Reconstruction of ttH final states using advanced machine learning techniques

Abschlussarbeit im Rahmen des Studienganges

Mathematik (M.Sc.)

an der Universität Göttingen

vorgelegt am: 1.12.2025

von: Jannis Rowold aus Lüneburg

1. Gutachter: Prof. Dr. Quadt

2. Gutachter: Prof. Dr. Wald

Universität Göttingen

Fakultät für Mathematik und Informatik

II.Physik-UniGö-MSc-2025/05

Abstract

This thesis deploys a graph attention network to reconstruct the $t\bar{t}(H \to b\bar{b})$ process. Simulations of the LHC Run 2 and a simulated ATLAS detector are used to obtain data. Reconstruction involves predicting the node classes that correspond to jet flavour and the edge classes that correspond to edges between nodes originating from the same parent particle. These predictions are further refined using a logistic regression model to determine the final candidates for each edge class. Overall, this approach reconstructed 17.4% of all events completely correctly.

Contents

1 Introduction

The Higgs boson, discovered in 2012 [1, 2], has been extensively studied to test the predictions of the Standard Model [3–12]. Theories can never be proven and must be confronted with facts, i.e. measurements made in experiments. Due to its strong connection with the top quark, the reconstruction of $t\bar{t}H$ events is particularly important for investigating new physics. However, the $t\bar{t}H$ production process is relatively rare, which increases the difficulty of observing and reconstructing these events.

Given the low rate of $t\bar{t}H$ production, the focus is on the Higgs bosons most probable decay mode, the decay into a bottom and anti-bottom quark pair. This leads to the specific final state, $t\bar{t}(H \rightarrow b\bar{b})$, which is the main target of this study. The process involves several distinct components, a Higgs boson decaying into a $b\bar{b}$ pair, and a top quark pair where one top quark decays leptonically (producing a b quark, a charged lepton, and a neutrino), while the other decays hadronically (producing a b quark and two other quarks, excluding b and t).

The detector cannot directly measure neutrinos, nor can it distinguish between particles and antiparticles. This limits the available information and makes the kinematic reconstruction of the original event structure challenging. Existing approaches, such as SpaNet [13], have attempted to address this using transformer-based methods.

This thesis presents an alternative approach using graph neural networks (GNNs) to reconstruct the $t\bar{t}(H \rightarrow b\bar{b})$ process. The main goal of the GNN is to classify jets and leptons in simulated collision data and to identify which of them originates from the same parent particle. The model is trained on simulated data consistent with the Standard Model and a simulated detector response, aiming to support further experimental analyses. Ultimately, the work seeks to develop a method that enables the reconstruction of this complex final state, providing a foundation for future studies of Higgs boson properties.

2 The Standard Model

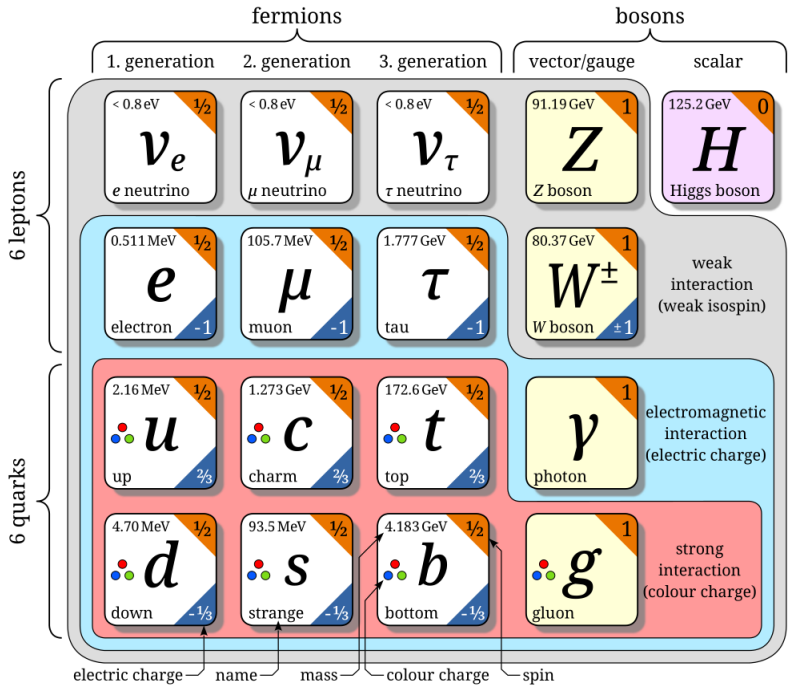

Figure 1: Elementary particles, their categorisation and properties, which are included in the Standard Model.

The Standard Model [3–12] is a widely used theoretical framework that describes the fundamental particles and the interaction between them. It combines the current understanding of quantum mechanics and quantum field theory as well as the electroweak interaction and quantum chromo dynamics into a single theory.

With the Standard Model, the fundamental particles are organised into two groups: fermions and bosons. The fermions, particles with a spin of $\frac{1}{2}$, can be seen on the left side of figure 1. The bosons, which have an integer spin, are located on the right-hand side of the graphic. Fermions make up all visible matter. Bosons mediate the fundamental forces between the fermions.

According to the current Standard Model, there are four vector Bosons with a spin of 1:

- g, gluons are massless bosons. They mediate the strong nuclear force.
- γ, photons are also massless bosons. They mediate the electromagnetic force.
- The Z bosons have a mass, a neutral charge and mediate the weak nuclear force.
- The W bosons also have a mass, a positive or negative charge and also mediate the weak nuclear force.

Furthermore, there is one scalar boson with spin-0, called the Higgs boson. It is the most recent addition to the Standard Model and the only scalar boson known to date. The Higgs boson is responsible for giving mass to other particles through the Higgs mechanism. The vector bosons mediate three of the four fundamental forces as described above: the strong nuclear force, the weak nuclear force, and the electromagnetic force. The fourth fundamental force, gravitation, is not explained by the Standard Model.

Fermions can be split up by the forces they interact with. Six of the fermions, the quarks, interact via the strong force. The other six fermions are leptons. They only interact through the electromagnetic and weak force.

Leptons can be further divided into electrons, muons, and taus and their corresponding neutrinos. Electrons, muons, and taus carry electric charge, while neutrinos are electrically neutral and interact only via the weak nuclear force. Additionally, every fermion has a corresponding antiparticle. These antiparticles have some properties inverted, for example charge.

2.1 The $t\bar{t}(H \to b\bar{b})$ Process

Proton proton collisions at high energy cause a multitude of different reactions and decay chains. The main focus of this thesis is the $t\bar{t}(H \to b\bar{b})$ process, which is relatively rare compared to some other decay processes. The occurrences of the different processes can be seen in figure 2.

The $t\bar{t}H$ production is mainly mediated through gluon-gluon fusion. Since gluons are massless and do not interact directly with the Higgs boson, the interaction needs to be mediated by a virtual quark loop. This quark is usually the heavy top quark. A graphical representation of the $t\bar{t}(H \to b\bar{b})$ process can be seen in figure 3, in the form of a Feynman diagram.

Both gluons involved in the fusion create, each, a top–antitop quark pair. The top quark from one gluon fusion and the antitop quark from the other interact with each other to produce a Higgs boson. This Higgs boson decays almost instantaneous into either a fermion–antifermion pair or a boson pair. The focus of this thesis is on the fermion–antifermion pair, in particular where the Higgs boson decays into a bottom-antibottom quark pair. This is also the most probable decay, with a branching ratio of approximately $\approx 58\%$.

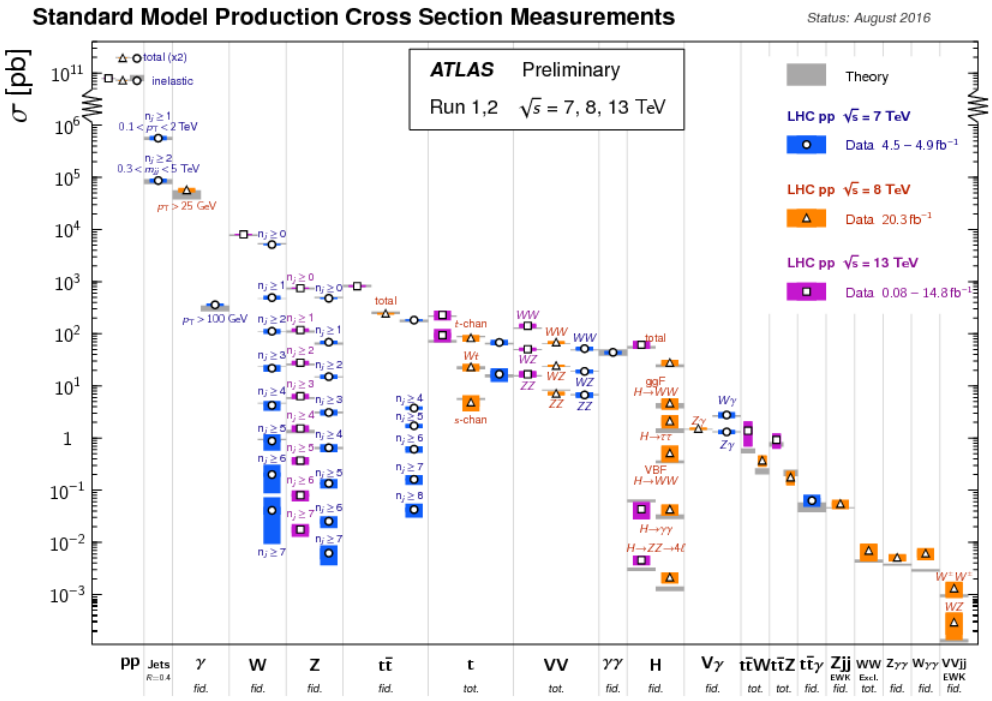

Figure 2: The figure shows the theoretical and measured (from different LHC runs) proton proton collision error sections, which indicate the respective production rate [14].

The other top and antitop quark, which are not producing a Higgs boson, decay further. The top quark decays into a W^+ boson and a bottom quark, while the antitop quark decays into a W^- boson and an antibottom quark.

W bosons decay leptonically or hadronically. In the leptonic case, the W bosons decays into an equally charged lepton and the corresponding neutrino. For example, a W^- might decay into an electron and an electron anti-neutrino. In the hadronic case, the W boson decays into a quark-antiquark pair. The total electrical charge of the quark-antiquark pair equals the charge of the W boson.

For example a W^+ would decay into one up-type quark and one anti down-type quark. Since there are three up/down type quarks. This yields a total of nine decay options, each occurring with the same probability. The data used in this thesis does not contain the case that the W boson decays into a bottom or an anti bottom quark, as well as top or anti top quark.

In many processes, such as top quark or Higgs boson decays, one W boson often decays leptonically while the other decays hadronically, which is the case in the data set used in this thesis. The quarks from hadronic decays produce particle showers known as jets. These jets can be measured by the ATLAS detector.

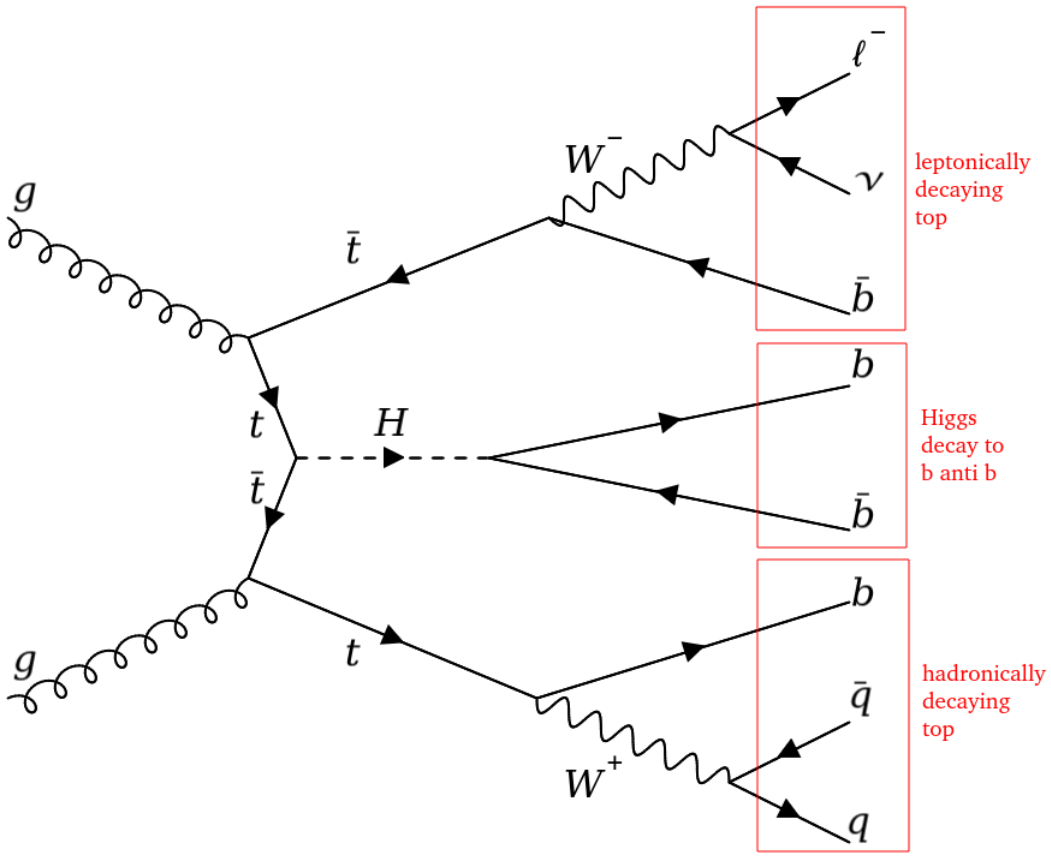

Figure 3: The figure shows one possible Feynman diagram of the $t\bar{t}(H \to b\bar{b})$ decay process. The W^- could also decay hardronically and the W^+ leptonically.

