Using Advanced Machine Learning Techniques to Study Poorly Modeled Processes in *pp* Collisions with the ATLAS Detector

Dissertation
zur
Erlangung des Doktorgrades (Dr. rer. nat.)
der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

von
Federico Guillermo Diaz Capriles
aus
Ft. Lauderdale, FL, USA

Bonn, 10.3.2023

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn

1. Gutachter:	Prof. Dr. Ian C. Brock
2. Gutachter:	Priv-Doz. Dr. Philip Bechtle
Tag der Promotion:	15.05.2023
Erscheinungsjahr:	2023

Acknowledgements

I would like to thank my family and friends for their continued support with the distinction of my mother, and best friend whose contributions are what made this possible. Prof. Dr. Ian Brock for the opportunity to take part in this field, the patience, the support, and the freedom to pursue my ideas. Finally, the friends that helped during these tough times. Corona in the office was rough, but at least being able to vent to each other made it a little more bearable.

Contents

1 Preamble **1**

2 Introduction **3**
2.1 The Standard Model . . . 3
2.1.1 Fermions . . . 4
2.1.2 Bosons . . . 6
2.2 Structure of a Proton . . . 13
2.3 Physics at Particle Colliders . . . 14
2.4 Top-Quark Physics . . . 17
2.5 Tau Physics . . . 22

3 The Large Hadron Collider and ATLAS **25**
3.1 The Large Hadron Collider . . . 25
3.2 ATLAS . . . 28
3.3 Reconstruction and Identification of Objects . . . 32

4 Data, Monte Carlo Simulation, and Event Selection **39**
4.1 Datasets . . . 39
4.2 Monte Carlo Simulation . . . 39
4.3 Signal and Sources of Background . . . 41

5 Machine Learning **49**
5.1 Introduction . . . 49
5.2 Artificial Neural Networks . . . 49
5.2.1 Architecture and Components . . . 50
5.2.2 Data Preparation . . . 53
5.2.3 Over- and Underfitting . . . 54
5.2.4 Regularization . . . 56
5.2.5 Batch Sizes and Hardware . . . 58
5.3 Supervision . . . 60
5.4 Weak Supervision . . . 60
5.4.1 Learning from Label Proportions . . . 60
5.4.2 Classification Without Labels . . . 61
5.5 No Supervision . . . 61
5.5.1 Autoencoders . . . 62
5.5.2 Masked Autoregressive Density Estimators (MADE) . . . 64

5.5.3 Normalizing Flows . . . 66
5.5.4 Masked Autoregressive Flows . . . 67
5.5.5 Anomaly Detection with Density Estimation . . . 68

6 Imprecise Modeled Processes **71**
6.1 Interference Between tW and $t\bar{t}$ . . . 71

7 Poorly Modeled Backgrounds **91**
7.1 Hadronic τ Leptons . . . 91

8 Summary and Conclusion **117**

A Machine Learning Package **119**

B Autoencoder Distributions **121**

C Data vs. MC Classifier for τ_{had} with a two b-jet selection **123**

D Networks which Included the Jet and Track Calorimeter Widths as Training Variables **127**

E Likelihood Fit **131**

F DR/DS Autoencoder Distributions **135**

Bibliography **145**

List of Figures **157**

List of Tables **161**

CHAPTER 1

Preamble

One of the most powerful, unifying forces is humanity's desire to understand and uncover truths. One of the prime examples is the grand achievement of particle accelerators such as the Large Hadron Collider (LHC) located in the Swiss-French border. A massive machine designed to smash the constituents of matter to better understand what we are made of and what the laws of our universe are. Over a hundred countries, represented by thousands of people, come together to analyze these collisions with tools like the ATLAS detector and see snapshots of the rules with every smash. These violent collisions are our window to understanding if the model we have, namely the Standard Model of Particle Physics, is accurate in describing the world as we see it.

The Standard Model (SM) is the set of rules that describe particle properties and dictate how they interact with each other. Not only does the model state how particles interact, it also predicts measurable quantities such as how particles decay, how often, and into what decay products. With tools like particle detectors, we can test the rigidity of our model as it should be able to predict a plethora of interactions seen in the detector. Over many years of measurements and research, humanity has made the SM one of the most robust theories. As accurate and extensive as the SM is, it does not describe everything. Phenomena such as dark matter and neutrino masses, for example, are not mentioned anywhere in the SM. These problems further drive us to unite and design experiments to measure and understand these processes.

Even with a strong understanding of SM particles, some exhibit interesting properties. One of such is the top-quark, the heaviest particle in the SM. Its large mass takes away the ability for it to create bound states as it decays much faster than forming composite particles does. The production and decay of top-quarks is an interesting study as its ability to be measured "alone" allows one to measure its SM parameters with great precision. Additionally, these parameters may give hints of beyond the Standard Model (BSM). Fortunately, the LHC is capable of producing massive particles like the top-quark frequently.

On a similar note, the Higgs boson has been one of the most interesting particles since its discovery as it has unique properties and purpose. Although the Higgs boson and top-quark are very different, they can be produced together via rare processes. Measuring such rare occurring productions opens up different parameters in the SM that can also be measured. However, this comes with difficulty in isolating these rare processes and estimation of unmodeled or poorly modeled backgrounds.

In this document, machine learning strategies are explored to aid such analyses in achieving their goal. One analysis attempts to measure the cross section of top-quark pair and single top-quark

production with a focus on measuring regions sensitive to their interference. This document shows the exploration of novel machine learning techniques that generate a variable that is highly sensitive to the interference of these two processes. The second analysis aims to measure the rare process of a single top-quark in association with a Higgs boson and a spectator quark. Here, many background processes mimicking hadronically decaying tau-leptons pose a challenge in the rare process measurement. The research presented herein shows a neural network acting as a discriminant between the modeled, desired processes and the undesired background. These machine learning applications have gone through several iterations trying to succeed at the posed challenge. This thesis documents some of the attempts and their conclusions.

CHAPTER 2

Introduction

2.1 The Standard Model

The best way to understand the universe we live in is to discover and model the rules that govern it. Humanity, through decades of work, has come up with a model that describes many of the particles and interactions that can be "seen" at the quantum scale. This model is called the Standard Model (SM) and beyond cataloguing interactions, it can be used to make predictions backed by experiment. The SM has predicted particles which were later discovered in high energy colliders such as the top-quark (1995, Tevatron), tau neutrino (2000, DONUT), and Higgs boson (2012, LHC). In addition, experiments aim to push the boundaries of precision measurements and compare them with the known model. In figure 2.1, one can see results published by the ATLAS collaboration which compare their cross-section measurements to theoretical expectations. In this image, one can see the predictive power of the SM as the experimental results are in agreement with the theoretical expected values. Although the SM is the most accurate model to date, it is not without its limitations. There are a few questions that remain unanswered, for example: dark matter and energy are not predicted or explained, neutrino masses are not generated, and gravity is also not implemented as there is no renormalizable way to include it. Even with its limitations, the SM is a powerful tool to understand the universe.

To clarify, the SM classifies *elementary particles* and their interactions. Unlike a proton or neutron, an elementary particle has no substructure and is point-like. This reduces the current number of particles to a handful of particles which are divided into *fermions* and *bosons*. Fermions are particles that form matter and bosons are the particles associated to the forces between the fermions. The dividing property between these two groups is spin as bosons have integer-spin and fermions have half-integer-spin. Although one can imagine spin in the classical sense of the word, for point-like particles this means an intrinsic angular momentum.

It is important to differentiate particles by their characteristics as that tells by which rules they are governed and what role they play. For example, the spin of the particle dictates the type of statistics the particle follow. Fermions follow Fermi-Dirac statistics and therefore must obey the Pauli exclusion principle. This means, roughly, that two fermions cannot occupy the same space at the same time with the same quantum numbers. On the contrary, bosons follow Bose-Einstein statistics and are not bound by the exclusion principle.

It should be noted this is not the only characteristic that particles have. Some properties give particles the ability to interact with forces or be invisible to them. Characteristics of particles such

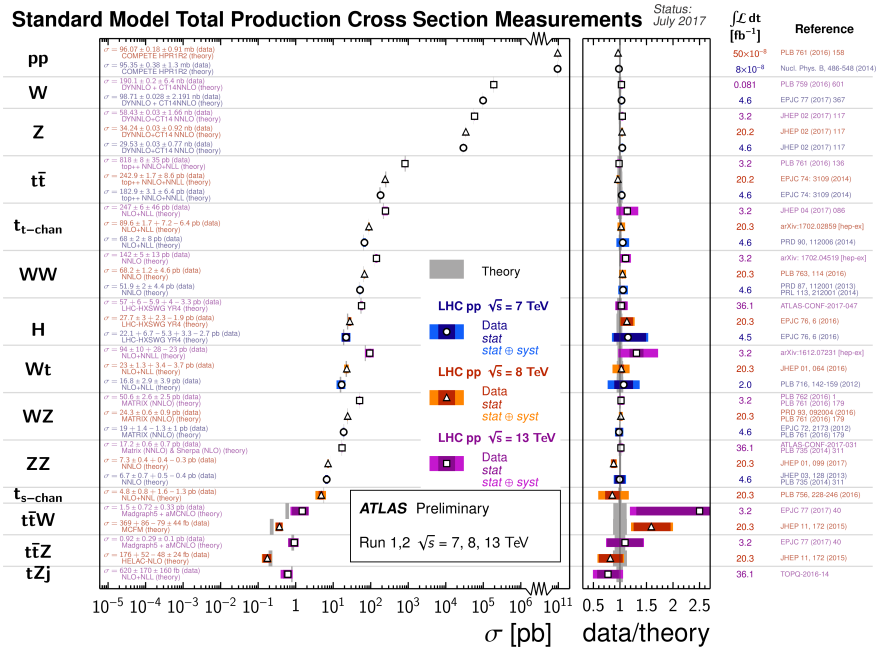


Figure 2.1: Public cross-section measurements for several physics processes and their theoretical values from the ATLAS collaboration [1]. The vertical axis is divided by production processes and the horizontal axis of the left-most plot is the cross-section in pico-barn. The right-most plot shows a ratio of the measured cross-section to the expected theoretical value with errors. Next to this plot is the integrated luminosity at the time of measurement, followed by the reference to the publication.

as electric, weak and color charge, flavor, mass dictate their behavior. Elementary particles, their characteristics and interactions are further developed in the coming sections. However, figure 2.2 shows all the elementary particles predicted by the SM with labels, classifications and some properties.

2.1.1 Fermions

As shown in figure 2.2, fermions come in three generations of increasing mass but share similar properties otherwise. Although it is not shown in the figure, each fermion has its own antiparticle which has the opposite charge and quantum numbers. Fermions exhibit different properties and are therefore further divided as *leptons* and *quarks*. There are a total of six quarks and six leptons which are classified by the generation.

Leptons

Leptons are split into charged and neutral with respect to electric charge. Of the charged leptons, the most well-known is the electron, e, which was discovered by J. J. Thomson in 1897. It has an electric charge of -1.602×10^{-19} C, or $-1\,e$ called the *elementary charge*, and a mass of $0.511\,\mathrm{MeV}/c^2$. As mentioned earlier, the muon and tau leptons, μ and τ, have the same charge as the electron but have increasing masses. The muon has a mass of about $105.7\,\mathrm{MeV}/c^2$ and the tau leptons have a mass of about $1\,776.8\,\mathrm{MeV}/c^2$, making it the heaviest lepton.

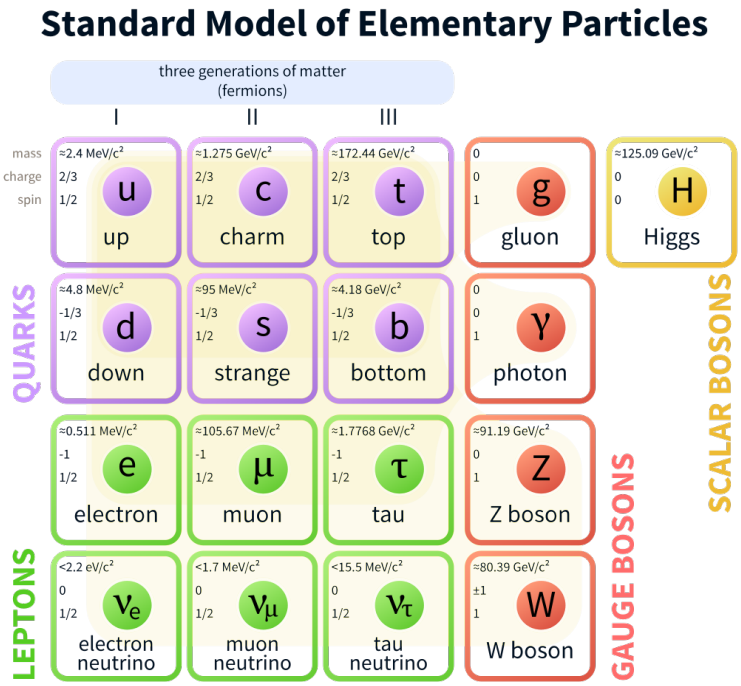


Figure 2.2: Schematic of the Standard Model of particle physics. The border color denotes what group the particle belongs, denoted by the color coded title. A colored background extends from bosons into fermions denoting what particles that boson interacts with. Each box contains the name, symbol, spin, mass, and charge of the particle [2].

Each charged lepton has a neutrally charged counterpart called the neutrino, ν_ℓ, which shares a lepton family number with their charged partner. Thus, the neutrino comes in three flavors: the electron neutrino ν_e, muon neutrino ν_μ and tau neutrino ν_τ. Neutrinos were postulated by Pauli after studying the energy spectrum of β decay. They were directly detected much later in nuclear reactors by a process called inverse beta decay where a proton interacts with an electron antineutrino and the resulting particles are a neutron and an positron. Experiments aimed at measuring neutrino emissions, e.g. solar neutrinos, have shown that neutrinos oscillate between families. That means that as a neutrino propagates through space, there is a chance that it will change its flavor. This phenomenon is called *neutrino oscillation* and it is a result of neutrinos having a non-zero mass, contrary to what is assumed by the SM. According to current experiments, no mass has been measured but instead upper limits have been imposed which are in the eV/c^2 range.

Quarks

Quarks share many similarities with leptons in terms of properties. Like leptons, quarks are organized into three generations of ascending mass where electric charge is identical between generations. These quarks are labeled as *up-type* and *down-type* quarks. Although all quarks are electrically charged, the charge is a fraction of the elementary charge. Here, up-type quarks have a charge of $2/3\,e$ and down-type quarks have a charge of $-1/3\,e$. Up-type quarks are, in ascending mass order, the up-quark

u, the charm-quark c, and the top-quark t. Similarly organized, down-type quarks are the down-quark d, the strange-quark s, and the bottom-quark b. Quarks with masses under 1 GeV/c^2 are called *light quarks* and these include the up, down, and strange quarks. The heaviest of the quarks is the top-quark with a mass of about 173.1 GeV/c^2, incidentally making it the heaviest particle in the SM.

Unlike leptons, quarks have another type of charge called *color charge* which allows quarks to interact via the strong force. Because of this property, quarks exhibit a peculiar phenomenon called *color confinement* where quarks are not allowed to exist in isolation. Notably, the top-quark decays too quickly to form bound states. This means it exists almost as if it were isolated, more detail in section 2.4. Quarks must form bound states called hadrons which must be "color neutral" to be physical particles. Color charge comes in three colors: red, green, and blue; with antiquarks having anticolors. Therefore, color-neutrality means a combination of all three colors (or anticolors) or a color with its anticolor. Combinations of all three colors are called *baryons* which are made of a triplet of quarks (qqq). Combinations of color and anticolor particles are called *mesons* which are made of a quark-antiquark pair ($q\bar{q}$). These two color combinations are the most common hadrons found in nature, though one can combine colors and anticolors to form groups of four or five quarks in a bound state. Such combinations are called tetraquarks ($q\bar{q}q\bar{q}$) and pentaquarks ($q\bar{q}qqq$). Such states have been found in the Large Hadron Collider experiment LHCb [3, 4].

2.1.2 Bosons

Bosons do not make up matter but are the mediators for interactions between them and thus are often called *force carriers*. In the SM, three out of the four forces of nature are explained: the strong, weak and electromagnetic (EM) force. Even though gravity's influence range is technically infinite, its strength compared to the other forces is orders of magnitude smaller. The comparison can be seen in table 2.1. Since elementary particles are so minute, the effect of gravity on them is negligible; specially in the context of current high energy experiments. As for the other forces, they all have one or more mediating particles which interact with other particles or, in some cases, themselves.

Force	Mediating Particle	Relative Strength
Strong	gluons (g)	10
Electromagnetic	photons (γ)	10^{-2}
Weak	W and Z bosons	10^{-13}
Gravity	Not in SM	10^{-42}

Table 2.1: The four fundamental forces of nature with their mediating particle and relative strength to gravity. Gravity has no mediating particle in the SM but there is a hypothetical *graviton* postulated by quantum gravity theories. [5]

The strong force is the reason why protons and other hadrons are formed and also binds them together to create composite objects like an atomic nucleus. This force is mediated by the gluon, which couples to the color charge of a particle (i.e. only quarks are affected) and is described by quantum chromodynamics (QCD). The weak force is responsible for particle decay and processes like nuclear fusion or fission reactions. The W and Z bosons are the mediators for this force and are the only force-carrier bosons with mass. This interaction has a short range of about 0.01 to 0.1 fm which is a consequence of how massive both of these bosons are. This force couples to all fermions and allows

them to "change flavor," a property described in more detail further in the section. Electromagnetism is the force which is most well-known as its force-carrier, the photon (γ), is perceived as light. This force couples to all particles which have an electric charge, including the charged W bosons.

Three main theories describe each of these forces: Quantum Electrodynamics (QED) for EM, Quantum Flavordynamics (QFD) for the weak force, and Quantum Chromodynamics (QCD) for the strong. These are quantum field theories, or theories which treat particles as excited states of some quantum fields. Their interactions are described by "interaction" terms in a *Lagrangian*, which is a function which characterizes a physical system. Given some Lagrangian, if one can find some transformation in which the Lagrangian is invariant, then such transformations yield symmetries or conservation laws. In the case of quantum field theories, such a transformation can lead to interaction terms between particles via some field, namely a boson. Therefore, each of the gauge bosons are a consequence of physical laws having some form of symmetry at all points of space-time. Specifically, the SM is a combination of three such transformations: $U(1) \times SU(2) \times SU(3)$, where the $U(1) \times SU(2)$ describes electroweak theory and $SU(3)$ represents the strong force.

Quantum Elecrodynamics

QED is the quantum field theory which describes the electromagnetic force. It details the interaction between charged particles as emissions and absorptions of an intermediary boson, the photon γ. Without doing the hard math, which can be found in physics books such as [5], one can derive the QED Lagrangian by starting from an equation which describes a free fermion, such as an electron. An equation which describes a free fermion is called *Dirac's equation* and is given by the following [5]:

$$\mathcal{L} = \bar{\psi}(i\hbar c\gamma^{\mu}\frac{\partial}{\partial x^{\mu}} - mc^2)\psi, \tag{2.1}$$

where m is the mass of the particle, c is the speed of light, $\hbar$ is the reduced Planck constant ($\frac{h}{2\pi}$), and ψ is a Dirac spinor (a four-dimensional description of a fermion). γ^{μ} are known as the *gamma matrices* which are a set of matrices with specific properties[1].

By introducing a transformation into this system, if invariant, one extracts conserved quantity. A global phase shift, such as:

$$\bar{\psi} \rightarrow \bar{\psi}e^{-i\delta}, \tag{2.2}$$

$$\psi \rightarrow e^{i\delta}\psi, \tag{2.3}$$

would clearly be a gauge invariant transformation as these are constants ($\mathcal{L} \rightarrow \mathcal{L}$). Such a trivial transformation does not introduce a new (or interesting) symmetry or conservation. However, if one introduces a local phase shift in the form of:

$$\bar{\psi} \rightarrow \bar{\psi}e^{-i\delta(x)}, \tag{2.4}$$

$$\psi \rightarrow e^{i\delta(x)}\psi, \tag{2.5}$$

then the partial derivative introduces new terms making our Lagrangian not invariant ($\mathcal{L} \rightarrow$

[1] To explain these matrices and their importance in detail, I would need to write an entire chapter just for them. Either way, they are not instrumental to the work in this thesis so I gloss over them and only acknowledge that they exist.

