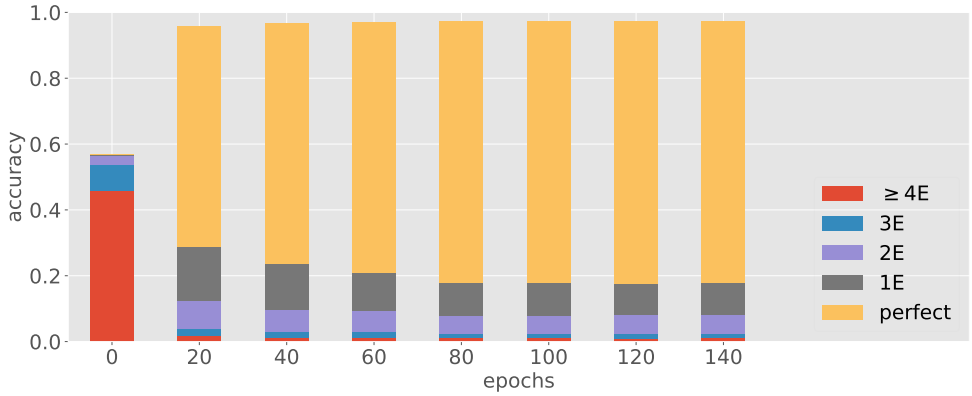


Figure 5.4: Fraction of number of prediction errors per event, normalized to the overall accuracy for X - H_{c}- B_{sig}- background training over training epochs on validation data set.

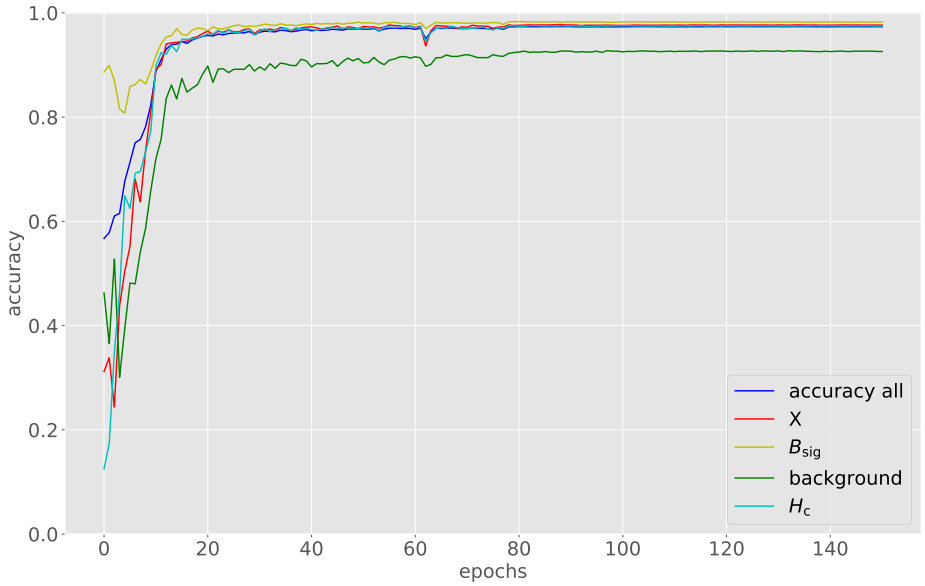


Figure 5.5: Category wise and overall accuracies for X - H_{c}- B_{sig}- background training over training epochs on validation data set.

5.2.3 Hadron ID Studies

In this study we test whether or not the simulated kaon and pion IDs can replace the mass hypothesis information for the DSIT model which is provided for each particle via its energy. In order to do this, two data sets based on the X - B_{sig}- background separation were created. For the first one all kaon masses were set to the pion mass m_π, resulting in all hadrons having the same mass. For the second data set the energy information was dropped from the input feature vector. Each data set was then used to train the DSIT model with four different sets of input variables. One with only the momenta (4-momenta respectively 3-momenta without the energy) and one each with the charge, hadron IDs and charge and hadron IDs in addition to the momenta information. See table 5.2 for the fraction of perfectly predicted events for each case. For both cases adding the charge information does not improve the DSIT model performance, while adding the hadron IDs improved the performance notably at least for the first case. For the case where the model was not exposed to the energy information, but only to the hadron IDs, no improvement was observed. It seems that the hadron IDs can only increase the performance of the DSIT model performance when the energy information is also provided, even if the masses are the same for all hadrons.

In comparison to table 5.1 where the X - B_{sig}- background yielded almost 92% perfectly predicted events one sees that both cases investigated here had worse performance. This means the mass/energy information can not be fully replaced by a deep representation that the DSIT model creates from the hadron ID features.

NN input variables	all $m_{\text{hadr.}} = m_\pi$	no energy passed
only momenta	76% (4-momentum)	68% (3-momentum)
+charge	76%	68%
+hadron IDs	87%	69%
+charge+IDs	87%	69 %

Table 5.2: Hadron ID study results for X - B_{sig}- background separation on validation data. The entries show the fraction of perfectly predicted events. The first column shows the input variables passed to the DSIT model, the second and third column show the different hadron ID study cases. For the second column all hadron masses were set to the pion mass m_π, for the third column the energy information was not provided to the NN and therefore no mass hypothesis.

5.3 Creation and Training on Real MC of the 6 Toy Modes

The DSIT model performance on the toy data so far was promising, therefore the six toy modes were simulated with the official Monte-Carlo creation software for Belle II called `basf2`, which was described in section 2.6. For this the six modes from figure A.1 were used for all separation cases that were tested. Each separation case was created once with the generated information and once with the reconstructed information, where detector and reconstruction effects were included. For both data sets the 4-momenta were used as input for the DSIT model. For the reconstructed information two extra data sets, where additionally the charge respectively the hadron IDs for kaons and pions were used as input, got created. This was done for the X - H_c - background and X - B_{sig}- background separation cases.

The fraction of perfectly predicted events for each case is presented in table 5.3. One can see a

significant decrease between the generated information and the reconstructed information. This is expected due to reconstruction uncertainties. Also notable is that the addition of charge or the hadron (pion and kaon) IDs to the input data increases the performance in all tested cases. This means that the DSIT model can better distinguish between the particle categories when it is provided with this extra information. In contrast during the hadron studies in section 5.2.3 no increase for the DSIT model's performance was found for adding the charge to the input feature vector. This shows that for the more complex simulated events created with `basf2`, the DSIT model needs more information to make meaningful predictions.

separation case	generated information	reconstructed information		
NN input variables	4 momentum		+charge	+hadron IDs
X - H_c - background	98%	40%	49%	44%
X - B_{sig}- background	99%	33%	38%	36%
X - H_c - B_{sig}- background	94%	13%	-	-

Table 5.3: Fraction of perfectly predicted events for the six decay modes simulated with `basf2`. For generated and reconstructed information. All cases with the 4-momentum as input for the DSIT model, last two columns with charge or hadron IDs as additional input.

CHAPTER 6

Application of Deep Semi-Inclusive Tagging on Monte-Carlo $B \to D^* \ell \nu$ Events

The previous studies aimed on verifying that the DSIT model is appropriate for the task of separating particles based on their ancestor particle. As shown in chapter 5 the DSIT model showed promising performance on the toy data and also on events which were simulated with `basf2`. Therefore this chapter aims to apply the DSIT model on a realistic reconstruction problem, namely the deep semi-inclusive tagging as described in section 4.2. For this the X - B_{sig}- background separation case is used for the DSIT model, while the H_{c} will be provided by the FEI. The tagging in this chapter is performed for $B_{\text{sig}} \to D^*(\to D^0\pi)\ell\nu$ ($\ell = e, \mu$) decays. The B_{tag} decays generically. Everything in this chapter is based on $B\overline{B}$ Monte-Carlo events.

6.1 Existing Semi-Inclusive Tagging for $B \to D^* \ell \nu$

The already existing semi-inclusive tagging method for $B \to D^*(\to D^0\pi)\ell\nu$ events from [9] will be used as a baseline to compare the DSIT model's performance. This method is not using any multivariate analysis tools, except for the reconstruction of the H_{c} with the FEI, which is exactly the case for this application as well. The tagging strategy of this approach is described in section 2.4. Notable is that the best candidate selection for the $\Upsilon(4S)$ is performed by taking the $\Upsilon(4S)$ candidate with the highest FEI `SignalProbability` for the used H_{c} candidate. Also only specific D^0 modes get reconstructed, namely $D^0 \to K\pi$, $D^0 \to K\pi\pi^0$, $D^0 \to K\pi\pi\pi$ and $D^0 \to K\pi\pi\pi\pi^0$.

6.1.1 Performance of Existing Semi-Inclusive Tagging for $B \to D^* \ell \nu$

Figure 6.1 shows the m^2_{miss}, as defined in equation (2.2), distribution of the reconstructed $\Upsilon(4S)$ mesons, reconstructed with the existing semi-inclusive tagging. In blue all events which passed all the applied selections are shown and in orange all events which additionally have a correctly reconstructed B_{sig}. The distributions are quite symmetrical and are centered at zero as it is expected for the reconstructed decay.
For the existing semi-inclusive tagging the efficiency is defined as the number of tagged $D^*\ell\nu$ events divided by the number of $D^*\ell\nu$ events in the analysed sample. The purity is defined as the fraction of truth matched $D^*\ell\nu$ events in the tagged sample. The efficiency of the existing method is 2.75% and the

purity is 70% when the `SignalProbability`, as described in section 4.1, for the FEI reconstructed H_c candidate is required to be at least 0.001.