3 ATLAS

It is essential to examine the ATLAS detector and its coordinate system (figure 4) used for data gathered with the detector to understand the data used in this thesis. The origin of the ATLAS reference frame is at the nominal interaction point, the point at which the collision is expected to occur. The ATLAS collaboration [15] uses a right-handed coordinate system. The x-axis points to the centre of the Large Hadron Collider ring, the y-axis points straight upwards and the z-axis points along the beam pipe.

Each jet measured after a collision is parameterised with a 4-tuple $x_i \in \mathbb{R}^4$ with $x_i \coloneqq (\phi_i, \eta_i, p_{t_i}, E_i)$. ϕ is the angle between the x-axis and the measured vector projected onto the x,y plane. It is $\eta \coloneqq -\ln\tan(\theta/2)$, with θ being the polar angle in the transversal plane. The range of η is restricted to $[-2.5, 2.5]$, since the detector only covers that range. On the other hand, $\phi \in [-\pi, \pi]$ is not restricted. The measured transversal momentum and energy are denoted as p_t and E, respectively.

Leptons have an additional parameter which describes the charge of the particle.

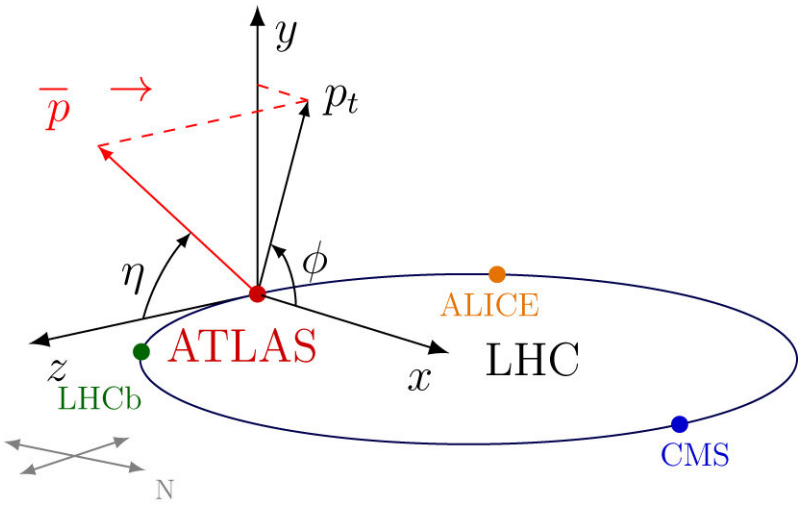

Figure 4: The figure shows a graphical representation of the coordinate system used at the ATLAS detector.

For each collision, the detector measures a variable number of jets and leptons. This variability poses the first challenge, as the number of detected particles differs from event to event, complicating data handling and processing. The variation arises because, like all measurement instruments, the ATLAS detector has constraints. For example, it has limited geometrical acceptance and limited efficiency. This means that particles at extreme angles outside the detectors range, or those with momentum below the detection threshold, may not be recorded.

In addition to missing particles, the detector can also reconstruct more particles than expected. This can result from the presence of additional nearby collision events, which contribute extra particles to the event.

3.1 Data Simulation

The data used in this thesis is obtained via Monte Carlo simulations as described in [16]. The first simulation is based on the theoretical assumptions given by the Standard Model. It is weighted to match the observed distributions in real data obtained from proton proton collisions, collected during Run 2 of the LHC with the ATLAS detector at a centre of mass energy $\sqrt{s} = 13$ TeV. The simulation samples multiple events to simulate both signal and background processes. Additionally, the simulation models uncertainties as well as effects of nearby proton proton collisions, called pileup. This simulation is used to obtain ground truth data.

The second type of data is obtained by further processing the first simulation using a fully

simulated ATLAS detector. This is done to simulate the real measurements as realistically as possible by including the flaws and physical limitations of the ATLAS detector.
These simulated events are highly detailed, including full information on every sensor measurement and particle track. This raw data is further processed to produce the 4-tuples described in the section 3.
This type of data is used as the input data, as it contains the final-state observables measured by the ATLAS detector, including contributions from background processes and measurement uncertainties.
The data used as ground truth includes intermediary states and particles, like the Higgs Boson, which decays almost instantly and cannot be observed directly by the detector. Given this ground truth, each jet of the input data can be decorated with labels. The task of matching the ground truth data to the input data is a rectangular assignment problem. An efficient way to solve these problems is given by [17].
Let $C \in \mathbb{R}^{N_R \times N_C}$ be the cost matrix with $N_C \leq N_R$. Each element $c_{i,j}$ represents the cost of assigning data point i to data point j, or in this case the distance from data point i to data point j. N_C and N_R correspond to the number of columns and rows, corresponding to the number of input and ground truth data points. The ground truth and input data is assigned to the column and rows such that $N_C \leq N_R$ holds. The rectangular assignment problem is then defined as:

$$
\begin{aligned}
X^* &= \arg\min_X \sum_{i=1}^{N_R} \sum_{j=1}^{N_C} c_{i,j} x_{i,j} \\
\text{subject to} \sum_{j=1}^{N_C} x_{i,j} &= 1 \forall i \\
\sum_{i=1}^{N_R} x_{i,j} &\leq 1 \forall j \\
x_{i,j} &\in \{0,1\} \forall i,j
\end{aligned}
$$

As a cost function, we use a modified version of the 2-norm, which takes into account that the distance is based on the two angles ϕ and η given in the 4-tuple of the jets. For example, consider two angles $\phi_1 = \pi - \epsilon$ and $\phi_2 = -\pi + \epsilon$, with $\epsilon < 0$. Since these two

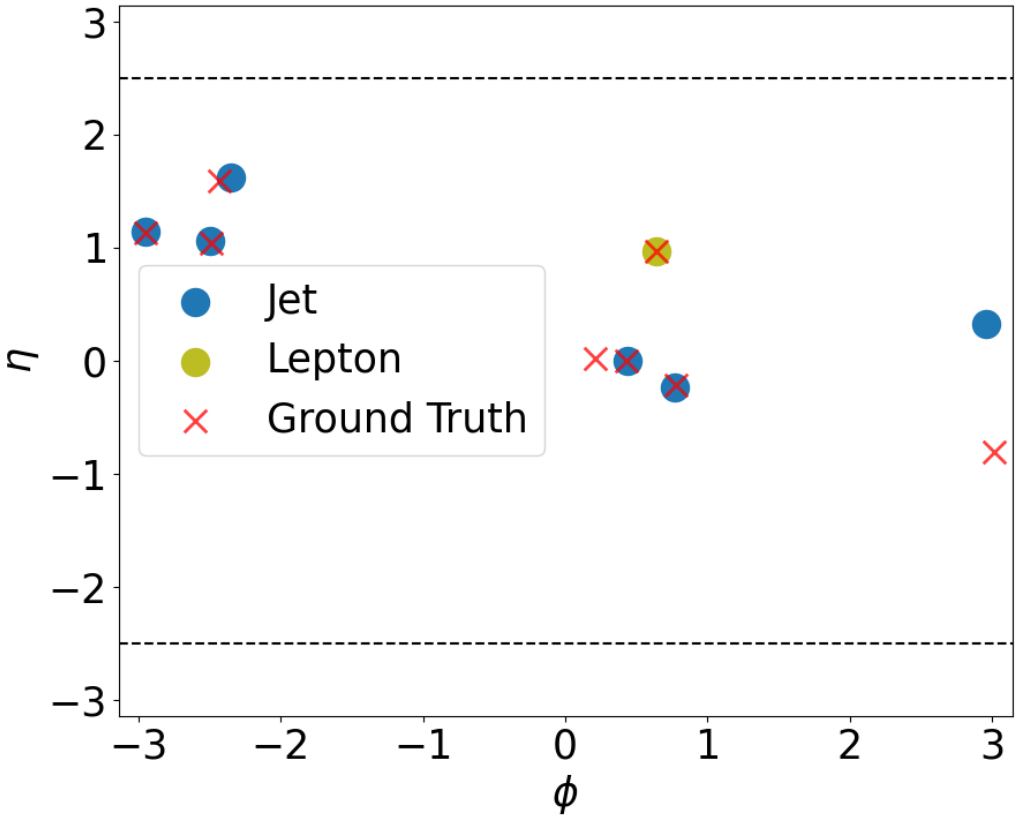

Figure 5: The figure shows the input data represented by circles, separated in jets and leptons, and the ground truth data as red crosses. The dashed black line indicates the boundary of the angle η, beyond which particles are not reconstructed by the detector. In this example, one jet is not matched. One of the unmatched red crosses corresponds to a neutrino in the ground truth. Since neutrinos cannot be detected by the ATLAS detector, they should not be matched.

angles are nearly identical on the circle, the distance should be

$$\lim_{\epsilon \to 0} d(\phi_1, \phi_2) = 0,$$

in contrast to the uncorrected Euclidean distance, which would yield to 2π, a value that incorrectly suggests the angles are nearly opposite, when in fact they are nearly identical on the circle.

The assignment is only accepted, if the cost is below a predefined threshold. If $x_{i,j} = 1$ and i is a jet, then $c_{i,j} \leq 0.3$. When i is a lepton, then $c_{i,j} \leq 0.1$. Furthermore, ground truth particles which are neutrinos are never matched. Even if a ground truth neutrino would be matched to an input particle, it can only be a random coincidence since neutrinos cannot be detected by the ATLAS detector. An example of an event can be seen in figure 5.

3.2 Flavour Tagging

Flavour tagging aims to classify the flavour of a jet. A jet is a shower of particles caused by quarks. The flavour of a jet is the type of quark, that initiated such jet. To identify the flavour, the ATLAS collaboration uses raw data, which includes information about each detector track. Only b- and c-quarks are tagged, since they have a relatively long lifetime and high decay multiplicity.

Traditional flavour tagging algorithm focus solely on b-tagging and use a two step approach. Firstly, specialised low level algorithm are applied to extract information about the trajectories in the tracks. In the second step, a classifier is applied to this information, for example by using DeepSets [18].

The data used in this thesis is decorated with the results from the GN2v01 tagger [19], a modern, single-step flavour tagging algorithm based on the transformer architecture. It further improves performance through auxiliary training objectives. The results from the GN2v01 tagger are calibrated into six different working points, each corresponding to a different efficiency for the target flavour e.g. for b-tagging (65%, 70%, 77%). A jet is tagged at a given working point if its probability exceeds a calibrated threshold required to pass that point. This means that the working point represents the fraction of true b- or c-jets that are successfully identified at that efficiency level. If none is passed, the jet is untagged. The exact specification can be seen in figure 6.