$\mathcal{L} - \hbar c \frac{\partial \delta(x)}{\partial x^\mu} \bar{\psi} \gamma^\mu \psi)$.

Using some foresight, one can define the phase invariance a bit differently such that things cancel out later.

$$\Lambda(x) \stackrel{\text{def}}{=} -\frac{\hbar c}{q} \delta(x), \tag{2.6}$$

where q is some constant which describes the strength of coupling to a boson we have yet to introduce. Later, this new constant revealed as the coupling constant to the EM force and is related to the charge of the fermion. However, the Lagrangian is still not invariant but one can introduce a new field which has some change given the local phase shift in the form of:

$$A_\mu \to A_\mu + \frac{\partial \Lambda(x)}{\partial x^\mu}, \tag{2.7}$$

where A_μ is the field and is introduced to the Lagrangian in the following form:

$$\mathcal{L} = \bar{\psi}(i\hbar c \gamma^\mu \frac{\partial}{\partial x^\mu} - mc^2 - q\gamma^\mu A_\mu)\psi. \tag{2.8}$$

It can be seen that this new Lagrangian is now invariant under a local phase transformation as, once introduced, the derivative from the first term cancels out with the new term. However, this is not the end of the tale as the new field must still be described as a free particle. One could sit around and try some different Lagrangian for different fields but, knowing the solution, one can simply use the Proca equation which describes spin 1 fields:

$$\mathcal{L} = -\frac{1}{16\pi} F^{\mu\nu} F_{\mu\nu} + \frac{1}{8\pi} \left(\frac{mc}{\hbar}\right)^2 A^\mu A_\mu, \tag{2.9}$$

where $F^{\mu\nu} \stackrel{\text{def}}{=} \partial^\mu A^\nu - \partial^\nu A^\mu$ and ∂^μ is the shorthand of $\frac{\partial}{\partial x_\mu}$. If one transforms the A^ν field as in equation 2.7, one can see that the $F^{\mu\nu} F_{\mu\nu}$ remains unchanged but $A^\mu A_\mu$ is not invariant. This should be an issue but it is not as the photon is massless and this part vanishes. Therefore, the free spin 1 field A^ν is invariant under a local phase transformation. Thus, the full QED Lagrangian can be written as:

$$\mathcal{L}_{QED} = \bar{\psi}(i\hbar c \gamma^\mu \frac{\partial}{\partial x^\mu} - mc^2 - q\gamma^\mu A_\mu)\psi - \frac{1}{16\pi} F^{\mu\nu} F_{\mu\nu}. \tag{2.10}$$

Feynman Diagrams and Predictable Observables

This is a neat exercise but what it tells is is much stronger than one sees on the surface. By using this Lagrangian, one can see which particles interact, how they interact, and how they propagate freely. For example, the $\bar{\psi} q \gamma^\mu A_\mu \psi$ tells us that a photon interacts with a charged fermion with strength proportional to the charge. Physically, this can be interpreted as a charged fermion emitting or absorbing a photon. Graphically, the term can be described using a powerful tool named *Feynman diagram* shown in figure 2.3. One should note that in this case, a photon is produced from a charged lepton and its antiparticle annihilating. The $\ell^+ \ell^- \gamma$ interaction (called a *vertex*) has the same mathematical "value," regardless of the orientation. The other terms in the Lagrangian are related to how these particles propagate through space.

The rules of Feynman diagrams can be found in [5] which explains in more detail. But in a nutshell,

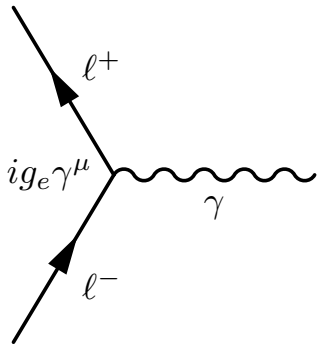


Figure 2.3: Feynman diagram depicting a charged lepton and its antiparticle annihilating into a photon. Diagram includes the Feynman rule for a vertex of this type.

solid arrow lines are fermions and wavy lines are electroweak bosons; both shown in figure 2.3. As a convention, a dashed line represents the Higgs boson and a curly line represent gluons. To add on, when *tree level* or *leading order* is used to describe a process, it means that the interaction has no higher order, or loop, corrections applied. These loop corrections can be done iteratively in higher orders for better precision and are often called *next-to-leading order* and adding *next-to* iteratively to denote higher and higher orders.

What is relevant or powerful about theories like QED is the predictive power it holds when compared to experiment. For example, energy and momentum conservation are known laws of the universe. Thus classically, given some input, one can predict an outcome. However, as one is working with quantum fields, the process of describing interactions is not identical to a classical mechanics problem of balls colliding. Instead, QED can be used to predict decays and scattering processes of arbitrary complexity. Or rather, it predicts the probability of decay and scattering. The scattering probability is referred to as a *cross-section*. Regardless of which of the two is the measurement desired, one needs two ingredients: the amplitude $\mathcal{M}$ (probability of scattering) – often called the *matrix element* – and the available phase space. Therefore, one can use QED to predict the cross-section of some scattering of charged particles, production, annihilation, decay, etc. and one can then build an experiment which tests the accuracy of the prediction. For example, the measurement of the anomalous magnetic moment of the electron showed the power of QED as its prediction was accurate to over ten significant figures [6].

Quantum Chromodynamics

Similar to how a photon carries the electromagnetic force between fermions, coupling to electric charge, a gluon g is the intermediary of the color charge. Like the photon, the gluon is massless and a spin 1 boson and that is where the similarities end. What sets the gluon apart from a photon is its ability to self-interact. This means is that the gluon carries a color charge and can therefore couple to other gluons. That is not the only interesting property that the strong force exhibits.

The strong force comes with interesting properties, one of which is *asymptotic freedom*. One can think of this as the opposite to gravity or electromagnetism's strength at short distances. When one gets closer to a gravitational or EM field, the force between a massive or charged object and the field

source increases. The strong force works in the opposite way by being weaker at short distances between color charged objects. This means that quarks freely move within a hadron without interacting with each other.

Another property unique to the strong force is called *color confinement*. Throughout several decades, physicists have been unable to find isolated quarks as these particles seem to be forcibly bound as hadrons. This phenomenon can be understood as a steady and practically infinite increase in the potential between two quarks as they are pushed apart. The further apart they are, the more energy needs to be put into the system to drive them outwards. At some point, there is enough energy to generate a quark-antiquark pair and therefore create two hadrons. The process of hadrons coming to being as quarks are separated is called *hadronization*.

Unlike an electric charge, in which a particle has it or it does not, color is a choice of three: red, green, and blue. Not to be confused with actual color, but rather a naming scheme for a three-element vector where a quark could have some value in one of the three elements. By construction, this comes from the type of transformation that is done to the Lagrangian in terms of color. Whereas QED comes from a unitary transformation $U(1)$, color charge is a special unitary transformation $SU(3)$. Without getting into the details, this translates to quarks having one of the colors and antiquarks having anticolor.

As mentioned earlier, gluons have color charge as well but dissimilar to quarks. Because of the structure of the $SU(3)$ transformation, eight massless gauge bosons appear. This is because gluons have both color-anticolor charge rather than one or the other. An example could be scattering of two, differently colored quarks – say red and blue – by an intermediary gluon. If the gluon is to couple to both of these quarks, it must therefore have a color-anticolor charge in some combination in order to couple to both. This is not as trivial as a gluon being "red-antiblue" but rather being a superposition "red-antiblue" and "blue-antired".

Quantum Flavordynamics and Electroweak Unification

The $W^{\pm}$ and Z bosons are the force-carriers in weak interactions and are the only gauge bosons to have mass. Specifically, their masses are $(80.385 \pm 0.015)\,\mathrm{GeV}/c^2$ for the W bosons and $(91.1876 \pm 0.0021)\,\mathrm{GeV}/c^2$ for the Z [7]. They have spin one like other gauge bosons and, as their notation suggests, the W bosons are electrically charged while the Z is not. The W boson, according to the SM, is the only boson that allows for flavor change in both within quarks or leptons through a mechanism known as mixing. This property can be seen in beta decay where a neutron decays into a proton and an electron and neutrino are emitted. Inside the nucleon (name given to a proton or neutron), one of the down-quarks that make up the neutron becomes an up-quark and a W boson is emitted which then decays into the electron-neutrino pair. Z functions similarly to the photon but with the ability to couple to the neutral leptons. It can be seen in neutrino-lepton scattering of different flavors because if the weak force only had W bosons then the cross-section of such a process would be smaller than measured.

Unlike both the strong force and EM, the weak force does not couple to an "easily describable" property of particles. The weak interaction comes from an $SU(2)$ transformation which couples to a particle's *weak isospin* and *hypercharge*; a property owned by all fermions. This means that although the neutrino cannot interact with the strong force by lack of color or the EM by lack of electric charge, it can interact with the weak force-carriers. Although not completely accurate, one could imagine that the W boson couples to a particle's flavor as quarks and leptons can be put into generational doublets.

For leptons, a charged weak interaction is well defined as a neutrino from a particular family may not interact with a W boson and emit a different-generation lepton. However, this does not explain the flavor mixing one observes in the quark sector as the decay $\Lambda \rightarrow p + e^- + \nu_e$ implies a flavor change from a strange-quark to a up-quark. Without going too deep into the history of quark discoveries, this phenomenon is explained by the Cabibbo-Kobayashi-Maskawa (CKM) matrix which has the following form:

$$\begin{pmatrix} d' \\ s' \\ b' \end{pmatrix} = \begin{pmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{pmatrix} \begin{pmatrix} d \\ s \\ b \end{pmatrix}, \tag{2.11}$$

where each element V_{ij} is the value of the CKM matrix at the interaction between the i-th and j-th flavor. For example, V_{cs} specifies the coupling of a charmed-quark to a strange-quark. Additionally, one can see that for each down-type quark, there is a corresponding down-type but with a prime (‘) attached. This implies that the physical (mass) states of down-type quarks one measures are a rotation of the “flavor states” that the weak force sees. The current values of the CKM matrix [7] are given in equation 2.12:

$$V_{CKM} = \begin{pmatrix} 0.97370 \pm 0.00014 & 0.2245 \pm 0.0008 & 0.00382 \pm 0.00024 \\ 0.221 \pm 0.004 & 0.987 \pm 0.011 & 0.0410 \pm 0.0014 \\ 0.0080 \pm 0.0003 & 0.0388 \pm 0.0011 & 1.013 \pm 0.030 \end{pmatrix}. \tag{2.12}$$

As mentioned earlier, the values of this matrix imply the coupling, or transition probability, of a quark to a different quark. One can see that the probability at the diagonal is close to one and off-diagonal terms tend to be smaller. Notably, the smallest being related to the transition between first and third generation quarks. Additionally, the V_{tb} value implies that the transition of the top-quark is almost always going to be to the bottom-quark. Off-diagonal values being small imply that mixing is not all too common in general. Analogous to quark mixing, neutrinos have been known to mix as well and for this reason the Pontecorvo-Maki-Nakagawa-Sakata (PMNS) matrix was introduced.

The unification of the electromagnetic and weak force was an achievement of physicists as it saw two different forces as two sides of the same coin. The first step for the unification is taking part of the rules for weak interactions and integrate it into the fermion spinors. From a fermion-fermion-W vertex, one has the following factor:

$$-i\frac{g_W}{2\sqrt{2}}\gamma^\mu(1-\gamma^5)[V_{ij}], \tag{2.13}$$

where g_W is the weak coupling constant and V_{ij} is the CKM matrix element – written in brackets as it can be omitted if the fermion is a lepton. γ^5 is defined as $\gamma^5 \overset{\text{def}}{=\!=} i\gamma^0\gamma^1\gamma^2\gamma^3$ and in the form of $\frac{1}{2}(1 \mp \gamma^5)$ it performs as a projection operator of *helicity*. By performing such projections on fermions, one now has chiral fermion states that can be used in electroweak theory. Left-handed chiral spinors, in the weak Lagrangian, behave similarly to QED where currents can be derived analogously. Additionally, the total spinor is just the sum of the chiral spinors and as such one can arrive at QED quite easily as EM does not care for chirality. Avoiding the hard math of unification, this formulation creates four bosons which boil down to the known γ, Z, and $W^\pm$ bosons.

In order for the Lagrangian to remain invariant under the $SU(2)$ transformation, the masses of the weak bosons must be zero. However, experiment shows that these are massive particles which also

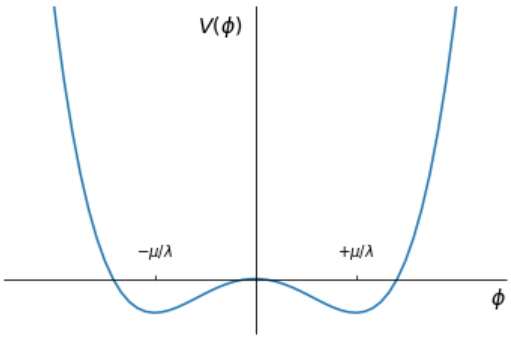


Figure 2.4: Sketch of the Higgs potential. It can be seen that the potential does not have a minimum about zero but instead has a minimum at $\pm\mu/\lambda$

imply the short acting distance of the weak force. Therefore there must be a mechanism which breaks the symmetry and gives them mass.

Higgs Mechanism and Particle Masses

The last boson in the SM is called the Higgs, H, and is the latest addition to the model. It has a mass of about 125 GeV/c^2, spin zero, and is both electrically and color neutral. It was first theorized by Peter Higgs in 1964 and found by the Large Hadron Collider in 2012. The Higgs field plays a unique role in the SM as it explains the mechanism by which some particles have mass.

In the SM Lagrangian, adding mass terms for some elementary particles breaks the symmetry needed for the interactions to work, for example the weak bosons. After all, the weak bosons are spin 1 particles which follow the Proca Lagrangian, shown earlier to not be invariant under local transformations. As before, to correct the problem one introduces a new field which, in this case, is a scalar field with the following form:

$$\frac{1}{2}(\partial_\mu\phi)(\partial^\mu\phi) + \frac{1}{2}\mu^2\phi^2 - \frac{1}{4}\lambda^2\phi^2, \tag{2.14}$$

where μ and λ are both real constants, and ϕ is the scalar Higgs field.

In the past, the equations of a free particle have had a motion component and a static component with a mass term. This scalar field is not different in that regard but what makes it different is that the ground state (non-kinematic minimum) is not when the field is zero. Figure 2.4 shows the Higgs potential and how the minimum is not at $\phi = 0$ but instead at $\phi = \pm\mu/\lambda$. The significance is that, so far, all other theories have been derived from excitations of the ground state. That means that the calculus one can perform is perturbative expansion of the field at these minimum. This particular field has two minima which, when one is chosen, symmetry is broken. This is referred to as *spontaneous symmetry breaking*.

This symmetry can be "repaired" by introducing a secondary, identical and independent scalar field. This changes equation 2.14 to the following:

$$\frac{1}{2}(\partial_\mu\phi_1)(\partial^\mu\phi_1)\frac{1}{2}(\partial_\mu\phi_2)(\partial^\mu\phi_2) + \frac{1}{2}\mu^2(\phi_1^2+\phi_2^2) - \frac{1}{4}\lambda^2(\phi_1^2+\phi_2^2), \tag{2.15}$$

where ϕ_1 and ϕ_2 are the independent scalar fields. In this case, the minimum is now a circle with a radius of size μ/λ Unfortunately, the symmetry is still broken but that can be mediated by redefining the scalar field as a complex field with the form

$$\phi \stackrel{\text{def}}{=} (\cos(\theta) + i\sin(\theta))(\phi_1 + i\phi_2), \tag{2.16}$$

with $\theta = -\tan(\phi_2/\phi_1)$ such that ϕ is a real field.

In this way, one uses a similar local transformation to what was done in QED to introduce a new spin 1 boson A_μ. With this new definition from equation 2.16, the Lagrangian becomes invariant under a local transformation. Furthermore, the new Lagrangian includes two important terms: an invariant mass term for the Higgs boson and another for the boson field A_μ. In the end, this spontaneous symmetry breaking and local gauge invariance are known as the *Higgs mechanism*. By introducing this new field in the SM Lagrangian, one now has a way to make massive spin 1 bosons.

2.2 Structure of a Proton

As explained earlier, quarks are unable to exist in isolation and therefore must form bound states with other quarks producing hadrons. Arguably, the most well known hadron is the proton, p, which is a baryon at the core of every atom. It has a positive charge and a mass slightly less than the neutron. However, one unintuitive fact of the proton is that its mass is much larger than the three valence quarks (uud) that compose it. For scale, the masses of the three quarks added together only contribute about 1 % of the total proton mass. The remaining mass comes from QCD binding energy, which is the kinetic energy of the quarks and energy of the gluon fields that bind them to each other.

One could consider the proton to only be composed of the three valence quarks to build the conceptual image of the proton. These three quarks continuously interact by exchanging gluons randomly or to keep them bound in the proton. In these exchanges, gluons may decay into quark-antiquark pairs (which recombine after a short time) or other gluons. These interactions are what make up the sea of partons inside a proton and contribute to the mass of the proton.

Considering that the proton has moving internal parts, each quark or gluon inside must contribute to the overall momentum of the proton. To describe how each *parton* (component of a hadron) contributes, physicists use a *parton distribution function* (PDF). This distribution explain how likely one is to "hit" a particular parton inside a hadron as a function of the energy scale. It should be noted that a PDF is dependent on the energy scale (often denoted as Q^2). This is defined by the longitudinal momentum between the proton and an exchanged boson in $e^{\pm}p$ collisions. At low Q^2, the valence quarks tend to be more dominant; in contrast at high Q^2, sea quarks begin to carry a small fraction of the momentum. The distributions of partons for a proton can be seen in figure 2.5 with energy scale at $10\,\text{GeV}^2$. These were determined at the ZEUS/H1 experiment over several measurements using a $e^{\pm}$ beam in collision with a proton in order to probe its structure. Also, other experiments contribute to the measurement of PDFs. PDFs are a useful tool as they are independent of the hard scattering process and can be extrapolated to higher energy scales.

The x value in the plot is called the *Bjorken x* and it described the longitudinal momentum fraction which a parton in the proton carries. It can be seen that when x is high the valence quarks are more likely to be the interacting parton. At lower x values, the gluons dominate with some probability of finding sea quarks. Thus, at higher energies one can consider the proton as a uniform sea of partons. In

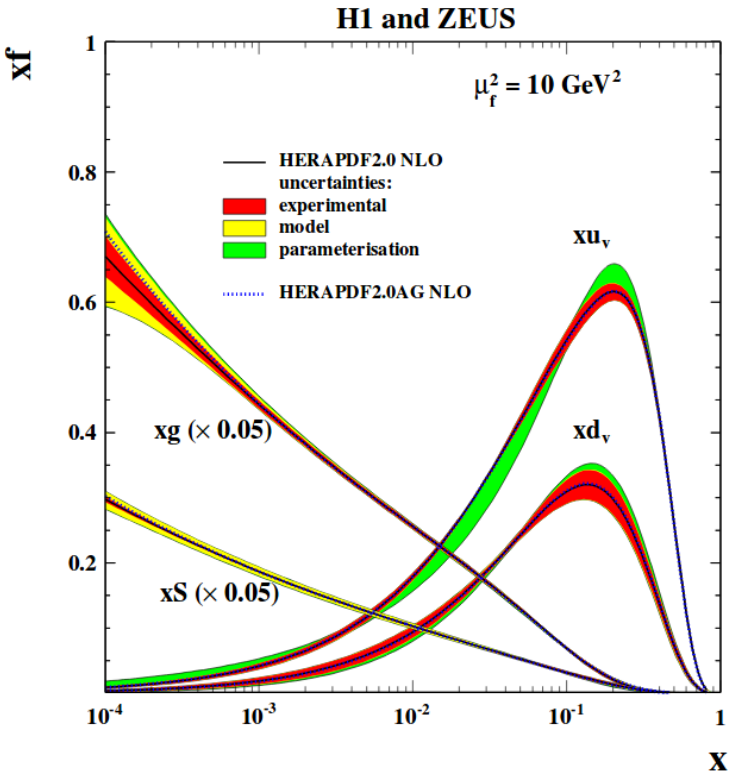


Figure 2.5: Parton distribution functions from a combination of measurements at HERA as a function of the longitudinal momentum fraction with uncertainties provided. The gluon and sea quarks PDFs have been scaled down by a factor of 20 [8].

order to probe in detail the structure of a proton, it is valuable to know the physics in particle colliders.