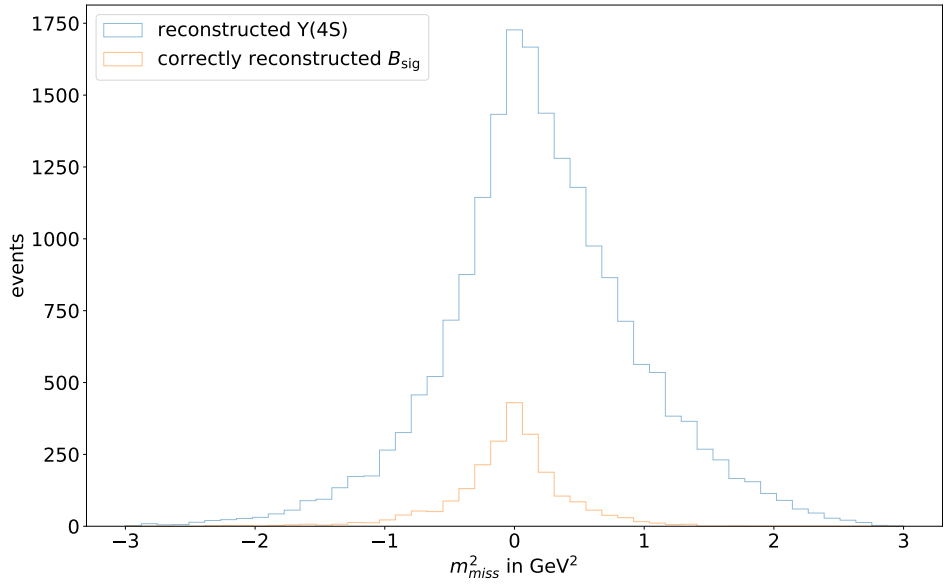


Figure 6.1: m^2_{miss}, as defined in equation (2.2), distribution for the existing semi-inclusive tagging method. In blue all events which passed all the applied selections are shown and in orange all events which additionally have a correctly reconstructed B_{sig}.

6.1.2 DSIT Model Performance on Data from Existing Semi-Inclusive Tagging Algorithm

To create a direct comparison between the already existing semi-inclusive tagging algorithm and the DSIT model, a training of the DSIT model was performed based on the final state particle selections the already existing algorithm is based on. Additionally the H_c reconstructed by the FEI was required to be perfectly reconstructed. This was done in order to ensure that the target labels were correct. If a not correctly reconstructed H_c candidate would be used for labeling the FSPs of an event, the labels would become ambiguous. This is because it is non-trivial to assume which of the particles actually existing in the event, the FEI tried to reconstruct as H_c. Only events for which the already existing method could reconstruct a $\Upsilon(4S)$ candidate were considered further.

A notable issue using the data for the DSIT model was assigning a mass hypothesis for each track. While the already existing method can handle multiple mass hypothesis candidates for each track, the DSIT model needs a fixed mass hypothesis because every track is only passed to the model once. The mass hypothesis was decided based on what the already existing method ended up using for the same decay event. For example if a track was used as a pion by the already existing method, this mass hypothesis was used in the DSIT model input feature vector. If the already existing method did not use a particle, the mass hypothesis was chosen randomly from the reconstructed candidates. Variables

used as input for the DSIT model are the particle's mass, charge, the particle IDs, namely pion ID, kaon ID, electron ID, muon ID and proton ID. Also the vertex information d_r and d_z are saved for each particle. The ECL cluster variables `clusterReg` and `clusterE9E21` are used too. In section 2.5 these variables are explained.

The used hyper parameters for the DSIT model were n_{hid} set to 512, n_{blocks} set to 4, the drop rate set to 0.3 and the batch size set to 64.

Table 6.1 shows the results of this comparison. Only the validation data events from the DSIT model are used here to compare the performance of the DSIT model and the existing method. The sample consists of $52 \cdot 10^3$ events. For this specific data set the DSIT model performs significantly better than the already existing method in terms of accuracy for all particles and in terms of prediction errors per event. However this is not translated into perfect predictions on an event basis. More than three quarters of events have more than 3 prediction errors. This problem will be addressed in the following sections.

case	perfect	1E	2E	3E	≥ 4E	accuracy
DSIT model	1.8%	4.2%	6.9%	9.2%	77.9%	76.3%
existing method	0.0%	0.0%	0.0%	0.0%	100%	56.6%

Table 6.1: Particle assignment errors per event and accuracy for the comparison of DSIT model performance and existing semi-inclusive tagging algorithm. Number of events: $52 \cdot 10^3$.

6.2 Training Data Creation

As shown in section 6.1.2, more adjustments were needed for feasible results. Training based on the data created here is the base for everything that follows in this chapter. In order to improve the results of the DSIT model specific data selections were applied for the training data.

6.2.1 Event and Particle based Selections

A generator level selection was applied to all events, selecting only $D^*(\rightarrow D^0\pi)\ell\nu$ events and the specific D^0 modes the already existing semi-inclusive tagging method is reconstructing. The H_c is required to be perfectly reconstructed and to have a B meson as mother particle. The mass hypothesis for each track was picked by using the `stdMostLikely` function of `basf2` as described in section 2.6.2. The top 0.5% of events in terms of number of particles per event got deleted in order to create training data with less outliers. This does not statistically affect our experiments, because the distribution of the number of particles per event has a very long tail towards high numbers of particles. The created data consisted of roughly $200 \cdot 10^3$ events. Those events were split into 80% training and 20% validation data.

6.2.2 Used Input Variables

Variables used as input for the DSIT model are the particle's mass, charge, the particle IDs, namely pion ID, kaon ID, electron ID, muon ID and proton ID. Also the vertex information d_r and d_z are saved for each particle. The ECL cluster variables `clusterReg` and `clusterE9E21` are used too. In

section 2.5 these variables are explained. Also provided were the H_c vertex coordinates. Because the general input data structure of the DSIT model is particle based, the H_c vertex coordinates were provided for every particle.

Variables which are not defined for all FSPs, e.g. the ECL cluster region variable is not assigned for particles which were not detected with the ECL, got imputed with -1.0. This allows the network to condition the existence of the feature or not because all variables which are not defined for all FSPs have ranges greater or equal to zero.

6.3 Hyper Parameters and Ablation Studies

6.3.1 Hyper Parameters

Different hyper parameters and their combinations were tested until the best combination was found. n_{hid} and n_{blocks} proved to be the most crucial hyper parameters, who already affected the DSIT model's performance significant if only small changes were made. The best values found were setting n_{hid} to 256, n_{blocks} to 4. In contrast the performance was only mildly affected by changes of drop rate and batch size. For this reason the drop rate was set to 0.1 and the batch size to 64.

6.3.2 Ablation Studies

In order to test which variables are the optimal input features, ablation studies were performed. Table 6.2 shows the distribution of prediction errors per event and the accuracy calculated for all FSPs for the different sets of input features. Note that the row "all variables" means that here all input variables as described in section 6.2 were provided to the DSIT model. In all the other cases, one variable or group of variables is excluded from the input. Only exception is the ablation case named "only 4-momentum", where only the 4-momentum was provided to the DSIT model.
It is surprising that most variables have only minor, unclear effects if left out. If `clusterReg`, pion ID, kaon ID or the H_c vertex are left out the DSIT model performs even slightly better than with all variables provided, although the effect is small. Only leaving out the charge has a significant effect by decreasing the accuracy of the DSIT model by 2 p.p. in comparison to passing all variables to the DSIT model. Interestingly is that even passing only the 4-momentum is achieving already 70.3% of accuracy in comparison to 76.5% if all variables get passed. This shows that the by far most important set of variables for the DSIT model is the 4-momentum, followed by far by the charge.
Because the best result was the one where the H_c vertex information was left out and in order to save the computation time needed to perform the vertex fit, it was decided that in the following sections the input variables will be used without the H_c vertex coordinates.

ablation case	perfect	1E	2E	3E	≥ 4E	accuracy
no H_c vertex	13.7%	17.2%	17.0%	15.3%	36.8%	76.8%
no `clusterReg`	13.7%	17.0%	17.1%	15.1%	37.1%	76.7%
no pion ID	13.5%	17.0%	17.1%	15.3%	37.1%	76.6%
no kaon ID	13.4%	17.0%	17.3%	15.1%	37.2%	76.5%
all variables	13.4%	17.0%	17.0%	15.4%	37.3%	76.5%
no proton ID	13.6%	17.1%	16.9%	15.1%	37.4%	76.5%
no electron ID	13.3%	17.1%	17.0%	15.2%	37.5%	76.5%
no mass	13.4%	16.9%	17.1%	15.4%	37.2%	76.5%
no d_z	13.5%	16.9%	16.9%	15.1%	37.6%	76.4%
no d_r	12.9%	17.0%	16.8%	15.2%	38.0%	76.2%
no `clusterE9E21`	13.2%	16.6%	17.1%	15.1%	38.0%	76.2%
no muon ID	13.0%	16.7%	17.1%	15.1%	38.1%	76.2%
no charge	10.8%	15.0%	16.4%	15.7%	42.2%	74.3%
only 4-momentum	7.0%	11.4%	14.5%	15.6%	51.5%	70.3%

Table 6.2: DSIT model performance on validation data for different ablation cases.

6.4 DSIT Model Performance

The performance of the DSIT model trained with the data described in section 6.2, the hyper parameters shown in section 6.3.1 and the input variables without the H_c vertex coordinates as motivated in section 6.3.2 get discussed in this section.