4 Methods

One problem, already mentioned, is that collision events can have different numbers of measured particles. An approach to handle this is to treat each event as a set of jets/leptons and apply a multilayer perceptron (MLP) independently to each jet/lepton of the set. However, this method discards a lot of information especially about relationships between particles. For example, in an image, if each pixel is treated individually, any size of image could be processed, but the model would fail to capture spatial correlations such as the ordering and relative positions of pixels. However, that is essential for meaningful pattern recognition. To address this issue, meaning to preserve the structural information, the input is concatenated and only then, an MLP is applied. This allows the model to process the full set. Still, sets of different sizes cannot be handled uniformly. For images

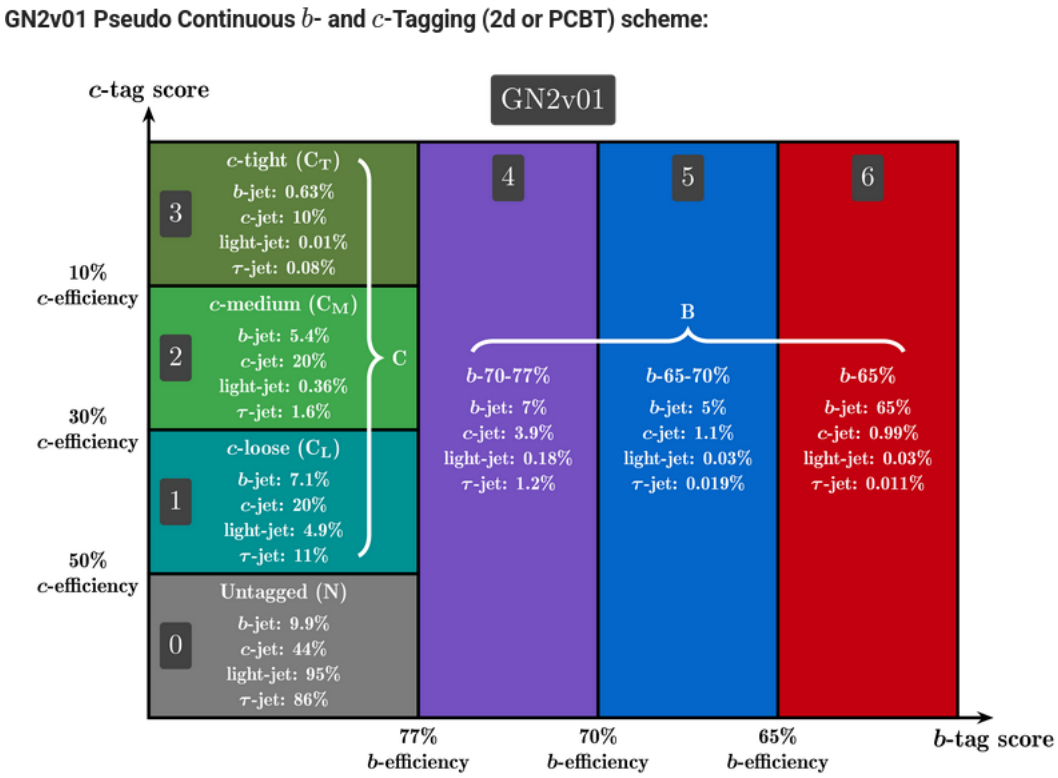

Figure 6: The figure shows the seven different working points, the GN2v01 tagger predicts. A particle is assigned in bin 0 if it is untagged. The bins 1-3 is for c-Tagging and bins 4-6 for b-Tagging. The exact percentages of particles, which pass the corresponding working point, is listed inside each bin.

of different sizes, this problem is addressed using convolution, enabling the model to generalise across different input dimensions. Convolution applies a kernel, a matrix for example of size 3×3 or 5×5, across the entire image by sliding it over local regions. Kernels consist of learnable parameters, the kernel weights. At each position, the kernel weights are multiplied element-wise with the corresponding pixel values and aggregated via summation or averaging to produce a single output value. This process generates a new, smaller feature map that retains spatial structure and works regardless of the size of the image.

The main difference between image data and the input data is that the set of pixels have a fixed order, forming a regular grid structure. Therefore, an image can be described as a graph leading to the conclusion that convolution can be achieved with a graph structure, called graph convolution. In the case of a 3×3 kernel each pixel is connected to its immediate neighbouring pixels, including diagonals. For a 5×5 kernel, the connectivity extends to two neighbours in each direction. The convolution is then defined as the

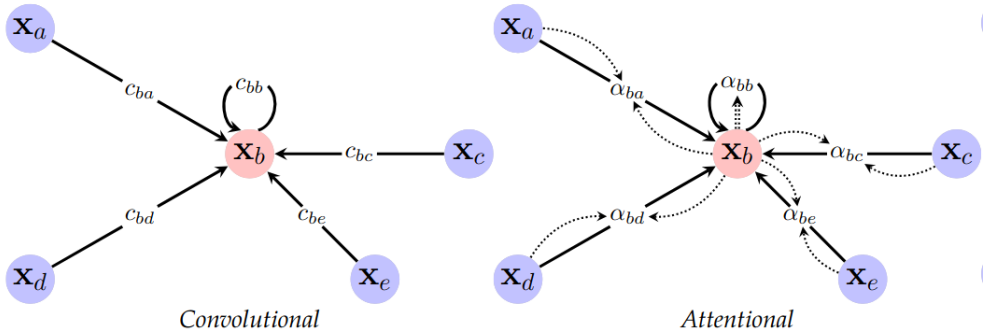

Figure 7: The figure shows a graphical representation of the two types of graph convolutional layers: On the left classic graph convolution and on the right graph attention.

weighted sum of the connected pixels. The key difference between standard convolution and graph convolution lies in how boundary pixels are handled. In standard convolution, the output size is smaller than the input because the kernel cannot be applied at the image edges as there are not enough neighbouring pixels. By using a padding, adding virtual pixels around the boundaries appropriate to the aggregation operator allowing the kernel to be applied uniformly across the entire image, these two approaches are equivalent.
Since the edges of a graph can have any structure, not only neighbouring structure of convolution, graph convolution is more flexible and generalisable than traditional convolution.

4.1 Graph Neural Networks

Let $k \in 1, \ldots, N$ be the k-th event and N the total number of events. Then, an undirected graph can be defined as $\mathcal{G}_k = (\mathcal{V}_k, \mathcal{E}_k)$, representing event k. Let N_k be the number of particles measured in event k. Then, the Graph $\mathcal{G}_k$ has N_k nodes $v_i \in \mathcal{V}_k$ and edges $(v_i, v_j) \in \mathcal{E}_k$, with $i, j \in \{1, \ldots, N_k\}$.
For this graph structure convolution layers can be defined. There are three types of layers Graph Neural Networks (GNNs) often use [20]. This thesis focusses on graph convolution and graph attention. Figure 7 shows a graphical representation of these two types, which will be introduced in the following.
Graph convolution is inspired by classical convolution but applies fixed weights, which are usually dependent of the adjacency matrix of the graph. Let $X \in \mathbb{R}^{N_k \times m}$ be the data of event k, $u, v \in \mathcal{V}_k$ and $\mathcal{N}_u$ be the set of vertices to which u is connected with an edge.

Then, the update for the node u is defined as

$$\tilde{x}_u = \phi\Big(x_u, \bigoplus_{v \in \mathcal{N}_u} c_{uv}\psi(x_v)\Big). \tag{1}$$

ϕ and ψ are learnable functions, for example simple linear layers. $\bigoplus$ represents any aggregation operation for example mean or sum and c_{uv} are the weights. The graph convolution layer is the most basic type of layer in Graph Neural Networks. While it effectively propagates information across the graph, the fixed weights can be limiting for complex data. To address this, more advanced architectures are needed such as attention. Graph attention implements a learnable self-attention mechanism, which computes the weights implicitly. The update formula with attention is defined as

$$\tilde{x}_u = \phi\Big(x_u, \bigoplus_{v \in \mathcal{N}_u} a(x_u, x_v)\psi(x_v)\Big). \tag{2}$$

The usual choice for the attention mechanism is given by

$$a(x_u, x_v) \coloneqq \frac{\exp\Big(\text{LeakyReLU}\Big(\boldsymbol{a}^T(Wx_u \parallel Wx_v)\Big)\Big)}{\sum_{v' \in \mathcal{N}_u} \exp\Big(\text{LeakyReLU}\Big(\boldsymbol{a}^T(Wx_u \parallel Wx_{v'})\Big)\Big)} \tag{3}$$

as defined in [21]. W is a linear transform parameterised by a weight matrix, $\boldsymbol{a}$ is a weight vector and $\parallel$ denotes the concatenation operation. This mechanism first applies a linear transformation to the node features, producing a new representation for each node. It then computes a score for each pair of nodes by taking the dot product of the transformed features with a learnable weight vector $\boldsymbol{a}$. The two linear layers W and $\boldsymbol{a}$ are applied sequentially without an activation function between them, effectively forming a single linear transformation. To improve model expressiveness, the authors of GATv2 [22] propose moving the weight vector $\boldsymbol{a}$ outside the LeakyReLU activation, thereby introducing a non-linearity between the two linear transformations. This architectural change allows the model to learn more complex, non-linear relationships.

The last type of layer is the message passing layers. Message passing layers generalise graph attention layers. As this type of layer is not used in this thesis, it is not defined here.

4.2 Spectral Graph Convolutions

The equation (1) is not particularly useful in a deep learning context due to their inefficiency and sequential, node-by-node update mechanism. However, just as in classical convolution, graph convolution can be efficiently implemented in the Fourier domain. In the following, the formulations derived in [23] are introduced.

For a graph with N vertices, adjacency matrix A and degree matrix D, which is the diagonal matrix containing the number of neighbouring nodes $D_{ii} = \sum_j A_{ij}$, the normalised graph Laplacian is defined as

$$L = I_N - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}. \tag{4}$$

I_N denotes the identity matrix of shape N. Since L is a real symmetric positive semidefinite matrix, it can be decomposed into $L = U\Lambda U^T$. U is the matrix of eigenvectors and Λ the diagonal matrix of the corresponding eigenvalues. U is also called the Fourier basis. The convolution of a signal $x \in \mathbb{R}^N$ with a filter g_θ, a diagonal matrix containing $\theta \in \mathbb{R}^N$, in the Fourier domain is defined as

$$g_\theta \star x = U g_\theta U^T x. \tag{5}$$

This formulation remains computationally expensive and and impractical for large graphs, since the computation of the eigendecomposition is infeasible for large graphs, requiring $\mathcal{O}(N^2)$ operations per layer.

As described in detail [24], the equation (5) can be approximated by Chebyshev polynomials $T_k(x)$ up to order K with

$$g_{\theta'} \star x \approx \sum_{k=0}^{K} \theta' T_k(\tilde{L}) x. \tag{6}$$

$\theta' \in \mathbb{R}^N$ are the coefficients for the Chebychev polynomials, which are defined recursively: $T_0(x) = 1$, $T_1(x) = x$, $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ and $\tilde{L} = \frac{2}{\lambda_{max}} L - I_N$. This approximation depends on nodes that are a maximum of K steps away and can be evaluated in $\mathcal{O}(|\mathcal{E}|)$.

If $K = 1$, this equation describes a single layer that is linear with respect to the graph Laplacian. Equation (6) can be written as

$$g_{\theta'} \star x \approx \theta'_0 x + \theta'_1 (L - I_N) x = \theta'_0 x + \theta'_1 D^{-\frac{1}{2}} A D^{-\frac{1}{2}} x \tag{7}$$

under the assumption that a network can scale the eigenvalues such that $\lambda_{max} \approx 2$. As described in [23], it might be beneficial to constrain the equation to one parameter instead of two to avoid overfitting and set $\theta = \theta'_0 = -\theta'_1$. Since the eigenvalues are in the range [0,2], it might also lead to numerical issues when applied repeatedly. This can be addressed by renomalisation to $\tilde{A} = A + I_N$ and $\tilde{D}_{ii} = \sum_j \tilde{A}ij$. This approach can also be generalised to multichannel signals which results in:

$$Z = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} X \Theta, \tag{8}$$

with $X \in \mathbb{R}^{N \times C}$ and $\Theta \in \mathbb{R}^{C \times F}$. Here, C corresponds to the number of channels and F to the feature size.

This formulation allows to quickly approximate a single convolution layer, which is also easy to implement.

5 Problem Statement

For each collision event, the goal is to reconstruct the underlying physics process from the measured final states. To achieve this, the problem is decomposed into two sub-tasks.

First, each jet and lepton is classified into one of seven flavour categories. For jets, the classes are unmatched, light quarks (up and down quark), strange quark, charm quark and bottom quark. For leptons, there are two classes, electrons and muons combined and taus.

The second task is the edge classification into four classes to determine whether pairs of jets (or a jet and a lepton) share a common parent particle. The edges between nodes, which originate from the same parent particle, are considered as an edge of that class. All other edges belong to the false edge class.

The first non false class is the edge between the two b-quarks originating from the Higgs boson. The second class is the edges between a lepton and a b-quark, which originate from the same top quark. The third class is the counter part to the second class and contains the edges between the two light quarks and the b-quark from the other top quark. The edge classes can be seen in figure 3.

Let $\mathcal{C}_k$ be the set of classes to be classified. For the classification, a generative approach is used. It models the class conditional probability densities $p(x|\mathcal{C}_k)$ and the class priors

$p(\mathcal{C}_k)$. Then, the Bayes' theorem can be used to compute the posterior distribution

$$p(\mathcal{C}_k|x) = \frac{p(x|\mathcal{C}_k)p(\mathcal{C}_k)}{p(x)} = \frac{p(x|\mathcal{C}_k)p(\mathcal{C}_k)}{\sum_j p(x|\mathcal{C}_j)p(\mathcal{C}_j)}. \tag{9}$$

Let

$$a_k \coloneqq \ln(p(x|\mathcal{C}_k)) + \ln(p(\mathcal{C}_k)), \tag{10}$$

such that equation (9) can be written as

$$p(\mathcal{C}_k|x) = f_k = \frac{\exp(a_k)}{\sum_j \exp(a_j)}. \tag{11}$$

Here, $f : \mathbb{R}^n \to \mathbb{R}^k$ is the model which approximates the posterior and $f_k = f_k(x)$ is the k-th value of the vector. An error function is obtained by calculating the negative log likelihood, which results in the cross entropy loss. Note that ln and exp are used to simplify the calculations of the negative log likelihood, not presented here. This can also be done for multidimensional inputs and classes, which results in the loss function

$$\mathcal{L} = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \ln(f_{nk}), \tag{12}$$

with t_{nk} being the one-hot encoded class labels. The model is defined as $f : \mathbb{R}^{n\times m} \to \mathbb{R}^{n\times k}$. The model performs both node and edge classification, applying the cross-entropy loss function to each task. The total loss is the sum of both cross entropy losses. Optimising this combined loss is done by using gradient-based descent methods.

6 Implementation Details

In the following chapter, all details regarding the model, including data preparation and training, will be discussed.

