



THE UNIVERSITY *of* EDINBURGH
**School of Physics
and Astronomy**

**Senior Honours Project
Distinguishing Longitudinally and
Transversely Polarised Weak Bosons using
Machine Learning**

Alexandra McAdam
September-November 2024

Abstract

A study into using Machine Learning methods, specifically Graph Neural Networks (GNNs), to distinguish polarisations of weak bosons is presented. Distinguishing polarisations provides a unique opportunity to confirm electroweak symmetry restoration, as longitudinal polarisation states vanish at restoration. The performance of GNNs trained on simulated $pp \rightarrow ZZ \rightarrow \ell\bar{\ell}q\bar{q}$ events with known longitudinal ($Z_L Z_L$) or transverse ($Z_T Z_T$) polarisation states was evaluated in both the lab and diboson rest-frame. A Deep Neural Network (DNN) was implemented for comparison. The lab-frame GNN achieved classification accuracies of 80% for $Z_L Z_L$ and 63% for $Z_T Z_T$, comparable to the rest-frame GNN and DNN. These results suggest that the GNN has potential but further development is required to fully exploit the GNN's unique ability to model complex relationships between particles.

Declaration

I declare that this project and report is my own work.

Signature:

Date: 29/11/2024

Supervisor: Dr. Dominik Duda

10 Weeks

Contents

1	Introduction	2
2	Background	4
2.1	The Standard Model of Particle Physics	4
2.2	Electroweak Theory	5
2.3	Machine Learning in Particle Physics	5
2.3.1	Deep Learning	6
2.3.2	Graph Neural Networks	7
3	Methodology	9
3.1	Event Selection	9
3.2	Monte Carlo Simulation and Data Pre-Processing	11
3.3	Node Feature Selection	12
3.4	GNN Architecture, Implementation and Training	13
3.4.1	Architecture	13
3.4.2	Implementation	14
3.4.3	Training	14
3.5	Performance and Evaluation Metrics	15
3.5.1	Model Comparison	16
4	Results and Discussion	17
4.1	Lab-Frame GNN	17
4.2	Diboson Rest-Frame GNN	19
4.3	DNN	21
5	Conclusion and Outlook	22
6	Acknowledgements	23
A	Monte Carlo Dataset	27
B	DNN Input Variables	28

1 Introduction

The Standard Model (SM) of particle physics is one of the most successfully tested theories in physics and with the discovery of the Higgs boson in 2012, it was confirmed that particles gain their mass through the Higgs mechanism [1, 2]. A key principle in the SM is the unification of the electromagnetic and weak forces at high energies into a single force, the electroweak (EW) force. As the Universe cooled following the Big Bang, the EW force underwent a process called spontaneous symmetry breaking, which is described by the Higgs mechanism, causing the electromagnetic and weak forces to become distinct [3]. As a consequence of EW symmetry breaking, the W and Z bosons, which are the mediators of the weak force and the only massive bosons, gained their masses and their longitudinal polarisation states [3].

High-energy particle collider experiments, such as those at the Large Hadron Collider (LHC) at CERN, are already exploring EW physics at and beyond EW restoration scales [4]. A potential way to probe EW restoration is by distinguishing the polarisation states of the W and Z bosons, as the longitudinal polarisation states will vanish once EW symmetry is restored [4]. Additionally, being able to determine polarisation state fractions at high energies can provide insight into Beyond Standard Model (BSM) physics, theories of which predict different fractions [5].

The polarisations of W and Z bosons are not directly measurable and hence a method of distinguishing polarisations from experimental data is needed. The aim of this project was to use a modern machine learning (ML) method to do this. Previous studies have also employed ML methods including Boosted Decision Trees (BDT), Deep Neural Networks (DNN) and theory-informed Wide Neural Networks [6, 4, 7, 8, 9, 10]. This project utilised Graph Neural Networks (GNNs), a type of ML method which performs computations on graph-structured data to exploit relational properties, which is a novel approach to distinguishing polarisations that could lead to improved performance over the other ML methods previously used. Figure 1 shows the fractions of various polarisation states at different boson transverse momenta (p_T). The GNN can be trained to learn the relationships between polarisation and p_T or other kinematic variables to infer the polarisation without being explicitly programmed.

This project focused on distinguishing the polarisations of Z boson pairs, known as dibosons (ZZ), which are produced in high-energy particle colliders mainly through quark annihilation and to a lesser extent, gluon-gluon fusion [11]. The specific focus was on events where a high p_T diboson decays to a lepton-antilepton pair and a quark-antiquark pair. Leading-order Feynman diagrams of these events are shown in Figure 2. Due to the strong force, each quark forms a stream of highly collimated particles called a jet before detection and at high p_T , these merge to form a single large radius jet [12]. Therefore, the experimental signature of these events are a lepton, an antilepton and a large radius jet. The GNN was trained on experimentally-available variables from simulated events where the initial polarisation state is known. The performance of the GNN was then evaluated, comparing performance using data in the lab reference frame and diboson rest reference frame. As a benchmark for GNN performance, a DNN, a ML technique which has been used in previous studies, was also implemented in this project.

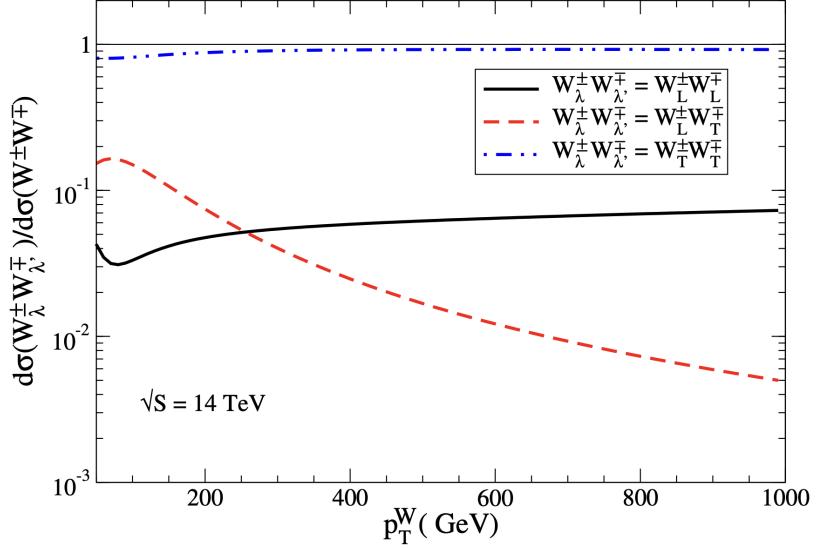


Figure 1: The ratio of differential cross section for a given polarisation and total differential cross section for $pp \rightarrow W^\pm W^\mp$ events are plotted against the W boson transverse momentum (p_T^W). The diboson polarisations shown are transverse (blue dash-dot-dot), mixed longitudinal and transverse (red dash), and longitudinal (black) [4].

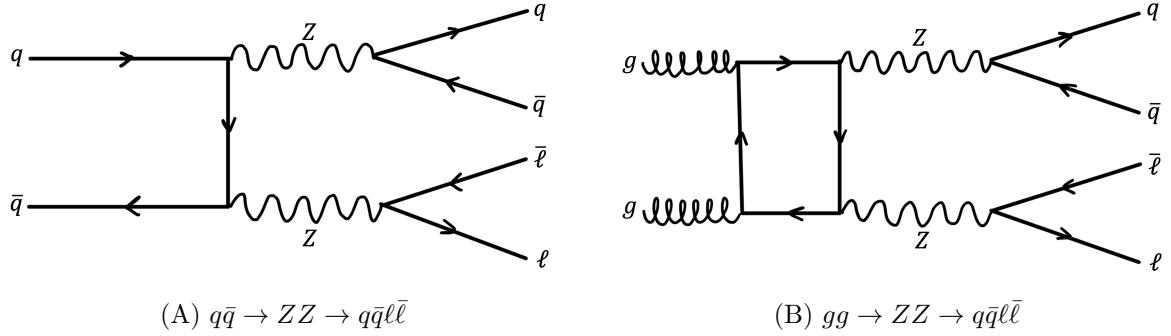


Figure 2: Leading-order Feynman diagrams for ZZ production and decay to a quark-antiquark pair ($q\bar{q}$) and lepton-antilepton pair ($\ell\bar{\ell}$). (A) shows ZZ production through quark annihilation and (B) shows ZZ production through gluon fusion.

This report is structured as follows: Section 2 details the relevant background of the Standard Model of particle physics and Electroweak Theory as well as an introduction into the use of Machine Learning in particle physics and the techniques used in this project. Section 3 describes the methods employed in this project, from event selection and data simulation to GNN development and evaluation metrics. Section 4 presents the project results alongside the corresponding discussion and Section 5 concludes with a summary of the project and outline for future research directions.

2 Background

2.1 The Standard Model of Particle Physics

The Standard Model is the foundation of particle physics, it describes the properties, behaviours and interactions of fundamental particles. Each fundamental particle has intrinsic properties such as mass, electric charge, spin and colour. Particles with half-integer spin (e.g. $s = \frac{1}{2}, \frac{3}{2}$) are known as fermions. These particles make up matter and can be divided into two groups, quarks and leptons. Particles with integer spin (e.g. $s = 0, 1$) are known as bosons, which are responsible for mediating the fundamental forces. Figure 3 illustrates the particle content and mass spectrum of the Standard Model.

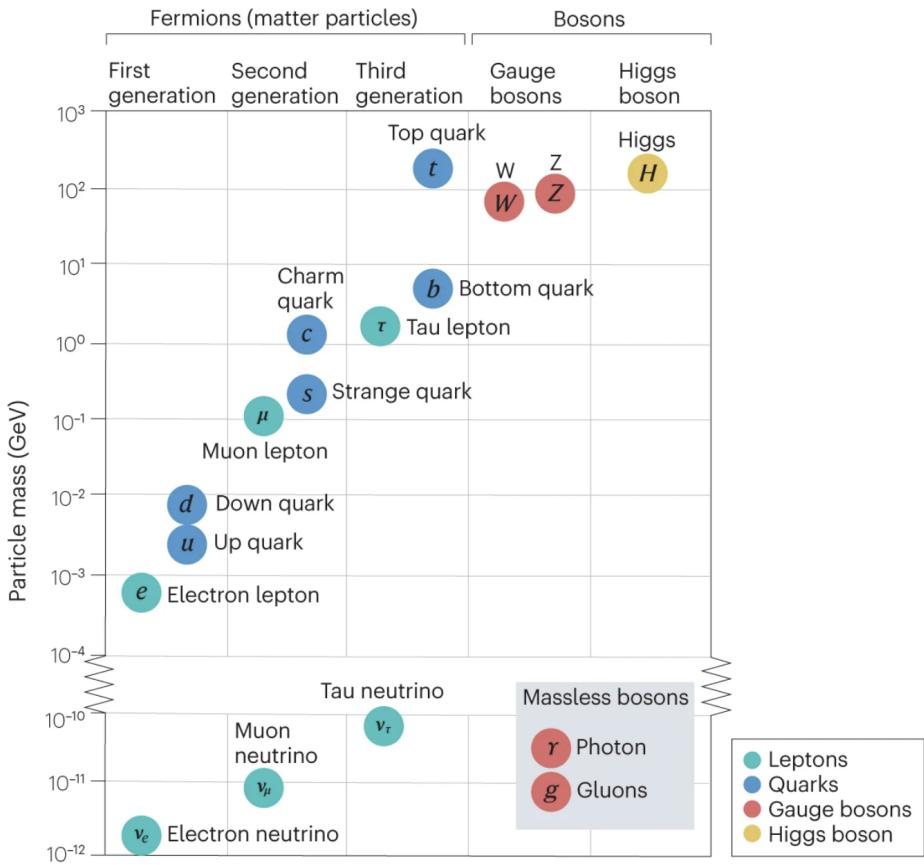


Figure 3: The elementary particles of the Standard Model are divided into fermions and bosons. The fermions are subdivided into generations, with particle masses increasing with increasing generation. The bosons are divided into the gauge bosons, which are the force carriers, and the Higgs boson [13].