6.4.1 Performance Indicators

Table 6.3 shows the distribution of prediction errors per event, the accuracy calculated for all particles and per category, namely X, B_{sig} and background. Notably are that the correct prediction rate for background of 68.3% is worse than for X with 80.1% and for B_{sig} with 77.2%. The fact that only 36.8% of all events have 4 or more prediction errors is a promising result.

perfect	1E	2E	3E	≥ 4E	accuracy	X acc.	B_{sig} acc.	bg acc.
13.7%	17.2%	17.0%	15.3%	36.8%	76.8%	80.1%	77.2%	68.3%

Table 6.3: Distribution of prediction errors per event, accuracy for all particles and accuracies for the three categories X - B_{sig}- background (bg) on validation data. Based on the best achieved training on the data described in section 6.4.

Figure 6.2 shows the fraction of correctly predicted FSPs per event on the y-axis and the number of FSPs per event on the x-axis. Note that each column is normalized to 1, therefore the colored entries show the fraction of events in the respective number of FSPs bin. For all four modes the fraction of correctly assigned FSPs per event worsens for higher numbers of FSPs per event. However the effect is not as bad as one might expect. The degrading effect is more severe as the number of D^0 daughters increases. This is expected because the higher number of D^0 daughters increases the number of B_{sig} particles per event. This increase of the combinatorial complexity makes the DSIT model more prone

to errors in comparison to smaller numbers of B_{sig} FSPs.
Figure 6.3 shows the summed momenta of wrongly predicted particles per event. Each color shows the distribution for different numbers of wrongly predicted particles per event from one to four. The plot indicates that for most events particles with relatively high momentum get assigned correctly and usually the particles with less momentum are those who get wrongfully predicted. If one looks at the peaks of the different distributions, the typical wrongly predicted particle appears to have a momentum of $p \approx 120\,\text{MeV}$.

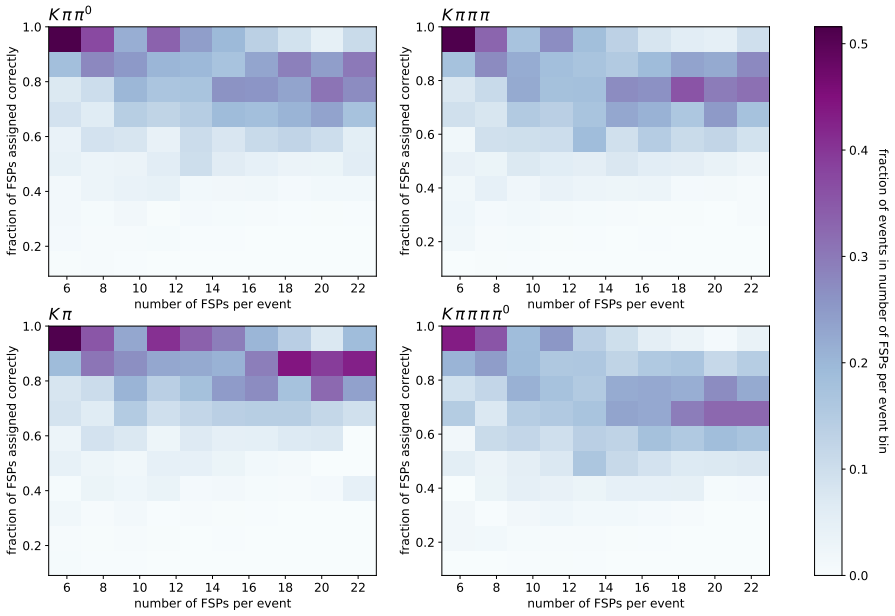


Figure 6.2: Fraction of FSPs per event predicted correctly by the DSIT model vs. number FSPs per event, normalized on each column of the x-xis (number of FSPs). The color indicates the fraction of events in number of FSPs per event bin. Displayed per D^0 mode.

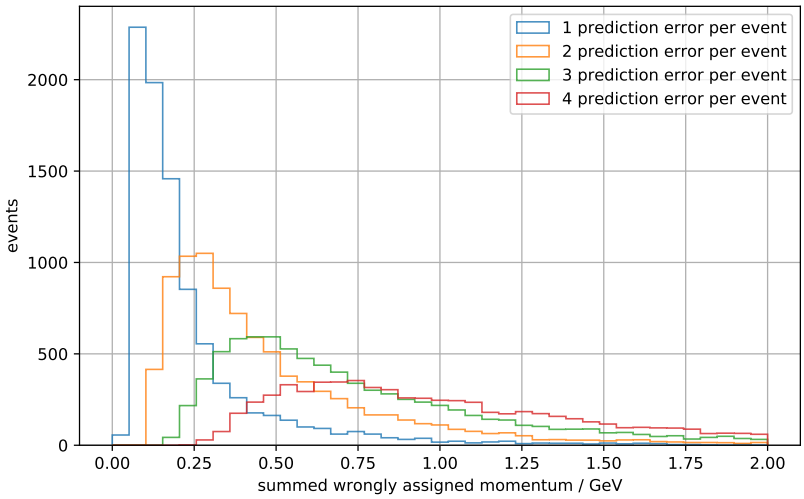


Figure 6.3: Summed momenta distributions of wrongly predicted particles per event, for different numbers of wrongly predicted particles per event.

6.4.2 Physics Variables directly from DSIT Model's Predictions

After the performance indicators described in the last section seemed promising, relevant physics variables got calculated based directly on the DSIT model's predictions for each FSP and the information of the already FEI reconstructed H_c.
Figure 6.4 shows the distribution of the M_{bc} of B_{tag}, as defined in equation (2.1). The distribution is shown for all events and for different numbers of prediction errors per event from zero to two. The peaks of the distributions are at 5.28 GeV and therefore in alignment with the B^0 meson mass of $m_B = 5.279\,\text{GeV}$ [3]. The m^2_{miss}, as defined in equation (2.2), distribution of the $\Upsilon(4S)$ reconstructed directly from the DSIT model's predictions is shown in figure 6.5, with an event selection of $M_{bc} \geq 5.0\,\text{GeV}$ for B_{tag}. The distribution is shown for all events and different numbers of prediction errors per event from zero to two. The mean and standard deviation are smaller for lower numbers of prediction errors per event. This dependency is depicted for a wider range of prediction errors per event, from zero to ten, in figure 6.6. Also for this wider range of prediction errors per event the same trend is visible. The higher the number of prediction errors per event the higher are the mean and standard deviation of the m^2_{miss} distribution.

The results look promising, but the decay reconstruction was not done exclusively. Therefore it is not possible to apply further selections on the reconstructed $\Upsilon(4S)$ candidates or their daughter particles. Because of this the explicit reconstruction based on the DSIT model's prediction will be performed in section 6.5.

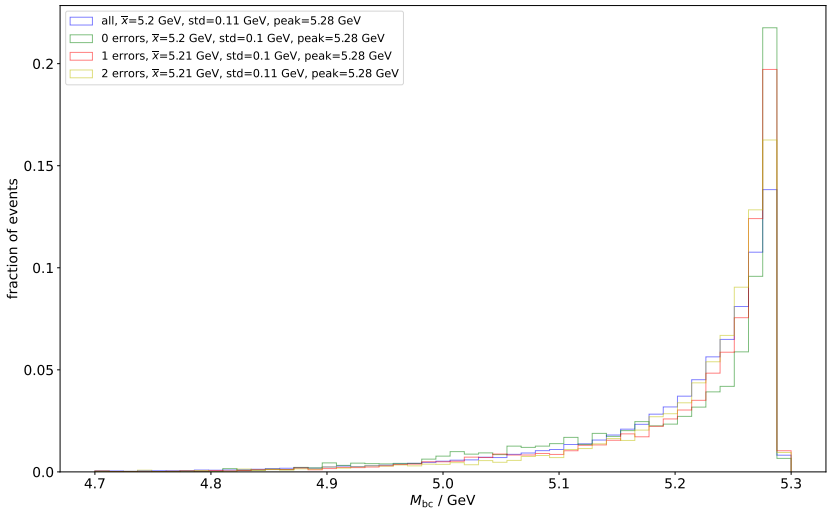


Figure 6.4: Distribution of M_{bc} of B_{tag}, as defined in equation (2.1), reconstructed directly from the DSIT model's predictions and the H_{c} reconstructed by the FEI. The distribution is shown for all events and different numbers of prediction errors per event from zero to two.

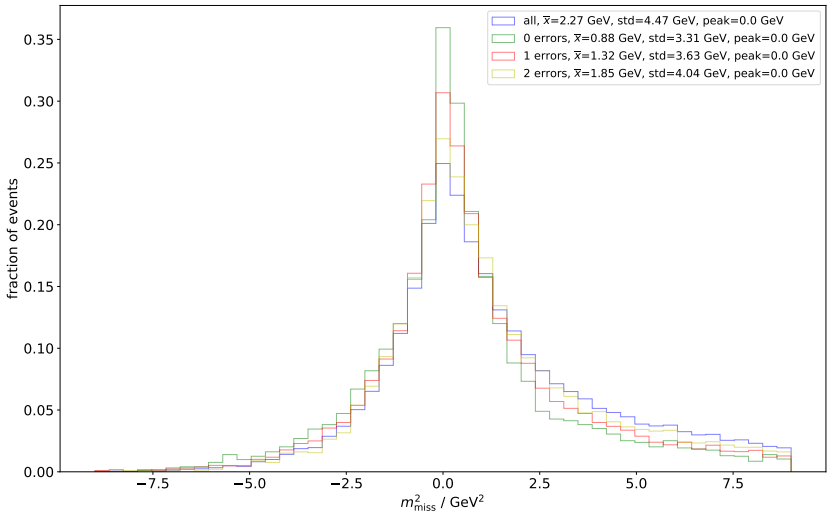


Figure 6.5: Distribution of m^2_{miss}, as defined in equation (2.2), reconstructed directly from the DSIT model's predictions and the H_{c} reconstructed by the FEI with a selection of $M_{\text{bc}} \geq 5.0\,\text{GeV}$ for B_{tag}. The distribution is shown for all events and different numbers of prediction errors per event from zero to two.