6.1 Data Preparation

The distributions of ϕ and η both have a mean of 0 and are bound, so no normalisation is needed. The transversal momentum and energy are normalised to a mean of 0 and a standard deviation of 1.

Table 1: The table shows the sum of all pairwise distances of the node features after each convolutional layer. Without residual connection the number of convolutional layers is severely restricted.

Convolutional layer	GCN	GAT	CGN w residuals	GAT w residuals
1	$< 10^{-7}$	2.4×10^{-2}	3.5	4.5
2	$< 10^{-7}$	1.4×10^{-6}	8	8.9
3	$< 10^{-7}$	$< 10^{-7}$	19.8	17.9
4	$< 10^{-7}$	$< 10^{-7}$	54.9	40.8

The ATLAS detector can easily distinguish between jets and leptons. It can even distinguish between different types of leptons. This is because leptons have very specific properties that the detector can recognise. Therefore, this additional information can reliably be used. This information is encoded with one-hot encoding for the three cases: jet, electron, or muon. Tau is not encoded. This adds a total of three binary parameters. The working points of each jet determined by the GN2v01 tagger, see section 3.2, are one-hot encoded. GN2v01 uses a total of seven different working points, adding seven parameters for all jets. Since there is no tagging information for leptons, we use a vector with seven zeros for all leptons. Therefore, each measurement in each event is parameterised as a 13-tuple.

6.2 Model

Graph neural networks are well suited for the reconstruction problem, as explained in section 4. However, the data which should be processed by a GNN requires an adjacency matrix. The graph structure for the events, described in section 4.1, does not contain an adjacency matrix. Therefore, we attempted to implement an architecture that would use a learnable adjacency matrix. However, the performance was not promising. Instead, graph attention layers are used and all the graphs are viewed using a fully connected adjacency matrix. This means that each node is connected to every other node. The idea behind this is that the implicit weighting in the form of the self-attention operator should be able to adapt the adjacency matrix to the problem. Furthermore, fully connected adjacency matrices have the issue that the node features tend to average out based on the formulation of equation (1). This problem also occurs in graph attention layers, but not as severely as in classical graph convolutional layers.

Table 1 shows the average difference of the nodes. To calculate this, we evaluated the

average pairwise distance based on the 2-norm after each layer of a trained model. The values in the table are the average of these summed pair-wise distances across all events. Graph convolutional networks without residual connections do not achieve good results, because a fully connected adjacency matrix causes all nodes to perform the same operations. Graph attention networks are capable of weighting the adjacency matrix, based on the nodes features and therefore slow this effect down. When the number of graph attention layers does not exceed 2, the model is able to learn meaningful representations for the nodes. Otherwise, they also treat all nodes equally. This significantly limits the overall complexity of the model.

This problem can be addressed using residual connections. These connections are learnable linear layers which are applied before the convolution. The results of these layers are added after the convolution. The effects can be seen in the right columns of table 1. As desired, the pairwise distances are no longer zero. However, the pairwise distances become larger and larger as the number of layers increases. This can lead to numerical problems, such as exploding gradients. However, this is preferable to being limited to a maximum of 2 graph attention layers without residual connections.

The model used to address the two problems, described in the section 5, first uses a simple linear layer to embed event information into the hidden size of 128. Then, four graph attention layers with additional residual connections and three heads each are applied. Each head is a different convolutional layer and the results are concatenated. For the node classification, an MLP is used which reduces the dimension to the number of classes. For edge classification, the output of the last convolutional layer is concatenated, based on the adjacency matrix. In the fully connected case, each node feature is concatenated with all other nodes. Then, another MLP is used to classify all edges and output a value for all four classes. All intermediate linear layers in the MLPs are followed by a ReLU activation function. ReLU activation functions between the convolutional layers led to impaired performance. Therefore, only linear layers are followed by ReLU. This is consistent with claims in [25]. A graphical representation of the model can be seen in figure 8.

Events can be reconstructed by combining node and edge classification. However, the model predicts several candidates for each edge class, which allows for many different reconstructions. There are several ways to select one of these possibilities. The simplest is to choose the possibility that the model is most confident about, i.e., the one with the

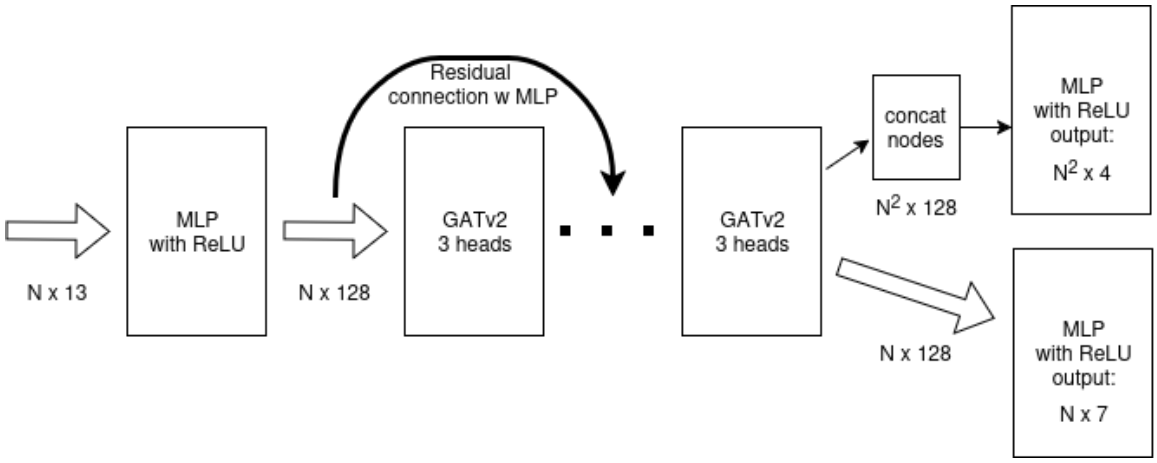

Figure 8: A graphical representation of the deployed model.

highest predicted probability. Another option is to set specific thresholds for the node and edge probabilities to reduce the number of different possibilities, and then use an algorithm, such as a logistic regression model, to select from the reduced set of possibilities. We tried both options, but were unable to find good thresholds for the second approach. Therefore, rather than applying thresholding, a logistic regression model is applied to all possible edges directly. The 4-tuple of each node is combined with the output of the trained model and then processed by a logistic regression, also trained on the training data set. This should learn the thresholds implicitly.

6.3 Training

The model was trained on a relatively small data set. The total of number of $t\bar{t}H$ events is 44,070. Since the focus of this thesis is the $t\bar{t}(H \to b\bar{b})$ process, these events are filtered to obtain a total of 28,262 different events. This number is then split up with a ratio of 80:20 for the training and validation data sets. For training, a batch size of 128 was used, along with the Adam optimizer [26] and a learning rate of 3×10^{-4}. Due to the model design, it is not a problem that the events have different number of measured jets. However, this poses an obstacle for batching. Batch processing is useful for faster training and also for regularisation through randomness. To process the data in batches nonetheless, each batch can be considered as a giant graph containing 128 unconnected subgraphs. For the batched graph, the node features are concatenated and the adjacency

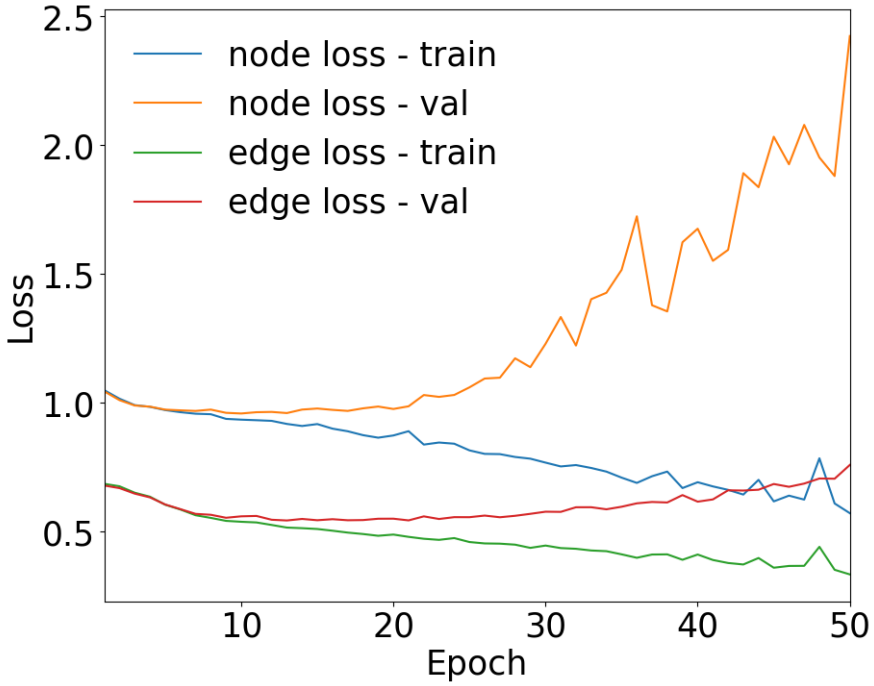

Figure 9: The figure shows the losses of the training and validation data sets during training. The total loss is divided into two parts: the node loss and the edge loss. The training and validation losses (val) begin to diverge after about 10 epochs. While the training loss continues to decrease, the validation loss almost stops decreasing. After about 20 epochs, the model begins to overfit and the validation losses increase again.

matrix can be expressed as a diagonal block matrix

$$A_b = \begin{pmatrix} A_1 & 0 & \dots \\ 0 & \ddots & 0 \\ \dots & 0 & A_n \end{pmatrix}.$$

This allows for computation with the equation (8).

Figure 9 shows the loss curves of the nodes and edges for training and validation data. The node loss decreases only slightly, while the edge loss decreases more significantly. Both losses begin to diverge at around 10 epochs and gradually decrease up to 20 epochs. After that, the model begins to overfit. The node loss overfits significantly more than the edge loss. With a dropout of 0.2, overfitting is delayed but it does not lead to better performance. The model with the lowest combined validation loss was selected.

Figure 10 shows the accuracy per epoch. The node accuracy starts relatively high and increases to around 65% in the first few epochs. After that, the node accuracy fluctuates randomly around 65%. The edge accuracy starts low compared to the node accuracy and

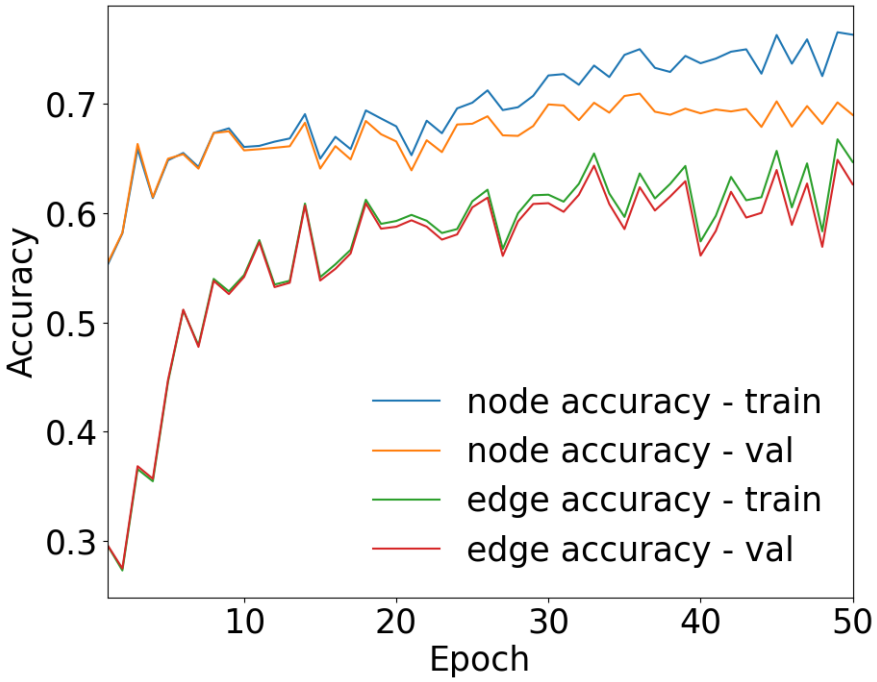

Figure 10: The figure shows the accuracies of the training and validation data sets during training. The accuracies of the training and validation data sets are relatively close to each other. In epoch 20, where overfitting begins, they start to diverge increasingly.

increases rapidly to 55% in the first 15 epochs. After that, there is no significant increase in accuracy.

7 Evaluation and Results

All results and analyses in this chapter are based exclusively on the 28,262 $t\bar{t}(H \to b\bar{b})$ events.

7.1 Data Analysis

To gain a better understanding of the data, the event characteristics are examined in the following. The distribution of the jet multiplicity is shown in the top part of figure 11. An event has at least 5 and at most 15 jets. The average is 7.37 jets. The most common jet multiplicity is 7, which makes sense since the $t\bar{t}(H \to b\bar{b})$ process leads to 7 (without neutrino) final states. About 42% of all events have 8 or more jets, which also means that at least 42% of events have one or more unassigned jets. This can be seen more clearly in the lower left part of the figure 11. It shows the distribution of unassigned jets in the events. 80% of events have at least one unassigned jet, with up to nine unassigned sets.