There are four fundamental forces: gravity, electromagnetism, the strong force and the weak force. Photons mediate the electromagnetic force, they are massless and travel at the speed of light (c). Gluons mediate the strong force, which is responsible for binding protons and neutrons together to form nuclei. The W and Z bosons mediate the weak force which governs processes such as beta decay. Unlike photons and gluons, which are massless, the W and Z bosons are massive, with experimentally measured masses of 80.3692 ± 0.0133 GeV for the W bosons and 91.1880 ± 0.0020 GeV for the Z boson

[14]. Additionally, the W bosons are electrically charged (W^\pm). To explain why these bosons have mass, the Higgs mechanism was proposed, which necessitated the addition of the Higgs boson to the Standard Model [15]. The experimental discovery of the Higgs boson in 2012 corroborated the Higgs mechanism, providing a greater understanding of how particles, including the W and Z bosons, acquire mass [1, 2].

Bosons also have a polarisation which refers to the orientation of their spin relative to their momentum. Polarisation is defined by the scalar product $h = \vec{S} \cdot \frac{\vec{p}}{|\vec{p}|}$, where \vec{S} represents spin and \vec{p} is momentum. For spin-1 bosons, such as W and Z bosons, there are three polarisation states; two transverse modes, corresponding to $S_z = \pm 1$, and one longitudinal mode, corresponding to $S_z = 0$ [16]. The longitudinal polarisation states of bosons also originate from the Higgs mechanism, this process is explained for W and Z bosons in Section 2.2.

2.2 Electroweak Theory

A central principle of the Standard Model is the unification of the electromagnetic and weak forces at high energies to become a single electroweak force [3]. As the universe cooled following the Big Bang, these forces became distinct, with the weak force becoming short-ranged and acquiring three massive bosons while the electromagnetic force remained long-ranged and with one massless boson [12].

According to the Glashow-Salam-Weinberg (GSW) model, the unified theory of the electroweak force, electroweak interactions are described by the symmetry group $U(1) \otimes SU(2)$. Symmetry groups represent the mathematical transformations under which quantum fields remain invariant. The $U(1) \otimes SU(2)$ group necessitates four massless gauge bosons which are related to the photon, Z^0 , W^+ and W^- [17]. As the universe cooled, spontaneous symmetry breaking occurred and the W and Z bosons gained mass through their interaction with the Higgs field [16]. Additionally, these bosons gained longitudinal polarisation states by absorbing the three massless degrees of freedom of the Higgs field, which are known as Goldstone bosons [16].

According to the Goldstone Boson Equivalence Theorem, at sufficiently high energies, i.e. significantly beyond the W and Z boson masses where electroweak symmetry is restored, the longitudinal polarisation states will vanish by reverting to Goldstone bosons and leave only transverse polarisation states of the W and Z bosons [3]. Therefore, distinguishing polarisation states of W and Z can provide a unique opportunity to test electroweak symmetry restoration in high-energy experiments.

2.3 Machine Learning in Particle Physics

Machine Learning (ML) enables computers to learn from data to make predictions or discover patterns without being explicitly programmed. This has a wide variety of applications and in high-energy particle physics (HEP) ML has become an essential tool in data analysis. For example, at the Large Hadron Collider (LHC) the amount of data collected in experiments is extremely large, as much as 50 terabytes per second, due to the number of collisions and the relative rarity of collisions which produce interesting

products [18]. The data is highly complex and multi-dimensional and combined with the sheer volume of data this means that traditional data analysis techniques become inefficient. Previously, ML techniques such as Boosted Decision Trees were prevalent in HEP which helped improve analyses but also had limited capability when compared to traditional methods [19]. The relatively recent practical development of an ML sub-field known as ‘deep learning’ has enabled the training of algorithms which can generalise to even greater dimensionality and complexity than previous ML algorithms, kick-starting a new era in ML and also in HEP data analysis [19]. The significance of this development is evidenced by the awarding of the 2024 Physics Nobel Prize to the researchers behind the foundations of deep learning [20].

2.3.1 Deep Learning

Deep learning utilises artificial neural networks, which are computational models inspired by the brain’s structure and function of neurons and synapses. In deep learning, these networks are trained on data to identify and learn complex patterns and relationships between features, allowing the model to make predictions based on new, unseen data. A Deep Neural Network (DNN) consists of an input layer and an output layer with multiple intermediate layers between them as shown in Figure 4 [21]. The intermediate layers are called ‘hidden’ layers because they do not directly receive input or produce output data, instead they transform and represent data through learned weights and biases, which the model adjusts to best capture relationships within the data [21]. Layers consist of units, these are artificial neurons which process input values through an activation function, such as the ReLU or sigmoid function, shown in Figure 5, which introduce non-linearity and allow the network to approximate complex mappings [21]. Layers can vary in type and function for example, a fully-connected layer connects each unit to every unit in the subsequent layer and a convolutional layer is good at processing spatial data like images, each bring different inductive biases to the network [22].

In HEP experiments, deep learning is used to help infer the underlying physics processes from data. To train artificial neural networks to do this, either supervised or unsupervised learning is used. This main difference between these methods is that in supervised learning the ‘truth labels’ of the training data are known and in unsupervised they are not. The truth labels depend on the model objective for example, in a classification problem the truth label is the known class to which an item belongs to and the aim of the model is to determine this correctly. Supervised learning is most commonly used in HEP for tasks such as jet classification and signal identification [23, 24], although unsupervised learning has also been used for clustering and anomaly detection [25, 26].

For supervised learning in HEP, simulated data is usually used to train models since the true physical parameters (the truth label) of the particles involved are known whereas in experimental data these are often unknown [27]. The model calculates a measure of accuracy, called the loss value, by comparing its predictions to these truth labels [28]. A large loss value indicates the prediction is far from the truth label. The aim of model training is to minimise the loss value as much as possible in a process called optimisation. A common optimisation method is stochastic gradient descent which updates the model in small steps following a method called backpropagation [28]. A key challenge during

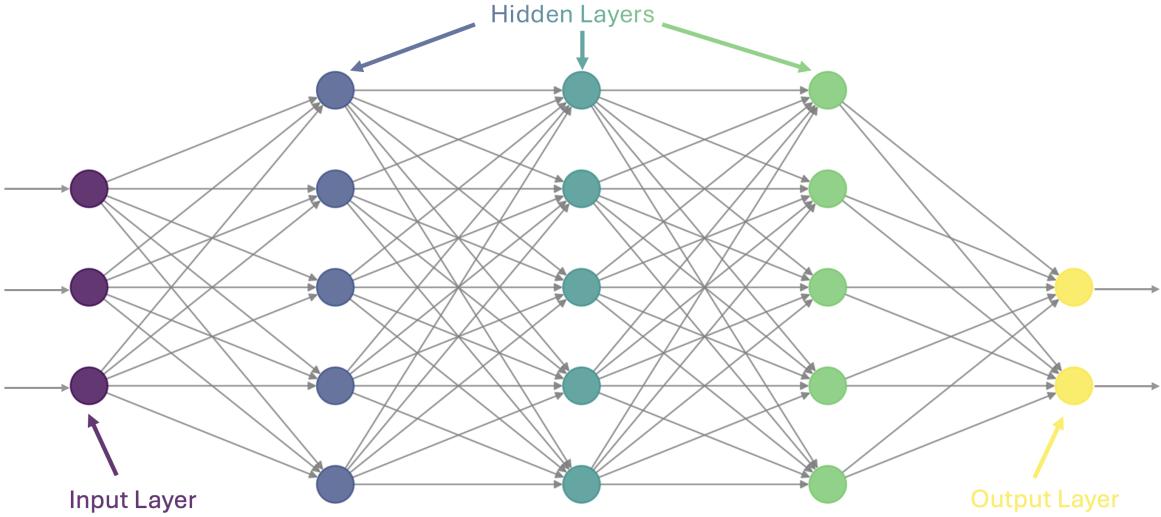


Figure 4: The architecture of a Deep Neural Network (DNN) is composed of input, hidden and output layers. Each layer is composed of a number of computational units, represented here as circles, which process either the input data if in the input layer or the output of the preceding layer's units if in the hidden or output layers. This diagram shows a fully-connected DNN where the output of each unit becomes the input of each unit in the next layer shown by the gray arrows.

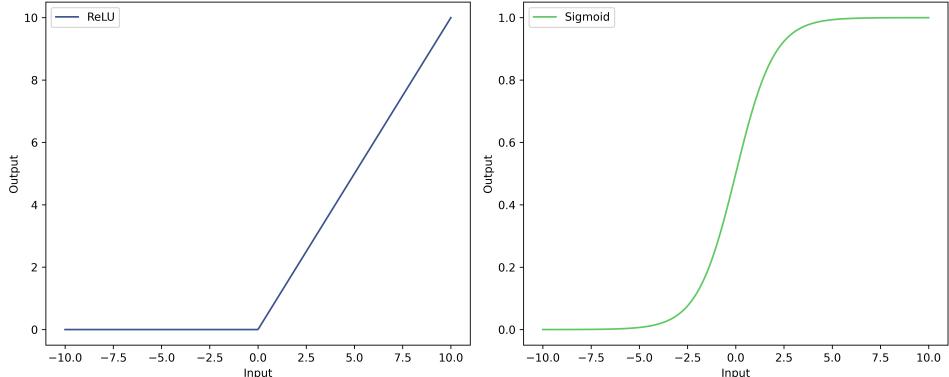


Figure 5: Activation functions are used to introduce non-linearity into artificial neural networks. Left plot shows ReLU activation function defined as $\max(0, x)$ and right plot shows Sigmoid activation function defined as $\frac{1}{1+e^{-x}}$.

training is avoiding overfitting, this is when the model performs well for the training data but cannot generalise to unseen data. Regularisation techniques such as dropout are used to reduce overfitting [29]. The choice of loss function, optimisation method and other hyperparameters are a central part of the design of deep learning algorithms and will vary considerably depending on task and data.

2.3.2 Graph Neural Networks

Graph Neural Networks (GNNs) are a type of DNN which are designed for graph-structured data. Graphs are a data structure characterised by objects (nodes) connected

with pairwise relationships (edges) as shown in Figure 6. Information can be encoded in both the nodes and edges of a graph, allowing complex relationships to be modelled between objects. This information is known as features and a feature vector is associated with each node or edge. Graphs are particularly suited to describe data generated from HEP experiments in which the objects, for example, particles produced in a proton-proton collision, are highly relational [27].

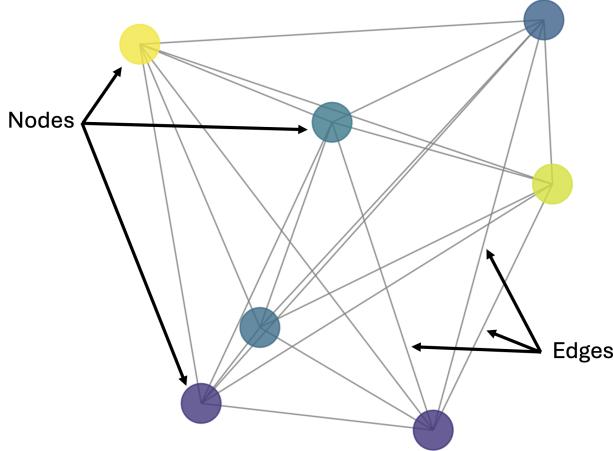


Figure 6: Graph data structures are characterised by nodes and edges which can both have information stored in them. Nodes represent objects and edges represent the relationships between objects.