2.3 Physics at Particle Colliders

Unlike the HERA accelerator which experimented with $e^{\pm}p$ collisions, a hadron collider clashes together hadrons to probe quark-(anti)quark (or gluon) interactions. There are many reasons to choose specific colliding particles for expected or desired interactions. For example, a $p\bar{p}$ collider like Tevatron would have most collisions come from $q\bar{q}$ interactions. At any rate, particle accelerators or colliders are massive tools used by scientists to push measurements to their limits. These giant machines accelerate two beams of particles to speeds near that of light and have them crash at interaction points where one usually sets a detector. They allow humanity to probe and measure physical properties of elementary particles as they collide.

As mentioned earlier, the cross-section of a particular process is one of such physical values that can be measured by using these machines. Earlier, the cross-section was briefly described in the context of how Feynman diagrams are useful. In the context of particle accelerators, one can measure a total (or inclusive) cross-section as well as a *differential cross-section*. The main difference is that where an inclusive cross-section measures the overall rate of production, the differential version is dependent on

some variable. For example, one could measure a certain production with respect to energy or angle of a particular produced particle. Such information is useful for models, or event generators, physicists use to test their theories. A cross-section is typically given in units of *barn* where 1 b is 1×10^{-28} m^2.

Speaking of units, a couple of mass measurements have been given in terms of GeV/c^2 instead of the commonly used g. This is because, as one can imagine, particles are incredibly light as 1 g is about 5.61×10^{23} GeV/c^2. To add on, physicists sometimes write units in what is called *natural units* where certain constants are set to one. This unit system is used as constants like the speed of light, reduced Planck's constant, and Boltzmann constant are everywhere and only provide unit conversion. As such, units of energy, momentum, mass, temperature, and decay width are given in electron-volts (eV). Conversely, time and lengths are given in units of inverse energy (eV^{-1}). This unit choice makes for cleaner equations but it also makes dimensional analysis easier. As such, most or all upcoming units in this document are given in natural units.

Another physical property of importance in high energy particle physics is the *decay width* or decay rate denoted as Γ. Similar to the cross-section which describes probability of production or interaction, the decay rate is a measure of a given particle decay. After all, as unstable particles are created they decay into more stable particles. However, most particles do not decay into one particular channel and therefore have a decay width that is a sum of all possible decays given by:

$$\Gamma = \sum_{i=1}^{N} \Gamma_i, \tag{2.17}$$

where Γ is the total decay width which is composed by summing over particular decay widths Γ_i for N possible decays. This total decay width is related to the *lifetime*, τ, of the particle or how long it lives given by:

$$\tau = \frac{1}{\Gamma}. \tag{2.18}$$

This relation is given in natural units, as one should use an $\hbar$ to transform energy units to time. Furthermore, one can derive a probability of decay to any particular channel, in percentage, by the *branching ratio*. This quantity is intuitively given by the ratio of a particular decay width divided by the total, or:

$$BR_i = \frac{\Gamma_i}{\Gamma}. \tag{2.19}$$

Given that one has the capability of measuring or estimating the cross-section, one is able to predict the number of events one sees given some experiment. That is to say, if one wanted to measure how many events is expected, or the rate of them, one needs only the cross section and the *luminosity* of the experiment. The *instantaneous* luminosity is a measure of the collider's capability for collisions. It is related to how many particles are collided at some frequency and how much area the particle beams cover. It is defined as:

$$\mathcal{L} \overset{\text{def}}{=} \frac{f n_1 n_2}{4\pi \sigma_x \sigma_y}, \tag{2.20}$$

where $\mathcal{L}$ is the luminosity, f is the frequency of particle crossing, n_1 and n_2 are the number of particles in each beam, and $\sigma_{x/y}$ is the spread of the beam in the x or y direction. This definition of luminosity can be integrated by time to get the *integrated* luminosity. Both of these versions of luminosity are useful for calculating the event rate or total events of a particular process with cross-section σ. Event

rate and total events of a particular process are given in:

$$\frac{dN}{dt} = \sigma \mathcal{L}, \tag{2.21}$$

$$N = \sigma \int \mathcal{L}\, dt. \tag{2.22}$$

Luminosity is useful to gauge the performance of a detector in addition to its integral over time (named integrated luminosity). At the LHC, this is done with the van der Meer method described in [9, 10].

As luminosity is directly correlated to yields of interesting events, one desires to increase this value. This can be done by lowering the Gaussian spread of the beam, increase the speed of the particles (not useful in high-energy colliders) or by injecting more particles in the beam. Alternatively, one could change the value of π but that seems out of the scope of this document[2]. One can certainly use more particles but that comes with underlying collisions which may make an event "muddy." Not only can more partons interact but other objects in the beam may also collide in tandem or simultaneously. This effect is called *pileup* and it comes in two types: in-time and out-of-time pileup. In-time pileup is when the additional collisions happen at the same time as the underlying event. Out-of-time happens when crossings before and after the underlying event leave electronic signals in the detector.

The accelerator is not the only component in such an experiment. A detector is required to observe and measure phenomena created by the colliding beams. It is useful to construct a coordinate system by which one can interpret the readings from a particle detector. By convention, this makes the beam direction the Z axis, while upwards is the Y axis. In a circular collider, the X axis is directed towards the center of the accelerator. From these coordinates, one can describe kinematic variables which are attributed to objects being detected. For example, the momentum in the transverse plane, the angular direction in which objects are ejected, among others. Using the kinematics of measured particles, one can create new variables which reconstruct the event such as the transverse m_{T} or invariant mass M. The invariant mass of particles is defined in equation 2.26 and the transverse mass of two particles is defined in equation 2.27.

$$p_{\mathrm{T}} = \sqrt{p_X^2 + p_Y^2}, \tag{2.23}$$

$$\phi = \begin{cases} \arctan(\frac{p_Y}{p_X}) & p_X \neq 0, \\ (\text{sign of } p_Y)\frac{\pi}{2} & p_X = 0. \end{cases} \tag{2.24}$$

$$y = \frac{1}{2} \ln\left(\frac{E + p_Z}{E - p_Z}\right), \tag{2.25}$$

$$M = \sqrt{E^2 - p_X^2 - p_Y^2 - p_Z^2}, \tag{2.26}$$

$$\begin{aligned} m_{\mathrm{T}} &= \sqrt{M^2 + p_X^2 + p_Y^2} \\ &\approx \sqrt{2 p_{\mathrm{T},1} p_{\mathrm{T},2} (1 - \cos(\phi_1 - \phi_2))}. \end{aligned} \tag{2.27}$$

The transverse plane (XY) is valuable as the colliding particles have nearly no *transverse momentum*

[2] In 1897, a bill named the "Indiana Pi Bill" attempted to set the value of π to be 3.2 legally via erroneous methods. The senate of Indiana (U.S.A.) thankfully rejected the bill after it passed their House of Representatives.

p_T and such a quantity is conserved. This implies that the projection of momentum on the transverse plane, defined in equation 2.23 of all objects must add up to zero. Any noticeable deviation from zero implies invisible particles, namely neutrinos. These particles only interact weakly and propagate through the material easily. Therefore, the negative sum of all transverse momenta is denoted as the *missing transverse energy* E_T^{miss}, a vector sum of all neutrinos.

If partons are the colliding particles, one does not know the exact boost along the Z axis as these have a fraction of the momentum of the accelerated particle. For this reason, the *rapidity* y defined in equation 2.25 is used as a measure of this boost. However, this requires the precise measurement of the momentum in the beam direction. When the energy of the particle is much larger than its mass, such that its energy is almost entirely momentum, a quantity called *pseudorapidity* η is used. This quantity, shown in equation 2.28, is an approximation of rapidity but it can be calculated from the polar angle θ rather than energy of the particle.

$$\eta = -\ln \tan\left(\frac{\theta}{2}\right). \tag{2.28}$$

From the polar angle and the pseudorapidity, one can measure the angular separation between two objects, named ΔR.

$$\Delta R \stackrel{\text{def}}{=} \sqrt{\Delta\eta^2 + \Delta\phi^2}, \tag{2.29}$$

where $\Delta\eta$ and $\Delta\phi$ imply the difference of η and ϕ between the two objects. The usefulness of ΔR is that it is invariant under boosts in the beam direction for massless objects. Although it is not invariant for massive objects, it is still a reasonable approximation at the high energies of colliders like the LHC.

When particles collide, they typically eject debris not in the form of a simple interaction but as a cloud of other partons which may further interact. The ejected particles typically radiate in the same direction and form a *parton shower*. Typically, the parton shower forms around a small ΔR around the high energy particles ejected from collisions. This spray of collimated particles is called a *jet*. Jets are complex objects which are identified by reconstruction algorithms which determine properties, and in some case identities, of the original partons. These algorithms must be sophisticated enough to distinguish pileup or debris from what is of interest. The typical algorithm used for jets in ATLAS is the *anti-k_t* algorithm [11]. Jets are further covered in the detector section 3.2.

2.4 Top-Quark Physics

One of the protagonists of this thesis is the most massive particle, the top-quark. Being an up-type quark, the top-quark has all the properties described in Sec 2.1 for this subset of quarks. It is one of the most interesting particles in the Standard Model as it cannot hadronize and instead decays immediately (about 0.5×10^{-24} s) almost always into a W boson and a b-quark. Given its unique signature and inability to form a bound state, it gives physicists the ability to measure this particle with great detail. Unlike other quarks, properties like its polarization may be derived from its decay products.

Production

One can produce t-quarks either singly via interactions with a W boson or as a pair ($t\bar{t}$) via the strong interaction. Although a Z or Higgs bosons can decay into a top-quark pair, the largest contribution

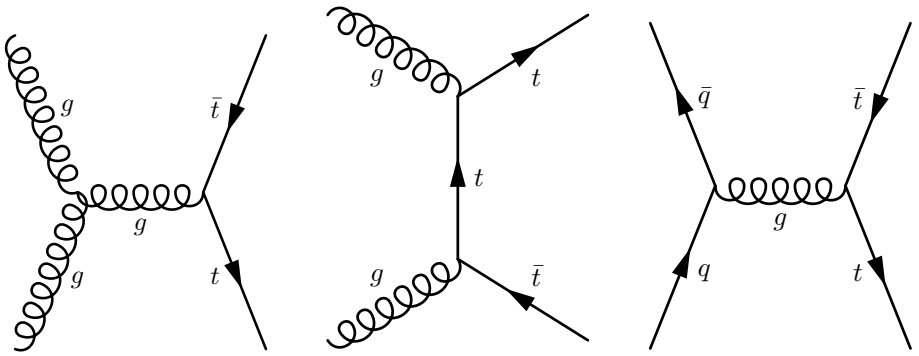


Figure 2.6: Feynman diagrams depicting a few top-antitop-quark pair production mechanisms with (a) gluon fusion and (b) quark-antiquark annihilation

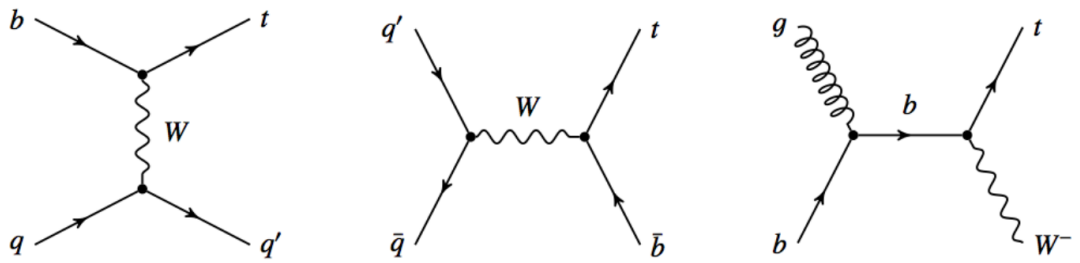
of events comes from the strong force. Incidentally, the most common way to create top-quarks is in pairs. In pp colliders like the LHC, the most common way to create top-quark pairs is via gluon fusion ($gg \rightarrow t\bar{t}$). Alternatively, in a $p\bar{p}$ collider like the Tevatron top-quark pair production happens most commonly via quark-antiquark annihilation ($q\bar{q} \rightarrow t\bar{t}$). Both processes are shown in figure 2.6. Theory predicts that the top-quark pair cross-section at center-of-mass energy $\sqrt{s}$ of the LHC (13 TeV) is $\sigma_{t\bar{t}} = 833.9\,^{+20}_{-30}\,(\text{scale})\,^{+21}_{-21}\,(\text{PDF})\,^{+23}_{-22}\,(\text{mass})\,\text{pb}$ [12–15]. This includes next-to-next-to-leading order (NNLO) in QCD and the resummation of next-to-next-to-leading logarithmic (NNLL) soft gluons and assumes the top-quark mass to be $m_t = 172.5\,\text{GeV}$.

One should not overlook single top-quark production as its various channels give physicists a different glimpse into properties of the top-quark. The three main single top-quark production processes are shown in figure 2.7. Each has its own signature, cross-section, and may give hints towards properties proposed in beyond the Standard Model physics. The predicted cross-section for these processes at $\sqrt{s}$ =13 TeV are [16–20]:

$$\begin{aligned}
\sigma_{t-\text{channel}} &= 216.99\,^{+6.62}_{-4.64}\,(\text{scale})\,^{+6.16}_{-6.16}\,(\text{PDF})\,^{+1.81}_{-1.81}\,(\text{mass})\,^{+0.39}_{-0.39}\,(\text{E}_{\text{beam}})\,\text{pb},\\
\sigma_{tW} &= 71.7\,^{+1.8}_{-1.8}\,(\text{scale})\,^{+3.4}_{-3.4}\,(\text{PDF})\,\text{pb},\\
\sigma_{s-\text{channel}} &= 10.32\,^{+0.29}_{-0.24}\,(\text{scale})\,^{+0.27}_{-0.27}\,(\text{PDF})\,^{+0.23}_{-0.22}\,(\text{mass})\,^{+0.01}_{-0.01}\,(\text{E}_{\text{beam}})\,\text{pb}.
\end{aligned}$$

The leading process, t-channel, happens when a spectator quark interacts weakly with a b-quark changing the flavor of both in the end. Seeing as how the bottom-quark is not valence in a proton, it must come from either the sea of quarks or from gluon splitting ($gb\bar{b}$ vertex). In this case, the outgoing particles are a spectator quark and a top-quark. The next process in terms of cross-section is the tW process, where a top-quark is produced in association with a W boson. Similarly to the t-channel, the initial b-quark comes from the sea or via splitting which is then excited and radiates a W boson and a t-quark. In this scenario, a W boson is emitted on-shell[3] alongside a top-quark. There are many ways

[3] On-shell is shorthand for *on the mass shell* which implies the particle obeys the energy-momentum relation ($E^2 = (pc)^2 + (mc^2)^2$). Colloquially, one can think of this as a particle which has the correct mass in accordance to its momentum or energy.



(a) t-channel (b) s-channel (c) Wt production

Figure 2.7: Feynman diagrams depicting the production of single top-quark via t-channel (a), s-channel (b), and t W channel (c). [21]

to draw the tW channel but the outgoing particles are always these two. However, if one considers the bottom quark in the proton PDF when describing this process (referred to as the *five-flavor scheme*), then the process includes a spectator b-quark. This is important as it plays a role in the relationship between tW and $t\bar{t}$. Lastly, the s-channel is produced by quark-antiquark annihilation into a virtual W boson which decays into a t-quark and a b-quark (one must be anti-quark). This is the rarest of processes at the LHC since finding an antiquark in the proton is harder than finding a gluon.

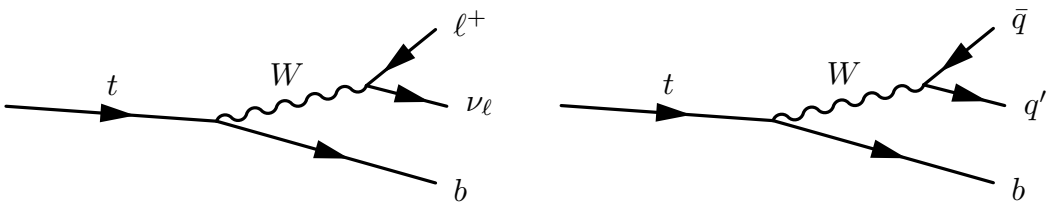
Although each of these three is interesting on its own right, the tW channel is a focus in this document. To be precise, it is not the tW channel by itself but its relationship with top-quark pair production. If one sees these two processes as separate, one should note the differences in cross-sections as tW production is not even one tenth of $t\bar{t}$. In a tW focused analysis, this poses a challenge as $t\bar{t}$ is one of the main backgrounds (if not *the* main background).

Decay

Earlier, it was mentioned that the top-quark decays almost always into a bottom quark and a W boson. This is due to the V_{tb} element of the CKM matrix being nearly one. Given that the top-quark mass is more than twice the sum of W and bottom quark masses, it can produce both particles on-shell. Moreover, the fast decay of the top-quark means that information based on spin can be drawn from its decay products since they will retain this information. Therefore, top-quark polarization can also be studied with more precision than other quarks.

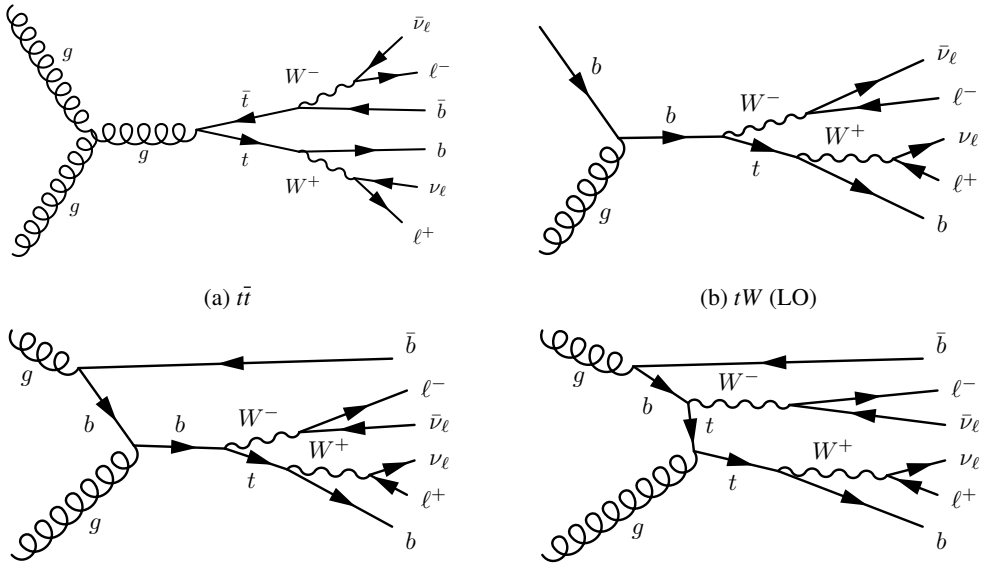
As the daughter particles of the top-quark are not inherently stable, they too decay into more stable particles. The b-quark typically forms long-lived hadrons which can be identified in experiments by their flight-time and decays among other properties. These quarks are, of course, not alone but in jets in real-life experiments. The W boson has two decay modes: leptonic and hadronic, shown in figure 2.8.