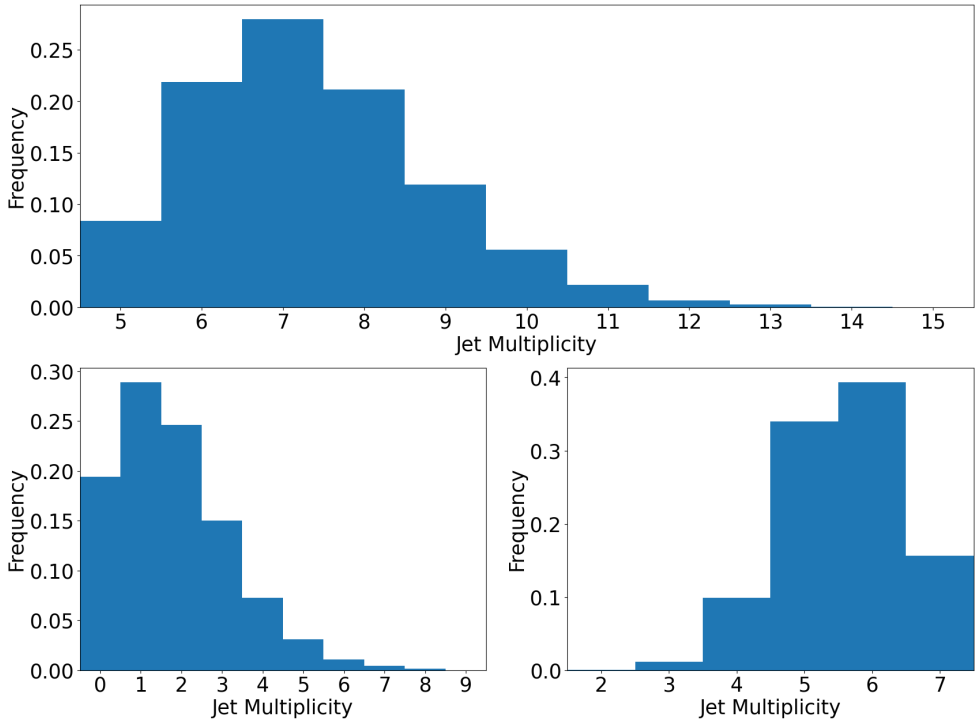

Figure 11: The top figure shows the distribution of jet multiplicity in an event. Events can have between 5 and 15 jets, with 7 being the most common number. The second figure (bottom left) shows the distribution of multiplicity of unmatched jets. Close to 80% of events have at least one unmatched jet. The third figure (bottom right) shows the distribution of matched jet multiplicity. The most common number is six matched jets, while only around 15% are fully matched/reconstructable.

The average number of unassigned jets is 1.79, which corresponds to a fairly large class. The counterpart to this can be seen on the lower right side. It shows the distribution of the number of assigned jets. Most events contain only five to six assigned jets, and the average is 5.58. To fully reconstruct the event, all seven jets must be assigned, which is only the case in 15% of all events. In all other cases, at least one of the non false classes cannot be reconstructed because it is not present in these events.

The next section takes a closer look at the data labels. Figure 12 shows the occurrence of the different node classes. The most common class is the bottom quark, accounting for approximately 45% of all nodes. This is perfectly reasonable, given that four of the seven final states are bottom quarks. Around a quarter of all nodes are unmatched. The strange, charm and light quarks all occur similarly, taking into account that the light quarks consists of up and down quarks. Considering the final state of the $t\bar{t}(H \rightarrow b\bar{b})$ process, there should be precisely one lepton and two non-b quarks, as well as four bottom

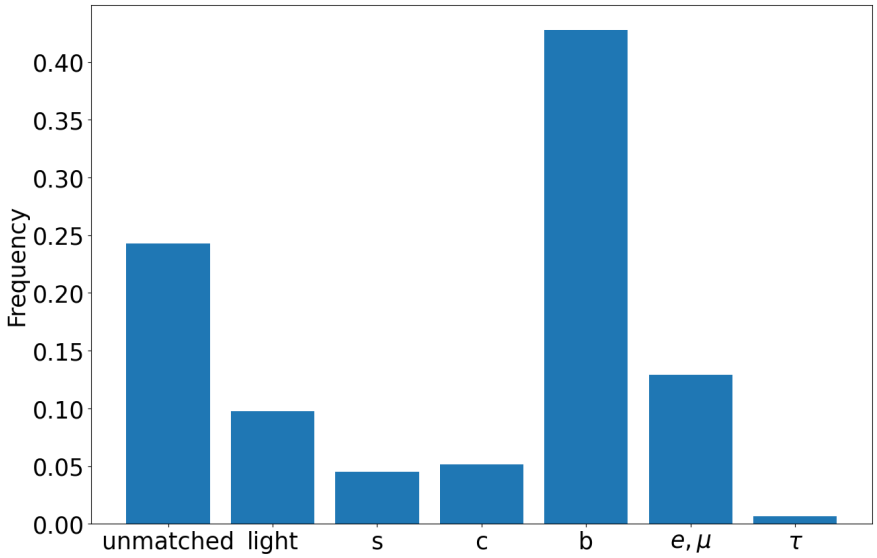

Figure 12: The figure shows the frequency of each node class across the entire data set. The most common node class is bottom quarks. This is as expected, since the process should have four. The second largest class is unmatched, which causes a lot of confusion for the model.

quarks. In the input data, light, strange and charm quarks account for a total of 19.4% of all nodes, while leptons account for 13.6%. But leptons should occur half as often as the other non-b quarks, which suggests that leptons in the input data are less frequently unmatched than quarks. A plot supporting this can be found in the appendix figure A1. Finally, the tau lepton is rather rare, accounting for only 0.64% of all leptons.

For the edge classes, the occurrences are dominated by false edges, see figure 13 on the left. The edge classes $H \to b\bar{b}$ and $W \to l, \nu$ span only two nodes. This results in two edges when both nodes exit. The $W \to q\bar{q}$ class has three nodes, leading to six different edges. All the other nodes are false edges. If there were seven nodes, there would be 49 possible edges, and if the event were fully reconstructable, this would lead to only $\frac{10}{49} \approx 0.2$ meaningful edges. This only holds if all three edge classes exist. The fraction of events in which the edge classes exist is shown in figure 13 on the right. In approximately 56% of all events, both bottom quarks, which originate from the Higgs boson, are matched. The $W \to l, \nu$ class exists in 80% of all events, which is significantly higher than the first class. This is because leptons are more likely to be matched. Lastly, the $W \to q\bar{q}$ class exists in only around 40% of events, which is because it consists of three nodes instead of two, making it less likely that all nodes will be matched.

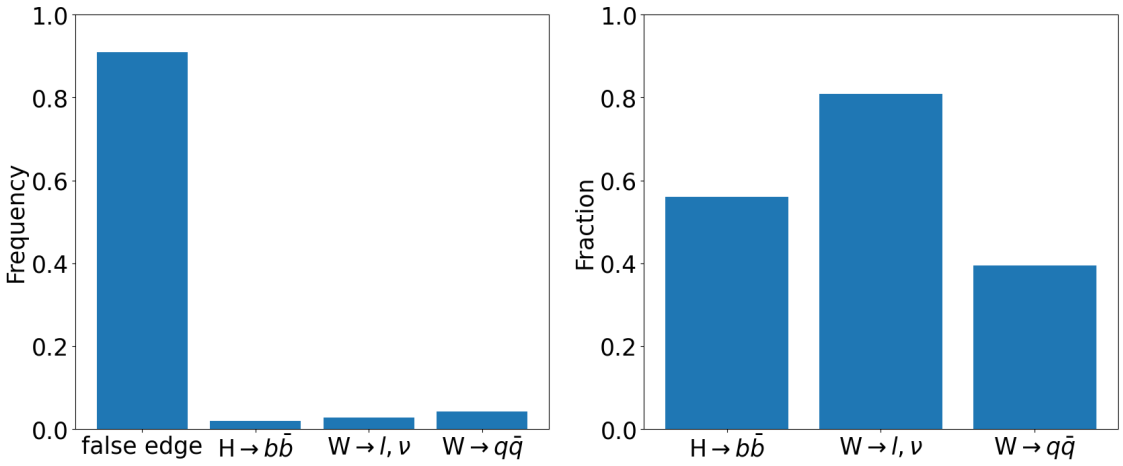

Figure 13: The first (left) figure shows the number of occurrences of each edge class. 90% of all edges are false edges. The second (right) figure shows how often the different edge classes occur in all events. This, combined with the number of possible edges per class, leads to the remaining 10% of edges in the left-hand plot.

7.2 Model Analysis

This chapter analyses the performance of the model. Unless specified otherwise, all plots are based on the validation data set. The overall accuracy for the node classes is 66.1%. The confusion matrix for the nodes can be seen in figure 14. The precision of b-quarks (80%) contributes significantly to the overall accuracy since it is by far the largest class. However, the b- and c-quark classes must be treated differently because information was obtained from the GN2v01 tagger. This information and the models output will be compared later.

The cluster of leptons at the bottom right of figure 14 stands out. The model can distinguish between electrons, muons and taus more accurately than by random guessing. If the leptons were not separated in electrons, muons and taus, there would be almost no confusion between any other classes. Another cluster of confusion is between the classes unmatched, light and strange quarks, with light quarks causing the most confusion. Furthermore, the model often mistakes charm quarks for light quarks.

The confusion matrix only shows the models final predictions, i.e. the class with the highest output value. This representation loses a lot of information. Another representation that provides more insight can be seen in figure 15. This illustrates how the various output values are distributed when the true label is unmatched. It provides a more detailed version of the first row of the confusion matrix by showing the models prediction probabilities of the different classes instead of only the most likely one.

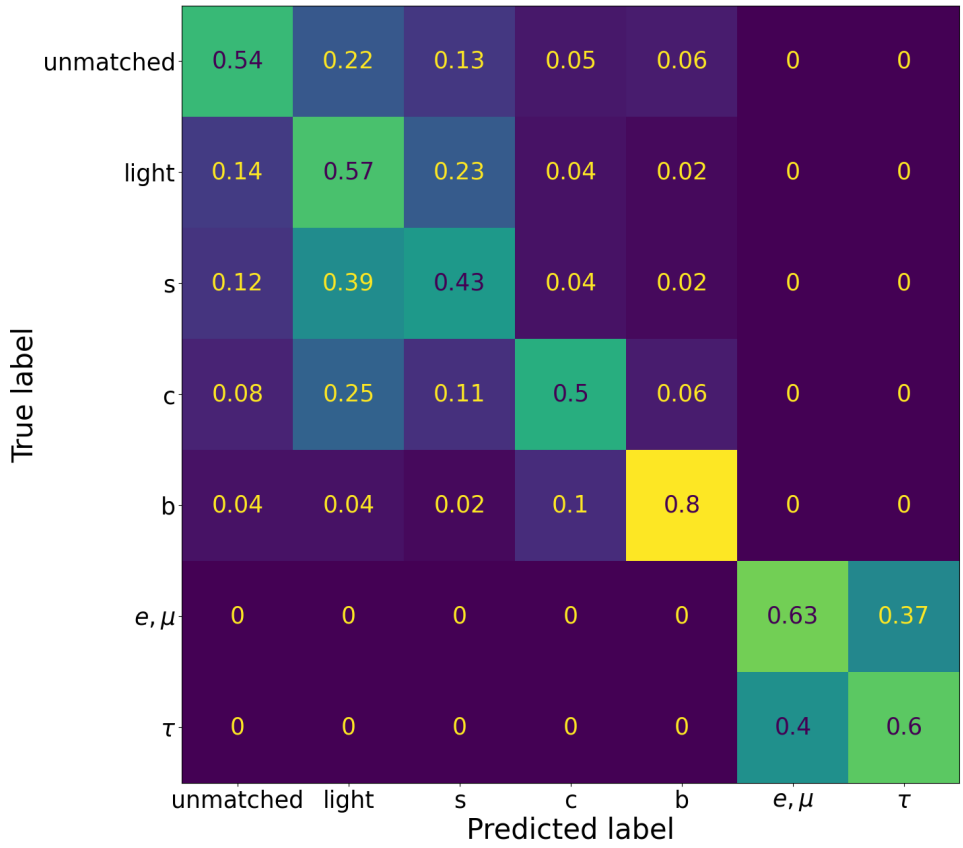

Figure 14: The figure shows the confusion matrix of the node classes. There is a lot of confusion between the classes unmatched, light quarks and strange quarks. Bottom quarks are rarely confused, except with charm quarks. Charm quarks are mostly confused with light quarks. There is also a lot of confusion between e, μ and τ, but leptons are never confused with any other class.

The mass of the unmatched class is evenly spread, whereas all the other classes have clear centres of mass. This may be because the unmatched jets originate from background or neighbouring processes. Meaning, they belong to an unknown class. They are therefore labeled as unmatched but might actually be part of another class. This would also explain the small b-quark peak at a predicted score of 0.85, as these are most likely jets, from neighbouring collisions, corresponding to b-quarks. The other plots of this type can be found in the appendix figures A2 and A3.