A typical GNN architecture for a classification task is shown in Figure 7. GNNs use graph-specific neural network blocks which are composed from multiple layers and operations [30]. GNN blocks process both node and edge information, iteratively aggregating and transforming features from neighbouring nodes and edges across the graph to update the current node’s feature vector. This process is called neural message passing and it enables nodes to receive information from their local neighbourhood, capturing local structure and relationships within the graph [13]. The architecture typically includes six core functions: three update functions and three aggregation functions, applied at the

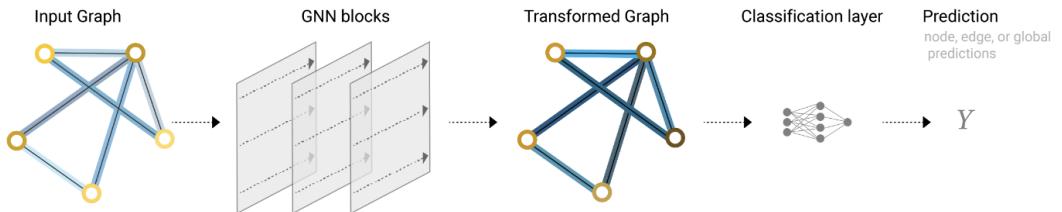


Figure 7: Typical architecture of a Graph Neural Network (GNN) for classification. Input graphs are fed into specialised GNN blocks which capture local structure and relationships within the graph to produce a transformed graph with updated features that incorporate information from neighbouring nodes and edges. Transformed graphs are then passed through a classification layer to pool the graph information and produce a final prediction [30].

edge, node, and global (graph) levels [22]. The update functions adjust features based on incoming messages, while the aggregation functions summarise information from surrounding nodes or edges, ensuring that each node’s features incorporate relevant local or global context [22]. After passing through the GNN blocks, a process called pooling is used to combine all information from the transformed graph to create predictions at graph, node or edge level [30].

3 Methodology

As discussed in Section 2.3.1, deep learning models can ‘learn’ from data during training to exploit information optimally, leading to greater performance than traditional approaches. To assess the potential of Graph Neural Networks (GNNs) for distinguishing the polarisation states of weak bosons, a GNN was trained on graphs constructed from the simulated 4-momentum of decay products of longitudinally and transversely polarised ZZ bosons. A GNN was chosen due to its ability to capture the inherent structure and relationships between particles, which should enable more accurate predictions than other deep learning architectures.

This section details how the GNN was developed, beginning with event selection, dataset generation and feature choice. This is followed by an explanation of the GNN architecture, implementation and training. Lastly, the metrics used to evaluate the GNN performance are detailed.

3.1 Event Selection

Weak bosons can be produced in high-energy particle colliders and decay through a number of different channels. To distinguish their polarisation states, firstly events where they are produced need to be identified. This project focussed on distinguishing Z polarisations in highly-boosted diboson events in the semi-leptonic decay channel, the reasons for which are explained below. The kinematic properties of the Z boson decay products are related to the Z polarisation and can be used to infer whether the boson was longitudinally or transversely polarised [31].

Highly-Boosted: For energies around and above electroweak symmetry restoration scales, particles have high transverse momenta and are hence highly relativistic, exhibiting significant Lorentz boosts. This causes particles produced in decays to be highly collimated in the lab-frame and to appear clustered in detectors.

Diboson: Diboson events involve the simultaneous production and decay of boson pairs (ZZ). Dibosons are mainly produced through quark annihilation with a small contribution from gluon-gluon fusion, as was shown in Fig. 2. Diboson events produce more data about each Z than single boson events and thus enable greater sensitivity for distinguishing polarisations [32].

Semi-Leptonic Decay Channel: In the semi-leptonic decay channel, one boson decays to a lepton-antilepton pair and the other decays to a quark-antiquark pair. The

quarks produced in these decays only exist freely for around $10^{-15}m$ before their strong interaction becomes so great that new quark-antiquark pairs are produced, these then bind to form many various hadrons [12]. This process is known as hadronisation and it results in the formation of jets, which are a cone of highly collimated particles that are the experimental signature of quarks [12]. In a highly-boosted decay of a Z boson to a quark-antiquark pair, the resultant jets can merge into a single large-radius jet as shown in Figure 8.

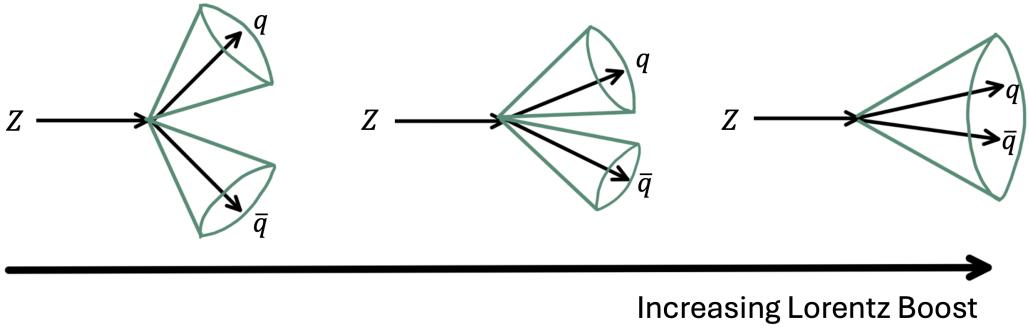


Figure 8: The green cones represent highly collimated streams of particles called jets which are produced by the hadronisation of quarks. As Lorentz boost increases with increasing particle momentum the individual jets merge to produce a single large radius jet in the lab-frame.

Studying this channel balances sensitivity with manageable backgrounds. For leptonic decays at high momenta, the leptons produced are highly collimated and can create indistinguishable signals in the detector leading to a loss of information. Whereas jets produced by hadronic decays can be effectively reconstructed at high momenta, helping to minimise information loss [33]. However, events from the fully hadronic channel are not considered in this project due to overwhelming background levels from multijet events, jets initiated by single quarks or gluons, which make them challenging to analyse [33]. Reduced sensitivity is introduced in both leptonic and semi-leptonic channels through τ lepton decays. As the heaviest lepton, τ leptons have a short decay time ($10^{-13}s$) and typically decay into other leptons or hadrons before detection unlike e^- and μ leptons [16]. Hadronic τ decays resemble jets which adds complexity to data analysis [34]. τ decays also produce neutrinos that cannot be directly detected in particle collider experiments due to their weak interactions [16]. A single neutrino's presence can be inferred from missing transverse energy, as the transverse energy of collision products should sum to zero (assuming the centre of mass frame) and any deviation from this indicates undetected particles [16]. However, if multiple neutrinos are produced information is lost.

By focussing on highly-boosted diboson events in the semi-leptonic decay channel, the aim was to balance the positive effects of greater sensitivity, effective jet reconstruction and reduced backgrounds with the negative effects of indistinguishable lepton signals and reduced information from τ decays to maximise the performance of the GNN at and beyond electroweak symmetry restoration, i.e. at energies significantly above the W and Z masses of 80GeV and 91GeV.

3.2 Monte Carlo Simulation and Data Pre-Processing

For initial GNN training, the dataset consisted of $Z_T Z_T \rightarrow \ell \bar{\ell} q \bar{q}$ and $Z_L Z_L \rightarrow \ell \bar{\ell} q \bar{q}$ events, where Z_T and Z_L represent transverse and longitudinal Z boson polarisation states, $\ell \bar{\ell}$ represents a lepton-antilepton pair and $q \bar{q}$ represents a quark-antiquark pair. A diagram of a proton-proton collision which produces these events is shown in Figure 9. This subset of highly-boosted diboson semi-leptonic decays was chosen to simplify development of the GNN. With only two initial states ($Z_T Z_T$ and $Z_L Z_L$), the GNN will be a binary classifier, predicting either transverse or longitudinal polarisation states. Expanding to more initial states such as $Z_T Z_L$ and $Z_L Z_T$ would require a larger training dataset, increasing computational demand, and add complexity to the GNN. Starting with this reduced dataset allowed for efficient model evaluation.

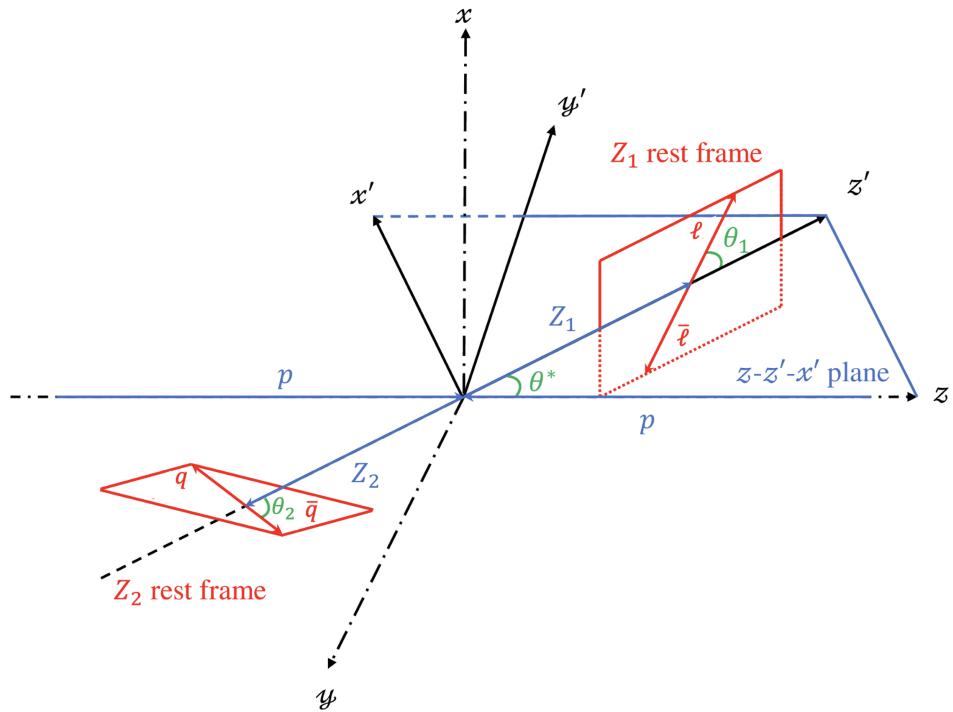


Figure 9: Diagram of a $pp \rightarrow ZZ \rightarrow \ell \bar{\ell} q \bar{q}$ event. The (x, y, z) coordinates are the lab reference frame (dot-dashed axis) with the z -axis aligned with beam direction. The (x', y', z') coordinates are defined so z' is in the direction of the Z_1 boson with respect to the diboson rest-frame, x' describes the reaction plane containing the lab z -axis and z' -axis, and y' is given the right-hand rule. θ^* is the Z production angle in the diboson rest-frame and θ_1 and θ_2 are the Z decay angles in the Z_1 and Z_2 rest-frames [35].