As mentioned in the previous section, two top-quark production processes of interest are tW and $t\bar{t}$ with their interplay as a focus. Real life colliders yield many different processes and physicists are unable to "trigger" specific interactions. For such reasons, one considers kinematic regions where the desired processes are enhanced. To explore tW and $t\bar{t}$'s interactions, one should look at the potential signature both of these processes have. Figure 2.9 shows the production and decay of both tW and $t\bar{t}$ with the decay showing what final state particles one would expect in an experiment. From these diagrams, one can see that tW and $t\bar{t}$ have similar final states when tW is at leading order; shown by



(a) (b)

Figure 2.8: Decay signatures of the top-quark. The W boson from the top-quark decay can decay leptonically (a) and hadronically (b).



(a) $t\bar{t}$ (b) tW (LO)

(c) tW (four-flavor) (d) tW (NLO)

Figure 2.9: Decay signatures of tW and $t\bar{t}$ with examples of tW and $t\bar{t}$ interference. The final state particles one would see in a detector are shown for $t\bar{t}$ (a), tW (b), and interfering diagrams (c) and (d). The final states of tW and $t\bar{t}$ differ only by the multiplicity of b-quarks at leading order (LO). However, when tW is generated in the five-flavor scheme or at NLO, the end-state particles are equivalent and therefore interfere. In the NLO case (d), a top-quark may be generated off-shell bringing up the question if this is a $t\bar{t}$ or tW diagram in the first place.

figures 2.9(a) and 2.9(b). In this case, the only difference lies in b-quark multiplicity as figure 2.9(b) shows only one. However, when tW is generated in the four-flavor scheme, a secondary spectator quark can be seen in figure 2.9(c) which makes the final state identical to that of $t\bar{t}$. If that were not enough, figure 2.9(d) shows a diagram of tW at NLO which has a secondary top-quark along with the same final state.

tW and $t\bar{t}$ Interplay

Since tW and $t\bar{t}$ have similar final states along with tW at NLO having a second top which may become on-shell, these two processes interfere. One could argue that diagrams like the one shown

in figure 2.9(d) is not tW or $t\bar{t}$ but this would be splitting hairs. For the most part, the labels tW and $t\bar{t}$ are insufficient and most physicists have chosen to call singly (tW) and doubly ($t\bar{t}$) resonant diagrams. As figure 2.9(d) generally has an off-shell top, this is considered a singly resonant diagram. Nonetheless, both processes fall under the umbrella term $WWbb$ which identifies them by the final product. Although there are many reasons to study each process independently, both are considered background processes in other top-quark analyses. When taken separately, the interplay between tW and $t\bar{t}$ introduces large uncertainties which typically affect the quality of such analyses.

One form of interplay between these two processes is their interference, as mentioned earlier. One can consider the $WWbb$ amplitude $\mathcal{A}_{WWbb}$ as the sum of both tW and $t\bar{t}$ shown in equation 2.30:

$$\begin{aligned} \mathcal{A}_{WWbb} &= \mathcal{A}_{tW} + \mathcal{A}_{t\bar{t}}, \\ |\mathcal{A}_{WWbb}|^2 &= |\mathcal{A}_{tW}|^2 + 2\mathrm{Re}|\mathcal{A}_{tW}\mathcal{A}_{t\bar{t}}| + |\mathcal{A}_{t\bar{t}}|^2 \\ &= \mathcal{S} + \mathcal{I} + \mathcal{D}, \end{aligned} \tag{2.30}$$

where one can see the squared amplitude of $WWbb$ is composed of a singly resonant term $\mathcal{S}$, an interference term $\mathcal{I}$ and a doubly resonant term $\mathcal{D}$. This amplitude is important as it is used to calculate the cross section of either the overall process or individual ones.

In this case, it becomes obvious that isolating either of these processes is non-trivial due to the interference term. For this reason, two treatments of the tW contribution were developed in [22] and briefly mentioned in [23]. In one of the treatments, the $t\bar{t}$ contribution is removed at the beginning of the calculation, leaving only the singly resonant contribution in the square amplitude. This is aptly called *Diagram Removal* (DR) and comes at the cost of not being gauge invariant as contributions from some parts of the phase space are removed. The second treatment attempts to subtract the $t\bar{t}$ contribution after squaring, called *Diagram Subtraction* (DS). The estimate of the doubly resonant contribution $\tilde{\mathcal{D}}$ and the actual contribution $\mathcal{D}$ are intended to cancel. For DS, the subtraction is a gauge invariant estimate which may mean the subtraction may be non-zero but as close as possible leaving it with some residual δ. Both of these definitions are shown in equations 2.31 and 2.32:

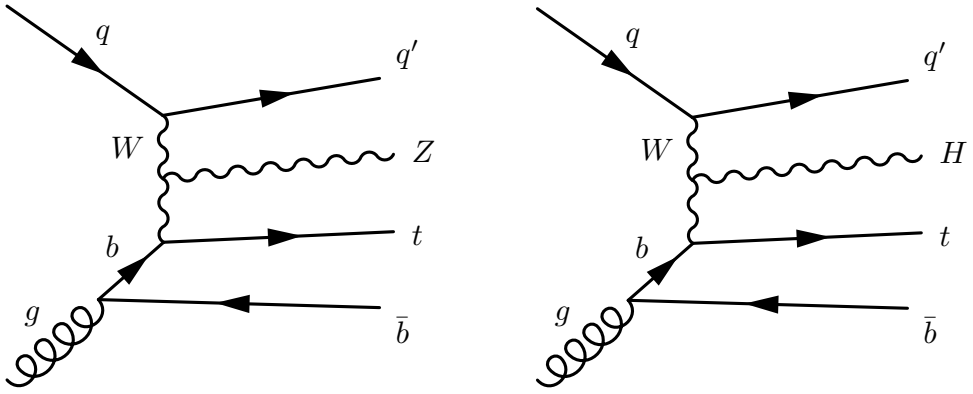
$$DR : |\mathcal{A}|^2 = \mathcal{S}, \tag{2.31}$$

$$\begin{aligned} DS : |\mathcal{A}|^2 &= \mathcal{S} + \mathcal{I} + \mathcal{D} - \tilde{\mathcal{D}}, \\ &\simeq \mathcal{S} + \mathcal{I} + \delta. \end{aligned} \tag{2.32}$$

Additionally, alternative models exist which shuffle initial/final state particles which are similarly named but enumerated: DR1 and DR2, along with DS1, DS2, DS3, and DS4. [24] Although tW analyses would prefer the interference to be minimal, it contributes about 10 % of the total $WWbb$ cross-section [25]. For this reason, the interference effects are considered interesting for study and is therefore a focal point in the analysis described herein.

Rare Top-Quark Processes

Two examples of rare single top-quark processes are t-channel production in association with either a Higgs or a Z boson; tHq and tZq respectively. Figure 2.10 shows both productions and their similarities. The naming convention comes from the final state containing a spectator quark, a top-quark, and a Higgs boson. Although the diagram shows a b-quark from gluon splitting, this is usually so forward



(a) tZq (b) tHq

Figure 2.10: Feynman diagram depicting one of many ways to produce tZq (a) and tHq (b) final states. A Higgs or Z boson could be radiated from any of the particles involved in this diagram but radiation from the W boson was chosen by preference. Both are rare processes where the associated bosons may decay leptonically (including τ) or hadronically.

(along the beam direction) that one cannot identify its flavor. It should be of note that the Higgs or Z bosons may radiate from nearly any particle in this t-channel diagram.

Both of these processes look nearly identical and are considered rare as their cross-sections are orders of magnitude smaller than other previously mentioned single top-quark production modes. For reference, the predicted cross-sections are [26, 27]:

$$\sigma_{tZq} = 904\,^{+49.7}_{-100.3}\,(\text{scale})\,^{+5.4}_{-5.4}\,(\text{PDF})\,\text{fb},$$
$$\sigma_{tHq} = 82.2\,^{+5.9}_{-9.0}\,(\text{scale})\,^{+0.32}_{-0.32}\,(\text{PDF})\,\text{fb}.$$

It is important to emphasize the units used here are fb and not pb like the previously stated cross-sections. Therefore, these rare processes benefit from using the full phase space for measurement purposes.

2.5 Tau Physics

Another important particle in the studies herein is the tau lepton. Where the top-quark is the most massive particle overall, the tau lepton is only the most massive of the leptons. With a mass of $1.777\,\text{GeV}/c^2$, it decays quickly (in about 2.90×10^{-13} s [7]) compared to other particles of its class. Its short lifetime means that, like the top-quark, it can typically only be detected by its decay products rather than directly.

Production

Tau leptons can be produced in the same way all other leptons can. Although tau leptons have the same probability to be emitted electroweakly, this changes for a Higgs boson decay. Figure 2.11 shows the branching ratios of Higgs boson's decay modes where the fourth most likely decay mode is to

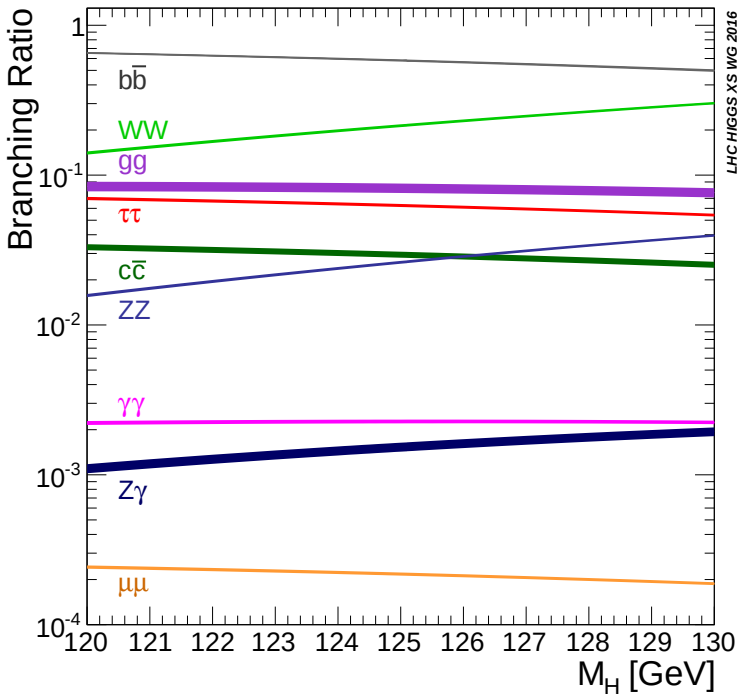


Figure 2.11: Higgs boson branching ratio and uncertainties near the measured Higgs boson mass. The figure shows a color coded probability of the Higgs boson to decay into the listed particles. It can be seen that the Higgs boson decaying into two tau leptons is the fourth most likely mode. [28]

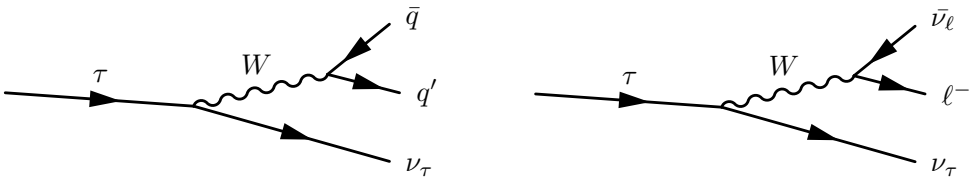
two tau leptons. This is orders of magnitude greater than muon decay, which is the last in the plot. This implies that processes where the Higgs boson is involved may contribute a greater number of tau leptons. In this case, rare processes like tHq benefit from including tau leptons in their final states for their measurements.

Decay

The tau lepton has a unique property compared to other leptons: its high mass allows it to decay hadronically. The tau lepton decays by emitting a W boson and a ν_τ, where the W boson may decay hadronically ($\simeq 65\,\%$) or leptonically ($\simeq 35\,\%$). Both of these decay modes are shown in figure 2.12.

In ATLAS analyses, leptonically decaying tau leptons are typically treated as the lighter lepton instead. This is because lepton signatures are clean when compared to hadronic signatures. Leptonic decays suffer from missing energy coming from two neutrinos per tau lepton decay. However, this is only about a third of all possible τ lepton decays. Hadronic decays are more complicated as tau leptons may decay into one or three charged hadrons in association with or without neutral hadrons.

In order to conserve charge, the tau lepton must decay into an odd number of charged particles; most commonly, one or three. In ATLAS, hadronically decaying tau leptons are classified as called one- or three-*prong* depending on how many charged particles are associated. This is useful as the amount of hadrons, charged or neutral, influence the kinematics of the event and therefore reconstruction faces an identification challenge. For context, table 2.2 shows the percentages of hadronic decay modes as



(a) hadronic (b) leptonic

Figure 2.12: Illustration of tau lepton decay modes. The tau lepton only decays via the charged weak bosons which subsequently decay hadronically (a) or leptonically (b). A neutrino is always emitted to conserve lepton quantum number.

Decay mode	Branching Ratio [%]
$h^\pm$	11.5
$h^\pm + \pi^0$	25.9
$h^\pm + \geq 2\pi^0$	10.6
$3h^\pm$	9.46
$3h^\pm + \geq 1\pi^0$	5.09

Table 2.2: Branching ratio of different hadronic tau lepton decay modes in percentages with respect to total tau lepton decay modes. $h^\pm$ denotes a charged hadron (pion or kaon) and π^0 is the associated neutral hadron. Neutrinos are implied in each of these decay modes. [7, 29]

percentages of the total possible tau lepton decay modes.

Given that hadronically decaying tau leptons have many signatures and kinematics, the problem of tau identification is not trivial. Processes where several hadrons are created may mimic the signature of a tau. For example, a simple multi-jet event where any number of neutral hadrons are in close proximity to one or three charged hadrons. In ATLAS, such tau leptons are identified by a sophisticated recurrent neural network (RNN) algorithm described in [30]. Without delving into the details of the RNN presented, the idea is summarized as using several variables available from the ATLAS detector to identify real tau leptons from mimicking backgrounds. A more in-depth summary can be found in section 3.3.

CHAPTER 3

The Large Hadron Collider and ATLAS

3.1 The Large Hadron Collider

On the French-Swiss border, the largest particle accelerator on the planet can be found. The Large Hadron Collider (LHC) is a proton-proton collider used by the "Conseil Européen pour la Recherche Nucléaire" (CERN), or European Organization for Nuclear Research. The LHC is a circular collider of 27 km in circumference and buried over 100 m underground. Not only is it the largest circular collider on earth, but also the one with highest energy collisions. By accelerating two proton beams in opposite direction at 6.5 TeV per beam, the LHC holds the world record of 13 TeV[1] center-of-mass energy collisions. The proton beams are carried in separate tubes with opposing flow which are kept at ultra-high vacuum[2].

Superconducting magnets are used to keep beams on their intended circular path and focus the beam. Dipole magnets are used to keep the beam in a circular trajectory while quadrupole magnets focus the beam with higher-order magnets used for corrections. These magnets are cooled to temperatures lower than 2 K using over 120 t of liquid helium. Given the quantity of magnets, this feat of engineering makes the LHC the largest cryogenic facility in the world [31].

Where magnets are used to curve the beam the acceleration of protons is handled by electric fields. Figure 3.1 shows a schematic of the different components in the LHC which accelerate protons in stages before entering the main ring. In the years spanning the last data-taking run, the linear accelerator 2 (LINAC 2) was used to feed 50 MeV protons to the Proton Synchrotron Booster (PSB). After 2020, the linear accelerator 4 (LINAC 4) is instead used to accelerate H^- atoms to 160 MeV and feed them to the PSB after stripping both electrons and leaving only a proton behind. The PSB accelerates protons to 2 GeV and then injects them into the Proton Synchrotron (PS) where they reach 26 GeV. Once here, the PS passes the protons to the Super Proton Synchrotron (SPS) which accelerates them to 450 GeV and they are then injected to the main ring. Finally, they reach the target energy of 6.5 TeV before collisions at the four designated interaction points where the experiments are housed.

The four main experiments at LHC interaction points can be seen in figure 3.2. These are A Toroidal LHC Apparatus (ATLAS) [33], Compact Muon Solenoid (CMS) [34], LHC-beauty (LHCb) [35], and A Large Ion Collider Experiment (ALICE) [36]. The first two experiments, ATLAS and CMS, are

[1] As of 2022, the center-of-mass energy achieved by the LHC has increased to 13.6 TeV.

[2] Ultra-high vacuum is a technical term which is defined as pressures lower than 100 nPa. These low pressures are useful as a particle's free mean path reaches lengths of about 40 km; ensuring no collisions inside the beam.

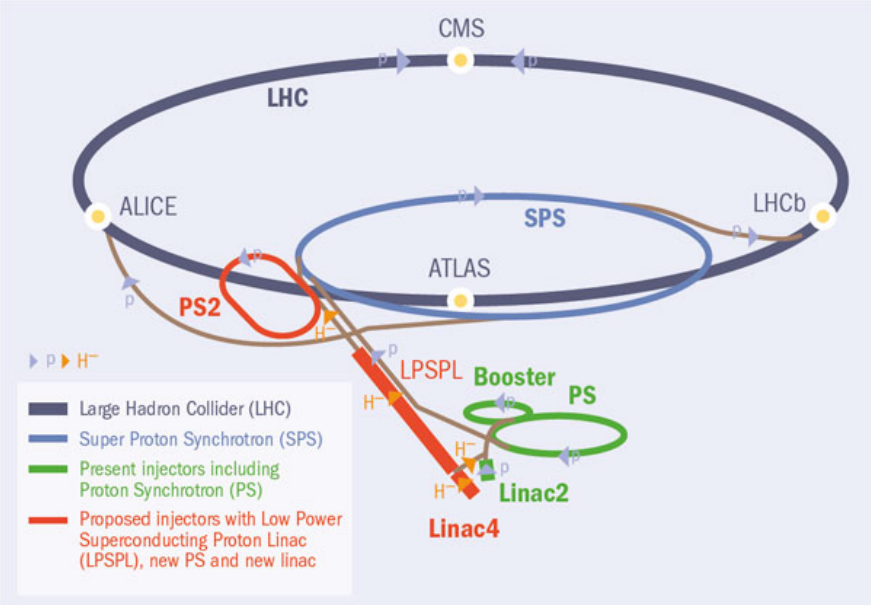


Figure 3.1: Schematic of the LHC and all its sequential accelerators [32].

general purpose detectors. These are used to cover a plethora of cases where one may find new physics, or explore properties of known particles and perform precision measurements of the SM. Additionally, having two such detectors allow for cross-confirmation of discoveries at the LHC. LHCb is a specific purpose detector with the goal of understanding matter-antimatter asymmetry. The experiment is designed to mainly detect rare b-quark bound states along the beam direction and study charge-parity violation. ALICE focuses mainly on heavy-ion physics with the goal of studying the formation of quark-gluon plasma. These collisions recreate similar conditions to those just after the Big Bang which may help in the understanding of confinement and hadron formation.

The goal of having a high energy particle collider like the LHC is to probe previously unseen physics, test models beyond the standard model, and measure properties of particles like the Higgs boson. In order to operate and allow physicists to perform analyses on data, the LHC must be backed with the computing power to handle such a monumental task. For example, the LHC has provided over tens of petabytes over its data-collecting periods which must be stored and processed somewhere. For this reason, the LHC has established the Worldwide LHC Computing Grid (WLCG) whose goal is to provide computing resources such as storage, distribution, and analysis. This computing grid is supported by over 170 sites in over 42 countries which provide over 1.4 million processing cores and 1.5 EB of storage.

So far, the LHC has had two data-taking runs appropriately named Run-1 (2010 to 2012) and Run-2 (2015 to 2018). Although the LHC recently ran with 13 TeV collisions, it did not start at that high energy. Run-1 saw a center-of-mass energy of 7 TeV from 2010 to 2011 and increased to 8 TeV in 2012 [37]. During this time period, the LHC delivered a total integrated luminosity of about 29.2 fb^{-1}. After Run-1 finished in 2013, the LHC was shut down for two years where the LHC and various experiments would receive upgrades [38].