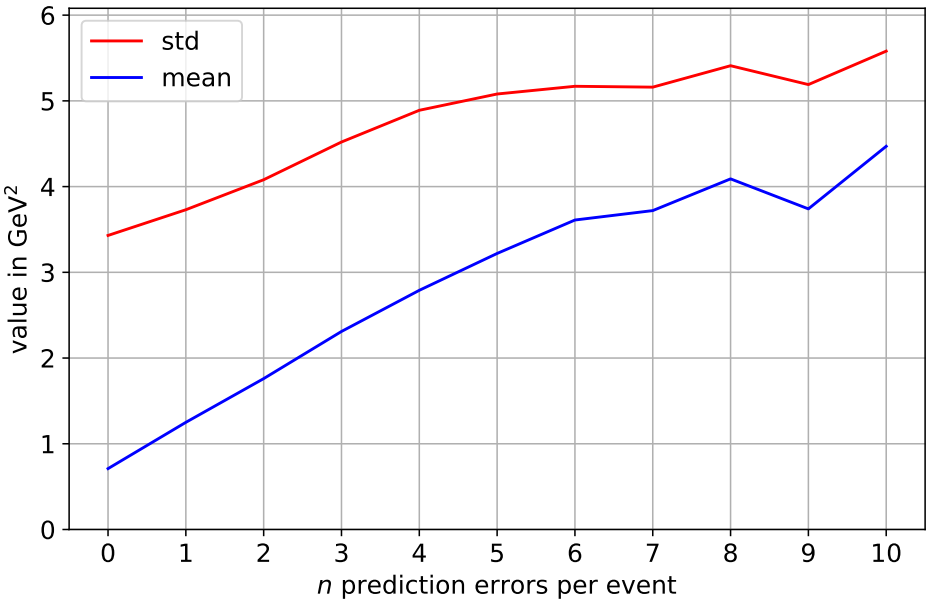


Figure 6.6: $n\cdot$ prediction errors dependance of std and mean in m^2_{miss} distribution in figure 6.5 ($n \in \mathbb{N}, 0 \leq n \leq 10$).

6.5 Explicit $\Upsilon(4S)$ Reconstruction based on DSIT Model's Predictions

The DSIT model gets applied on explicit reconstruction of the $\Upsilon(4S)$ meson. For this the H_c will be reconstructed by the FEI, the B_{sig} will be explicitly reconstructed by the DSIT model's predictions and the X will be taken inclusively also based on the DSIT model's predictions. The explicit reconstruction will make more reconstructed variables available, which then will be used to improve the reconstruction quality by being able to apply selections on these variables.

This explicit reconstruction will also be done under more realistic circumstances then before. No more generator level selections on $D^*(\to D^0\pi)\ell\nu$ events or the D^0 modes which are meant to be reconstructed are applied. The FEI reconstructed H_c is no longer required to be perfect or to have a B meson mother, but is instead selected with a best candidate selection. Note that the DSIT model trained with the data described in section 6.2, the hyper parameters shown in section 6.3.1 and the input variables without the H_c vertex coordinates as motivated in section 6.3.2 gets used in this section. This means the model used was trained only for $D^*(\to D^0\pi)\ell\nu$ events and the specific D^0 modes. Therefore the reconstruction performed also needs to discriminate between the $D^*(\to D^0\pi)\ell\nu$ events which are meant to be reconstructed and the other decay modes in the sample.

The reconstruction is based on $200\,\text{fb}^{-1}$ of $B\overline{B}$ Monte-Carlo events. The FSPs used for the reconstruction are selected with the `stdMostLikely` function like it was done to create the training data. The mass hypothesis decided by the `stdMostLikely` function will be retained throughout the reconstruction which is mostly important for the B_{sig} reconstruction.

6.5.1 Custom `basf2` Module

In order to explicitly reconstruct the $\Upsilon(4S)$ meson based on the DSIT model's predictions it is necessary to get the predictions during the `basf2` reconstruction process. For this a custom `basf2` module was created which applies the DSIT model. The module makes the predictions available for the further `basf2` reconstruction by saving it for every particle in a integer variable called `extraInfo(NN_prediction)`. Here 0 stands for a predicted background particle, 1 for a X particle and 2 for a B_{sig} descendant.
The custom module requires three arguments to be initialized before it can be added to the `basf2` path. First is a list containing all particle lists the DSIT model shall be applied to. Note that these lists can still, but do not have to, contain the FSPs which are daughters of the H_c candidate selected beforehand. In both cases the custom module expects the particles to have an assigned variable `extraInfo(Hc_used)` which indicates if a particle is a H_c descendant binary. The second argument for the custom module is a list containing the input variable names the DSIT model will use. This variable list has to have the same order the model's input variables had during training. The last argument is the DSIT model itself. For this the model gets initialized with the wanted hyper parameters and the weights get loaded from a training which was already performed.

6.5.2 H_c Best Candidate Selection

As already mentioned the H_c is no longer required to be perfectly reconstructed or to have a B mother. Instead a best candidate is selected based on the FEI `SignalProbability`. For the `SignalProbability` of the H_c a minimal value of 0.001 was used exactly like the method introduced in section 6.1. For this `SignalProbability` value and two more strict minimal values

the effectiveness of the H_c best candidate selection was investigated. For this the fraction of events which pass the corresponding `SignalProbability` selection were determined. Additionally the fraction of these events with any perfect H_c candidate (perfectly reconstructed and B mother) was determined. Then the best candidate selection was performed by selecting the H_c candidate with the highest `SignalProbability` value. After this selection the fraction of events where a perfect H_c candidate was selected were calculated.
Table 6.4 shows the results. The higher `SignalProbability` selections of 0.01 and 0.1 reject a lot of events, by only letting 80% respectively 38% of events pass. The fraction of events with a perfect H_c after the `SignalProbability` selection increases notably. For all three selections the fraction of perfect H_c events after the H_c best candidate selection seems a reasonable trade off for the simplicity of the performed best candidate selection.

It was decided to use the `SignalProbability` selection of 0.001, in order no to loose too many events due to the best candidate selection.

`SignalProbability` minimal value	0.001	0.01	0.1
fraction of events which pass the selection	100%	80%	38%
fraction of events with perfect H_c for at least one candidate	17%	20%	34%
fraction of perfect H_c events after H_c BCS	12%	15%	29%

Table 6.4: Best candidate selection for H_c based on the FEI `SignalProbability`.

6.5.3 Best Candidate Selection for $\Upsilon(4S)$ Candidates

The explicit $\Upsilon(4S)$ reconstruction with `basf2` often produces multiple $\Upsilon(4S)$ candidates per event. Therefore a best candidate selection for the $\Upsilon(4S)$ needs to be performed. The best candidate selection was based on the m^2_{miss}, as defined in equation (2.2), distribution, as this distribution is meant to peak at zero and this attribute was used to rate the different techniques tested.
Three different best candidate selection approaches were investigated. The first approach was to select the candidate with the minimal absolute value of m^2_{miss} of all candidates of the respective event. The second approach was to take the candidate with the smallest value of E_{extra} which was presented in [21]. E_{extra} is defined as the summed energy of photons with $E_\gamma \geq 50\,\text{MeV}$ which were not used to reconstruct the $\Upsilon(4S)$ candidate. The third approach was to perform a random candidate selection. The m^2_{miss} distributions for all three approaches are shown in figure 6.7. The best candidate selection based on the minimal absolute value of m^2_{miss} gives the best peak. Therefore this best candidate selection approach was selected.

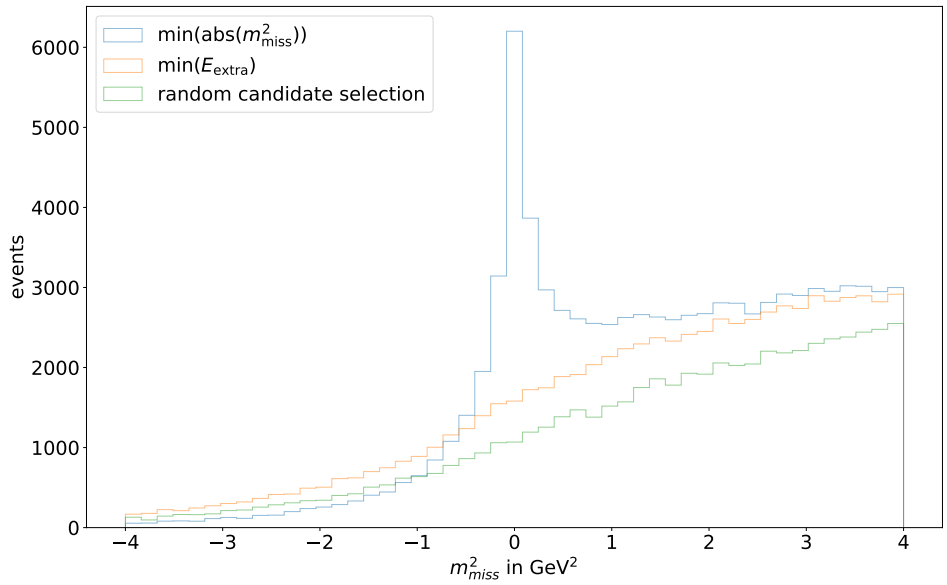


Figure 6.7: m^2_{miss}, as defined in equation (2.2), distribution from explicit $\Upsilon(4S)$ reconstruction in `basf2` based on the DSIT model's predictions and the H_c reconstructed by the FEI. Shown are the best $\Upsilon(4S)$ candidates for each of the three different tested best candidate selection options.