To implement a supervised learning method as described in Section 2.3.1, simulation was used to generate physically-realistic data to train the GNN. The generated dataset included the truth labels of $Z_T Z_T$ or $Z_L Z_L$, defined in the partonic rest frame, which are not directly accessible from experimental data. Monte Carlo simulation packages were used to generate a diverse sample of events, accounting for the probabilistic nature of particle interactions and decays, from a proton-proton collision at centre-of-mass energy $\sqrt{s} = 14\text{TeV}$. `MadGraph_aMC@NLO 3.5.1` was used to compute the matrix elements, which describe the probabilities of particle interactions [36]. `LHAPDF 6.3.0` was used to evaluate the parton distribution functions (PDFs), which describe the probability of

finding partons (quarks or gluons) in a hadron as a function of the fraction of the proton’s momentum carried by the parton [37]. **Pythia** 8 was then used to simulate the parton showers, the cascade of particles and radiation produced in a high-energy collision, producing the final states and momenta of collision products [38]. Generated events were subsequently passed through the fast detector simulation package **Delphes** 3.5.0 to model the response of a generic high-energy particle collider detector [39]. The resulting dataset consisted of high-level observables, which include reconstructed objects such as jets, and detailed information about the particle constituents of each event. The following requirements were placed on the simulated data to select $pp \rightarrow ZZ \rightarrow \ell\bar{\ell}q\bar{q}$ events:

- One large radius jet with $p_T > 200\text{GeV}$, $m > 50\text{GeV}$ and $|\eta| < 2.0$
- Exactly two charged leptons, each with $p_T > 25\text{GeV}$ and $|\eta| < 2.5$

where p_T is transverse momentum, m is invariant mass and η is pseudorapidity which is defined below.

The specific variables included in the dataset provided for this project are listed in detail in Appendix A. Each event included a 1-dimensional array of observables and a 2-dimensional array of constituent information. For each constituent, the kinematic information (i.e. 4-momentum), charge and particle type (lepton or hadron) was included.

In high-energy particle colliders, spherical coordinates (r, ϕ, θ) are commonly used to describe particle trajectories. Additionally, rapidity defined as $y = \ln(\frac{E+p_L}{E-p_L})$ where E is the particle energy and p_L is the momentum parallel to the beam axis, is used due to its Lorentz invariance. For highly-boosted particles, rapidity approximates pseudorapidity, defined as $\eta = -\ln(\tan\frac{\theta}{2})$. In the dataset, the 4-momentum $p_\mu = (E, \vec{p})$ is given in terms of transverse momentum p_T (momentum perpendicular to beam axis), azimuthal angle ϕ and pseudorapidity η . For highly-boosted events, particle masses are negligibly small compared to their energies. Constituent particle energy can therefore be approximated as $E = p_T \cosh \eta$. The variables p_T , ϕ and η are given in both the lab-frame and diboson rest-frame. All variables are given in natural units, which sets constants to one, for example the speed of light ($c = 1$) and the reduced Plank constant ($\hbar = 1$).

3.3 Node Feature Selection

The dataset must be processed into input graphs for the GNN. Each event in the dataset was represented as a graph where the constituent particles form the graph nodes with the associated node feature vectors constructed from both from the observables and constituent information. Selecting appropriate node features is critical to the GNN’s ability to make accurate predictions. The choice of features was based on those used in ParticleNet, a GNN developed for jet classification, which has close parallels with the task of polarisation classification [40]. The high accuracy of ParticleNet showed that its choice of features is successful, which suggested that similar features could work well for this GNN.

The GNN node features are shown in Table 1. The first seven variables were calculated from the 4-momentum of each constituent particle and the final two taken directly from

the dataset. These features aim to capture subtle kinematic and relational properties within the data which the GNN can exploit to determine the initial boson polarisation state.

Feature	Description
$\Delta\eta$	Difference in pseudorapidity ($\eta_{jet/lepton} - \eta$)
$\Delta\phi$	Difference in azimuthal angle ($\phi_{jet/lepton} - \phi$)
$\log p_T$	Logarithm of particle transverse momentum
$\log E$	Logarithm of particle energy
$\log\left(\frac{p_T}{p_{T(jet/lepton)}}\right)$	Logarithm of transverse momentum ratio
$\log\left(\frac{E}{E_{(jet/lepton)}}\right)$	Logarithm of energy ratio
R	Angular separation ($\sqrt{\Delta\eta^2 + \Delta\phi^2}$)
isLep	Lepton indicator (isLep==1 if lepton, isLep==0 otherwise)
charge	Electric charge

Table 1: Each node, representing a constituent particle, in a GNN input graph is associated with a node feature vector of these variables. Depending on the particle type and the frame of reference different dataset variables are used to calculate the features.

3.4 GNN Architecture, Implementation and Training

3.4.1 Architecture

The GNN architecture is based on ParticleNet, a Dynamic Graph Convolutional Neural Network (DCGNN) designed for jet classification as mentioned above. DCGNNs are a type of GNN that dynamically re-construct graphs after every node feature update [40]. ParticleNet uses GNN blocks called EdgeConv which specialise in capturing local geometric features from graphs [41]. Each EdgeConv block consists of multiple EdgeConv layers, where each layer applies a linear transformation, followed by batch normalisation and then a ReLU activation. The number of events from the dataset which are processed at one time is called the batch size and batch normalisation is a technique which stabilises the training process [42].

The architecture starts with four stacked EdgeConv blocks. A k-nearest-neighbours (knn) algorithm is used to construct the graphs for each block, the first EdgeConv block calculates the distances in $\phi - \eta$ space between nodes and connects each node to its 16 nearest neighbours to form the initial graph. Subsequent EdgeConv blocks construct the graph using the transformed node feature vectors. A visualisation of the knn graphs formed by the first EdgeConv block is shown in Figure 10. The EdgeConv blocks are followed by a global average pooling layer to aggregate the transformed features over the graph. This is followed by a fully-connected layer with 128 units and dropout to reduce overfitting. Finally, another fully-connected layer with two units generates the binary graph classification.

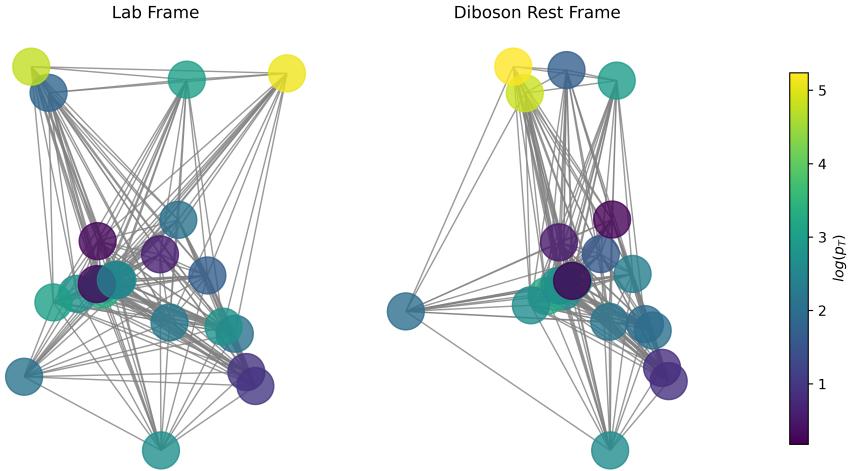


Figure 10: Visualisations of GNN input graphs in $\phi - \eta$ space generated using the k-nearest neighbours algorithm with $k=16$. The same event is shown in the lab-frame and the diboson rest-frame.

3.4.2 Implementation

The GNN is implemented using Python Machine Learning libraries Scikit-learn and PyTorch, with PyTorch Geometric for handling graph operations [43, 44, 45]. The implementation was adapted from the GitHub repository given in Ref. [46], which implements a GNN based on ParticleNet.

3.4.3 Training

Once implemented, the GNN was trained on the simulation dataset which was split into training, validation and testing subsets. The training and validation subsets consist of 90% of the full dataset, of which 90% is for training and the remaining 10% for validation. The testing subset was not used in the training process so that it was statistically independent from the training and validation subsets to provide an unbiased evaluation of the GNN performance.

Training was performed using the supervised learning method (as described in section 2.3.1), with loss calculated using the cross-entropy loss function which is suited to classification tasks [47], and optimised using the Adam stochastic gradient descent method [48]. Each training iteration of the dataset is called an epoch and the number of epochs was capped at 10. To help prevent overfitting and minimise unnecessary computations, early stopping was implemented. This stops training before the maximum number of epochs if the validation accuracy had not improved in the past three consecutive epochs and it saves the best model up to that point. Validation accuracy, which measures the GNN’s prediction accuracy on unseen data, was used to ensure the model can generalise effectively. The batch size was set to 256 to balance learning stability with computational efficiency.

The GNN has many hyperparameters, tunable values, that can influence its performance. However, since the implementation was adapted from another GNN, which was trained on a similar dataset, learning rate and dropout probability were identified as the most important hyperparameters to tune and other hyperparameters were kept constant.

Learning rate determines how much the model is updated during training. If the learning rate is too small, the GNN could take a very long time to reach peak validation accuracy however, if the learning rate is too high, the GNN might not converge to the optimal solution and thus have reduced validation accuracy [21]. Therefore, the best learning rate must be found to minimise the number of training epochs for computational efficiency whilst still allowing the GNN to reach peak validation accuracy.

Dropout is a layer in the GNN which works by randomly ignoring the inputs from some fraction, the dropout probability, of units in the preceding layer. This helps reduce overfitting to the training data, helping the model to generalise to unseen data [29]. Dropout probability was tuned since it is important that the GNN can perform well on unseen data.

GNN models were trained using different combinations of the values shown in Table 2 to determine the optimal combination of hyperparameters.

Hyperparameter	Search Values
Learning Rate	0.001, 0.005, 0.01
Dropout	0.3, 0.4, 0.5

Table 2: Values used to perform a course search over tunable hyperparameters. Nine different GNN models, with different combinations of learning rate and dropout, were trained to find which achieved the best performance.

3.5 Performance and Evaluation Metrics

GNN performance was analysed in several ways to provide insights into how well the GNN is able to predict boson polarisations. The evaluation metrics detailed below were applied to the GNN outputs based on the testing dataset. Each metric highlights a different aspect of model performance.

Confusion Matrix: The confusion matrix is a table used to clearly visualises the performance of classification models. It displays the level of correct and incorrect predictions for each true class, longitudinal ($Z_L Z_L$) or transverse ($Z_T Z_T$).

Receiver Operating Characteristic (ROC) Curve and Area Under the Curve (AUC): The ROC curve is a plot of the true-positive rate (TPR) against false-positive rate (FPR) which are defined as:

$$TPR = \frac{TP}{TP + FN}, FPR = \frac{FP}{FP + TN} \quad (1)$$

where TP is the number of true positive classifications, and FN, FP, TN the equivalents for false negative, false positive and true negative respectively. For the GNN, $Z_L Z_L$ can be considered the positive class and $Z_T Z_T$ the negative class. The area-under-curve (AUC) is equal to 1 for a perfect classifier and will be around 0.5 for a completely random classifier. This gives a way to evaluate the model performance as a whole.

Discriminant: The discriminant is a calculated value which assess the confidence of the GNN in its predictions. It was defined as:

$$D_{Z_L Z_L} = \log \left(\frac{p_{Z_L Z_L}}{p_{Z_T Z_T}} \right) \quad (2)$$

where $p_{Z_L Z_L}$ and $p_{Z_T Z_T}$ are the probabilities the GNN assigns to the $Z_L Z_L$ or $Z_T Z_T$ classes for a given event. Higher values of $D_{Z_L Z_L}$ indicate confidence in the longitudinal class and lower values in the transverse class, values around zero indicate the GNN is not confident in either.

Operating Point: An operating point is a specific value of the discriminant ($D_{Z_L Z_L}$) which is used to define an optimal threshold at which to separate classes to achieve a desired value of a metric such as accuracy. The operating points can be found for different values of invariant diboson mass (m_{ZZ}), to assess how the model performs at different invariant masses (corresponding to different energies). Invariant diboson mass is calculated using the 4-momentum

$$m_{ZZ} = \sqrt{(E_{jet} + E_{lepton})^2 - \vec{p}_{jet} \cdot \vec{p}_{lepton}} \quad (3)$$

where $E_{jet/lepton}$ is the energy and $\vec{p}_{jet/lepton}$ is the momentum of the jet or lepton system.