In 2015, the LHC began Run-2 with collisions at 13 TeV. By the end of 2016, the integrated luminosity delivered by the LHC had already surpassed the luminosity given by Run-1. Each

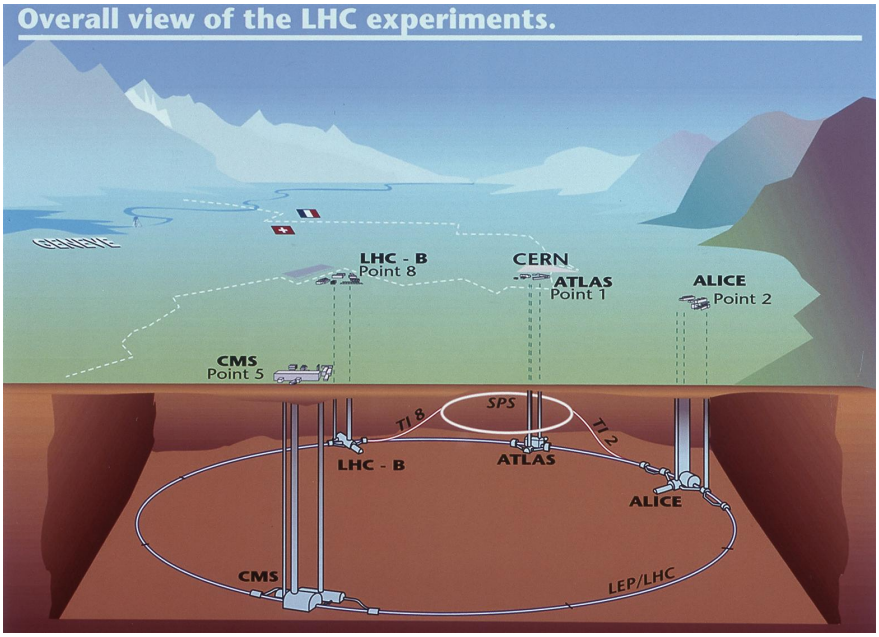
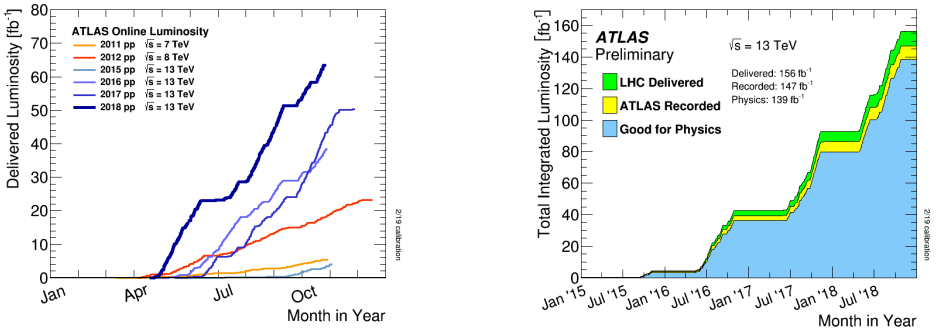


Figure 3.2: Drawing of the LHC and its four different detectors.



(a) yearly

(b) consecutive

Figure 3.3: Plot showing the delivered integrated luminosity per year at the LHC. [40] Each line in (a) is color-coded to separate Run-1 from Run-2 and shaded for years between each run. The areas of (b) show how much was delivered by the LHC, how much was recorded by ATLAS, and how much of that was useful for analyses.

subsequent year would increase the luminosity and deliver greater quantities of data shown in figure 3.3(a). Altogether, Run-2 provided about 139 fb^{-1} of integrated luminosity shown in figure 3.3(b). Run-3 officially began in July of 2022 [39] with collisions with center-of-mass energy at 13.6 TeV with ATLAS recording collisions in August of the same year.

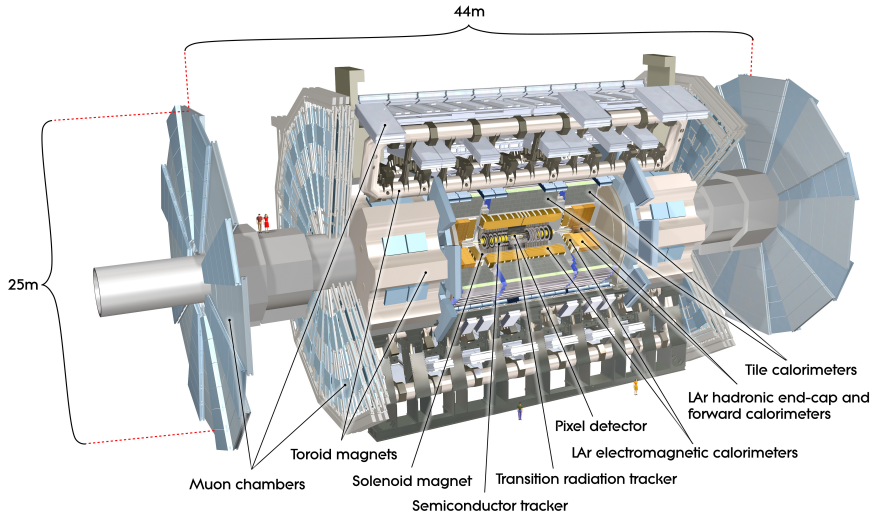


Figure 3.4: Detailed depiction of the ATLAS detector with labeled components [41].

3.2 ATLAS

ATLAS is the largest particle detector of all LHC experiments and largest volume detector to date. Named after the mythological colossus Atlas, this detector towers above humans at 25 m tall and is measured at 44 m long. Weighing at about 7 000 t and with over 3 000 physicists as members from over 175 institutions in 38 countries, this massive piece of machinery is a marvel of engineering and a monument to what can be achieved by collaborating countries. The behemoth is shown in figure 3.4 with two average sized humans for comparison. In addition, the detector is sliced such that each component is visible and labeled.

As shown in figure 3.4, the ATLAS detector is composed of many sub-components which serve a multitude of purposes. Some components which serve the same purpose are typically grouped together under one umbrella term such as the inner detector (ID). The calorimeter system is divided by the particles which they are designed to interact with and are therefore named the electromagnetic and hadron calorimeters. As charged particles bend in a magnetic field, a magnet system is installed such that one may better identify the properties of charged particles. In the outer part of the detector, the muon detectors can be found which are necessary as muons penetrate further than electrons. An in depth coverage of the ATLAS detector during Run-1 can be found in [42]. Details on subcomponents and the differences between Run-1 and Run-2 are described in the coming sections.

After the long shutdown of 2018, the ATLAS detector received significant upgrades and had maintenance work done in preparation for Run-3. These upgrades are briefly described in section 3.2. That said, the following sections cover the details of the ATLAS detector during Run-2 as the data used in this documents was recorded then. A readable breakdown of these upgrades can be found in [43].

Inner Detector

Beginning only a few centimeters from the colliding beams is where the inner detector can be found. The main purpose of the ID is to track charged particles as they pass through its subcomponents. Because of the magnetic field which covers the ID and calorimeter systems, charged particles are bent and therefore their momentum and charge can be measured. With the ability to see tracks of charged particles, one can be associate them to a vertex from which they came; implying an interaction point. Furthermore, the ID is composed of subdetectors which are the pixel detector (Pixel), semiconductor tracker (SCT), and transition radiation tracker (TRT). The ID's subsystems cover up to a range of pseudorapidity $|\eta| < 2.5$.

The ID uses two different types of particle detectors for tracking: silicon detectors (Pixel and SCT) and straw trackers (TRT). Silicon detectors are a silicon semiconductor placed in an external electric field which creates a potential inside the material. As charged particles pass through the silicon, they interact with electrons in the material which cause electron-hole pairs. These are attracted to opposite sides of the field and once they accumulate at the extremes, then they produce a current. The Pixel detector layer is composed of many of these small silicon detectors which are placed in a two-dimensional grid with separate electronics for precision. The SCT differs from the Pixel by using strips of material as contacts instead of individual pixels.

The second type of detector used in the ID are straw trackers. These are long tubes which are filled with some gas that can become ionized as particles pass through. At the center of the tube is a wire which runs the entire length of the tracker. By charging walls of the tube (or the wire), a potential differential is created inside the tube. Similar to the silicon in the previous detector, charged particles ionize the material inside the electric potential. In this case, the ionized gas atoms drift towards the wall or wire depending on charge and collide with other gas atoms. With each collision, more gas particles are ionized; a process which may happen several times. This avalanche of ionized gas particles generate a current once they contact the wire or wall of the detector. These straw tubes are arranged such that when multiple are struck, a track can be reconstructed.

The Pixel detector is first to be struck by incident particles as it is the closest to the beam. It is composed of three radial layers and three disk layers on each end-cap. During the upgrade before Run-2 began, the Insertable B-Layer (IBL) was introduced to the innermost part of the detector. The purpose of this new layer was to improve both track and vertex reconstruction in the ID [44]. The IBL is composed of 224 modules with each module containing about 26 000 pixels. Each layer thereafter is instead composed of 1 744 modules with each module containing 16 readout chips and about 47 000 pixels. The IBL pixels measure $50\,\mu\text{m} \times 250\,\mu\text{m}$ where the other layers have a size of $50\,\mu\text{m} \times 400\,\mu\text{m}$. Given their purpose, all components must be radiation hardened to be useful in the experiment. As precise as these pixels are, they have intrinsic inaccuracies in the azimuthal $(R - \phi)$ direction for all layers of $10\,\mu\text{m}$. Similarly, the inaccuracies in the axial (z) direction for the barrel and along the radial R direction are $115\,\mu\text{m}$.

Second radially after the Pixel detector is the SCT which performs a similar function as the Pixel layer. As mentioned before, these are not individual pixels but instead strips which measure $80\,\mu\text{m} \times 12\,\text{cm}$. The SCT has four double layers of silicon strips which span an area over $60\,\text{m}^2$. The accuracy per module in the $(R - \phi)$ direction for both barrel and end-caps are $17\,\mu\text{m}$. The axial (z) uncertainty is $580\,\mu\text{m}$ for the barrel while the uncertainty of the disks in the radial R direction is $580\,\mu\text{m}$.

The outermost part of the ID is the TRT. The TRT is composed of several straw trackers which measure 4 mm in diameter and can be as long as 144 cm. About 298 000 straws are used in this layer

of the detector with a potential of about −1 500 V. Between the straws there is transition radiation component of the detector. This material has varying indices of refraction which cause the high energy particles to produce transition radiation and therefore leave stronger signals in the straws. The amount of radiation emitted depends on the speed of the particle with the greatest radiation coming from lighter particles. By construction, the TRT only provides information in the $R - \phi$ plane. Each straw has an uncertainty of 130 µm.

Calorimeters

In order to measure the energy of particles, physicists use calorimeters. These devices are typically made of materials which use specific interactions to stop certain particles. In general, these materials interact with incident particles and cause cascades of secondary particles. ATLAS uses two calorimeters which surround the solenoid magnet system encasing the ID. Where the ID tracks particles as they pass through, the calorimeters absorb these particles in order to measure the deposited energy. The two calorimeters, as mentioned before, are the electromagnetic calorimeter (EM) followed radially by the hadron calorimeter (HCAL). In contrast to the ID, the calorimeter system covers a greater range in pseudorapidity up to $|\eta| < 4.9$.

The EM calorimeter measures the energy deposited primarily by photons and electrons. It is divided into three parts that cover different η ranges: The central region is covered by the barrel in the range of $|\eta| < 1.475$ while the end-caps have two coaxial wheels each that cover $1.375 < |\eta| < 3.2$ This calorimeter is composed of two materials: one which absorbs made of lead and stainless steel, and another which samples made of liquid argon. These materials make it so photons go into pair production which cause a cascade in the material that can be measured. Similarly, single electrons or positrons induce the same cascade effect when interacting with the detector material. Additionally, the EM calorimeter covers the complete polar angle without any gaps in the azimuthal direction by using an accordion geometry.

The inner region of the EM calorimeter, ($|\eta| < 2.5$), is divided into three sections for improved precision. Each of these sections give a high level of detail, decreasing in granularity outward. The best resolution is from the innermost strip with $\Delta\eta \times \Delta\phi = 0.003 \times 0.1$. This part acts as a pre-shower detector which provides precise measurements in the η in order to help with particle identification. In general, the accuracy in the end-caps is worse than central regions but it is still useful in jet reconstruction and missing transverse energy measurements.

The hadron calorimeter measures particles which pass through the EM calorimeter but interact strongly. Like the EM calorimeter, the HCAL is composed of an absorbing material, steel, and scintillating tiles which sample the barrel region. These are called tile calorimeters and they encompass the region $|\eta| < 1.7$. The mid-outer region ($1.5 < |\eta| < 3.2$) has an end-cap which is composed of liquid argon like the EM calorimeter. However, this component uses copper as the absorbing material instead. The forward region ($3.1 < |\eta| < 4.9$) has a high-density forward calorimeter composed of copper in the first layer followed by tungsten thereafter.

Muon Spectrometer

The component furthest from the beam is the muon spectrometer (MS). As its name suggest, this component is intended to detect muons which pass through the entirety of the detector. This is because

muons penetrate most of ATLAS's subcomponents without depositing much energy. Therefore, muons leave tracks in the ID as they are charged and then in the muon spectrometer.

The muon spectrometer is composed of three components: three toroidal magnets which provide magnetic fields, several chambers designed to measure muon tracks, and a set of triggering chambers with accurate time resolution. Similar to the magnets surrounding the ID, the magnets in this stage curve the muon tracks in order to measure their momentum. The muon is tracked by drift tubes similar to those of the TRT. However, the resolution of these drift tubes is instead 80 µm in the inner chamber and 60 µm in the forward region. In the barrel region, the Resistive Plate Chambers (RPC) trigger when a muon passes through. The Thin Gap Chambers (TGC) perform the same task but in the forward region of the detector. Their combined coverage is only up to $|\eta| < 2.4$.

Magnet System

These magnets are critical for the experiment and it is composed of two magnet systems: the central solenoid magnet and a toroidal system. The central solenoid magnet is right outside the ID and provides a field strength of 2 T. The strength is necessary to bend high-energy charged particles traversing the ID for tracking purposes. The bending is proportional to the momentum of the particle and proportional to the field strength.

The toroid magnet is made up of three parts with one in the center region and one at each end-cap. This system has eight separate coils that generate a magnetic field up to 4 T. Both magnet systems need to be placed in a cryostat as their maximum temperature to work is about 4.5 K [45]. Each barrel toroid has its own separate cryostat system due to their size.

Triggers and Data Acquisition

Earlier in section 3.1, the massive amount of data generated by the ATLAS detector was mentioned. However, not all data that the detector reads out is useful for physics analysis. The discrimination of events is done by the trigger and data acquisition system (TDAQ). The TDAQ selects about one event in every 200 000 that would be interesting for analysis. This selection is done by passing two triggers.

The first trigger used in Run-2 is completely hardware based. The definition of "interesting" is based on information of reduced granularity components in the detectors. Typically, these are high transverse momentum objects like leptons and jets. Additionally, large missing and total transverse energy plays a role in the selection. The decision about an event's quality is reached in only 2 µs after the interaction takes place. As an event passes this trigger, the full detector information is read from electronics into readout drivers and then buffers.

The second trigger filters events based on the reconstruction at a rate of 75 kHz. The decision to keep an event at this point is based purely on the information of the first trigger. A passing event is then fully reconstructed and transferred to storage associated with the event filter. The reconstruction and transference takes between 1 to 10 ms to accomplish. Lastly, the final selection is made where passing events are kept for offline analysis. The information is, at this point, reduced by a factor of 10 from the second trigger level. At the end of the chain, the rate of recorded events reaches about 300 MB s^{-1}. More detail in [46].

Run-3 Upgrades

The ATLAS detector has undergone an upgrade between Run-1 and Run-2, as mentioned in section 3.1. Just as before, during the shutdown from the end of 2018 the ATLAS detector received several upgrades over 3.5 years mentioned in [43]. One such upgrade was the addition of the New Small Wheel Detector (NSW) which sits in the inner layer of the forward muon spectrometer. This upgrade is intended to stay for the long run (estimated 20 years) and be present for High Luminosity LHC (HL-LHC) operations. The NSW is composed of two gaseous detector technologies called micromegas (MM) and small-strip thin-gap chambers (sTGC). The upgraded resolution of this component is instrumental for the TDAQ system as it is intended to reduce the saving of undesirable background events. Each wheel is composed of 16 sectors of which has 2 million MM and 350 000 sTGC readout channels.

Additionally, new muon chambers intended to be used in the HL-LHC were installed. These include eight small-diameter monitored drift tube (sMDT) modules and 16 next-generation RPCs. The RPC layers are introduced in the inner layer of the muon spectrometer and provide a higher granularity than previous generations. The new sMDT modules, as their name suggest, are half the size in diameter. Their decrease in size provide a higher rate capability.

The liquid argon calorimeters are now fitted with new digital electronics which improve trigger selection. This improves the resolution for the EM calorimeter's triggers. These electronics also increase the information available at trigger level which improves the calorimeter's ability to reject non-electron or non-photon objects. In total, 5 000 new optical fiber cables were added. Additionally, 1 524 new readout boards were refurbished and 124 new trigger readout boards were introduced. New "super-cells" were introduced which provide information from every calorimeter layer at a higher granularity. These cells provide over 23 TB s^{-1} of data.

The TDAQ system, with the HL-LHC approaching, also received hardware and software upgrades. The new calorimeter triggers are large-scale Field-Programmable Gate Arrays (FPGAs). These arrays allow for more complex algorithms whose purpose is particle identification and missing energy calculation. The NSW detector, as mentioned earlier, also improves trigger ability and reduces background rates. Furthermore, the TDAQ trigger software was rewritten with multi-threading in mind. As luminosity increases, so do potential events and therefore serialized software meets a ceiling of computational potential. With the increase of data, the analysis framework used by ATLAS must also improve in performance. For this reason, a refactoring of code has been performed to make the framework multi-threading safe. Although not part of this thesis, it is worth mentioning that I contributed to the refactoring of jet reconstruction modules.

3.3 Reconstruction and Identification of Objects

Using the components described earlier, ATLAS is a detector built to detect and measure several different types of objects and their properties. Figure 3.5 depicts how some particles would behave as they traverse the detector. The curved lines in this diagram show the effect of the magnet system has on charged particles. As neutral particles cannot ionize gases of the TRT or interact with the Pixel or SCT systems, they are invisible and leave no tracks in the ID. That is with the notable exception of neutrinos which do not interact with the detector and instead are "seen" as missing transverse energy, $E_{\mathrm{T}}^{\mathrm{miss}}$. The design of the EM calorimeter implies that both photons and electrons are stopped at this point while muons and hadrons move on. Although most of the energy of hadrons is deposited in the

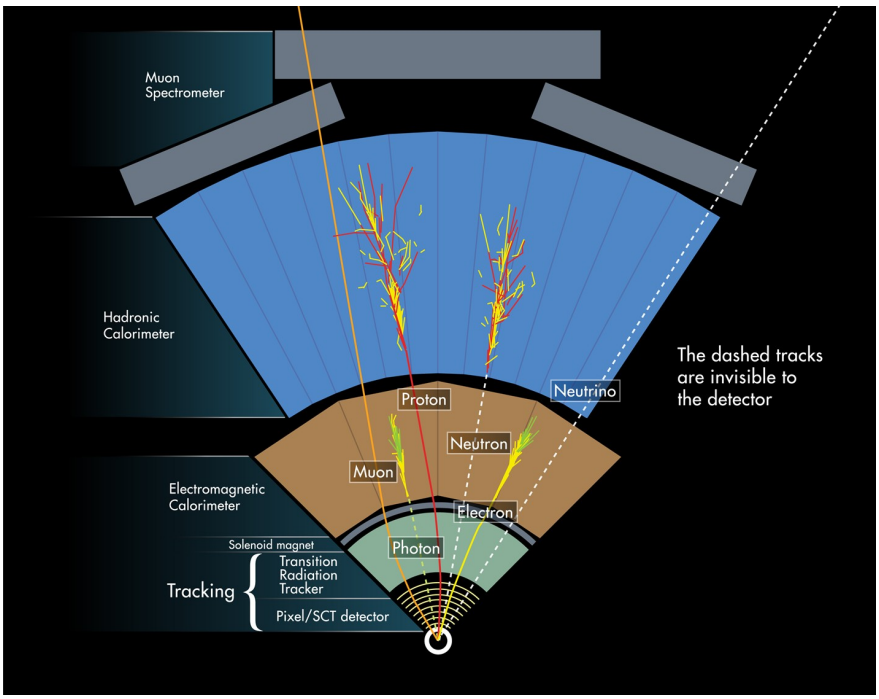


Figure 3.5: Cross sectional image of the ATLAS detector showing how different types of particles are detected and measured [47].