The model uses information obtained from the GN2v01 tagger. To compare the taggers results with those of the model, different working points are used as hard labels. As the working points for b-efficiency overlap, specific working points are used for different b-efficiencies. Specifically, working point 6 is used for a b-efficiency of 65%, working points 5 and 6 are used for a b-efficiency of 70%, and working points 4, 5 and 6 are used for

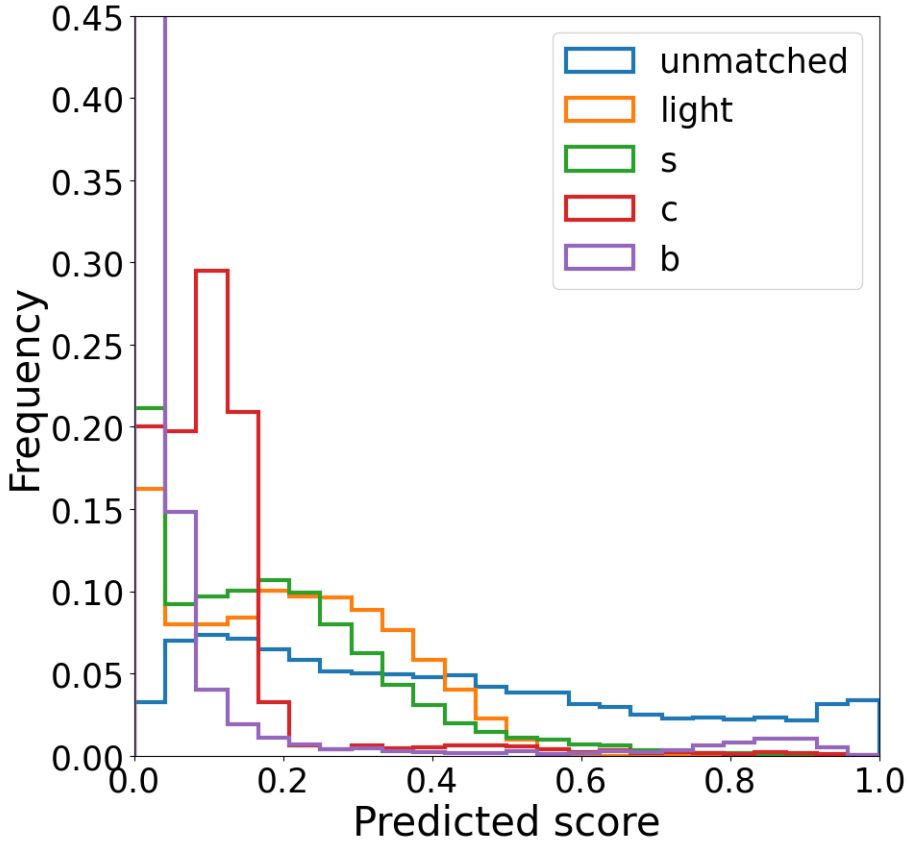

Figure 15: The figure shows a more continuous representation of the first column of the node confusion matrix. One histogram is plotted for each class. The leptons are not shown, as all their mass would be in the first few bins. Taking a closer look at the b-quark histogram, a small peak can be seen around 0.8. This suggests that some unmatched jets are actually bottom quarks. The same can be seen on a smaller scale with charm quarks at 0.5.

a b-efficiency of 77%. The results can be seen in figure 16. Similarly, hard labels are obtained for c-efficiency, as shown in figure 17. For b and c quarks, the model yields the same results as the tagger for a b-efficiency of 77% and a c-efficiency of 50%.

The confusion matrix for the edge class prediction can be seen in figure 18. The false edge class has the lowest precision at 51%, and is therefore confused with the other classes a lot. This is to be expected, since even if the model could predict b-quarks perfectly, it would be very difficult to distinguish them from each other. Additionally, the detector cannot differentiate between fermions and anti-fermions. Therefore, the model is probably stuck in a local minimum that classifies all edges with the correct node partners and all possible b quarks. The high level of confusion with the false edge class is problematic, as they occur around ten times more frequently than the other classes. This is illustrated in figure 19. The number of predicted edges in the non-false classes is more than four times

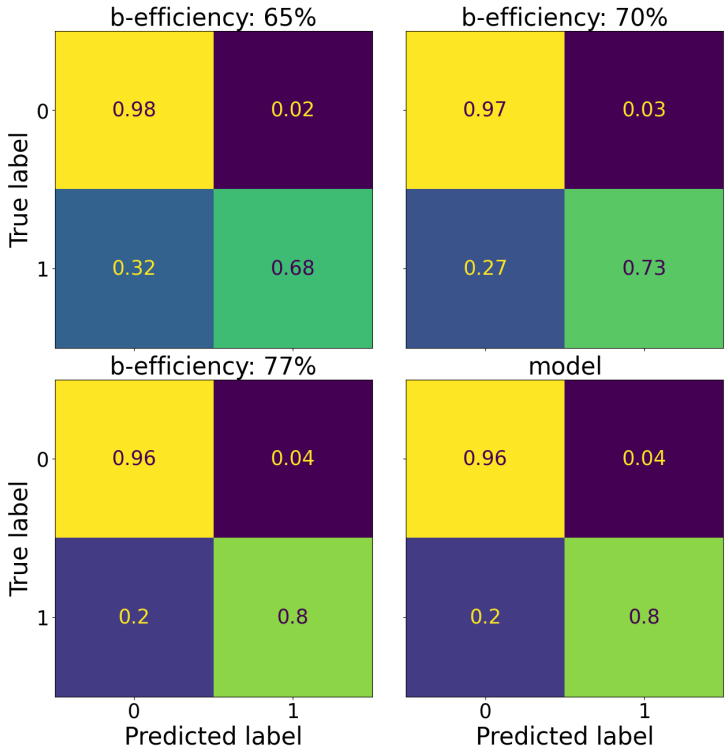

Figure 16: The figure shows smaller confusion matrices containing only true or false. The matrix in the top left corner predicts b-quarks based on the sixth working point of the GN2v01 tagger. The matrix in the top right corner predicts b-quarks based on the fifth and sixth working points of the GN2v01 tagger. The matrix in the bottom left corner predicts b-quarks based on the fourth, fifth and sixth working points of the GN2v01 tagger. The matrix on the bottom right shows the b-quark prediction based on the model. The performance is the same, showing that the model heavily relies on the tagging information.

higher than expected, which negatively affects performance in the reconstruction task. The only confusion between the three non-false classes is between $H \to b\bar{b}$ and $W \to q\bar{q}$, and none at all with the $W \to l, \nu$ class.

7.3 Reconstruction Results

Table 2 considers the validation data set, with the additional constraint that the two nodes originating from the Higgs boson need to be matched. This results in a total of 3,175 data points. The accuracy results are shown both overall and for each number of matched jets in an event. 'Highest edge score' selects the edge with the highest score based on the model. 'Logistic regression' selects the edge with the highest score from the logistic regression model. Overall, using logistic regression improves accuracy by around 3%.

In Table 3, the data set is filtered such that the two nodes from the $W \to l, \nu$ class exist.

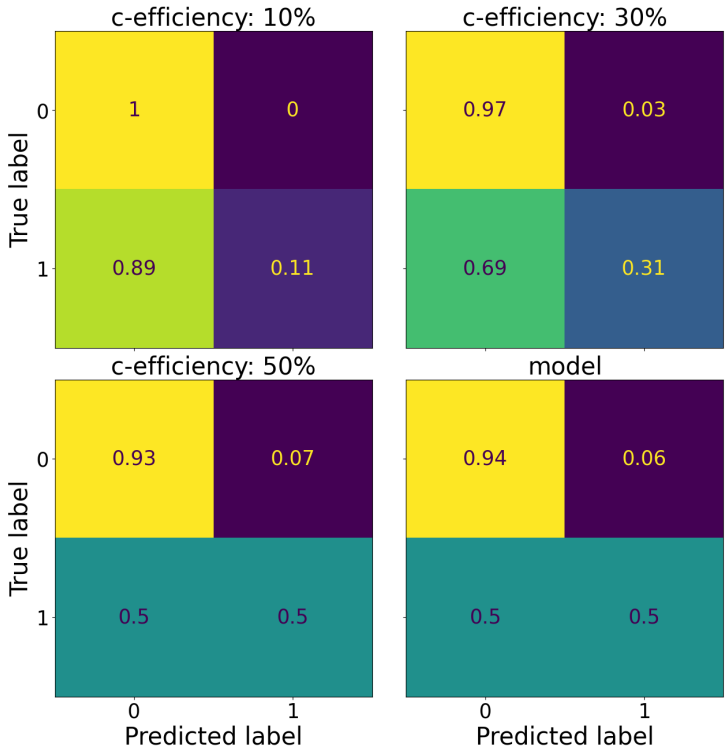

Figure 17: The figure shows smaller confusion matrices containing only true or false. The matrix in the top left corner predicts c-quarks based on the third working point of the GN2v01 tagger. The matrix in the top right corner predicts c-quarks based on the second and third working points of the GN2v01 tagger. The matrix in the bottom left corner predicts c-quarks based on the first, second and third working points of the GN2v01 tagger. The matrix on the bottom right shows the c-quark prediction based on the model. The performance is the same, showing that the model heavily relies on the tagging information.

This leaves 4,611 data points. Once again, the refined model performs better, with an accuracy improvement of around 2%.

Table 4 displays the data set filtered so that the edges of the $W \to q\bar{q}$ class exist. With three nodes, this is significantly smaller than the other two, containing 2,212 data points. The highest edge probability is not calculated because the average of all possible permutations would have to be taken. Instead, logistic regression was used.

Ultimately, all three classes should exist. Meaning, all seven jets must be matched. This makes the entire event reconstructable. A total of 896 events are selected in this way. A simple strategy is deployed to reconstruct the events completely. The edges with the highest chance of success is selected first, as can be seen in the seven matched row in the tables. Therefore, first, an edge from the $W \to l, \nu$ edges (connecting two nodes) is selected, then one from the $W \to q\bar{q}$ edges (connecting three nodes) and lastly from

Table 2: The table shows the proportion of events in which the $H \to b\bar{b}$ edge is correctly predicted. This is divided based on the number of jets matched in the events. Using the 'highest edge score', the edge that the model predicts the highest score is selected. The accuracy results are shown in the corresponding column. The 'logistic regression' column shows the accuracy of the model outputs when refined using an additional logistic regression model. Logistic regression achieves better results than the highest edge score option for all numbers of matched jets.

Matched jets	count	Highest edge score	Logistic regression
3	6	1	1
4	96	0.4896	0.5417
5	697	0.4046	0.4247
6	1480	0.2622	0.3101
7	896	0.2567	0.2656
combined	3175	0.3002	0.3310

Table 3: The table shows the fraction of events, where the edge $W \to l, \nu$ is correctly predicted. This is divided based on the number of jets matched in the events. Using the 'highest edge score', the edge that the model predicts the highest score is selected. The accuracy results are shown in the corresponding column. The 'logistic regression' column shows the accuracy of the model outputs when refined using an additional logistic regression model.

Matched jets	count	Highest edge score	Logistic regression
3	24	0.6667	0.5833
4	324	0.6111	0.6235
5	1401	0.5774	0.5832
6	1966	0.5468	0.5738
7	896	0.5123	0.5368
combined	4611	0.5545	0.5730

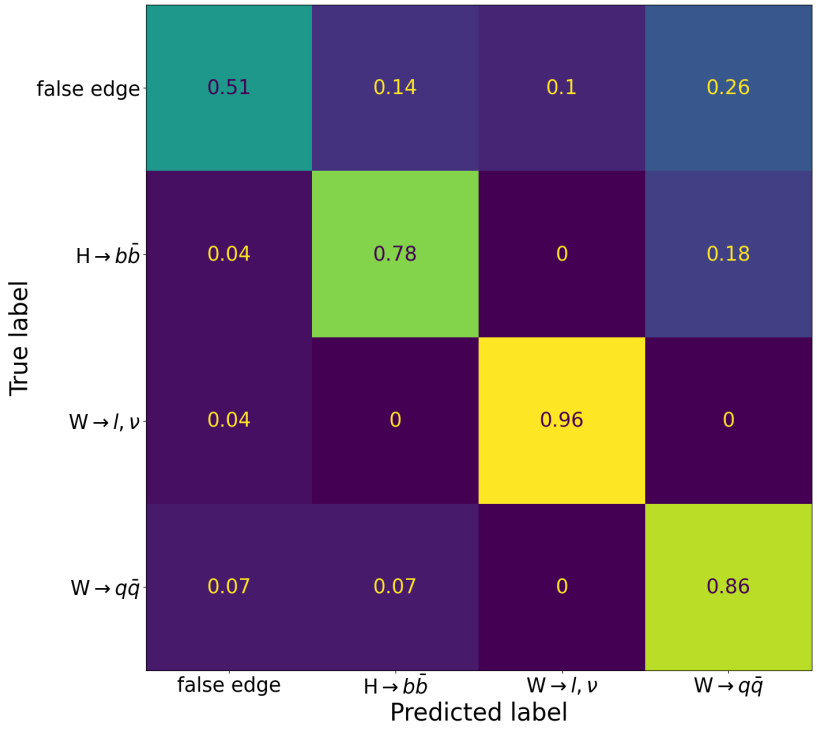

Figure 18: The figure shows the confusion matrix for the node class prediction. Most of the confusion is between false edges and the other classes. The only other confusion is between $H \to b\bar{b}$ and $W \to q\bar{q}$. It is also noteworthy that there is almost no confusion when the edge class is $W \to l, \nu$.

the remaining edges $H \to b\bar{b}$ (connecting two nodes). Some examples can be seen in the figures 20 and 21. The main confusion appears to be centred on the distinction between the bottom quarks. While the model has successfully learned which combinations of nodes are possible, it struggles with b-quarks. Most of the errors occur because the b-quarks are confused.