3.5.1 Model Comparison

GNN models were trained using node features either in lab-frame or diboson rest-frame to asses how changing reference frame affected the performance.

A feed-forward Deep Neural Network (DNN) was also implemented to classify boson polarisations using high-level features as input, as explored in earlier studies [7, 8, 9], as a comparison to the GNN performance. The GNN should show better performance than the DNN since it can use inherent structure and relationships between constituent particles which the DNN doesn't have access to. The DNN was implemented using code adapted from the GitHub repository given in Ref. [46], Machine Learning libraries Scikit-learn and PyTorch were used [43, 44]. The DNN architecture is composed of an input layer, then 4 hidden layers each with 64 units that consist of a fully-connected layer, batch normalisation and the ReLU activation. This is then followed by dropout and an output layer.

The DNN takes a vector of observables as input, since the dataset included 26 different observables, the optimal subset of these which maximised DNN performance was found using a metric called Mutual Information (MI). MI between two variables is a measure of their dependency, it is equal to zero if they are completely independent and higher values indicate higher dependency [49]. By calculating the MI of each observable with

the DNN output, the optimal features were found. The DNN was initially trained on all observables, then these were ranked based on their MI with the DNN output. The observable with lowest MI was removed from the input variables and the DNN retrained. This was repeated until the DNN validation accuracy peak was discovered. The final input vector consisted of 22 observables which are detailed and ranked by importance in Appendix B. Transverse momentum p_T values and Z rest-frame decay angles $\cos \theta_1$ and $\cos \theta_2$, had the highest MI which was expected as longitudinal and transverse states cause different momentum imbalance in jet constituents and different decay angles [32].

4 Results and Discussion

This section details the results and discussion of GNN performance evaluation.

4.1 Lab-Frame GNN

The GNN was trained on a dataset with a total of 171,756 events. 102,256 events had $Z_L Z_L$ initial states and 69,500 events had $Z_T Z_T$ initial states. The optimal hyperparameters were found to be a learning rate of 0.005 and a dropout probability of 0.3, with which the GNN achieved a validation accuracy of 0.73. These values were found through training multiple models with the node features in the lab-frame. Evaluation metrics for the performance of the GNN on the testing dataset are presented in Figure 11.

Confusion Matrix Fig.11(A): The GNN classified $Z_L Z_L$ correctly in 80% of cases and $Z_T Z_T$ in 63%. A true $Z_L Z_L$ was mistaken for a $Z_T Z_T$ in 20% of cases and a true $Z_T Z_T$ for a $Z_L Z_L$ in 37%. The GNN is better at predicting $Z_L Z_L$ and generally favours classifying events as $Z_L Z_L$ as shown by the higher rate of $Z_T Z_T$ misidentifications. This is potentially caused by the imbalance in the numbers of each class in the training dataset. The GNN can learn to expect more $Z_L Z_L$ events and when unsure about which class to predict for a given event, chooses $Z_L Z_L$ simply on the basis that it expects more. This is called model bias and is problematic because the GNN should rely only on the graph information. When presented with experimental data, where class imbalance can be quite significant and with an expected zero $Z_L Z_L$ events above electroweak symmetry restoration, this bias can lead to more inaccurate predictions.

ROC Curve Fig.11(B): The ROC curve demonstrates reasonably good overall performance, with an area under the curve (AUC) of 0.795. This is well above the random classifier value (AUC=0.5) but leaves room for improvement towards the perfect classifier value (AUC=1).

Discriminant Distribution Fig.11(C): The discriminant distribution shows $Z_L Z_L$ and $Z_T Z_T$ peaking at positive and negative values respectively however, both peaks are quite close to zero which indicates low confidence in many predictions. The overlap in distributions shows the number of events that were misclassified. The best operating point for classification was found by maximising balanced accuracy. Balanced accuracy is a metric which accounts for imbalanced classes and is used when both classes are

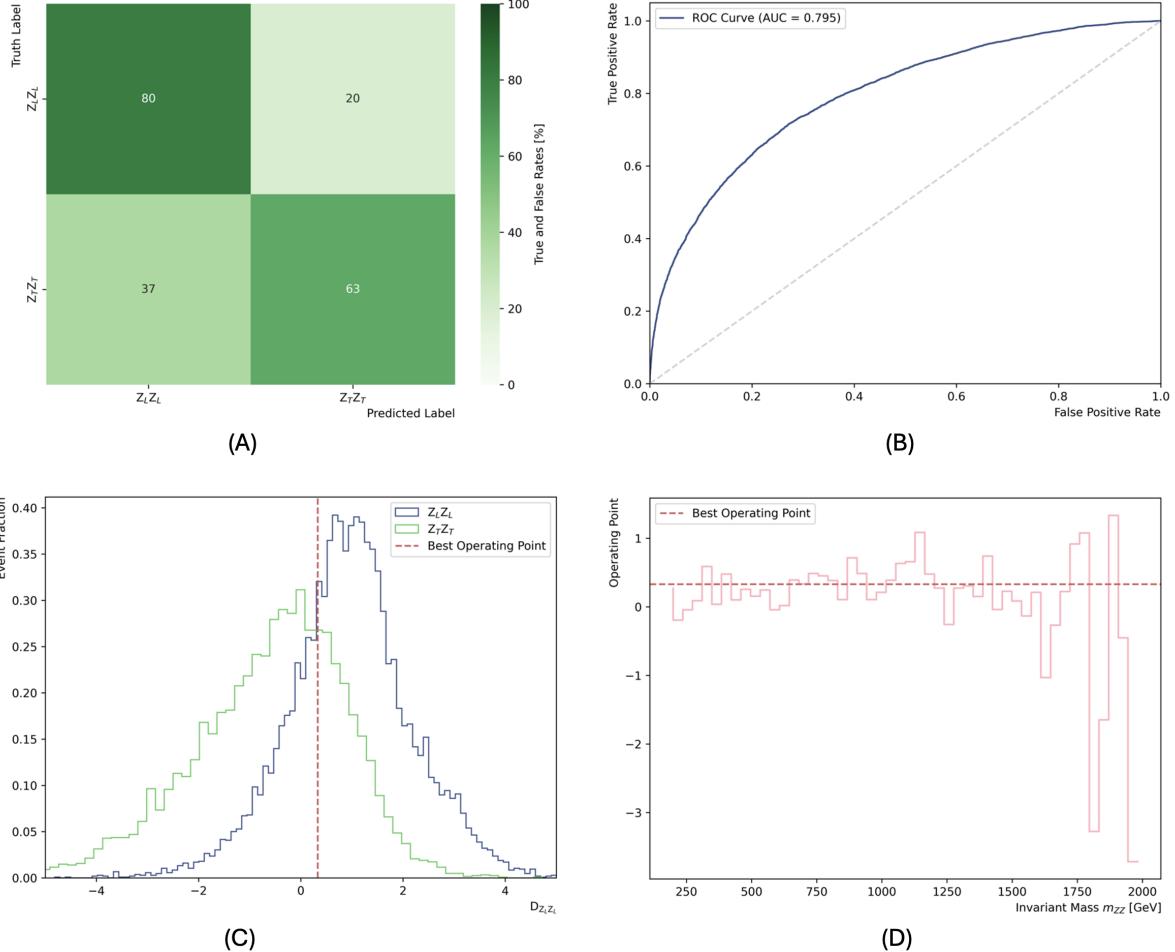


Figure 11: Evaluation metrics, as described in Section 3.5, for the GNN trained using a learning rate of 0.005, dropout probability of 0.3 and node features in the lab-frame. Plot (A) is the confusion matrix. Plot (B) is the ROC curve with the AUC value inset. The grey dashed line shows the ROC curve for a GNN with random predictions. Plot (C) is the discriminant $D_{Z_LZ_L}$ plotted separately for each true class, $Z_L Z_L$ (blue) and $Z_T Z_T$ (green). The best operating point to maximise balanced accuracy is shown by the red dashed line. Plot (D) is the operating points for different diboson invariant mass intervals with the best operating point again shown by the dashed red line.

considered equally important. Balanced accuracy is defined as:

$$Balanced - Accuracy = \frac{TPR + TNR}{2} \quad (4)$$

where TPR is the true positive rate and TNR is the true negative rate. Using this metric, the best operating point was found to be $D_{Z_LZ_L} = 0.33$. Classifying events with $D_{Z_LZ_L} > 0.33$ as longitudinally polarised and those with $D_{Z_LZ_L} < 0.33$ as transversely polarised achieves the highest balanced accuracy.

Invariant Mass Dependent Operating Points Fig.11(D): The optimal operating points were evaluated across 50 equal intervals of $m_{ZZ} \in [180, 2000]\text{GeV}$. As invariant mass increases the optimal operating points fluctuate further from the best operating point. This behaviour can be interpreted by looking at the number of events in each mass interval (see Fig.12), sparser numbers of events at higher masses correspond to

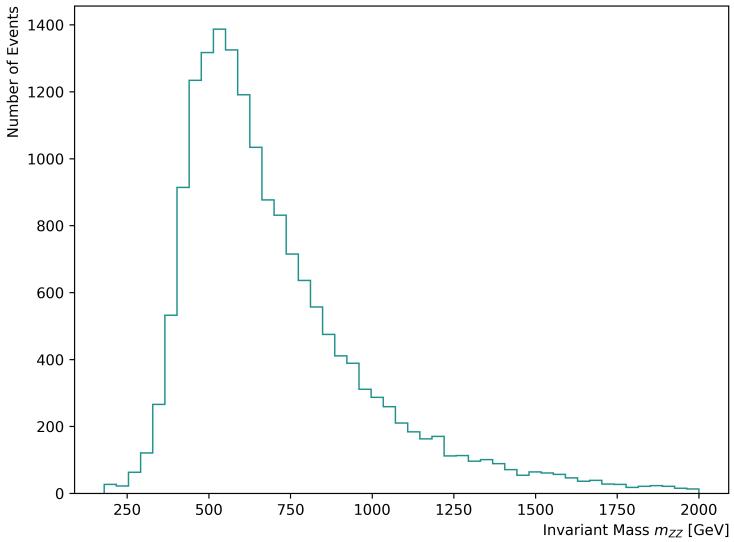


Figure 12: Histogram showing the number of events in each invariant mass (m_{ZZ}) interval for the GNN testing dataset.

larger fluctuations from the best operating point.

Overall, the GNN demonstrates reasonably good performance as shown by correctly classifying $Z_L Z_L$ in 80% of cases and $Z_T Z_T$ in 63%, with a bias towards $Z_L Z_L$ events due to imbalanced training data. An AUC of 0.795 indicates reliable classification however, the discriminant distributions reveal low confidence in many predictions. Optimal operating point fluctuates at higher m_{ZZ} due to reduced event statistics and hence there is less reliable performance at these values.

4.2 Diboson Rest-Frame GNN

Using the same dataset and hyperparameters, the GNN was retrained with the node features shifted into the diboson rest-frame by transforming the values of p_T , ϕ and η . Figure 13 presents the evaluation metrics for this GNN.

Confusion Matrix Fig.13(A): This GNN classified $Z_L Z_L$ correctly in 77% of cases and $Z_T Z_T$ in 65%. A true $Z_L Z_L$ was mistaken for a $Z_T Z_T$ in 23% of cases and a true $Z_T Z_T$ for a $Z_L Z_L$ in 35%. This GNN also shows a bias towards $Z_L Z_L$, it is unlikely that this is caused by reference frame effects and is most likely due to imbalanced training data as in the lab-frame GNN.

ROC Curve Fig.13(B): The ROC curve has an AUC of 0.788, slightly lower but comparable to the lab-frame GNN's AUC of 0.795 indicating they have similar performance.