6.5.4 Results

The events where a $\Upsilon(4S)$ candidate was found based on the DSIT model's predictions were categorized in background and truth matched events. Note that the definition for truth matched here is quite loose. It only requires the event to be a real $D^*(\to D^0\pi)\ell\nu$ event and the lepton from the $D^*\ell\nu$ decay to be correctly reconstructed. A more strict definition of truth matched did not let many events pass, therefore the more loose definition was chosen.
The reconstruction was not really successful. Which can exemplary be shown for the mass distribution for the reconstructed D^0 meson for the four different reconstructed decay channels shown in figure 6.8. The background events in orange and the truth matched events in blue are stacked on one another in the plots. Background and truth matched events both peak at the same value. Except for the $D^0 \to K\pi\pi\pi\pi^0$ mode, which also happens to have by far the lowest statistics, all peaks are far from the D^0 mass $m_{D^0} = 1.865\,\text{GeV}$ [3] which is indicated with the red vertical line in the figure. The m^2_{miss}, as defined in equation (2.2), distribution of the $\Upsilon(4S)$ meson selected best candidates is shown in figure 6.9. The distribution is shown once with no selections and with different applied selections which are listed with their respective boundaries in table 6.5. The single selections applied are the B_{tag} M_{bc}, as defined in equation (2.1), B_{tag} ΔE, as defined in equation (2.3), $p_{\text{t, lepton}}$ and the p_{t,D^*} selection. Also shown are the distributions if all those selections are applied at once and this in addition with the D^0 mass or the D^* mass selection. Again the background events in orange and the truth matched events in blue are stacked on one another in the plots. For no selections and all different selections applied the m^2_{miss} distribution has a much bigger tail in the positive range. Especially the

$p_{\text{t, lepton}}$ selection is quite effective by reducing the background events. Applying all selections at once and either the D^0 mass or the D^* mass selection additionally reduces the background notably and creates a much a more symmetrical distribution. But this comes with the cost of reducing the statistics significantly. This issue is depicted in more detail in figure 6.10. The bars shows efficiency and purity for all different selections from table 6.5 which got applied subsequently from left to right. The efficiency is defined by the number of events which were tagged divided by the number of expected events n_{expected}. In the $200\,\text{fb}^{-1}$ of data are $N_{\text{all}} = 101.9 \cdot 10^6$ $B\overline{B}$ events. Of those $n_{\text{expected}} = N_{\text{all}} \cdot \epsilon_{D^*\ell\nu} \cdot \epsilon_{D^*\to D^0\pi} \cdot \epsilon_{D^0\text{modes}} = 2.08 \cdot 10^6$ are the events the reconstruction should tag. With $\epsilon_{D^*\ell\nu} = 9.9\%$ being the branching fraction of $B \to D^*\ell\nu$, $\epsilon_{D^*\to D^0\pi} = 67.7\%$ being the branching fraction of $D^* \to D^0\pi$ and $\epsilon_{D^0\text{modes}} = 30.49\%$ being the combined branching fraction of the reconstructed D^0 modes. The purity is defined as the number of truth matched events, as described above, divided by the number of reconstructed events. Figure 6.10 shows that the efficiency is getting extremely small with a value below 10^{-4} once all selections got applied. At the same time the purity is increasing with every added selection, but with a highest value of only 47%.

Compared to the efficiency of 2.75% and the purity of 70% of the already existing semi-inclusive tagging method described in section 6.1.1, one has to conclude that the DSIT model based approach performs much worse than the already existing one. The purity definition used for this thesis is more strict than for the already existing semi-inclusive tagging method, which does not require the truth matched lepton.

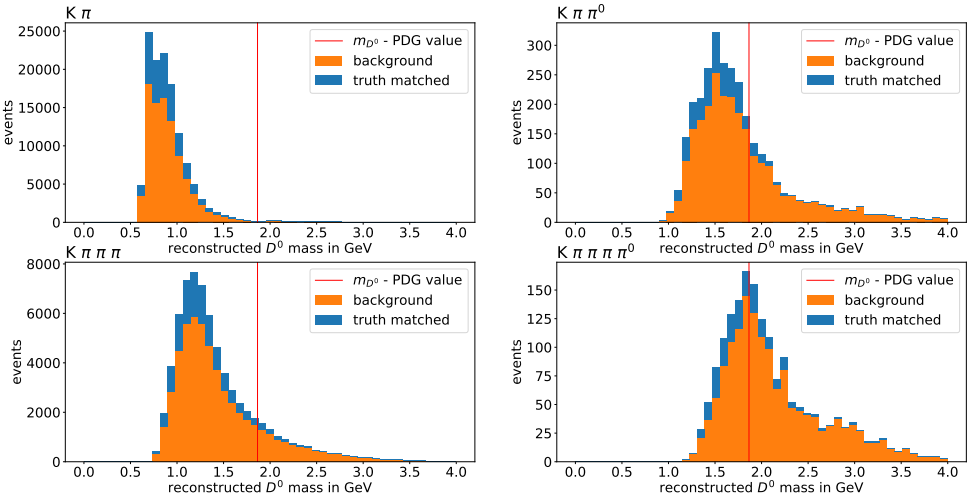


Figure 6.8: D^0 mass for each reconstructed D^0 decay mode from explicit $\Upsilon(4S)$ reconstruction in `basf2` based on the DSIT model's predictions and the H_c reconstructed by the FEI. The (loosely) truth matched (blue) and background (orange) events are stacked on each other, the red vertical line shows the PDG value of the D^0 mass.

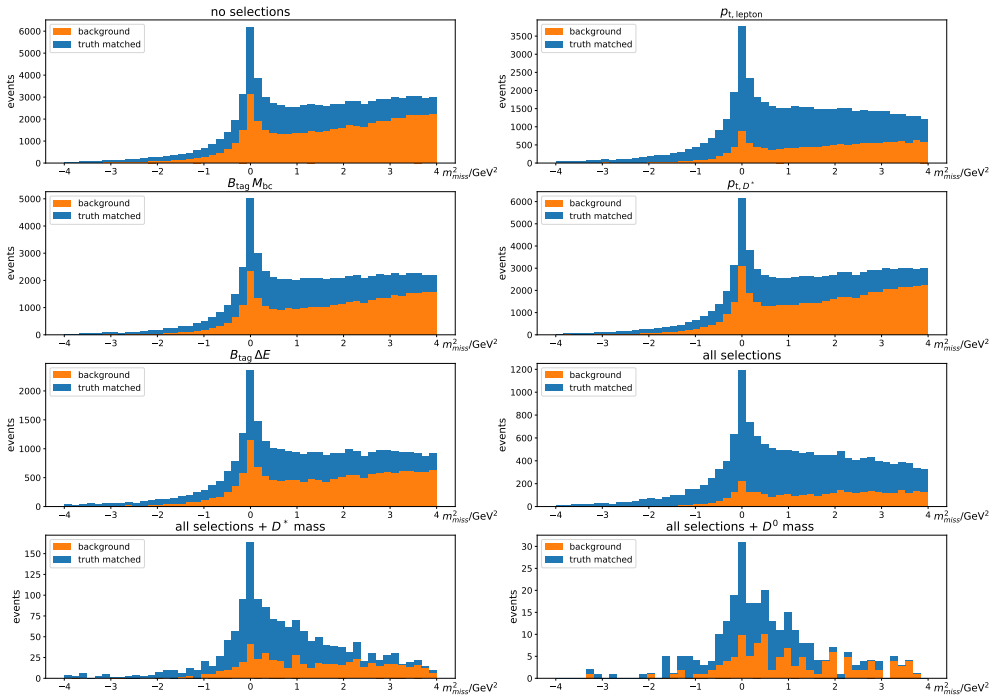


Figure 6.9: m^2_{miss}, as defined in equation (2.2), distribution with different applied selections, which are explained in table 6.5, from explicit $\Upsilon(4S)$ reconstruction in `basf2` based on the DSIT model's predictions and the H_c reconstructed by the FEI. The (loosely) truth matched (blue) and background (orange) events are stacked on each other.

selection name	lower limit	upper limit
B_{tag} M_{bc}	5.0 GeV	-
B_{tag} ΔE	-2.0 GeV	-1.0 GeV
`foxWolframR2`	-	0.3
p_{t,D^*}	-	2.4 GeV
$p_{\text{t, lepton}}$	0.7 GeV	-
D^0 mass	1.8 GeV	1.95 GeV
D^* mass	1.8 GeV	2.2 GeV
hadronic B_{tag}	1	-

Table 6.5: Selections used in figure 6.9 and 6.10. Hadronic B_{tag} indicates binary whether the B_{tag} is fully hadronic or not. `foxWolframR2` is the ratio of the second and zeroth Fox Wolfram moments, see [8] for more details. See section 2.5 for M_{bc} and ΔE definitions.