The left figure in figure 20 shows a complete reconstruction, where the model reconstructed everything correctly, including the node classes.

The reconstruction example on the right side of figure 20 is partially correct. Despite the node classes being wrong, the $W \to q\bar{q}$ class is correct. The model often confuses unmatched nodes with quarks in the reconstruction. In this example, the model predicted two unmatched nodes and only three b-quarks. This illustrates why an additional model for the reconstruction task is sensible, since an edge between an unmatched jet and a bottom quark is not possible with hard thresholds, given that the threshold for the b-quark is high enough.

In the both examples in figure 21, all of the edges are reconstructed incorrectly. In all of these examples, all non b quark nodes are predicted correctly and there was only confusion

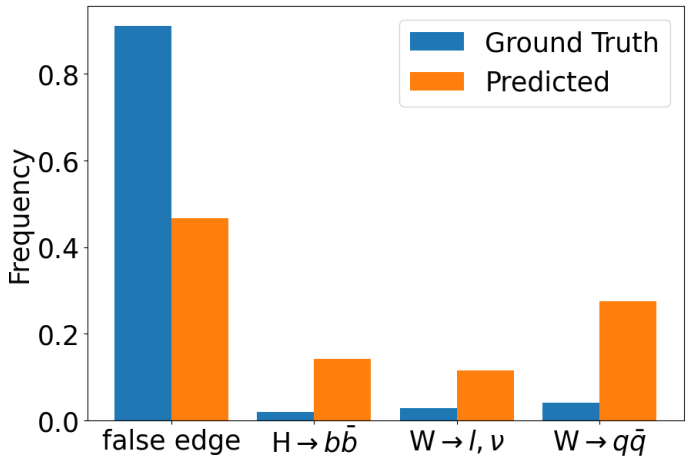

Figure 19: The figure shows the frequencies of the ground truth and predicted edge classes. This illustrates the impact of the low precision on the false edge class, since it is by far the largest class. The prediction for non-false classes contain at least four times the expected amount.

Table 4: The table shows the fraction of events in which the edge $W \to q\bar{q}$ is correctly predicted. This is divided based on the number of jets matched in the events. The 'logistic regression' column shows the accuracy of the model outputs when refined using an additional logistic regression model.

Matched jets	count	Logistic regression
4	13	0.2308
5	291	0.3058
6	1012	0.3360
7	896	0.3013
combined	2212	0.3174

between the b-quarks.

In all 896 events, the edges selected for $W \to l, \nu$ contained a lepton. This suggests that the model can distinguish b-quarks. Assuming the model can predict b-quarks with an accuracy of 100%, meaning it can distinguish them from unmatched jets, it would predict exactly four b-quarks. The probability of randomly selecting the correct b-quark corresponding to the class is $\frac{1}{4}$, which is significantly lower than the 0.5368 achieved by the model. Therefore, the model is two times more accurate than random guessing. Furthermore, the models accuracy of 53.68% is based on not all b-quarks being identified correctly.

The same applies for the $W \to q\bar{q}$ class. In 66.6% of all events, the model correctly predicts the two quarks. In 28.8% of events, one quark is predicted correctly, and in only around 4.6% of events, no quark is predicted correctly. Therefore, it is around twice as

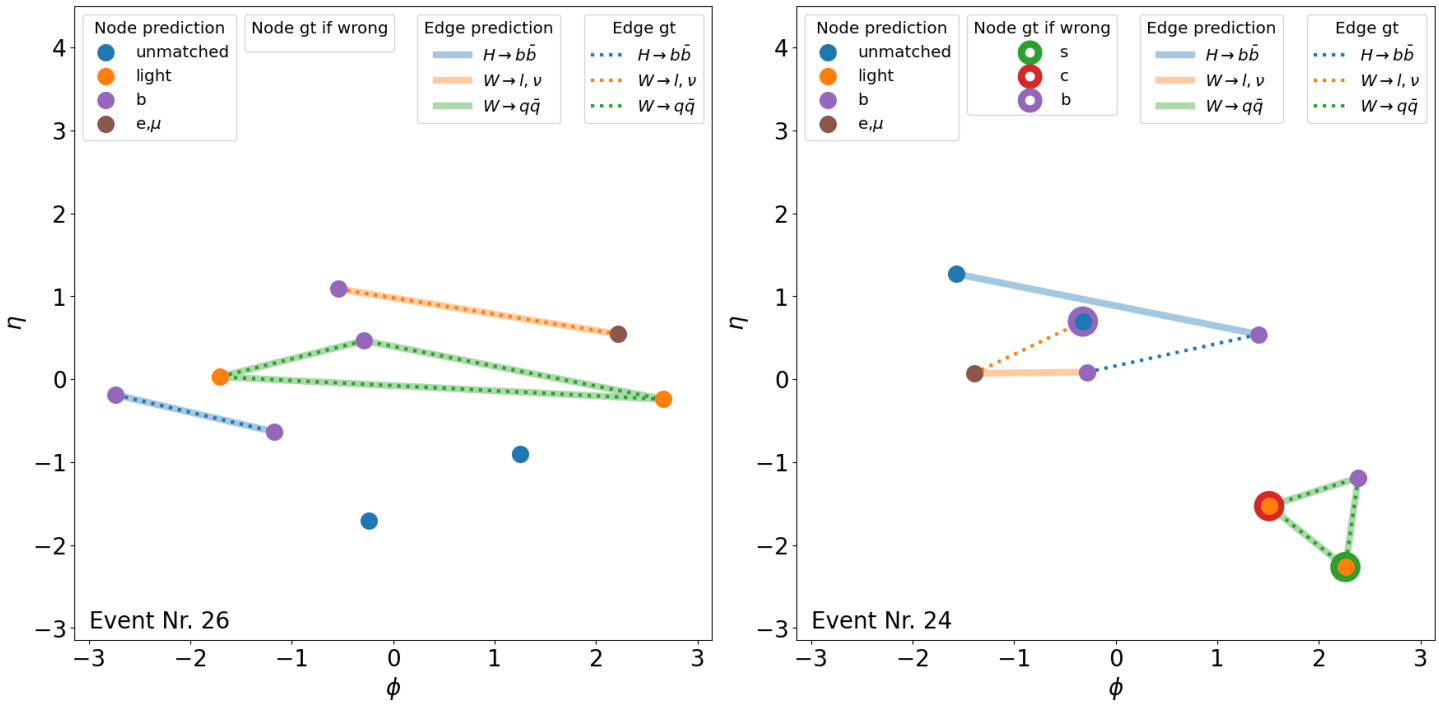

Figure 20: The left part of the figure shows an example of a correct reconstruction. Not only are all edges predicted correctly, but all nodes are also correctly classified. The figure on the right shows an example of a partially correct reconstruction. The $W \to q\bar{q}$ edge is predicted correctly, but the nodes are mixed up. The other edges are incorrectly predicted. The model confuses unmatched jets with bottom quarks, causing a connection to be made between an unmatched jet and a bottom quark, which should not be possible.

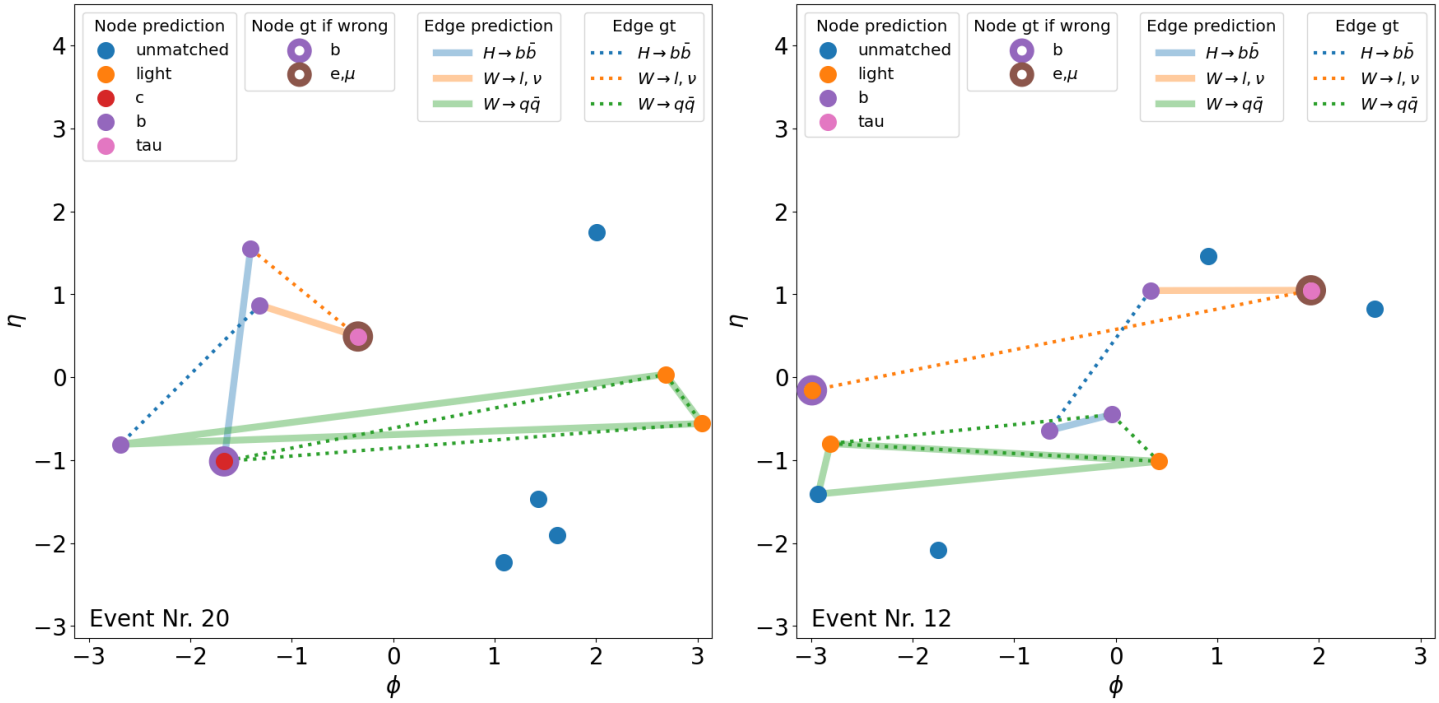

Figure 21: The left figure shows an example of an incorrect reconstruction. All edges are predicted incorrectly. However, all non-bottom-quark nodes are found correctly. The model is unable to allocate the bottom quarks to the correct edges. The same type of mixup of b quarks can be seen in the right figure.

Table 5: The table shows the reconstruction accuracies for each of the three non false edge classes.

Jet multiplicity	count	$H \to b\bar{b}$	$W \to l, \nu$	$W \to q\bar{q}$	all combined
== 7	260	30%	55%	43%	27%
== 8	285	26%	50%	29%	15%
>= 9	351	25%	55%	22%	13%
all	896	27%	54%	30%	17%

likely to select the correct b-quark. In approximately two-thirds of events, the model predicts the two correct non-b quarks. If then one of the four b quarks is selected at random, one would correctly predict only one-sixth of the events, as opposed to the 0.3013 that the model achieves. For the $H \to b\bar{b}$ class, the model has the lowest score, but it is still higher than a random guess. The probability of random guessing two out of the four b quarks is $\frac{1}{6} = \binom{4}{2}^{-1}$. The model achieves a score of 0.2656, which is better than random guessing but less significant than the scores of the other two classes. This also explains why the fraction of correctly predicted edges decreases as the number of matched jets increases. The more jets that are matched, the higher the chance of more bottom quarks. In 17.4% of all reconstructable events, all three edges are correctly predicted, as can be seen in table 5. In only 6.9% of events, two edges are correctly predicted. In the majority of events (44.5%), only one edge is predicted correctly, while in 31.1% of events, all edges are predicted incorrectly.

These results are worse compared to the results of SpaNet [13]. They achieve 49%, 69% and 49% accuracy for the respective edge classes and an overall accuracy of 40%. This is 20% higher than the results achieved in this thesis. However, the data SpaNet is trained on contains a magnitude of 10-100 times more events than used in this thesis. Additionally, the training times are significantly longer. SpaNet trained 24 hours on a single machine with AMD EPYC 7502 CPU and 4 Nvidia 3090 GPUs [13]. Training for this thesis was conducted on a basic laptop with a CPU for 90 minutes. This shows that similar results could be possible with larger training and some model improvements.