Discriminant Distribution Fig.13(C): The discriminant distributions for $Z_L Z_L$ and $Z_T Z_T$ are comparable to those of the lab-frame GNN, with peaks at positive and negative values respectively. Again they are close to zero, indicating low prediction confidence. The best operating point, found from maximising balanced accuracy, was $D_{Z_L Z_L} = 0.25$.

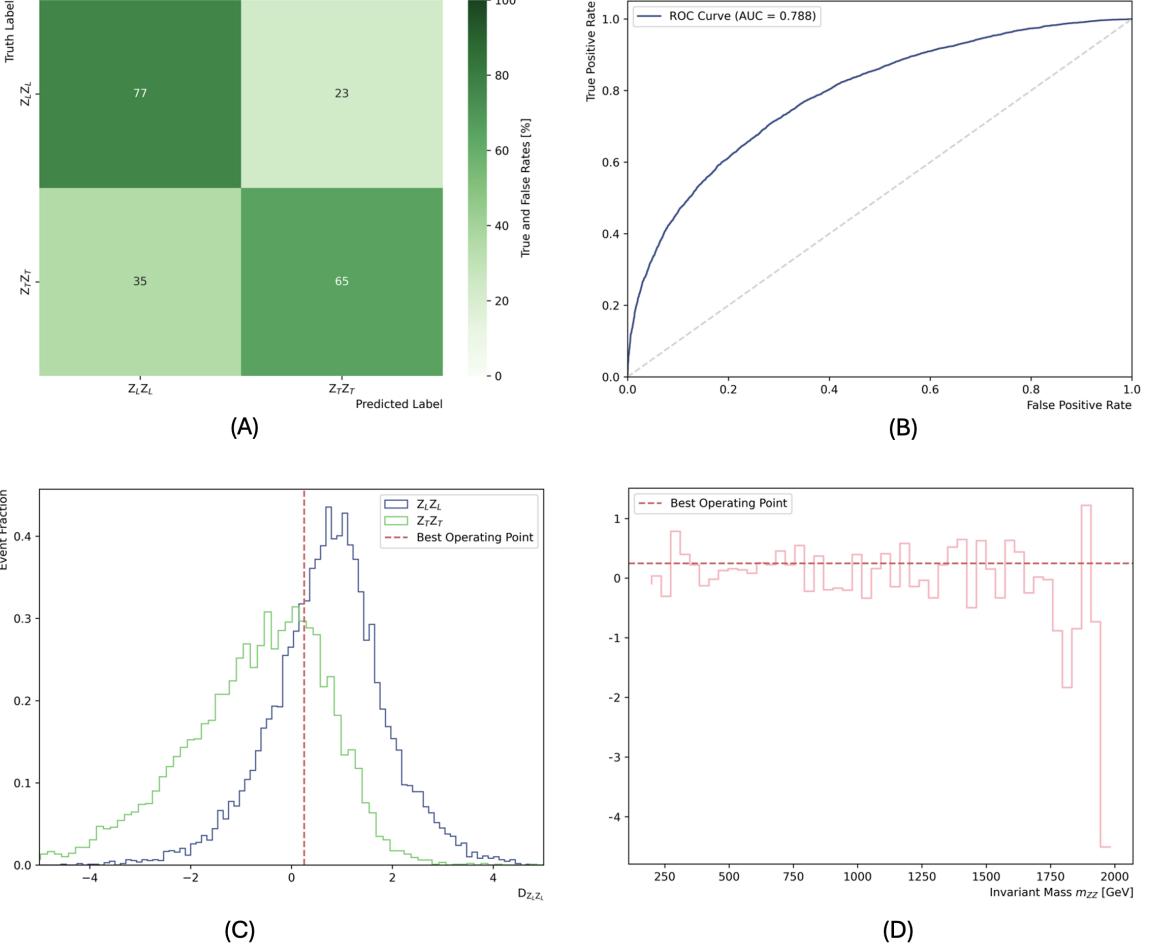


Figure 13: Evaluation metrics for the GNN trained using a learning rate of 0.005, dropout probability of 0.3 and node features in the diboson rest-frame. Plot (A) is the confusion matrix. Plot (B) is the ROC curve with the AUC value inset. The grey dashed line shows the ROC curve for a GNN with random predictions. Plot (C) is the discriminant $D_{Z_LZ_L}$ plotted separately for each true class, Z_LZ_L (blue) and Z_TZ_T (green). The best operating point to maximise balanced accuracy is shown by the red dashed line. Plot (D) is the operating points for different diboson invariant mass intervals with the best operating point again shown by the dashed red line.

Invariant Mass Dependent Operating Points Fig.13(D): The optimal operating points were calculated for 50 equal intervals in $m_{ZZ} \in [180, 2000]\text{GeV}$. As invariant mass increases, the optimal operating points fluctuate further from the best operating point, consistent with what is observed for the lab-frame GNN. However, deviations only become more severe above around 1750GeV. This implies that event sparsity at higher m_{ZZ} may not be the only factor contributing to operating point fluctuations. The diboson rest-frame GNN may perform better at higher m_{ZZ} but this requires further investigation.

These results are notable as it was expected that shifting to the diboson rest-frame would have a larger effect on the GNN performance and instead they were highly comparable. However, it should be noted that the hyperparameters optimised for the lab-frame GNN, learning rate and dropout probability, may not be optimal for the diboson rest-

frame GNN and require further tuning.

4.3 DNN

The DNN was trained using the same dataset as the GNNs. As described in Section 3.5.1, the DNN inputs were simple vectors of observables rather than graphs of particle constituents. To address the class imbalance, oversampling was performed by randomly duplicating $Z_T Z_T$ events until the classes were balanced [50]. The distributions of each DNN input using the oversampled dataset are shown in Appendix B. Evaluation metrics for the DNN are presented in Figure 14.

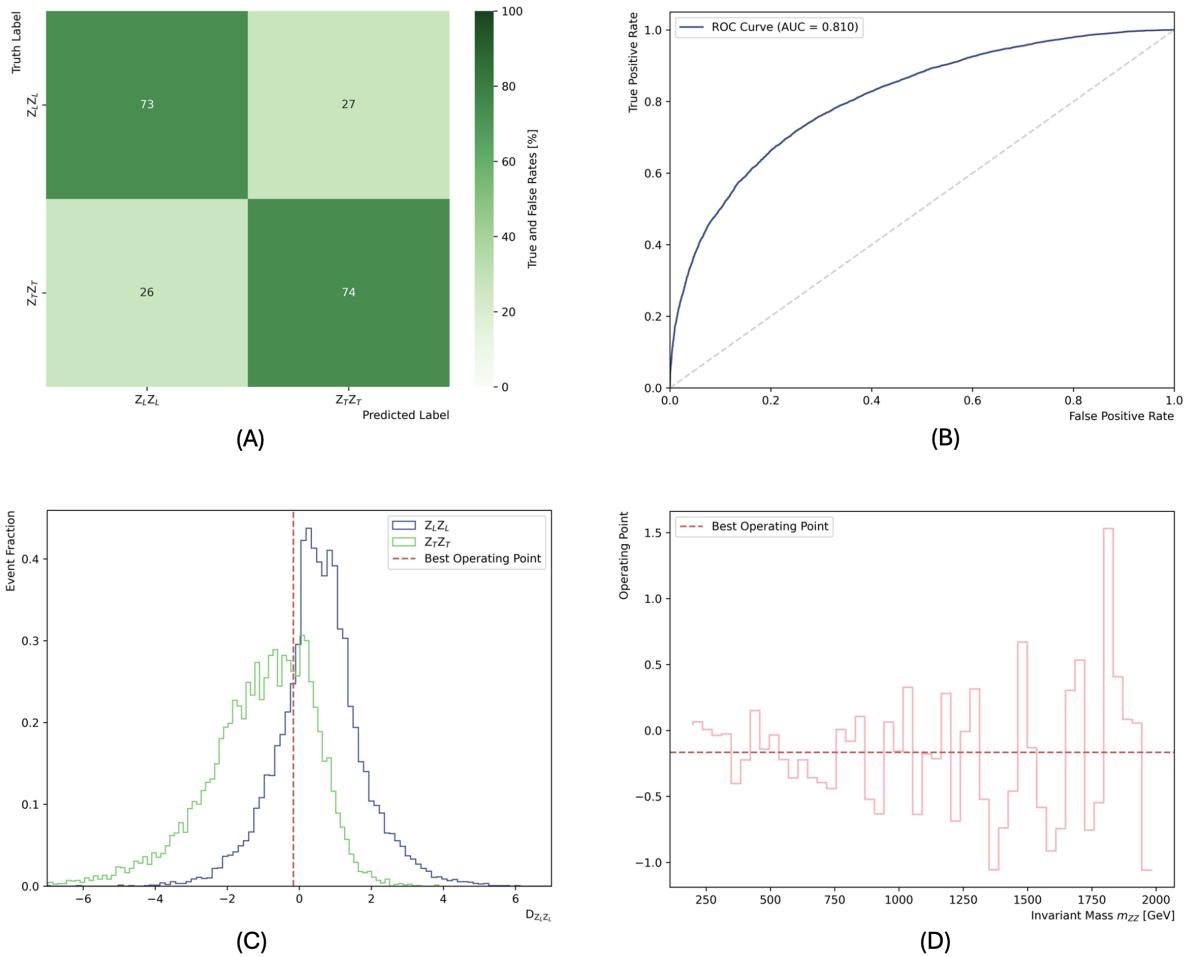


Figure 14: Evaluation metrics for the DNN. Plot (A) is the confusion matrix. Plot (B) is the ROC curve with the AUC value inset. The grey dashed line shows the ROC curve for a DNN with random predictions. Plot (C) is the discriminant $D_{Z_L Z_L}$ plotted separately for each true class, $Z_L Z_L$ (blue) and $Z_T Z_T$ (green). The best operating point to maximise balanced accuracy is shown by the red dashed line. Plot (D) is the operating points for different diboson invariant mass intervals with the best operating point again shown by the dashed red line.

Confusion Matrix Fig.14(A): The DNN shows equal performance in each class, $Z_L Z_L$ was classified correctly in 73% of cases and $Z_T Z_T$ in 74%. A true $Z_L Z_L$ was mistaken for a

$Z_T Z_T$ in 27% of cases and a true $Z_T Z_T$ for a $Z_L Z_L$ in 26%. This result is what is typically expected for a binary classifier and supports the hypothesis that the imbalanced dataset is the cause of the observed bias in the GNNs. Oversampling was not implemented in the GNN training due to the increased complexity of handling graphs and time constraints limited the exploration of other options such as simulating more $Z_T Z_T$ events. Further discussion of this limitation is addressed in Section 5.

ROC Curve Fig.14(B): The DNN achieved an AUC of 0.810, outperforming both lab and diboson rest-frame GNNs which had AUC values of 0.795 and 0.788 respectively.

Discriminant Distribution Fig.14(C): The discriminant distributions of $Z_L Z_L$ and $Z_T Z_T$ resemble those of the GNNs, with peaks centred around a positive and negative values and close to zero. The $Z_T Z_T$ distribution is more spread towards the negative side of the plot which suggests that the DNN is more confident in its $Z_T Z_T$ predictions compared to the GNNs. The best operating point was determined from maximising balanced accuracy to be $D_{Z_L Z_L} = -0.17$, closer to zero than either GNN as expected with balanced classes.

Invariant Mass Dependent Operating Points Fig.14(D): Optimal operating points were evaluated for 50 equal intervals of $m_{ZZ} \in [180, 2000]\text{GeV}$. Fluctuations from the best operating point increased with invariant mass but were less pronounced than in the GNNs. The largest magnitude of optimal operating point was around $|D_{Z_L Z_L}| = 1.5$, as opposed to around 4.5 and 3.5 for the GNNs. This was likely due to the DNN being trained on more events, which helped to reduce event sparsity at higher m_{ZZ} .