HCAL, charged hadrons may deposit some of their energy in the EM calorimeter as they pass through it.

Given that the top-quark and tau leptons decay before reaching the calorimeters, they still decay into objects that may be reconstructed nonetheless. The details of how such objects are detected are explained in the upcoming sections. As tau leptons may decay leptonically, such decay are typically counted as the lighter lepton instead. To elaborate, typical ATLAS analyses make no difference from a muon from the primary interaction or a muon which came from a tau lepton in the primary interaction. This does not hold true for hadronically decaying tau leptons as they are reconstructed instead of being treated as jets.

Charged Tracks in the ID

The inner detector and muon spectrometer of ATLAS provides the means to measure charged particles as they traverse the material. Their momentum and charge are derived from the curvature of the bent tracks in the ID. Several quantities are used from this information to identify and correctly label objects.

Two such quantities are the impact parameters, d_0 for transverse and z_0 for longitudinal. An impact parameter is a measure of the distance between the point of closest approach of the track to a reference, a line parallel to the beam spot. The tracks are also measured in both azimuthal and polar angles, ϕ and θ respectively. Lastly, the transverse momentum is measured and labeled p_{T} with an added sign to denote the charge of the particle.

Many charged particles pass through the ID as collisions happen, making the reconstruction of

tracks is non-trivial. As a particle traverses the Pixel or SCT, these detectors generate a 3-D space set of points called *hits*. From a seed cluster of three hits, new hits are added from other layers to form many track candidates. Using the information from the hits such as position and covariances, the next hit can be predicted and associated. These tracks are evaluated by number of hits in the track, the goodness of fit, among other criteria. The evaluation of tracks is quantified by a *score*. The tracks are extended into the TRT and refit with their score reevaluated.

When tracks originate from the same point, this point is called a *vertex*. It is important to differentiate between primary and secondary vertices. A primary vertex is a vertex associated to the collision. Secondary vertices appear from other processes such as heavy flavor decays. Vertex finding algorithms [48] are employed to help identify desired physics processes. More information can be found in [49–51] with some extra content about Run-3 tracking in [52, 53].

Calorimeter Clustering

Outside of the ID is the calorimeter system which stops most particles and measures their energy. These particles are not seen as individual hits but rather as deposits of energy at various localized spots in the detector. For this reason, these deposits are grouped into clusters which correspond to the striking particles. These clusters are used to reconstruct all objects that calorimeters absorb and calculate the missing transverse energy. As these clusters are not necessarily isolated, boundaries may need to be drawn between clusters. Jet clustering algorithms are used jet reconstruction given clusters of eenergy deposits.

The standard jet clustering algorithm is the anti-k_t algorithm with a radius parameter of 0.4 for normal small-R jets [11]. The clustering identifies the smallest distances between entries in the calorimeter, d_{ij}, and between entries and the beam (B), d_{iB}. When the minimum distance is d_{iB}, entry i is labeled a jet and removed from the list of entries. The minimum distances are recalculated and this procedure is repeated for all entries until there are none left. The definition of these distances are as follows:

$$d_{ij} = \min(k_{ti}^{2p}, k_{tj}^{2p}) \frac{\Delta_{ij}^2}{R^2}, \tag{3.1}$$

$$d_{iB} = k_{ti}^{2p}, \tag{3.2}$$

where k_{ti} is the transverse momentum and R is a radius parameter. The Δ_{ij}^2 is the defined as $(y_i - y_j)^2 + (\phi_i - \phi_j)^2$ where y is the rapidity and ϕ is the azimuthal angle. The parameter p is added to manage the importance of energy and geometrical scales. In the case where one sets $p = 1$, one gets the k_t algorithm [54] and if $p = 0$ then one gets the Aachen/Cambridge algorithm [55]. For the anti-k_t algorithm, this parameter is set to -1 turning k_{ti}^{2p} in the equation to $1/k_{ti}^2$.

This algorithm clusters soft particles with hard[3] ones and the radius parameter R tells the algorithm how far to look for neighbors. If there is a hard particle within R and $2R$ of the softer particle, the two jets are distinct but the one with highest energy is conical. If they are both of equal energy, then a straight line is drawn between the jets and neither is conical. In general, this case of equal energy jets is practically impossible. In the case where the hard jet is within the R distance in which case both are considered one cluster centered around the one with highest energy. If the two hard jets are of

[3] Hard and soft is physics jargon for high and low energy in some context.

relatively similar energy, then the cluster shape is adapted and non-conical. The key feature of the algorithm is that soft jets do not change the shape of hard jets. The idea is that this adds resilience with respect to soft radiation and flexibility with hard radiation. More information can be found in [56–58] with extra content on Run-3 performance and upgrades in [59]. After Run-3, the long shutdown will be used to prepare the detector for higher luminosity by using machine learning for real-time processing of calorimeter signals [60].

Electrons

As one can imagine, tracks in the ID and energy deposits in the EM calorimeter do not inherently indicate that the association is an electron. A strict set of criteria is imposed on the electron candidate in order to give some guarantee the identification is correct. This criteria comes in the form of impact parameter selection, likelihood identification, isolation, etc.

First, the electron must have EM calorimeter deposits associated with ID tracks. The minimum transverse momenta is 10 GeV with pseudorapidity ($|\eta| < 2.47$) but not in the crack which is $1.37 < |\eta| < 1.52$. Electron candidates must pass a selection based on impact parameters (or their likelihood):

- $|\text{sig.}(d_0)| < 5$,
- $|\delta z_0 \sin(\theta)| < 0.5\,\text{mm}$.

The term $|\text{sig.}(d_0)| < 5$ denotes the significance of the d_0 parameter. The usage of the polar angle in $\sin(\theta)$ for the longitudinal parameter is done such that tracks with expected larger error in the forward region are rejected. Next, a likelihood identification is made at several working points: tight, medium, and loose plus B-layer (central $e^{\pm}$). This likelihood discriminant is calculated with properties of the event like the shower shape and ratio of deposited energy to tracker momentum [61, 62].

The term *working point* with its various degrees are used in other identification algorithms as well. A working point (sometimes abbreviated as WP) is a reference to a particular performance setting. Each degree defines the performance setting where *tight* is the highest performance and *loose* is the weakest. Therefore, when one is using a "tight working point" it means that the identification performance is deemed the best at some metric; for example, background rejection.

Electron candidates must also be isolated from jets to some extent to be counted as electrons. Track isolation is calculated from the sum of p_{T} of all objects in a ΔR cone that are not the candidate divided by the p_{T} of the electron candidate. The `PromptLeptonImprovedVeto` (PLIV) algorithm improves the existing machine learning working points by adding the isolation of the lepton candidate and the lifetime to veto non-prompt[4] leptons. Calorimetric isolation is defined as the sum of transverse energies deposited in the calorimeters within a ΔR cone. More detail on isolation can be found in [63–65]. Regrettably, a paper on PLIV is not available as of the writing of this document.

Muons

As muons penetrate the entire detector, they do not deposit much energy in the calorimeters and can be seen in the muon spectrometer. Although hard to fake, muon candidates are still required

[4] Non-prompt leptons are real leptons which do not come from the primary interaction.

to pass rigorous criteria. Some of these are similar to those of the electron but working points are understandably different.

To begin, muon candidates must have muon spectrometer tracks match with ones in the ID. The tracks in the MS are reconstructed from hits as the ones in the ID would be. However, the segments are identified in the individual components of the MS via a Hough transform [66]. Then, these tracks are fit and given a χ^2 score. Identification of high-quality muons are selected based on number of hits in the ID and MS stations, track fit properties, and variables related to the measurements in the two detectors. Based on these identification parameters, different working points are created to be used in several analyses: loose, medium, tight, LowPt, and HighPt. Similarly to electrons, muon isolation is dependent on p_{T} (at least 20 GeV), must be in the inner detector region ($|\eta| < 2.5$), and are subject to PLIV working points. Additionally, the impact parameter criteria for muons is $|\mathrm{sig.}(d_0)| < 3$ instead of 5 for electrons; while the criteria on z_0 is the same. More information on muon identification can be found in [67].

Jets

Jets are a plentiful object in the ATLAS detector, specially with the luminosities reached by the LHC [58]. With higher luminosities come a higher rate of pile-up. *Pile-up* is caused by additional pp collisions not related to the desired collision. These come in two flavors: *in-time pile-up* when secondary collisions happen in the same bunch crossing and *out-of-time pile-up* when these collisions happen in previous or subsequent crossings. For that reason, their accurate reconstruction and flavor tagging is necessary for many analyses.

Jets are reconstructed using the anti-k_t algorithm [11] with a R parameter of 0.4. In regards to isolation, jets are rejected if they are within $\Delta R < 0.2$ of an accepted electron and have $p_{\mathrm{T}} > 25$ GeV. Previously, tracking and calorimeter information were separate but these two had different resolution ranges. That is to say, the curvature of low energy particles gave an accurate reading of energy from the ID but not in the calorimeter system. Conversely, high energy particles are easier to see in the calorimeters but harder to measure in the ID. Therefore, an algorithm called *Particle Flow* (PFlow) was employed to combine this information [68–70]. Here, each track is associated to one or more clusters in the calorimeter system. The ratio of energy and momentum is calculated and the energy is removed in the calorimeter system. Remnants are removed if they are consistent with fluctuations.

In addition, a jet vertex tagger (JVT) is used to identify and reject pile-up jets [70]. The JVT is built as a two-dimensional likelihood from two variables related to the momenta of tracks and jets from primary hard-scatter vertices, and the number of pileup vertices. The JVT also has working points as medium and tight with similar options for forward ($2.5 < |\eta| < 4.5$) jets called fJVT. Once reconstructed, jets must go through calibrations and corrections for pile-up, MC corrections, detector effects, energy leakage, etc. with the details in [71].

As the top-quark almost always decays into b-quarks, which are seen as jets, flavor tagging becomes a priority. These jets which contain b-quarks are called b-jets and tagging jets as containing such b quarks is called b-tagging. As b mesons are often long-lived, decay weakly and have large invariant mass, a multivariate discriminant can be generated [72, 73]. The current b-tagging algorithm is the DL1r [74] which uses the above-mentioned information along with the information from a tagger using a recurrent neural network (RNN) [75]. The DL1r algorithm yields several working points which result in specific b-jet efficiency. This efficiency is set in incremental percentages denoting tighter to looser: 60 %, 70 %, 77 %, 85 %. It should be noted the percentages are smaller for higher

b-jet efficiency as higher background is rejected at the cost of real b-jets. For more information of b-tagging for ATLAS can be found in [76, 77].

Taus

Typically, tau leptons which decay leptonically are treated as their lighter brethren. Hadronically decaying tau leptons are another issue entirely, as mentioned in section 2.5. Similar to other objects, reconstruction of tau jets must pass some prerequisites in order to become candidates. First, the p_{T} of the jets must exceed 25 GeV and be in the barrel region ($|\eta| < 2.5$) but not in the crack. As mentioned in section 2.5, the tau lepton decays into odd-numbered charged hadrons (prongs) such that the total charge is ±1. Therefore, the jets which are candidates for tau lepton decay must be associated to one or three tracks in the ID. A five- or higher-order pronged tau appear rarely with a branching ratio of less than 0.2 % [7]. A detailed breakdown of trigger, reconstruction and energy calibration of hadronic tau candidates can be found in [78, 79].

After a candidate is chosen, it is fed into a RNN [30] which is designed to classify hadronic tau leptons. The RNN is trained using several variables from the ID tracks and calorimeter clusters as well as observables derived from these measurements. From the ID tracks, the following variables are used: transverse momentum, impact parameters, angular distance to the hadronic tau axis, and the number of hits in the IBL, Pixel, and SCT layers. From the calorimeter clusters, the transverse energy, angular distance the the hadronic tau axis and the cluster moments [80] are used. A detailed breakdown of the variables and network architecture used can be found in [30, 78, 79]. From the RNN response, several working points (tight, medium, loose, and very loose) are created with varying background rejection efficiencies.

Missing Transverse Energy

As neutrinos freely pass through the detector without interacting, their energy cannot be directly measured. However since particles collide with no transverse momentum, any missing transverse momentum can be attributed to invisible particles. Therefore, the negative vector sum of the transverse momentum of all visible particles is called the *missing transverse energy*, $E_{\mathrm{T}}^{\mathrm{miss}}$ or MET. The $E_{\mathrm{T}}^{\mathrm{miss}}$ of each event is calculated from user-selected high transverse momentum objects, with ID tracks and cell clusters not used for these objects. The unused tracks and clusters are used for a soft-term MET calculation as well.

CHAPTER 4

Data, Monte Carlo Simulation, and Event Selection

To foreshadow what is to come, the analysis in this thesis focuses on machine learning techniques applied in the context of $WWbb$ and tHq analyses. Various techniques, developed in chapter 5, are used to deal with problems these analyses face or aid in their sensitivity for some desired quantity. Before discussing these in detail, the data and simulation used are described in the following sections 4.1 and 4.2, respectively. Additionally, section 4.3 details cuts and region definitions for these analyses.

4.1 Datasets

The data used in this analysis was collected by the ATLAS collaboration during Run-2; spanning the years 2 015–18. The total integrated luminosity of the collected data is of approximately 139 fb^{-1}. As mentioned in section 3.2, not all of the data collected by the detector is useful for physics analysis. The noted integrated luminosity is that of interesting events when the detector was operating optimally. This data is often called *good for physics* and can be seen in figure 3.3(b). One should note the conditions were not identical for every year or even day. For example, pile-up conditions differed between the years, shown with averages in figure 4.1.

4.2 Monte Carlo Simulation

Physicists use models of processes to predict what one can detect in experiments. These models are not blackboard sketches but simulations using Monte Carlo (MC) techniques. From known distributions and parameters, events can be generated and physics processes can be simulated. Such distributions are the parton distribution functions and parameters may be masses of quarks and intermediary bosons. Ideally, these distributions and parameters describe reality well enough for the simulation to be comparable to data. Therefore, the accuracy of MC models is important for an accurate measurement.

To begin, MC simulation is a computational algorithm which samples distributions to obtain a numerical result. In high-energy physics (HEP), MC methods are used to generate simulated events from input distributions, as mentioned earlier. These distributions may differ when models or parameters are changed. In this case, a new set of sampled events can be generated for comparison with a nominal set. However, the process of generating simulated events is non-trivial.

In a nutshell, MC simulation is done in two main steps: event generation and detector simulation.

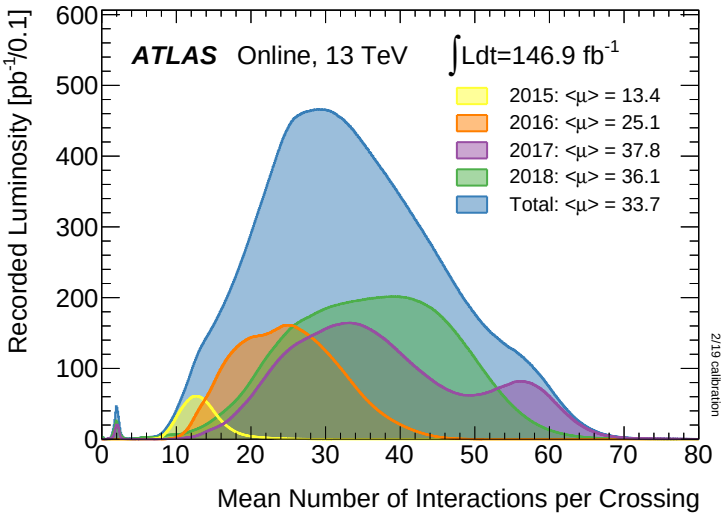


Figure 4.1: Total number of intersections per crossing for Run-2 recorded by ATLAS. The mean value of each year is given in the legend as $\langle\mu\rangle$. The total distribution is also shown in a light blue. [40]

Event generators [81] allow physicists to create events of hard scattering processes in high energy experiments. Such generators additionally parametrize initial and final state radiation (ISR/FSR), develop parton showering and hadronization. ISR and FSR are radiative emissions of particles outside of the primary interaction. For example, in a $q\bar{q} \rightarrow q\bar{q}$ interaction a gluon may be radiated from the initial (anti)quark or the final (anti)quark. Furthermore, the generators must also take pile-up conditions into account and choose decay modes of particles. From here, the generator establishes the trajectory and energy of outgoing particles. Examples of event generators can be found in [82–84].

As the generated particles are computer simulated, these cannot be input into the hardware directly. Therefore, a simulation of the ATLAS detector's material and geometry are used, called GEANT4 [85]. It should be noted that such a detailed simulation costs computation time and resources. For this reason, there exists four levels of simulation: full, library, fast, and parametric. In this document, the library and parametric simulation levels are omitted as they are not used.

Of the two relevant MC simulations, the one made with a full detector simulation is referred to as *full sim* (FS). This MC simulation is used as a comparative to data recorded with the ATLAS detector. Other uses of such samples are estimation of uncertainties introduced by the detector. Examples of which are the jet energy scale (JES), jet energy resolution (JER), lepton identification, calorimeter calibration, among others.

On the other hand, fast simulation samples are generated without the full detector simulation as the comparison ends early in the simulation chain. As the name implies, this makes samples faster to generate and are often called *ATLAS Fast* or `AFII`. The speed increase comes from parametrizing the calorimeter components to duplicate energy profiles with a fine E/η grid [86]. Often, these samples are used to estimate uncertainties introduced by modeling. For example, there exists more than one MC generator and therefore differences between them are considered uncertainties. In general, `AFII` simulates the energy response, average lateral shape, and energy fractions. However, it is unable to simulate particle decays nor leakage into the muon spectrometer.

The main difference between the two levels of simulation is the complexity of the detector simulation.

It should be noted, the simulation of particle transport in calorimeters uses the largest computation time. The complex geometries in the high-granularity calorimeters cause a large number of calculations. These computations are not entirely necessary as only the final deposits are necessary in the final product. Since the output should be analogous to pixel coloring corresponding to calorimeter deposits, a generative machine learning model can be applied. This model would be used for fast simulation where the network can produce the detector response from some incoming particle [87]. As of this document, such a procedure is still under development.

All simulation topics covered so far do not cover the conditions of the LHC, like pile-up. This is usually done by overlaying simulation with other collisions from QCD generators. Most single top-quark and all top-quark pair processes used in this work were generated with Powheg [88] in conjunction with Pythia8 [89] as the pile-up generator [90]. Noting that the generation of the DR and DS schemes for the tW-channel are also created with these generators. Additionally, a $WWbb$ sample named `bb4l` is also generated but in `AFII` instead of full sim. Some Higgs boson samples are also generated with Powheg and Pythia8. However, some rare processes like tHq and tZq are not made with the same combination. tHq samples are made with MadGraph5_aMC@NLO [91] with Pythia8 and tZq MC is created with MadGraph5_aMC@NLO with Herwig7 [92, 93].

For W and Z bosons in association with jets (W/Z + jets), Sherpa2.2.1 [94] is used. MC which simulates two electroweak bosons, called *diboson*, is created with Sherpa2.2.2 [94, 95]. Both Pythia and Sherpa generators are multi-purpose, leading order generators. Pythia generates QCD events, Sherpa is a multi-parton generator that includes hadronization, and Powheg is a next-to-leading order (NLO) generator that creates NLO QCD events. Details on MC generators used in the ATLAS experiment with extensive details on particular processes can be found in [96].