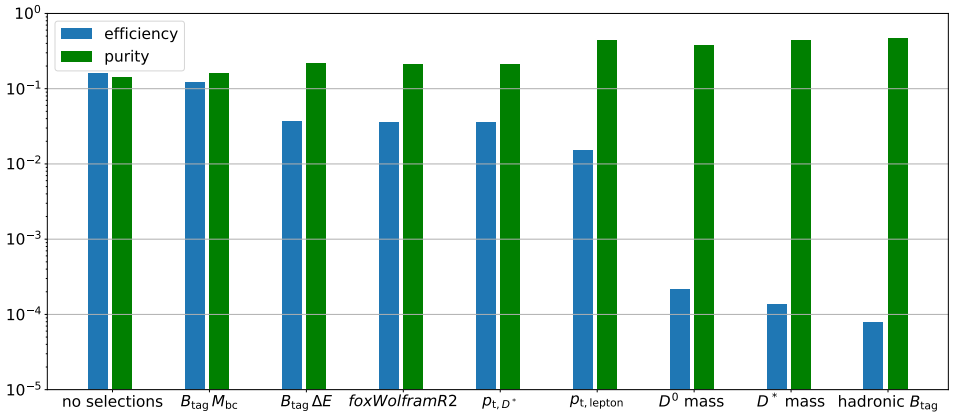


Figure 6.10: Efficiency and purity from explicit $\Upsilon(4S)$ reconstruction in `basf2` based on the DSIT model's predictions and the H_c reconstructed by the FEI for subsequently applied selections from left to right, which are explained in table 6.5. The purity is defined for truth matched $D^*(\rightarrow D^0\pi)\ell\nu$ events which also have a correctly reconstructed lepton in $B \rightarrow D^*\ell\nu$ decay.

6.5.5 Possible Performance Restrictions of the DSIT Model in Comparison to the Already Existing Method

Because the already existing semi-inclusive tagging method described in section 6.1.1 had notably better purity and efficiency as shown in section 6.5.4, possible reasons as wells as possible improvements for the deep semi-inclusive tagging approach are discussed here.

A big advancement of the already existing semi-inclusive tagging method is a higher flexibility regarding the H_c. Multiple H_c candidates get combined to multiple $\Upsilon(4S)$ candidates which then get selected with adjustable selections. The DSIT model approach on the other hand, at least in its current form, needs a decision on which H_c candidate to choose even before the DSIT model can predict the particles' categories.

This flexibility of a common `basf2` reconstruction also gets lost if the DSIT model is used for example in π^0 reconstruction. The DSIT model's approach does not allow for multiple π^0 candidates but only allows for a π^0 to be reconstructed if both of its photons got predicted correctly.

The DSIT model is designed to get each FSP it should predict fed into the network once. This leads to the necessity to decide on the mass hypothesis for each FSP in an early stage of the reconstruction process. This might have lead to performance restrictions for the B_{sig} reconstruction, where the decay is exclusively reconstructed and therefore the particle type matters. Especially the bad reconstruction results for the D^0 as shown in figure 6.8 support this statement.

Although apparent, a higher accuracy of the DSIT model would also increase the reconstruction quality for the deep semi-inclusive $\Upsilon(4S)$ reconstruction. Maybe additional input variables or an improved neural network design might improve the accuracy. With the current level of accuracy, the higher flexibility of the already existing semi-inclusive tagging method outperforms the DSIT model strongly.

CHAPTER 7

Rest Of Event Clean Up

As shown in chapter 6 the deep semi-inclusive tagging had a bad performance for the full $\Upsilon(4S)$ reconstruction. However the method netherless showed potential by predicting the category of FSPs. Therefore a simpler problem is investigated in this chapter. Goal is to train the DSIT model in order to clean up the rest of event FSPs once the B_{sig} is reconstructed. After the B_{sig} is reconstructed the next step is to reconstruct the B_{tag} inclusively. In order to help doing this the DSIT model shall categorize the FSPs in a meaningful way.

7.1 Training Data Creation

7.1.1 Best Candidate Selection of B_{sig}

The B_{sig} reconstruction usually produces multiple candidates. Therefore a best candidate selection is necessary and was taken from section 3.1 of this paper [22]. For every B_{sig} candidate a vertex fit gets performed, candidates which do not have a successful fit get deleted. If multiple candidates survive, the candidate with the smallest χ^2 corresponding to the fit gets chosen.

7.1.2 Particle and Event based Selections and Separation Cases

For this task only $B_{\text{sig}} \to D^*(\to D^0\pi)\ell\nu$ events and the specific D^0 modes already used in section 6.2 are used and get selected based on generator level information. The B_{sig} gets reconstructed then and if necessary the best candidate selection is performed as described in section 7.1.1. On all remaining FSPs selections are applied which shall already reduce the background particles. Those selections are physics motivated and taken from another analysis and are listed in the appendix in section A.3. Three different separation cases were then created from these FSPs, which are not already used in the chosen B_{sig} candidate, to test the DSIT model for different scenarios:

1. B_{tag}- combined background (everything not B_{tag} related)
2. all B related FSPs - background
3. B_{sig} (FSPs not already used for the B_{sig} candidate) - B_{tag}- background

7.2 Results

7.2.1 Hyper Parameters

Different hyper parameters and their combinations were tested until the best combination was found. n_{hid} and n_{blocks} proved to be the most crucial hyper parameters, who already affected the DSIT model's performance significant if only small changes were made. The best values found were setting n_{hid} to 256, n_{blocks} to 4. In contrast the performance was only mildly affected by changes of drop rate and batch size. For this reason the drop rate was set to 0.2 and the batch size to 64.

7.2.2 DSIT Model Performance

The distribution of prediction errors per event, the overall accuracy and the respective category wise accuracies for the three tested separation cases are shown in table 7.1. Notably is that the B - background separation has the highest accuracy, while the B_{sig}- B_{tag}- background separation has the lowest. This is expected because the latter has an additional category to be predicted. Also the B - background separation has a higher accuracy for predicting background than the B_{tag}- combined background separation. Putting all B related particles in one category seems to be easier for the network than putting B_{sig} and background in a combined non B_{tag} related category.

separation case	perfect	1E	2E	3E	≥ 4E	accur.	bg	B_{sig}	B_{tag}	B
B_{tag}- combined bg	5.2%	11.9%	17.1%	18.5%	47.2%	75.3%	60.7%	-	81.6%	-
B - bg	7.8%	18.2%	23.3%	20.5%	30.2%	81.5%	81.3%	-	-	81.6%
B_{sig}- B_{tag}- bg	2.2%	6.7%	11.8%	15.1%	64.2%	68.4%	77.4%	46.6%	72.7%	-

Table 7.1: Performance for the three different separation cases the DSIT model got trained for the rest of event clean up. Accur. stands for accuracy. Background (bg), B_{sig}, B_{tag} and B stand for the respective category wise accuracies.

Table 7.2 shows the number of FSPs N in the complete data set, the fraction of those particles predicted as background by the DSIT model and the corresponding amount $N_{\text{predicted as bg}}$ for all three separation cases and every category separately. The number $N_{\text{predicted as bg}}$ for the first category of each separation case is therefore the number of true positives for predicting the background. The number $N_{\text{predicted as bg}}$ for the remaining category/categories are therefore the false positives for predicting the background. For the B - background and the B_{sig}- B_{tag}- background separation cases the false positives outweigh the true positives, which means more particles are wrongfully predicted to be background than true background got predicted correctly. For the B_{tag}- combined background separation case the number of true positives of $220 \cdot 10^3$ outweighs the number of falsely predicted background of $137 \cdot 10^3$ almost by a factor of two.

separation case	category	N	fraction predicted as bg	$N_{\text{predicted as bg}}$
B_{tag}- combined background	combined bg	$362 \cdot 10^3$	60.7%	$220 \cdot 10^3$
	B_{tag}	$744 \cdot 10^3$	18.4%	$137 \cdot 10^3$
B - bg	bg	$147 \cdot 10^3$	81.3%	$120 \cdot 10^3$
	B	$960 \cdot 10^3$	18.4%	$177 \cdot 10^3$
B_{sig}- B_{tag}- bg	bg	$147 \cdot 10^3$	77.4%	$114 \cdot 10^3$
	B_{tag}	$744 \cdot 10^3$	15.2%	$113 \cdot 10^3$
	B_{sig}	$215 \cdot 10^3$	15.7%	$34 \cdot 10^3$

Table 7.2: Amount in each category N and fraction and amount $N_{\text{predicted as bg}}$ of particles predicted as background (bg) for each category of the three tested separation cases.

7.2.3 Physics Variables from DSIT Predictions for the B_{tag}- combined background Separation Case

The B_{tag}- combined background separation case got further investigated. This was done because for this separation case an clear application was the easiest, because B_{tag} was already clearly predicted by the DSIT model and the performace was better than for the B_{sig}- B_{tag}- background separation. Also the B_{tag}- combined background separation case was the only case where the true positive background predictions outweighed the false positives. Relevant physics variables got calculated based directly on the DSIT model's predictions for each FSP and the information of the already reconstructed B_{sig}. Figure 7.1 shows the distribution of the M_{bc} of B_{tag}, as defined in equation (2.1), calculated based on the DSIT model's predictions. The distribution is shown for all events and for different numbers of prediction errors per event from zero to two. The peaks of the distributions are at 5.27 GeV and 5.28 GeV and therefore in alignment with the B^0 meson mass of $m_B = 5.279\,\text{GeV}$ [3]. The m^2_{miss}, as defined in equation (2.2), distribution of the $\Upsilon(4S)$ reconstructed directly from the DSIT model's predictions is shown in figure 7.2, with an event selection of $M_{\text{bc}} \geq 5.0\,\text{GeV}$ for B_{tag}. The distribution is shown for all events and different numbers of prediction errors per event from zero to two. Figure 7.3 shows the summed momenta of wrongly predicted particles per event. Each color shows the distribution for different numbers of wrongly predicted particles per event from one to four. The plot indicates that for most events particles with relatively high momentum get assigned correctly and usually the particles with less momentum are those who get wrongfully predicted.