The DNN's overall performance was comparable to that of the GNNs. This contradicted expectations that the GNN would outperform the DNN due to its ability to exploit relational information through graphs, information which was inaccessible to the DNN. This suggested that the GNN had not reached its peak performance and with further development the GNN could have the potential to become even better at distinguishing longitudinal and transverse polarisations of weak bosons.

5 Conclusion and Outlook

This project demonstrated the potential of Graph Neural Networks (GNNs) to distinguish longitudinal and transverse polarisations of Z bosons, and hence provide an opportunity to test electroweak symmetry restoration. The GNN was trained using experimentally-realistic simulation data, with input graphs constructed from the 4-momentum of the decay products of $ZZ \rightarrow \ell\bar{\ell}q\bar{q}$ events with known initial polarisation states. The lab-frame GNN achieved classification accuracies of 80% for longitudinal polarisations ($Z_L Z_L$) and 63% for transverse polarisations ($Z_T Z_T$). The bias towards $Z_L Z_L$ was attributed to class imbalance in the training dataset. Performance was found to be comparable to both a GNN trained using graphs in the diboson rest-frame and to a DNN trained using high-level observables which suggested that the GNN is not optimally exploiting graph information. Further development could improve the GNN's ability to distinguish polarisations, particularly at electroweak symmetry restoration scales.

Future work to develop the GNN should firstly look at increasing event statistics and addressing class imbalance. While balanced classes may improve training, it should also be taken into account that experimental data will have significant class imbalance. Incorporating theoretical predictions of polarisation fractions, as was done in Ref. [10], could help optimise the GNN for specific energies. The GNN would also benefit from further hyperparameter tuning. This project only tested nine combinations of learning rate and dropout probability, a much finer grid search around the identified optimal values should be performed using more powerful computers to reduce training times. Once performance has been optimised, the GNN could be extended to classify additional polarisation states such as $Z_L Z_T$, and eventually be applied to W polarisation states as well. This would make the GNN a more versatile tool for studies into electroweak symmetry restoration.

6 Acknowledgements

I would like to express my gratitude to my supervisor Dr. Dominik Duda for his guidance and support. I really enjoyed and have learnt so much throughout this project.

References

- [1] ATLAS Collaboration, “Observation of a new particle in the search for the standard model higgs boson with the ATLAS detector at the LHC,” *Physics Letters B*, vol. 716, no. 1, pp. 1–29, 2012.
- [2] CMS Collaboration, “Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC,” *Physics Letters B*, vol. 716, no. 1, pp. 30–61, 2012.
- [3] G. Burdman, “Quantum field theory and the electroweak standard model,” in *Proceedings of the 2023 CERN Latin-American School of High-Energy Physics*, CERN Yellow Reports: School Proceedings, CERN, 2023.
- [4] L. Huang, S. D. Lane, I. M. Lewis, and Z. Liu, “Electroweak restoration at the LHC and beyond: The Vh channel,” *Physical Review D*, vol. 103, p. 053007, 2021.
- [5] D. Pappadopulo, A. Thamm, R. Torre, and A. Wulzer, “Heavy vector triplets: bridging theory and data,” *Journal of High Energy Physics*, vol. 2014, pp. 1–50, 2014.
- [6] ATLAS Collaboration, “Studies of the energy dependence of diboson polarization fractions and the radiation-amplitude-zero effect in WZ production with the ATLAS detector,” *Physical Review Letters*, vol. 133, p. 101802, 2024.
- [7] ATLAS Collaboration, “Observation of gauge boson joint-polarisation states in W production from pp collisions at $\sqrt{13}$ TeV with the ATLAS detector,” *Physics Letters B*, vol. 843, p. 137895, 2023.
- [8] J. Lee *et al.*, “Polarization fraction measurement in ZZ scattering using deep learning,” *Physical Review D*, vol. 100, p. 116010, 2019.
- [9] J. Lee *et al.*, “Polarization fraction measurement in same-sign WW scattering using deep learning,” *Physical Review D*, vol. 99, p. 033004, 2019.

- [10] M. Grossi, M. Incudini, M. Pellen, and G. Pellicciol, “Amplitude-assisted tagging of longitudinally polarised bosons using wide neural networks,” *The European Physical Journal C*, vol. 83, no. 759, 2023.
- [11] ATLAS Collaboration, “ $ZZ \rightarrow \ell^+ \ell^- \ell' + \ell' -$ cross-section measurements and search for anomalous triple gauge couplings in 13 TeV pp collisions with the ATLAS detector,” *Physical Review D*, vol. 97, p. 032005, 2018.
- [12] D. Griffiths, *Introduction to Elementary Particles*. Wiley-VCH, 2017.
- [13] G. DeZoort, P. W. Battaglia, C. Biscarat, and J.-R. Vlimant, “Graph neural networks at the Large Hadron Collider,” *Nature Reviews Physics*, vol. 5, pp. 281–303, 2023.
- [14] S. Navaset *et al.*, “Review of particle physics,” *Physical Review D*, vol. 110, p. 030001, 2024.
- [15] P. W. Higgs, “Broken symmetries and the masses of gauge bosons,” *Physical Review Letters*, vol. 13, pp. 508–509, 1964.
- [16] M. Thomson, *Modern Particle Physics*. Cambridge University Press, 2013.
- [17] L. Maiani, *Electroweak Interactions*. CRC Press, 2016.
- [18] A. Radovic *et al.*, “Machine learning at the energy and intensity frontiers of particle physics,” *Nature*, vol. 560, p. 41–48, 2018.
- [19] D. Guest, K. Cranmer, and D. Whiteson, “Deep learning and its application to LHC physics,” *Annual Review of Nuclear and Particle Science*, vol. 68, pp. 161–181, 2018.
- [20] The Nobel Committee, “The Nobel Prize in Physics 2024.” <https://www.nobelprize.org/prizes/physics/2024/press-release/>, 2024. Awarded to John J. Hopfield and Geoffrey Hinton for foundational discoveries and inventions that enable machine learning with artificial neural networks.
- [21] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016.
- [22] J. Shlomi, P. Battaglia, and J.-R. Vlimant, “Graph neural networks in particle physics,” *Machine Learning: Science and Technology*, vol. 2, p. 021001, 2021.
- [23] ATLAS Collaboration, “Graph neural network jet flavour tagging with the ATLAS detector,” tech. rep., CERN, Geneva, 2022. <https://cds.cern.ch/record/2811135>. Accessed: 24-11-24.
- [24] ATLAS Collaboration, “Transformer neural networks for identifying boosted higgs bosons decaying into $b\bar{b}$ and $c\bar{c}$ in ATLAS,” tech. rep., CERN, Geneva, 2023. <https://cds.cern.ch/record/2866601>. Accessed: 24-11-2024.
- [25] J. Bardhan *et al.*, “Unsupervised and lightly supervised learning in particle physics,” *European Physical Journal Special Topics*, 2024.
- [26] ATLAS Collaboration, “Search for new phenomena in two-body invariant mass distributions using unsupervised machine learning for anomaly detection at $\sqrt{s} = 13$ TeV with the ATLAS detector,” *Physical Review Letters*, vol. 132, p. 081801, 2024.

- [27] S. Thais *et al.*, “Graph neural networks in particle physics: Implementations, innovations, and challenges,” in *Proceedings of the US Community Study on the Future of Particle Physics (Snowmass 2021)*, Snowmass, 2021.
- [28] A. Tanaka, A. Tomiya, and K. Hashimoto, *Deep Learning and Physics*. Springer Nature, 2021.
- [29] N. Srivastava *et al.*, “Dropout: A simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.
- [30] B. Sanchez-Lengeling, E. Reif, A. Pearce, and A. B. Wiltschko, “A gentle introduction to graph neural networks,” *Distill*, 2021. <https://distill.pub/2021/gnn-intro>. Accessed: 15-11-2024.
- [31] D. Liu and L.-T. Wang, “Prospects for precision measurement of diboson processes in the semileptonic decay channel in future LHC runs,” *Physical Review D*, vol. 99, p. 055001, 2019.
- [32] A. Giannini, “Experimental perspectives for polarisation tagging.” Presentation slides, 2024. Presented at the COMETA Workshop 2024, Toulouse, September 2024.
- [33] ATLAS Collaboration, “Search for diboson resonances in hadronic final states in 139 fb^{-1} of pp collisions at $\sqrt{s} = 13 \text{ tev}$ with the ATLAS detector,” *Journal of High Energy Physics*, vol. 2019, no. 91, 2019.
- [34] CMS Collaboration, “Performance of reconstruction and identification of τ leptons decaying to hadrons and ν_τ in pp collisions at $\sqrt{s} = 13 \text{ TeV}$,” *Journal of Instrumentation*, vol. 13, no. 10, p. P10005, 2018.
- [35] ATLAS Collaboration, “Evidence of pair production of longitudinally polarised vector bosons and study of CP properties in $ZZ \rightarrow 4\ell$ events with the ATLAS detector at 13 TeV,” *Journal of High Energy Physics*, vol. 2023, no. 107, 2023.
- [36] J. Alwall *et al.*, “The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations,” *Journal of High Energy Physics*, vol. 2014, no. 79, 2014.
- [37] A. Buckley *et al.*, “LHAPDF6: parton density access in the LHC precision era,” *The European Physical Journal C*, vol. 75, no. 132, 2015.
- [38] C. Bierlich *et al.*, “A comprehensive guide to the physics and usage of PYTHIA 8.3,” *SciPost Physics Codebases*, p. 8, 2022.
- [39] J. de Favereau *et al.*, “DELPHES 3: a modular framework for fast simulation of a generic collider experiment,” *Journal of High Energy Physics*, vol. 2014, no. 57, 2014.
- [40] H. Qu and L. Gouskos, “Jet tagging via particle clouds,” *Physical Review D*, vol. 101, p. 056019, 2020.
- [41] Y. Wang *et al.*, “Dynamic graph CNN for learning on point clouds,” *ACM Transactions on Graphics*, vol. 38, no. 146, pp. 1–12, 2019.

- [42] S. Ioffe and C. Szegedy, “Batch normalization: accelerating deep network training by reducing internal covariate shift,” in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML’15, 2015.
- [43] F. Pedregosa *et al.*, “Scikit-learn: Machine learning in python,” *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011.
- [44] A. Paszke *et al.*, *PyTorch: an imperative style, high-performance deep learning library*. Curran Associates Inc., 2019.
- [45] M. Fey and J. E. Lenssen, “Fast graph representation learning with pytorch geometric.” <https://arxiv.org/abs/1903.02428>, 2019. Accessed: 24-11-2024.
- [46] S. Widyanto, “Feasibility study to search for higgs boson decays to gluons at the LHC using machine learning,” 2024. [Source Code] <https://github.com/Satriawidy/HggNN>. Date Accessed: 25-11-2024.
- [47] PyTorch Contributors, “Cross Entropy Loss.” <https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html>, 2023. Accessed: 15-11-2024.
- [48] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization.” <https://arxiv.org/abs/1412.6980>, 2017. Accessed: 15-11-2024.
- [49] A. Kraskov, H. Stögbauer, and P. Grassberger, “Estimating mutual information,” *Physical Review E*, vol. 69, p. 066138, 2004.
- [50] G. Menardi and N. Torelli, “Training and assessing classification rules with imbalanced data,” *Data Mining and Knowledge Discovery*, vol. 28, no. 1, pp. 92–122, 2014.

A Monte Carlo Dataset

Tables 3 and 4 detail the variables which were included in the Monte Carlo dataset used to train the GNNs and DNN. Table 3 is the high level features for each event, which includes reconstructed objects and observables. Table 4 is the 4-momentum information for a given particle constituent in an event. Up to 60 constituents were recorded for each event.