8 Summary and Outlook

This thesis attempts to reconstruct the $t\bar{t}(H \to b\bar{b})$ process. The term 'reconstruction' refers to the classification of the jets and the selection of those which originated from the

same parent particle. In the $t\bar{t}(H \to b\bar{b})$ process, the two b-quark jets originating from the Higgs boson belong together, as well as the two jets originating from the leptonic decay of the top quark ($W \to l, \nu$) and the three jets originating from the hadronic decay of the top quark ($W \to q\bar{q}$).

To identify these connections, we deployed a Graph Attention Network that was trained using simulated data from the ATLAS detector. The overall reconstruction accuracy of the complete process is 17.4%. The leptonically decaying top quark is reconstructed correctly in 54% of the events, while the hadronically decaying top quark and the $H \to b\bar{b}$ decay are reconstructed correctly in only 30% and 27% of the events, respectively. The model mostly mixes up the assignment of the bottom quarks.

For further improvement, some additional steps can be considered. For example, training on more data should improve the ability to generalise, and the training and validation loss/accuracy should converge. Having more data would also allow training to be restricted to events that can be fully or partially reconstructed. Furthermore, residual connections can be added. This allows more complex model structures to be implemented such as an adaptation of the U-Net [27] structure for graph convolutional networks or a masked autoencoder [28] structure. A masked autoencoder could potentially be used to predict missing jets in events with a matched jet multiplicity of less than seven. Another addition to the model could be to implement hyper-edges in order to predict the probabilities directly for the $W \to q\bar{q}$ edge class. Lastly, the logistic regression model that is used for refinement could be replaced with a more complex model, such as an MLP or DeepSets, based on the GNN model scores, for example, on the set of the top 10 edges.

In comparison to the model deployed in [13], the GNN performs significantly worse. However, with some of the suggested improvements and a proper comparison with regards to the amount of training events and training time, the approach proposed in this thesis might achieve similar results.

References

Bibliography

[1] ATLAS Collaboration. 'Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC'. In: *Phys. Lett. B* 716 (2012), pp. 1–29. DOI: 10.1016/j.physletb.2012.08.020.

[2] CMS Collaboration. 'Observation of a New Boson at a Mass of 125 GeV with the CMS Experiment at the LHC'. In: *Phys. Lett. B* 716 (2012), pp. 30–61. DOI: 10.1016/j.physletb.2012.08.021.

[3] S. L. Glashow. 'Partial-symmetries of weak interactions'. In: *Nucl. Phys.* 22.4 (1961), pp. 579–588. DOI: https://doi.org/10.1016/0029-5582(61)90469-2.

[4] S. Weinberg. 'A Model of Leptons'. In: *Phys. Rev. Lett.* 19 (1967), pp. 1264–1266. DOI: 10.1103/PhysRevLett.19.1264.

[5] A. Salam and J. C. Ward. 'Electromagnetic and weak interactions'. In: *Phys. Lett.* 13.2 (1964), pp. 168–171. DOI: https://doi.org/10.1016/0031-9163(64)90711-5.

[6] G. 'tHooft. 'Renormalization of massless Yang-Mills fields'. In: *Nucl. Phys. B* 33.1 (1971), pp. 173–199. DOI: https://doi.org/10.1016/0550-3213(71)90395-6.

[7] G.'t Hooft. 'Renormalizable Lagrangians for massive Yang-Mills fields'. In: *Nucl. Phys. B* 35.1 (1971), pp. 167–188. DOI: https://doi.org/10.1016/0550-3213(71)90139-8.

[8] G. 't Hooft and M. Veltman. 'Regularization and renormalization of gauge fields'. In: *Nucl. Phys. B* 44.1 (1972), pp. 189–213. DOI: https://doi.org/10.1016/0550-3213(72)90279-9.

[9] G. 't Hooft and M. Veltman. 'Combinatorics of gauge fields'. In: *Nucl. Phys. B* 50.1 (1972), pp. 318–353. DOI: https://doi.org/10.1016/S0550-3213(72)80021-X.

[10] S. L. Glashow, J. Iliopoulos and L. Maiani. 'Weak Interactions with Lepton-Hadron Symmetry'. In: *Phys. Rev. D* 2 (1970), pp. 1285–1292. DOI: 10.1103/PhysRevD.2.1285.

[11] D. J. Gross and F. Wilczek. 'Ultraviolet Behavior of Non-Abelian Gauge Theories'. In: *Phys. Rev. Lett.* 30 (1973), pp. 1343–1346. DOI: 10.1103/PhysRevLett.30.1343.

[12] H. D. Politzer. 'Reliable Perturbative Results for Strong Interactions?' In: *Phys. Rev. Lett.* 30 (1973), pp. 1346–1349. DOI: 10.1103/PhysRevLett.30.1346.

[13] M. J. Fenton et al. 'Reconstruction of unstable heavy particles using deep symmetry-preserving attention networks'. In: *Commun. Phys.* 7.1 (2024), p. 139.

[14] C. Gemme. *Latest ATLAS results from Run 2*. 2016. DOI: https://doi.org/10.48550/arXiv.1612.

[15] ATLAS collaboration. 'The ATLAS Experiment at the CERN Large Hadron Collider'. In: *JINST* 3 (2008).

[16] ATLAS collaboration. 'Measurement of the associated production of a top-antitop-quark pair and a Higgs boson decaying into a $b\bar{b}$ pair in pp collisions at $\sqrt{s} = 13$ TeV using the ATLAS detector at the LHC'. In: *Eur. Phys. J. C* 85.210 (2025).

[17] D. F. Crouse. 'On implementing 2D rectangular assignment algorithms'. In: *IEEE Transactions on Aerospace and Electronic Systems* 52.4 (2016), pp. 1679–1696.

[18] M. Zaheer et al. 'Deep Sets'. In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/f22e4747da1aa27e363d86d40ff442fe-Paper.pdf.

[19] ATLAS collaboration. 'Transforming jet flavour tagging at ATLAS'. submitted to Nature Communications. 2025. DOI: https://doi.org/10.48550/arXiv.2505.19689.

[20] M. M. Bronstein et al. 'Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges'. preprint: Work in progress. 2021. DOI: https://doi.org/10.48550/arXiv.2104.13478.

[21] P. Veličković et al. 'Graph Attention Networks'. In: *ICLR*. 2018. DOI: https://doi.org/10.48550/arXiv.1710.10903.

[22] S. Brody, U. Alon and E. Yahav. 'How Attentive are Graph Attention Networks?' In: *ICLR*. 2022. DOI: https://doi.org/10.48550/arXiv.2105.14491.

[23] T. N. Kipf and M. Welling. 'Semi-Supervised Classification with Graph Convolutional Networks'. In: *ICLR*. 2017. DOI: https://doi.org/10.48550/arXiv.1609.02907.

[24] D. K. Hammond, P. Vandergheynst and R. Gribonval. 'Wavelets on graphs via spectral graph theory'. In: *Appl. Comput. Harmon. Anal.* 30.2 (2011), pp. 129–150.

[25] F. Wu et al. 'Simplifying Graph Convolutional Networks'. In: *Proceedings of the 36th International Conference on Machine Learning*. Vol. 97. PMLR, 2019, pp. 6861–6871.

[26] D. P. Kingma and J. Ba. 'Adam: A Method for Stochastic Optimization'. In: *ICLR*. 2015. URL: https://arxiv.org/abs/1412.6980.

[27] O. Ronneberger, P. Fischer and T. Brox. 'U-Net: Convolutional Networks for Biomedical Image Segmentation'. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Springer International Publishing, 2015, pp. 234–241.

[28] K. He et al. 'Masked Autoencoders Are Scalable Vision Learners'. In: *IEEE, CVF, CVPR*. 2022.

A Appendix

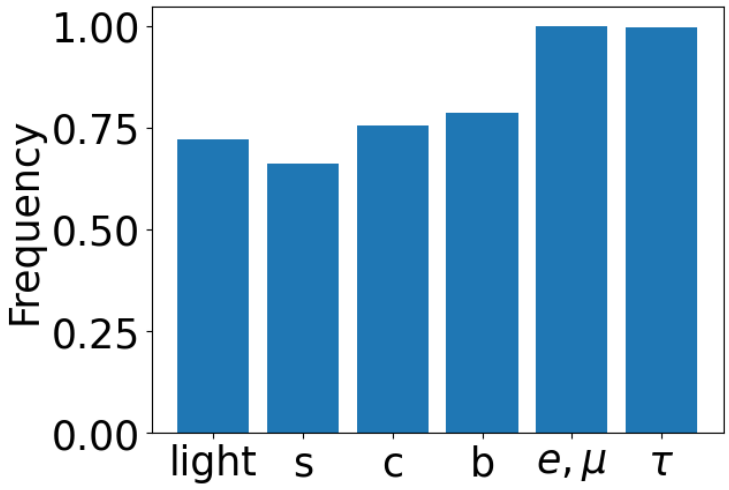

Figure A1: The figure shows the frequency, how often the classes get matched.

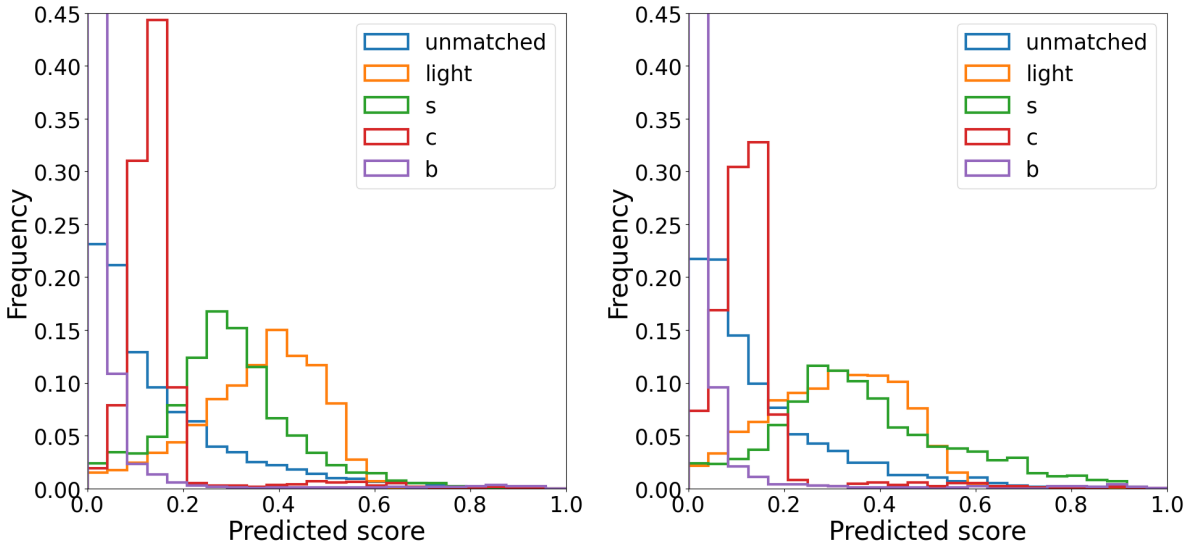

Figure A2: Left figure shows the histograms for the true label light quark. The right figure has the true label strange quark

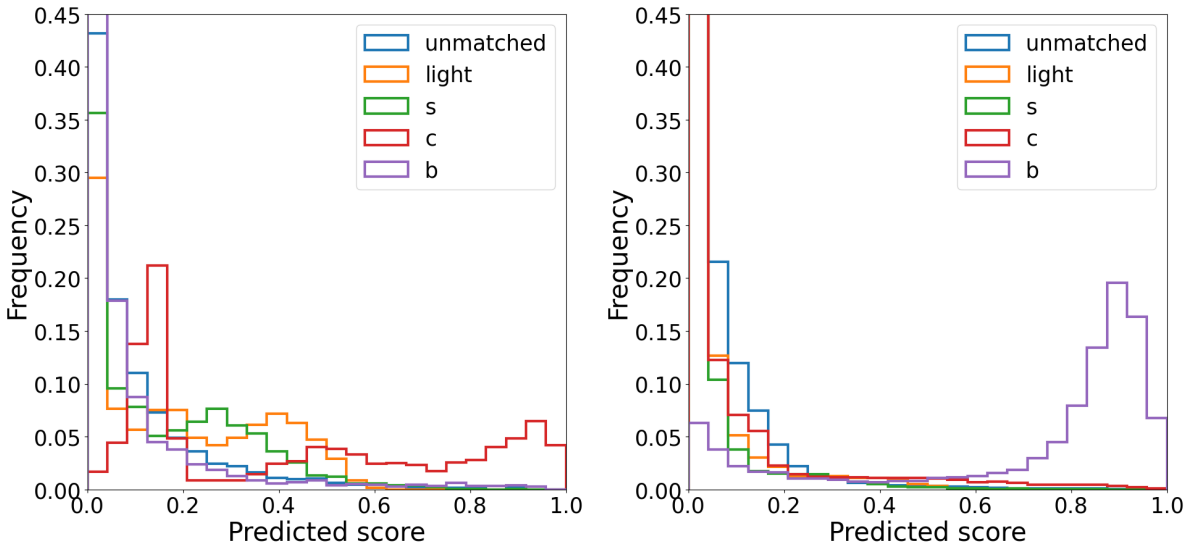

Figure A3: Left figure shows the histograms for the true label charm quark. The right figure has the true label bottom quark