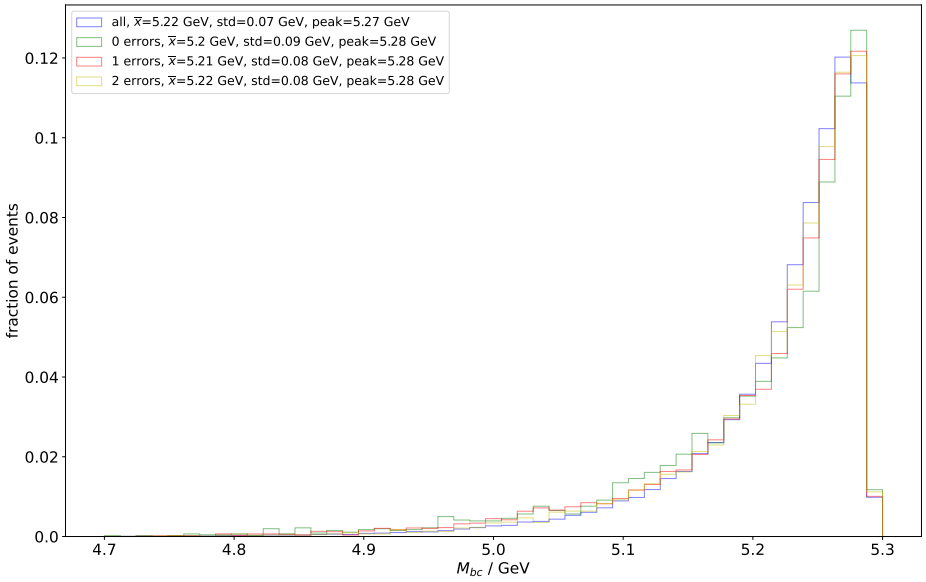


Figure 7.1: M_{bc} of B_{tag}, as defined in equation (2.1), distribution directly from the DSIT model's predictions for B_{tag}. The distribution is shown for all events and different numbers of prediction errors per event from zero to two.

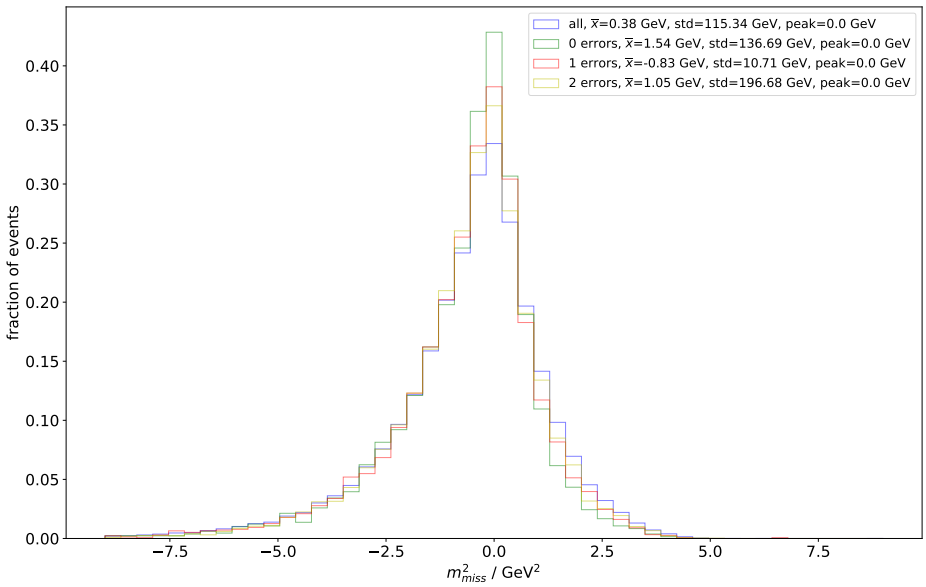


Figure 7.2: m^2_{miss}, as defined in equation (2.2), distribution directly from the DSIT model's predictions for B_{tag} and the B_{sig} reconstructed with `basf2` with an event selection of $M_{\text{bc}} \geq 5.0\,\text{GeV}$ for B_{tag}. The distribution is shown for all events and different numbers of prediction errors per event from zero to two.

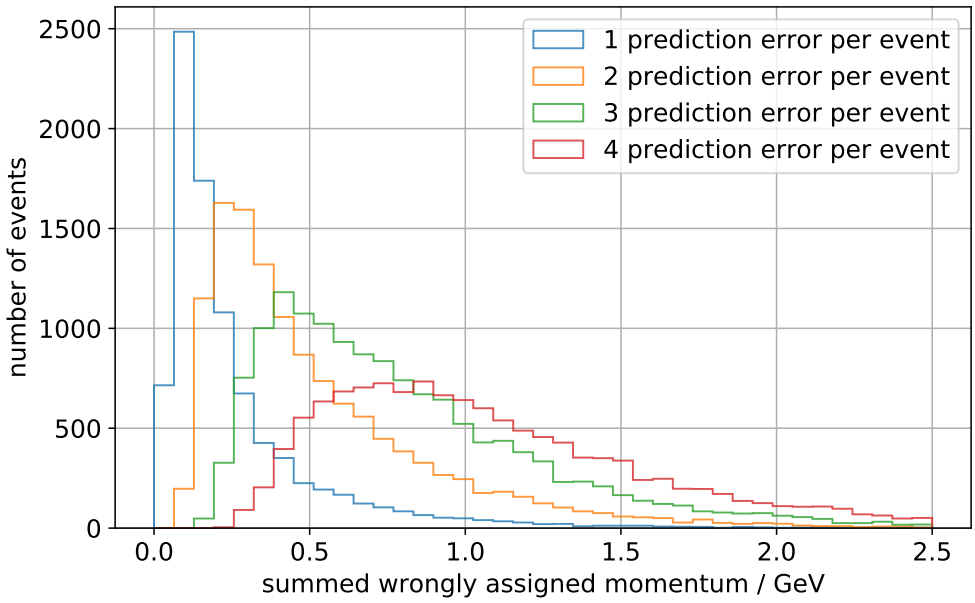


Figure 7.3: Summed momenta distributions of wrongly predicted particles per event, for different numbers of wrongly predicted particles per event.

CHAPTER 8

Conclusion

This thesis covered three major topics.

Deep Semi-Inclusive Tagging Toy Model

Data for the toy model was self created by simulating phase space events with `PhaseSpace` and `basf2`. Each tested data set consisted of six different decay modes. Different separation cases were tested. The toy model proved that the DSIT model can learn the patterns of $\Upsilon(4S)$ decays in order to categorize the final state particles based on their ancestor particles. It was shown that the performance worsens if more categories are required to be predicted by the network.

Deep Semi-Inclusive Tagging Applied on Official Monte-Carlo Data

For signal B mesons from $B \to D^*(\to D^0\pi)\ell\nu$ decays and generic tag B mesons the X - B_{sig}- background separation was trained on official Monte-Carlo data. Event and final state particle based selections necessary for good DSIT model training results were developed. The accuracy of the DSIT model reached up to 76.8%. The best combination of input variables for the DSIT model was found in ablation studies. For the explicit reconstruction of the $\Upsilon(4S)$ a best candidate selection for the FEI reconstructed H_c was developed. The explicit reconstruction of the $\Upsilon(4S)$ with the deep semi-inclusive tagging was found to be non compatible. The purity and efficiency of the deep semi-inclusive tagging were significantly lower than for a presented already existing semi-inclusive tagging method. Possible reasons for this or further improvements of the DSIT model's performance were discussed.

Rest Of Event Clean Up

Three different separation cases were created and tested for the rest of event clean up for events where a signal B meson was already reconstructed. This was performed for $B \to D^*(\to D^0\pi)\ell\nu$ decays. Accuracies between 68% and 82% were reached by the trained DSIT model. The B_{tag}- combined background separation was then chosen to calculate physics variables based on the DSIT model's predictions.