Index	Variable Name	Description
0	FJ_eta	Pseudorapidity of the fat jet
1	FJ_flavour	Flavour of the fat jet
2	FJ_mass	Mass of the fat jet
3	FJ_pT	Transverse momentum of the fat jet
4	FJ_phi	Azimuthal angle of the fat jet
5	LeadingSubJet_Eta	Pseudorapidity of the leading subjet
6	LeadingSubJet_Phi	Azimuthal angle of the leading subjet
7	LeadingSubJet_pT	Transverse momentum of the leading subjet
8	Lep_pT_balance	Momentum balance between leptons
9	NegLep_Eta	Pseudorapidity of the negatively charged lepton
10	NegLep_Phi	Azimuthal angle of the negatively charged lepton
11	NegLep_pT	Transverse momentum of the negatively charged lepton
12	Phi	Azimuthal angle of the event
13	Phi1	Azimuthal angle difference between objects
14	PosLep_Eta	Pseudorapidity of the positively charged lepton
15	PosLep_Phi	Azimuthal angle of the positively charged lepton
16	PosLep_pT	Transverse momentum of the positively charged lepton
17	SubLeadingSubJet_Eta	Pseudorapidity of the subleading subjet
18	SubLeadingSubJet_Phi	Azimuthal angle of the subleading subjet
19	SubLeadingSubJet_pT	Transverse momentum of the subleading subjet
20	Vlep_eta	Pseudorapidity of the lepton system
21	Vlep_mass	Mass of the lepton system
22	Vlep_pT	Transverse momentum of the lepton system
23	Vlep_phi	Azimuthal angle of the lepton system
24	cosThetaStar	Cosine of the angle between bosons in the rest-frame
25	costhetal1	Cosine of the decay angle between leptons
26	costhetal2	Cosine of the decay angle between quarks
27	polarisation_type	Polarisation type of the Z bosons

Table 3: High level features.

Index	Variable Name	Description
0	constituent_D0	Impact parameter in the transverse plane
1	constituent_DZ	Impact parameter along the beam axis
2	constituent_VVrest_eta	Pseudorapidity in the diboson rest-frame
3	constituent_VVrest_phi	Azimuthal angle in the diboson rest-frame
4	constituent_VVrest_pt	Transverse momentum in the diboson rest-frame
5	constituent_Vrest_eta	Pseudorapidity in the boson rest-frame
6	constituent_Vrest_phi	Azimuthal angle in the boson rest-frame
7	constituent_Vrest_pt	Transverse momentum in the boson rest-frame
8	constituent_charge	Electric charge of the particle
9	constituent_eta	Pseudorapidity of the particle
10	constituent_isLep	Equal to 1 if the particle is a lepton or 0 otherwise
11	constituent_phi	Azimuthal angle of the particle
12	constituent_pt	Transverse momentum of the particle

Table 4: Constituent level features.

B DNN Input Variables

Input Variable	Normalised Mutual Information
Vlep_pT	1.000000
NegLep_pT	0.558485
LeadingSubJet_pT	0.556843
PosLep_pT	0.527191
FJ_mass	0.467681
costheta1	0.347315
costheta2	0.210773
SubLeadingSubJet_pT	0.188753
FJ_pT	0.186516
cosThetaStar	0.081754
FJ_eta	0.038180
LeadingSubJet_Eta	0.037009
Vlep_eta	0.033148
SubLeadingSubJet_Eta	0.025650
SubLeadingSubJet_Phi	0.020790
PosLep_Eta	0.015405
Phi	0.002557
PosLep_Phi	0.001541
LeadingSubJet_Phi	0.001414
Vlep_mass	0.000000
NegLep_Eta	0.000000

Table 5: List of high level features used as input to the DNN. They are ranked by their feature importance using Mutual Information, features with higher importance have more classification power. Vlep_mass and NegLep_Eta are not zero, but very small.

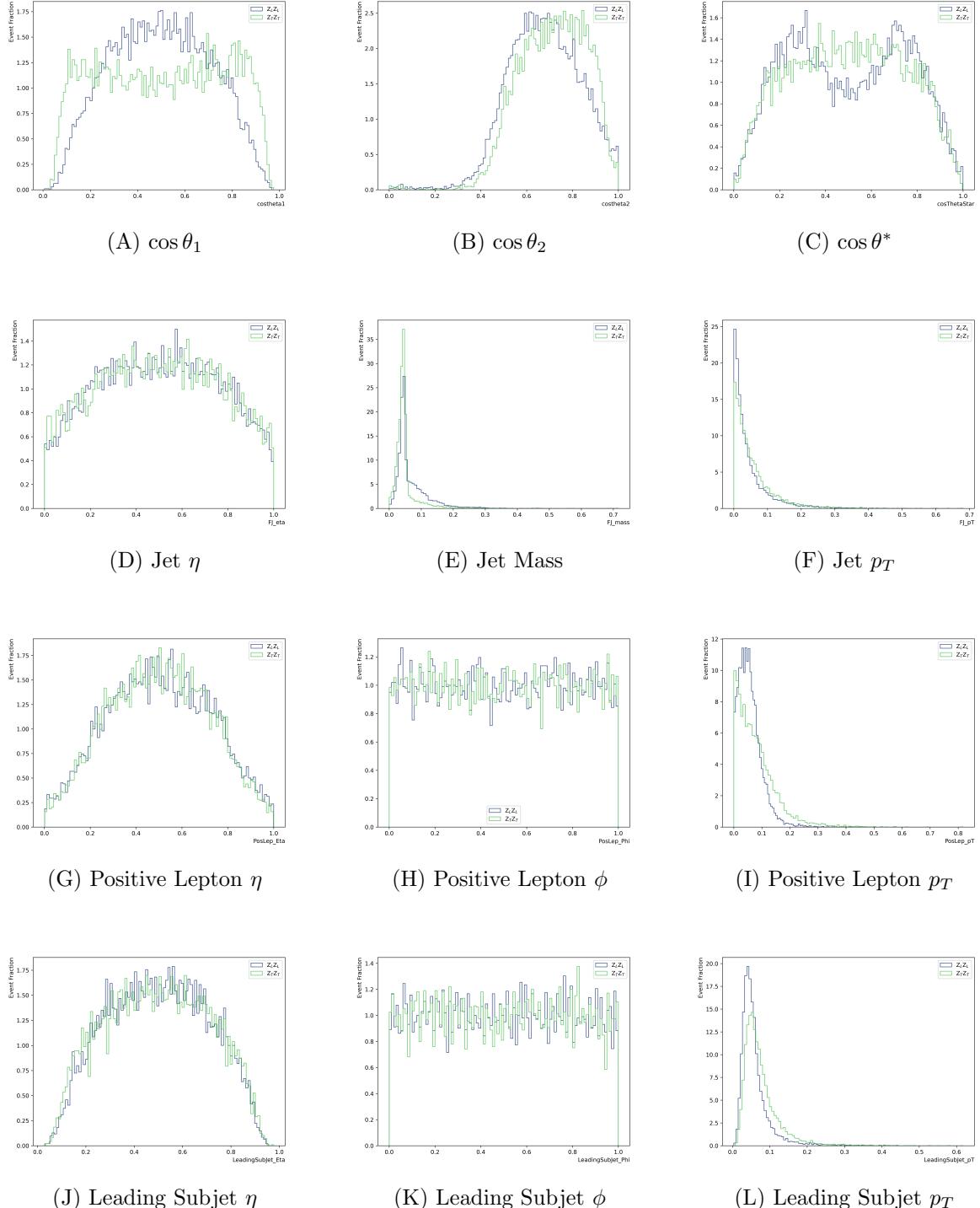


Figure 15: Normalised distributions of DNN input features for each class, Z_LZ_L (blue) and Z_TZ_T (green) (1/2).

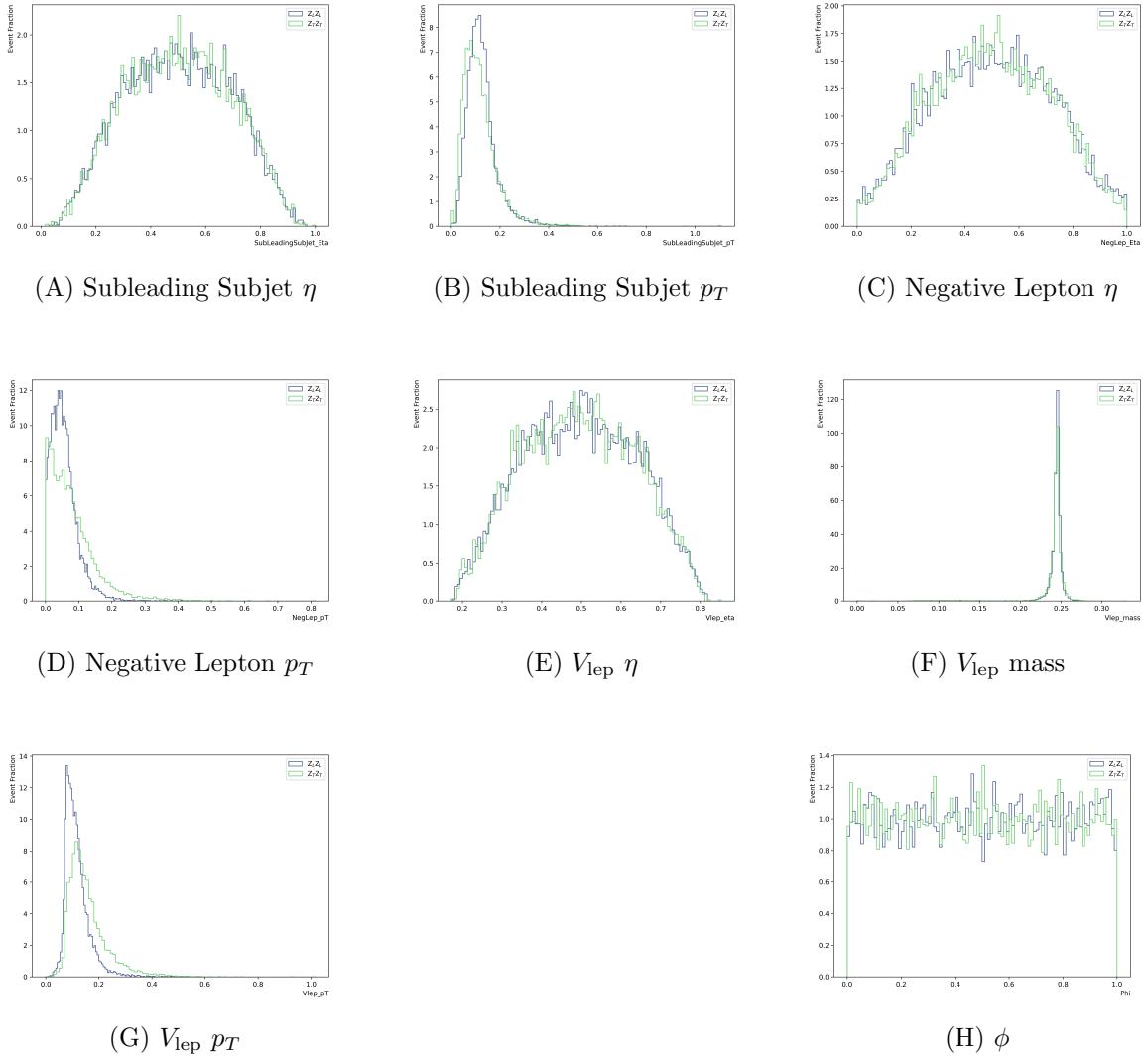


Figure 16: Normalised distributions of DNN input features for each class, $Z_L Z_L$ (blue) and $Z_T Z_T$ (green) (2/2).