Lastly, the generators for samples of importance are usually compared to other models as a systematic uncertainty. As an example, `AFII` samples of the nominal Powheg and Pythia8 single and double top-quark samples are made. These same processes are generated with a combination of Powheg and Herwig7, or MadGraph5_aMC@NLO and Pythia8. Then when compared to the nominal, any differences are attributed to the use of a different generator. The recommendations of handling modeling systematics can be found in detail in [97].

After events are generated, they are given weights which normalize the total to theoretical cross-sections. In particular, the pile-up reweighing is also done to properly compare the MC distributions to the data gathered under different pile-up conditions. Three MC campaigns were created for this purpose: MC16a matching to the years 2 015–16, MC16d for 2 017, and MC16e for 2 018. The MC that is generated needs to have some expected value as input such as physical quantities like the masses of particles. One of such parameters is the mass of the top-quark which is set to 172.5 GeV/c^2 for nominal samples.

4.3 Signal and Sources of Background

With MC events generated, one can focus on enhancing the purity of some desired process. Often times, this is done via a series of direct cuts on kinematic variables or object multiplicity. For example, a $t\bar{t}$ analysis would see the produced top-quark pair decay into two b-quarks and two W bosons with subsequent decays. These are seen in the detector as two b-jets and some combination of jets or leptons. If the chosen multiplicity of leptons is two, i.e. both W bosons decay leptonically, then the final state particles are two leptons, two b-jets, and some missing transverse energy. Such a selection

would mitigate most W + jets events as a secondary lepton would be from misidentification; an unlikely occurrence.

As this document contains machine learning applications to two analyses, selections are explained in the coming sections. The $WWbb$ selection is simple and the signal is clean enough to warrant little description of background processes. On the other side, a rare process measurement such as tHq with hadronic τ leptons has several background processes contaminating signal regions. The difficulty of hadronic τ identification is exacerbated by unmodeled or poorly modeled backgrounds which mimic hadronically decaying τ leptons.

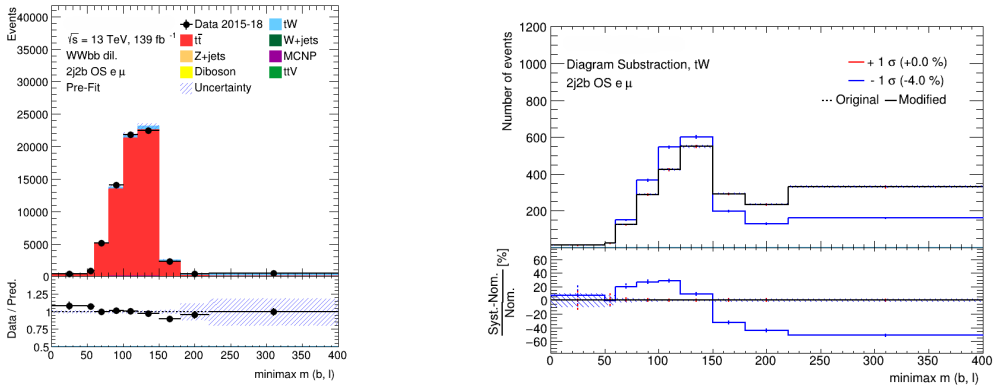
WWbb Interference Enhanced Region

In the $WWbb$ analysis, the goal is to perform a differential cross-section measurement on the $t\bar{t}$ and tW processes together. Additionally, the focus is on kinematic regions where the interference between these two processes is enhanced. As mentioned in section 2.4, these two processes interfere which means they have identical final states shown in figure 2.9. The interference is maximal in a region dominated by $t\bar{t}$ as shown in [21]. For that reason, the focus of the analysis presented here uses cuts that enhance interference and remove backgrounds as efficiently as possible.

To begin, top-quarks decay nearly always into a W boson and a b-quark. Furthermore, the W boson decays hadronically most commonly ((67.60 ± 0.27) %) but with about a third of decays being leptonic ((32.57 ± 0.28) %) [7]. One could consider a single lepton or purely hadronic analysis which would increase the raw number of events expected. Another positive of a single lepton analysis is the ability to reconstruct both W bosons as only one neutrino is emitted. This gives an analyst the ability to reconstruct the top-quarks involved in these processes to further probe their kinematics. However, the inclusion of these jets introduce potential contaminants from other processes like W + jets and fake leptons. Additionally, jets have typically higher uncertainties than leptons. Lastly, the $t\bar{t}$ and tW cross-section is high enough that choosing a decay mode with fewer events does not negatively impact the analysis.

Given the previous arguments, the $WWbb$ analysis uses a dilepton selection. Both leptons must be considered tight in ID and isolation and have a transverse momentum of at least 28 GeV. A third lepton veto is implemented to reject events with with a loose third lepton with $p_{\mathrm{T}} > 15$ GeV is present. The lepton pair must then be of opposite flavor ($e\mu$) to reduce Z + jets background. Lastly, the leptons must be of opposite sign to ensure they come from $t\bar{t}$. Afterwards, the multiplicity of jets is chosen such that a $t\bar{t}$ enriched region is defined. In this way, the number of jets must be two and they must also be b-tagged; this region is called `2j2b`. The nomenclature `XjYb` implies X total jets of which Y must be tagged as b-jets. The jets must have p_{T} of at least 25 GeV and the b-tagging working point is the tightest at 60 %. As mentioned in jet reconstruction section, the jet vertex tagger (JVT) [70] gives a value related to the likelihood of correct jet reconstruction. A minimum JVT value of 0.5 is required for jets which are under 60 GeV in traverse momentum and are central, corresponding to the tight working point. Of course, these cuts are made after the preselection of data and particle identification described in section 3.3.

In analysis with leptons, one typically performs a fake and non-prompt lepton estimate. Fakes are objects which are identified as an electron or muon but is not actually a lepton at all. This effect comes from QCD interactions mimicking the signal of a light lepton. Although rare in occurrence, the cross-section of such processes is orders of magnitude higher than the ones described in this document. Non-prompt leptons are real light leptons from secondary decays like heavy hadrons



(a) $m_{b\ell}^{\text{minimax}}$ stack plot

(b) DR/DS comparison

Figure 4.2: $m_{b\ell}^{\text{minimax}}$ distribution in the $WWbb$ region with highest interference. (a) shows the distribution compared to data with all included MC samples; using DR as the nominal tW sample. The uncertainty shown here is only statistical and from the interference systematic (DS). (b) is the plot comparing the DR and DS schemes to show the effect of interference as a function of this variable. The black line is the DR sample and the blue line is the DS.

decaying leptonically. Typically, one estimates fakes and non-prompt contribution using the Matrix Method or similar data-driven methods [98, 99]. This procedure is not done here as $WWbb$ is particularly clean and lepton identification in ATLAS has improved since the inclusion of algorithms like PLIV. However using MC, leptons which are charge-flipped or are tagged as non-prompt in the sample are treated as non-prompt and included as such instead.

Figure 4.2(a) shows graphically how clean the signal is for this analysis. The event yields are also found in table 4.1 to give numerical detail. Both the table and the plot show the agreement between data and MC with nearly all events estimated to be tW and $t\bar{t}$. Here, the tW DR scheme is used as signal with the DS scheme as a systematic uncertainty. However, the difference between these two can be interpreted as the interference effect. This can be seen in the figure 4.2(b) where the two lines which compare DR and DS grow in separation at higher values. The variable shown is referred to as *minimax m (b, l)*, $m_{b\ell}^{\text{minimax}}$, which is defined as:

$$m_{b\ell}^{\text{minimax}} = \min[\max(m_{b_1\ell_1}, m_{b_2\ell_2}), \max(m_{b_1\ell_2}, m_{b_2\ell_1})] \tag{4.1}$$

This variable is a measure of a top-quark's off-shell-ness and therefore sensitive to the interference effects.

In this exclusive region, tW and $t\bar{t}$ dominate the expected events as other processes contribute less than 1 % to the total. The effect of opposite lepton flavor selection can be seen in table 4.1. Comparatively, one loses about half of the events in total but one decreases the inclusion of Z+ jets a hundred-fold and mitigates one third of MCNP events. In this region, W+ jets is completely mitigated as this process is unable to generate two prompt leptons. Instead, a second lepton must come from a misidentified jet which is unlikely, as discussed previously.

With the selection of $WWbb$ events defined, one could continue onward to the later chapters about machine learning and what physics problems are addressed in this document. As stated earlier, the work contained in this document also pertains to the difficulties faced by the tHq analysis. Although

Process	Total Events					
	2j2b OS OF			2j2b OS		
tW	2 281.2	±	69.8	4 115.7	±	137.8
$t\bar{t}$	65 974.2	±	1 452.3	118 349.0	±	2 604.6
W + jets	<0.1	±	< 0.01	<0.1	±	< 0.01
Z + jets	7.4	±	3.6	693.2	±	27.9
MCNP	403.3	±	11.5	679.5	±	17.7
Diboson	2.9	±	0.4	6.4	±	0.6
ttV	32.2	±	1.0	58.7	±	1.5
Total	68 701.2	±	1 538.6	123 902.5	±	2 790.1
Data 2015-18	68 002			123 225		

Table 4.1: Event yields in the $WWbb$ high interference region after cuts. The labels OS signifies a selection of opposite sign flavors and OF denotes opposite flavor ($e\mu$) selection. The label MCNP is an acronym of *Monte Carlo Non-Prompt*. These are events from all MC samples which contain non-prompt leptons or leptons which have a wrongly assigned charge. The uncertainty noted is only due to statistics and the difference between DR and DS schemes.

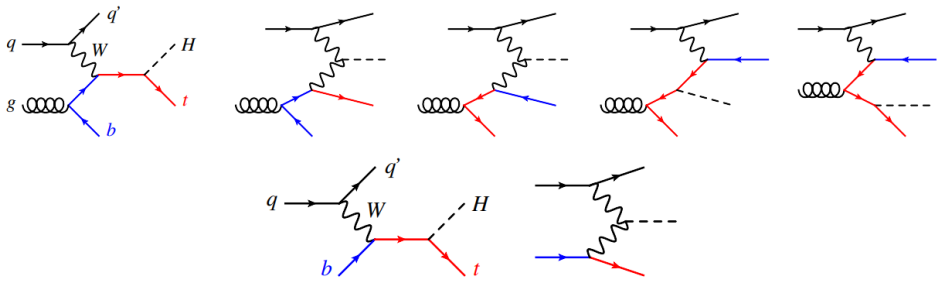


Figure 4.3: Feynman diagrams depicting various tHq production modes [27]. These are only leading order diagrams a t-channel production, producing the expected spectator quark. Note that the top diagrams are in the four flavor scheme and the bottom in the five flavor scheme.

the strategies and problems differ, it is important to establish a selection and facts in a similar manner before continuing onward.

tHq with a Hadronically Decaying τ

The goal of the tHq analysis is to experimentally measure the tHq cross-section and Higgs boson properties. As a rare process, tHq production is difficult to measure given the potential for more common processes to populate any given region. For this reason, the inclusion of tau leptons in the final state is considered. As explained in section 2.5, the tau is the only lepton which may decay hadronically and leptonically. The hadronic decay covers about 65 % of all tau lepton decays and therefore cutting such events reduce the phase space of this process.

Figure 4.3 shows various production modes of the tHq process. This image also shows that the

Higgs boson may couple to either the top-quark or the intermediate W boson. The spectator quark is detected as a forward ($|\eta| > 2.5$) jet while the other particles (t-quark and H) decay into other particles. The spectator b-quark is typically too low energy or too forward for detector acceptance. The top-quark's decay is exactly the same as stated earlier in the $WWbb$ selection.

The Higgs boson instead decays according to figure 2.11 which shows different likely decay modes. Of note, there is a greater likelihood that the Higgs boson decays into a $b\bar{b}$-pair. Additionally, the Higgs boson may still decay into pairs of Z or W bosons, or τ leptons which may decay leptonically or hadronically. As mentioned earlier, leptonic decay modes rather than hadronic allow for cleaner signal regions. For this reason, a typical analysis would choose a tri-lepton final state where the Higgs boson's and top-quark's decay must be leptonic. Given the rarity of this process, hadronic decay modes are also considered.

Although a light lepton only measurement is possible, there is something to gain from including tau lepton production in the selection. For one, the likelihood of a tau lepton produced from the Higgs boson is orders of magnitudes higher than that of other leptons. Should the Higgs boson decay into a pair of W bosons, these are more likely ((67.60 ± 0.27) %) to be hadronic decays rather than leptonic. In this case, the contribution from $H \to WW \to \ell\nu_\ell\ell\nu_\ell$ decays is smaller than that of $H \to \tau\tau\nu_\tau\nu_\tau$. Even if it is a low contribution, for such rare processes it is significant for it to be accounted.

As mentioned earlier, measuring the tHq cross-section with various final states are considered. These are:

- two same-charge light leptons with no taus (2ℓ SS),
- two light leptons with one hadronically decaying tau lepton ($2\ell + 1\tau_{\text{had}}$),
- one light lepton with two hadronically decaying tau leptons ($1\ell + 2\tau_{\text{had}}$),
- and three light leptons (3ℓ).

This document focuses on the problems faced by the $2\ell + 1\tau_{\text{had}}$ channel.

The final state of importance in this document requires a signature of one hadronically decaying tau lepton, two light leptons, one b-jet, and a spectator jet. These objects are subject to the same criteria underlined in section 3.3, similar to $WWbb$. It should be noted that when one hadronically decaying tau (τ_{had}) is required, the influence of events where the Higgs boson decays into two W bosons grows. That is because the tau lepton may come from the top-quark decay and the light leptons from the Higgs boson's daughter W bosons. In this case, one can check the charge of the light leptons. When these are oppositely charged, they are likely to come from the W bosons which come from the Higgs boson decay. On the other hand, if these are of same charge then assigning which lepton comes from Higgs boson decay is a non-trivial matter. However, this assignment is not pertinent to the work in this thesis and therefore it is ignored. Either way, it should be noted that the two light leptons' charge (opposite or same) are subject to different backgrounds. Therefore, one can split the $2\ell + 1\tau_{\text{had}}$ depending of this; as a convention, SS for same sign and OS for opposite sign. For the studies discussed in this document, this distinction is made if a further selection on light lepton charge is made. Lastly, a secondary and analogous region was used where there was only one light lepton and one τ_{had}, named $1\ell + 1\tau_{\text{had}}$. This region is used to explore strategies as it has greater statistics than the other channels discussed before.

Although similar reconstruction is done on the detectable objects, some selection criteria differ. For this reason, a general overview on cuts and selections is given. To begin, light leptons must have a transverse momentum greater than 20 GeV and are subject to the same reconstruction criteria

previously outlined. However, the leading light lepton must have at least $p_T > 27$ GeV. The upper limit on jet pseudorapidity imposed by the $WWbb$ selection is extended to $|\eta| < 4.5$ in this analysis to include the spectator quark. These jets must have at least 20 GeV in transverse momentum as well and be subject to the tight JVT working point. Differing from $WWbb$, forward jets ($4.5 > |\eta| > 2.5$) have an additional forward JVT (fJVT) cut of fJVT < 0.4 when their transverse momentum is less than 120 GeV. An inclusive jet multiplicity is made with the number of jets being minimum two and maximum six. Additionally, the b-tagging is done the same as $WWbb$ but here one or two b-jets are allowed. Tau leptons are required to be central but not in the crack, have a $p_T > 20$ GeV, and have one or three associated tracks. The RNN requirement for hadronically decaying tau leptons is the loose working point. Lastly, the missing transverse energy must be in a range of 5 GeV $< E_T^{miss} < 800$ GeV.

Unlike $WWbb$, a tHq of with this selection is not as clean. There are several backgrounds which must be accounted for and which contribute the majority of events. Most of the contamination comes from misidentification of τ_{had} in the event. For example, should additional jets from $t\bar{t}$ be considered a τ_{had}, then this event would be included. As $t\bar{t}$ has a cross-section orders of magnitude greater than tHq, this process is significant. Similarly, Z + jets events may qualify in the same way as $t\bar{t}$ events or the Z may decay as two τ leptons ((3.3658 ± 0.0023) % [7]).

Process	OS+SS	OS	SS
tHq	2.08 ± 0.04	1.259 ± 0.032	0.823 ± 0.026
tZq	41.3 ± 0.5	34.8 ± 0.5	6.6 ± 0.5
$t\bar{t}$	4 709 ± 15	4 631 ± 15	78 ± 5
tW	226 ± 6	223 ± 6	3.6 ± 2.1
W + jets	5.1 ± 1.2	2.0 ± 0.7	3.1 ± 2.8
Z + jets	2 850 ± 70	2 840 ± 70	8 ± 9
Diboson	147.6 ± 2.0	137.8 ± 2.0	9.9 ± 1.2
$t\bar{t}$ + W	55.4 ± 0.7	37.5 ± 0.6	17.8 ± 1.1
$t\bar{t}$ + Z	113.8 ± 1.0	97.1 ± 0.9	16.7 ± 1.0
$t\bar{t}$ + H	52.27 ± 0.23	36.49 ± 0.18	15.8 ± 0.4
tWZ/tWH	20.85 ± 0.13	17.56 ± 0.11	3.29 ± 0.19
Other	10.6 ± 1.8	8.9 ± 1.8	1.7 ± 1.1
Total background	8 230 ± 70	8 070 ± 70	164 ± 4

Table 4.2: Event yields of the $2\ell + 1\tau_{had}$ channel. [100] Additionally, these are split by the light lepton charge combination of same sign (SS) and opposite sign (OS).

Table 4.2 shows the event yields expected in the $2\ell + 1\tau_{had}$ channel. From here it can be seen that tHq is a small fraction, less than one percent, of the total expected events. Dominant processes are $t\bar{t}$ and Z + jets as suggested earlier. Additionally, rare $t\bar{t} + X$ processes are the next greater background following similar logic as above or with heavy bosons decaying as tau leptons. As $t\bar{t}$ and tW are two sides of the same coin, it makes sense that tW also is a background for such an analysis. tWZ and tWH have a three lepton final state if either the associated W boson or the one from the top-quark decay leptonic and the other hadronically. Diboson events, such as ZW, can pass a tri-lepton selection where one may be a tau lepton. tZq is the most similar to tHq with a radiated neutral boson which may decay leptonically. Lastly, W + jets is unlikely to be in this selection yet it still contributes marginally.

In another analysis, it would be an ignorable background yet here it contributes about twice that of signal events.

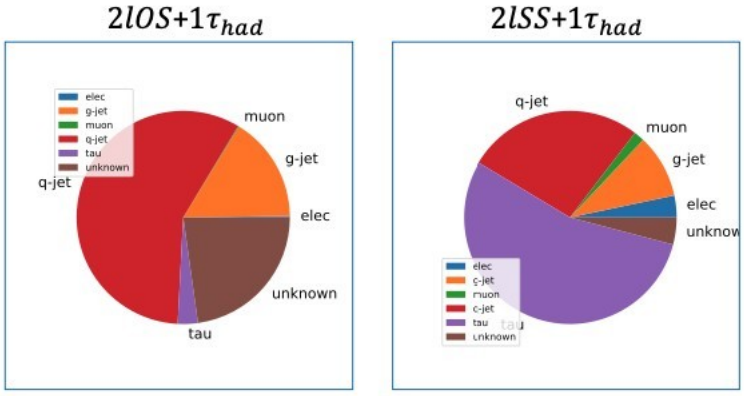


Figure 4.4: Estimated composition of tau candidates in the $2\ell + 1\tau_{\text{had}}$ region. [100] Tau candidates are broken down into their sources where *tau* implies true tau leptons without associated truth jet(s). Electron and muon are light leptons without truth jet match but reconstructed as a τ_{had}. Quark and gluon imply a respectively initiated jet reconstructed as a tau lepton but with truth ID of such objects. Unknown is designated to other objects which have no truth object associated.

A significant portion of the events which pollute this primary selection come from τ_{had} misidentification. This can be seen in figure 4.4 where tau candidates are split by their sources. These sources are gathered from MC truth information and used in the Template Fit method. In brief, MC is split, using truth information, according to the type of object which mimics a lepton of interest. This estimate of background is equivalent to a scaled background template to match data rates in background enriched regions. These templates are histograms of discriminating variables used to derive scale factors which can be used to estimate fake τ_{had}. This thesis contains an alternative estimation method using advanced machine learning techniques.