APPENDIX A

Useful Information

A.1 Variables used by basf2

See [10] for more details.

thetaInCDCAcceptance: binary indicates if particle is in CDC θ acceptance of $17^\circ < \theta < 150^\circ$

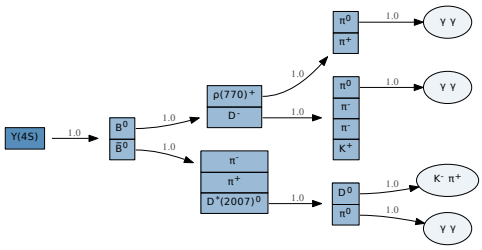
nCDCHits: the number of CDC hits associated to the particle

clusterReg: integer indicating the cluster region in the ECL

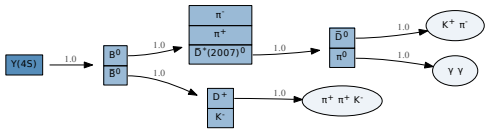
clusterNHits: sum of weights of the crystals the ECL cluster was measured

clusterTiming: time of the cluster, measured starting at the time of the event

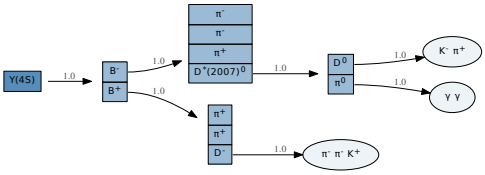
A.2 Used Decay Trees for Simulated Data



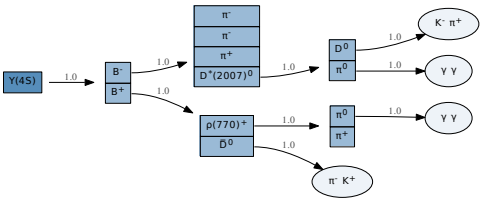
(a) Mode 0: B_{sig}: $\overline{B}^0$, H_{c} :D^-



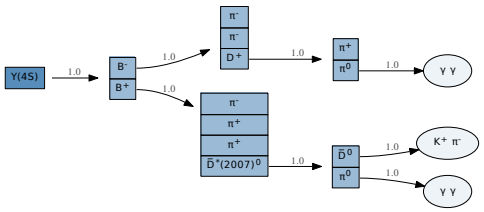
(b) Mode 1: B_{sig}: B^0, H_{c}: D^+



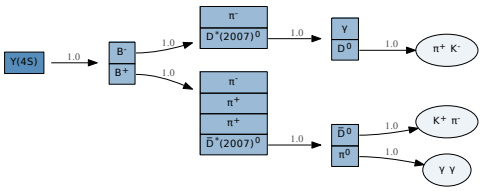
(c) Mode 2: B_{sig}: B^-, H_{c}: D^-



(d) Mode 3: B_{sig}: B^-, H_{c}: $\overline{D}^0$

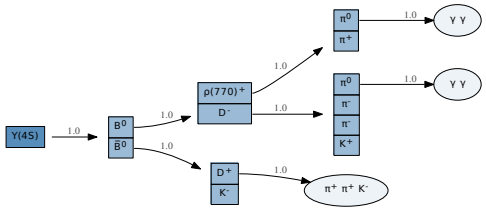


(e) Mode 4: B_{sig}: B^+, H_{c}: D^+

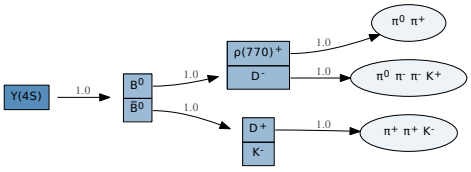


(f) Mode 5: B_{sig}: B^+, H_{c}: D^{*0}

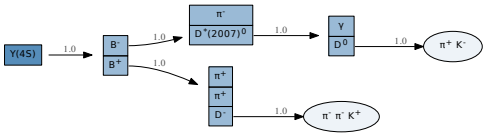
Figure A.1: Modes which were simulated with `basf2` for X - B_{sig}- background, X - H_{c}- background and X - H_{c}- B_{sig}- background separation and were simulated with `PhaseSpace` for X - H_{c}- B_{sig}- background separation



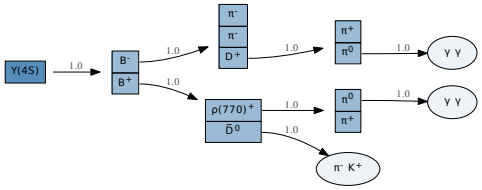
(a) Mode 0: B_{sig}: $\overline{B}^0$, H_{c} :D^-



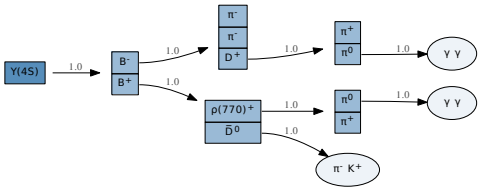
(b) Mode 1: B_{sig}: B^0, H_{c}: D^+



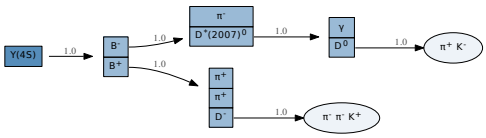
(c) Mode 2: B_{sig}: B^-, H_{c}: D^-



(d) Mode 3: B_{sig}: B^-, H_{c}: $\overline{D}^0$



(e) Mode 4: B_{sig}: B^+, H_{c}: D^+



(f) Mode 5: B_{sig}: B^+, H_{c}: D^{*0}

Figure A.2: Modes which were simulated with `PhaseSpace` for X - B_{sig}- background and X - H_{c}- background separation

A.3 Rest Of Event Clean Up Final State Particle Selections

The final state particle selections used for the data production in chapter 7 are listed here. The variables and their meanings get explained in section 2.5 and A.1.
Tracks are subject to these selections:

- $p_{\text{t}} > 0.05\,\text{GeV}$
- `thetaInCDCAcceptance` == 1
- $|d_{\text{z}}| < 3.0$
- $|d_{\text{r}}| < 1.0$
- $E < 5.5\,\text{GeV}$

Photons are subject to these selections:

- (`clusterReg`==1 and $p_t > 0.03\,\text{GeV}$) or (`clusterReg`==2 and $p_t > 0.04\,\text{GeV}$) or (`clusterReg`==3 and $p_t > 0.06\,\text{GeV}$)
- `thetaInCDCAcceptance` == 1
- $E < 5.5\,\text{GeV}$
- `clusterNHits` > 1.5
- |`clusterTiming`| < 200

Bibliography

[1] J. Hewett, *The Standard model and why we believe it*, arXiv preprint hep-ph/9810316 (1998) (cit. on p. 1).

[2] M. Kobayashi and T. Maskawa, *CP-violation in the renormalizable theory of weak interaction*, Progress of theoretical physics **49** (1973) 652 (cit. on p. 1).

[3] P. Zyla et al., *Review of Particle Physics*, PTEP **2020** (2020) 083C01 (cit. on pp. 3, 17, 31, 36, 43).

[4] I. Heredia de la Cruz, *The Belle II experiment: fundamental physics at the flavor frontier*, Journal of Physics: Conference Series **761** (2016) (cit. on p. 3).

[5] T. A. et al., *Belle II Technical Design Report*, 2010, arXiv: 1011.0352 [physics.ins-det] (cit. on p. 4).

[6] Recent Access: 03.03.2022, URL: https://www.facebook.com/belle2collab/photos/cross-section-of-the-belle-ii-detectorcredit-kek/2087517408199161/ (cit. on p. 4).

[7] Recent Access: 07.03.2022, URL: https://software.belle2.org/sphinx/light-2110-tartarus/analysis/doc/Variables.html?highlight=clusterreg#variable-clusterE9E21 (cit. on p. 7).

[8] B. I. Collaboration et al., *The Belle II physics book*, Progress of Theoretical and Experimental Physics **2019** (2019) ARTN (cit. on pp. 7, 17, 38).

[9] G. Mönig, *Development of a Semi-Inclusive Tagging Algorithm and Implementation of a Continuum Suppression in the Full Event Interpretation for the Belle II Experiment*, MA thesis: Bonn, University Bonn, 2021 (cit. on pp. 7, 25).

[10] The Belle II Collaboration, *Belle II Analysis Software Framework (basf2)*, URL: https://github.com/belle2/basf2 (cit. on pp. 7, 49).

[11] C. C. Aggarwal et al., *Neural networks and deep learning*, Springer **10** (2018) 978 (cit. on p. 9).

[12] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*, vol. 4, 4, Springer, 2006 (cit. on p. 9).

[13] R. Hecht-Nielsen, "Theory of the backpropagation neural network", *Neural networks for perception*, Elsevier, 1992 65 (cit. on p. 10).

[14] Recent Access: 11.03.2022, URL: https://openclipart.org/detail/290666/neural-net-graph (cit. on p. 10).

[15] J. Shlomi, P. Battaglia and J.-R. Vlimant, *Graph neural networks in particle physics*, Machine Learning: Science and Technology **2** (2020) 021001 (cit. on p. 10).

[16] T. Kipf, E. Fetaya, K.-C. Wang, M. Welling and R. Zemel,
"Neural relational inference for interacting systems",
International Conference on Machine Learning, PMLR, 2018 2688 (cit. on p. 11).

[17] I. Tsaklidis, P. Goldenzweig, I. Ripp-Baudot, J. Kahn and G. Dujany,
Demonstrating learned particle decay reconstruction using Graph Neural Networks at Belle II,
MA thesis: Strasbourg, Universitè de Strasbourg, 2020 (cit. on p. 11).

[18] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library",
Advances in Neural Information Processing Systems 32, ed. by H. Wallach et al.,
Curran Associates, Inc., 2019 8024, URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
(cit. on p. 11).

[19] T. Keck,
Machine learning algorithms for the Belle II experiment and their validation on Belle data,
PhD thesis: Karlsruhe, Karlsruher Institut für Technologie (KIT), 2017 (cit. on p. 13).

[20] A. Puig and J. Eschle, *phasespace: n-body phase space generation in Python*,
Journal of Open Source Software (2019), URL: https://doi.org/10.21105/joss.01570
(cit. on p. 17).

[21] J. Lees et al., *Measurement of an excess of B*
$\to$ *D (*)* τ*- *ν
τ *decays and implications for charged Higgs bosons*, Physical Review D **88** (2013) 072012
(cit. on p. 35).

[22] F. Abudinén et al., *Studies of the semileptonic* $\bar{B}^0 \to D^{*+}\ell^-\bar{\nu}_\ell$ *and* $B^- \to D^0\ell^-\bar{\nu}_\ell$ *decay processes with 34.6* fb^{-1} *of Belle II data*, (2020), arXiv: 2008.07198 [hep-ex]
(cit. on p. 41).

List of Figures

List of Tables