CHAPTER 5

Machine Learning

5.1 Introduction

In particle physics, data analysis comes with a set of challenges with increasing complexity proportional to ambition. For example, if the desired signal is orders of magnitude smaller than background processes; or if the identification of final state particles influences the results and systematic uncertainties. These challenges become impossible for a human to overcome as the size of data increases. As humans become unable to classify "image-by-image" over hundreds of thousands of events, Machine Learning (ML) algorithms are employed.

Machine Learning is a branch of computer science that aims to mimic humans' decision making with algorithms. In general, this is done by encoding input information as numbers or arrays and passing it through a series of functions with floating parameters. With some desired outcome as the task of the machine, it modifies its parameters to connect the input to its desired output. In this way, it can perform some decision based on the parameters it has chosen, given some input. An example of a decision making algorithm is an Artificial Neural Network (NN).

5.2 Artificial Neural Networks

Inspired by the brain's architecture, a neural network takes similar form to perform decision tasks. A NN has nodes and connections between them just as a brain has neurons and synapses. The brain takes information from the senses that are then processed as electrical signals between neurons until it comes up with a reaction. Similarly, a NN takes in some input and passes it through its nodes via "synapses" until it yields some decision. The easiest example of an Artificial Neural Network is a feed-forward Neural Network, shown in figure 5.1.

Feed-forward NNs are used most commonly in regression or classification problems. A network tasked with regression is supposed to predict some numerical value(s) from some input data. In physics, a regression network can be used to predict a particle's kinematic properties from those of its decay products. On the other hand, a neural network tasked with classification aims to correctly label its input data as some category decided by the user. Using such a network as the basis, important terms are described and definitions given in the following paragraphs.

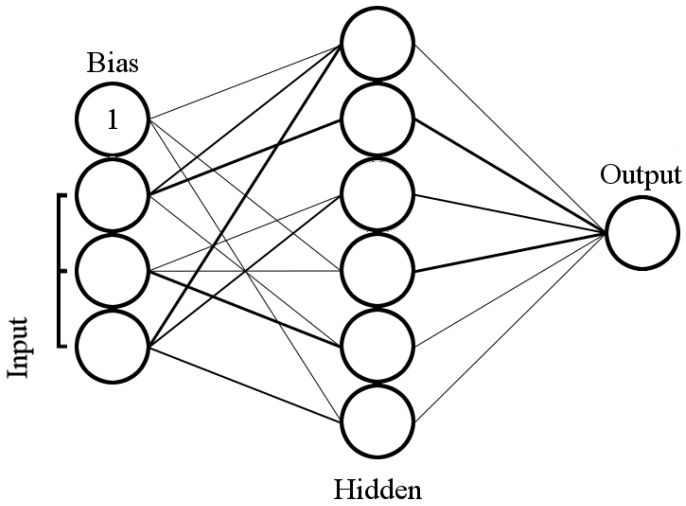


Figure 5.1: Illustration of a NN's architecture. Each circle represents a node and each line is a connecting synapse. The input nodes receive the input features, a per-event vector. The bias node is added to increase the flexibility of the learning and takes no input. Each node is connected by a synapse whose significance is measured by a parameter called weight. In this sketch, the thickness illustrates the size of the weight. The output node gives the event a value that determines the network's decision.

5.2.1 Architecture and Components

This architecture maps some set of features, a per-event vector ($\vec{x} \in \mathbf{X}$), to a target ($y \in \mathbf{Y}$). In the context of particle physics, these features can be the kinematics of final state particles in an event and their target identifies it as signal or background. The input vector, $\vec{x}$, is pushed through each column of nodes, called a *layer* or *hidden layer*. When a network has more than one hidden layer, it is often called a *deep* neural network. The output of each layer is pushed to the following layer until the last layer, the *output layer*, yields a predicted target ($\hat{y}$).

As mentioned before, this type of neural network has a series of nodes and synapses that connect them. Each node represents an *activation function* with a weight (w) parameter that modifies the output of previous nodes. They are called activation functions as they "fire" if the input value is over some critical number. More specifically, these functions yield small values (or zero) for small input values and large values otherwise. Activation functions are necessary as they add non-linearity to the network and allow it to learn more complicated functions. Without such functions, a feed-forward NN can be re-factored as a simple linear operation that would only work for simple mappings. Needless to say, a NN without activation functions would be insufficient for particle physics' complex problems.

An additional node is typically introduced to each layer (except the output layer) called the *bias* node (b). These are added to increase the flexibility of the model to fit the data. More specifically, it allows for the model to learn even when input features are set to zero.

An explicit form of the activation function can be seen in equation 5.1. The output of the ith node in the nth layer, h_i^n, is a function of the connection ($\vec{w}^n$) to previous nodes, their output ($\vec{h}^{n-1}(\vec{x})$), and the bias value of a particular layer (b^n). The activation function, g, is an arbitrary function that adds

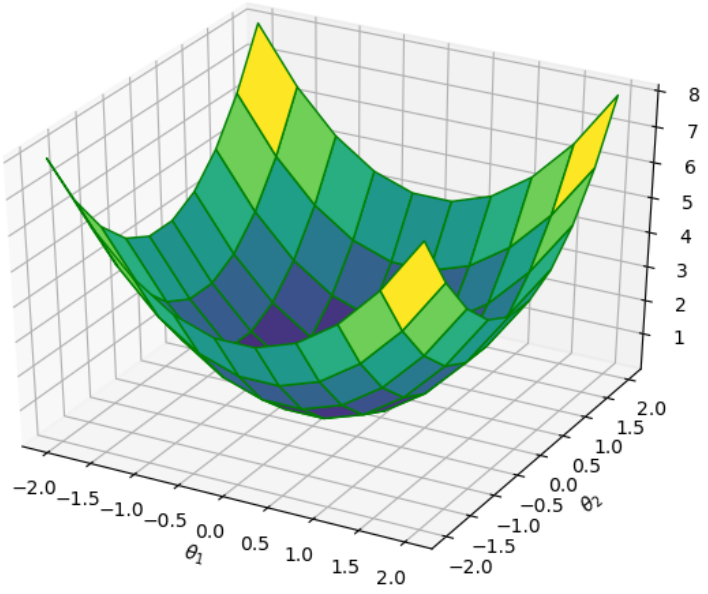


Figure 5.2: Example two-parameter loss function. The example loss curve is illustrated as the surface that has a minimum at $(0, 0)$ and spreads upwards quadratically.

the aforementioned non-linearity to the network.

$$h_i^n(\vec{h}^{n-1}, \vec{w}^n, b^n) = g(\vec{w}^n \cdot \vec{h}^{n-1}(\vec{x}) + b^n). \tag{5.1}$$

By performing this calculation iteratively towards the output node after N layers, one can get an estimator of the desired targets, $\hat{y}$:

$$\hat{y} = \vec{w}^N \cdot \vec{h}^{N-1} + \vec{b}^N. \tag{5.2}$$

Now that the network has parameters, it needs to know how to change them so that it creates the correct mapping $\mathbf{X} \rightarrow \mathbf{Y}$. This can be achieved by adding a loss function and an optimization procedure. A *loss function* is a user-defined function that compares the output of the network with the desired outcome. This means the loss function measures how similar the estimated output, $\hat{y}$, and desired output, y, are. Loss functions are commonly chosen such that similar output and target yield values close to zero. The optimization procedure then modifies the network's parameters with the aim of minimizing the loss function. Additionally, the non-linearity of a feed-forward neural network causes most complex loss functions to be non-convex [101]. Because of the loss function's shape, these types of networks are trained using a gradient-based optimizer.

An optimizing algorithm should seek to bring the loss function to a minimum as close to zero as possible. Since we know that loss functions depends on the network parameters, one could – with some difficulty – imagine that this spans some space. As a simple illustration, let's consider a two-parameter loss function that looks as figure 5.2. Consider that the initial parameters in our network are not ideal, in this case we find ourselves somewhere on the incline. In this case, our optimizer can take the negative gradient of the loss, with respect to the parameters, and get the direction to lower loss

values. From here the optimizer can change the parameters some distance, called a *learning rate*, in this direction. This procedure is repeated such that we follow the gradient downhill until we reach a minimum. Explicitly, the parameter updating is done with the following equation:

$$w_i^n(t) = w_i^n(t-1) - \eta \frac{\partial \mathcal{L}}{\partial w_i^n}(t-1), \tag{5.3}$$

where the parameter being updated in question is the weight, w, from the ith node in layer n at time step t. η represents the learning rate and it is multiplied by the partial derivative of the loss with respect to the node.

It should be noted that doing a gradient descent alone in such a manner may become inefficient or non-convergent for "ravines" or saddle points. For this reason, one can modify the weight optimization equation by adding a *momentum* component. The momentum adds a value proportional to the previous change in the parameter. This, over the course of a few updates, becomes additive in the direction towards the minimum; even when in a ravine. Explicitly:

$$w_i^n(t) = w_i^n(t-1) - \eta \frac{\partial \mathcal{L}}{\partial w_i^n}(t-1) - \mu v(t-1), \tag{5.4}$$

where the momentum factor μ is a number from 0 to 1 and $v(t)$ is the velocity at time-step t. A comparison between optimization algorithms is shown in figure 5.3 in a simple example. One can see that after 25 updates, the algorithm with momentum is closer to the lowest point in the surface.

Updates with this setting are the same as a vector sum of the gradient and momentum direction. The same way one would calculate the trajectory of a ball rolling down a ravine. A variation of momentum was introduced called Nesterov momentum or Nesterov Accelerated Gradient (NAG) [102]. In brief, the gradient component ($\frac{\partial \mathcal{L}}{\partial w_i^n}$) is done with respect to the parameter and the velocity rather than just the parameter. Sometimes, this is referred to as a "lookahead" gradient. Using NAG allows a faster change in velocity leading to faster and more stable convergence when compared to classical momentum. The effect of NAG is most obvious when the momentum parameter is high. In such a case, without NAG the model could be stuck "circling the drain" rather than falling into the minimum. For more information on various gradient descent optimization algorithms, refer to [103].

With a large enough network, one can see that this space becomes harder to imagine as its dimensions grow. As mentioned in the optimization step, one needs to calculate the gradient of the loss function with respect to the parameters. In neural networks, the gradient with respect to each parameter is a lengthy calculation where parameters influence each other by their connections. One can propagate the gradient backwards through the network from the output layer to the first. Using the chain rule of calculus, one can derive any particular weight in the neural network. This weight update method is called *backpropagation*.

One can see how this is performed explicitly in equation 5.5:

$$\frac{\partial \mathcal{L}}{\partial w_i^n} = \frac{\partial \mathcal{L}}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \vec{w}^N} \frac{\partial \vec{w}^N}{\partial \vec{w}^{N-1}} \cdots \frac{\partial \vec{w}^{n+1}}{\partial w_i^n}, \tag{5.5}$$

where N denotes the total hidden layers, n is a particular layer, i is the index of the node in that layer. The desired weight's gradient is denoted as w and all the weights connected to it are indexed but set to vectors, $\vec{w}$, for simplicity. It is obviously inefficient if one wants to calculate gradients of various

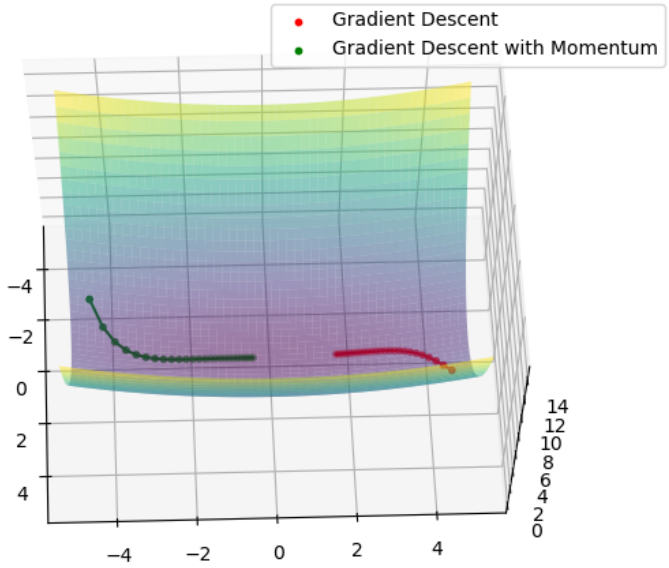


Figure 5.3: Example comparison of an optimizer with (green) and without (red) momentum. The loss curve is illustrated as the surface that has a minimum at $(0, 0)$ and is symmetric in both axes. It spans quadratically in both directions but with a coefficient along one axis that makes it slope "slower", causing a ravine. Both optimization algorithms were given the same learning rate and 25 time steps to reach $(0, 0)$. The initial points for both optimizers are identical but in different quadrants for visibility in the plot.

weights as one depends on the weights from the later layers. One can do each calculation once by evaluating each gradient layer-per-layer in a backwards manner. This means that one calculates the gradient for all nodes in layer N, then uses those to calculate the gradients in layer $N - 1$ and so on.

5.2.2 Data Preparation

It should be noted that values passed from the input layer to the first layer may be problematic if they are too different from each other. After all, the ranges of physical variables like transverse momentum and their angular direction differ in order of magnitude. If one gives a network this raw data, then the values passed between layers become biased towards the largest values. This means that variables like momentum "overpower" the nodes containing azimuthal and polar angle information. To overcome this, one must prepare data in such a way that a network can digest it easily. The procedure where one prepares data for the network is called *preprocessing*.

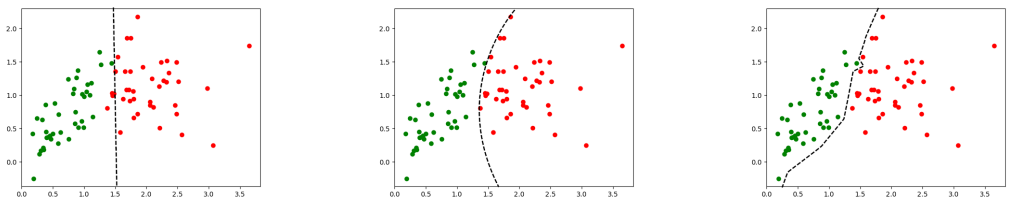
The simplest preprocessing one can do is to scale all distributions between 0 and 1. This *scaling* preserves the shapes of all distributions without changing any information embedded in the original data. A benefit of retaining shapes is the inclusion of small standard deviations of features and preserving zeroes in sparse data. However, this type of scaler does not reduce the importance of outliers in the data. Furthermore, some other aspects of machine learning may assume, or require, that the input data is Gaussian-like. In such a case, a scaler that retains shapes is insufficient and for that reason one employs a *standard scaler*. Such a scaler standardizes data, defined as removing the mean

and scaling the variance of each variable in the input data.

Another step one should take when training a machine learning algorithm is to split their data such that they can test their model. The aim of a such a network should not be to know exactly what its input data is, but rather draw general conclusions for points it has not seen before. Therefore, one can divide their data into a training and test sample. The training sample is used in the network during the training step to update weights and get the correct mapping of $\vec{x} \in X \rightarrow y \in Y$. At the same time, one uses the testing data to observe if the labels and other metrics, further discussed in section 5.2.3, are consistent between samples. To expand on this concept, one can make one more subdivision and create a validation set. When one constructs a neural network, its *hyperparameters* such as number of nodes and layers, activation functions, etc. may be changed to improve the model. The validation set is usable to compare differently constructed models' performance and make an educated decision on which to use.

5.2.3 Over- and Underfitting

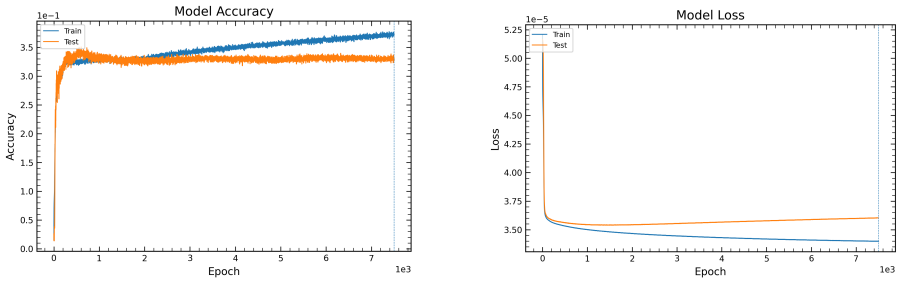
A neural network that fails to classify its data to the desired labels is often said to *underfit* the data. This means that the network could still improve if given more time or hyperparameters are tweaked to help it learn. On the other hand, an *overfitted* network is one that has learned the training set but fails to generalize to new data. An illustration of over- and underfitting models can be seen in figure 5.4 and a well-fitted model is given for comparison. It can be seen that an underfitted network classifies events loosely and makes broad mistakes. This basic vertical line succeeds with some classification but fails to account for some of the top-most points. In this case, the fitted function is too general and fails to provide any discriminating power. The overfitted model, instead, has opted to fit around dots in an "un-physical" way. Such a model also yields unreliable labels as it treats outliers or unlikely events as likely; thus, cutting into categories in a way it should not. The model in sub-figure 5.4(b) shows what a reasonable fit should be. The fit, in this case, shows a general but accurate classification of given categories.



(a) Underfitted model (b) Well-fitted model (c) Overfitted model

Figure 5.4: Examples of over- and underfitting and comparison to a well-fitted model. All three figures show the same randomly generated data as the colored dots. The images exemplify a classification task for a machine learning model where points are scattered by some unknown distribution. Individual figures show a dashed line that represents the model's fit.

A two-dimensional example is simple to visualize but, in practice, the dimensions may increase depending on the problem. One should use metrics of a model's performance that are independent of the dimensionality of the problem. Simple metrics like the accuracy of the labeling can be used to see how good the fit is. The accuracy metric can be defined as the ratio of correct predictions to the total



(a) Accuracy metric

(b) Loss value

Figure 5.5: Example metrics from a network in training. Both accuracy and loss are evaluated at each epoch of training for both training and test samples. The metrics show a model that improves over the first few epochs and then begins to overfit. This is seen as the loss and accuracy lines begin to diverge after 100 epochs.

predictions, or

$$\text{acc.} = \frac{N_{\text{correct}}}{N_{\text{total}}} = \frac{N_{\text{TP}} + N_{\text{TN}}}{N_{\text{TP}} + N_{\text{TN}} + N_{\text{FP}} + N_{\text{FN}}}, \tag{5.6}$$

where the subscript TP and TN mean true positive and true negative predictions, respectively. Similarly, the subscripts with F denote false positives and negatives.

As discussed in section 5.2.2, one divides data into training and test samples. This metric can be evaluated in both samples as a way to detect overfitting. An ideal model would show similar high accuracy for both samples whereas an overfitted model would show a training accuracy much higher than that of the test sample. This metric is typically evaluated at each complete pass-through of the data during training, called an *epoch*. Over the training time of the neural network, one can plot the value of accuracy per epoch and see the improvement over time. An example accuracy plot can be seen in figure 5.5(a).

As the network evaluates the loss for its weight update procedure, one can plot it as another diagnostic metric. This is shown in figure 5.5(b). Similar to the accuracy metric, the desired trend is that both training and test lines remain similar and improving. In the case of accuracy, this means increasing in value and decreasing for loss. Figure 5.5 shows rapid improvement up to about 100 epochs. After this point, both lines diverge and show evidence of overfitting.

As an important caveat, using only one or two metrics to evaluate a model is not enough. Evaluating the performance solely from the accuracy can lead the user to make the wrong decision with regards to their model. For this reason, other metrics are used in evaluating the performance of networks. One commonly used metric is called the *receiver operating characteristic* (ROC) curve and its area. The ROC curve is the plot of true positive rate (TPR) versus false positive rate (FPR). Recall that the true and false positive rates are defined as

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{5.7}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}. \tag{5.8}$$

with the same definitions as the accuracy definition in equation 5.6.

One can evaluate these rates for both training and test samples to get another measure of overfitting.

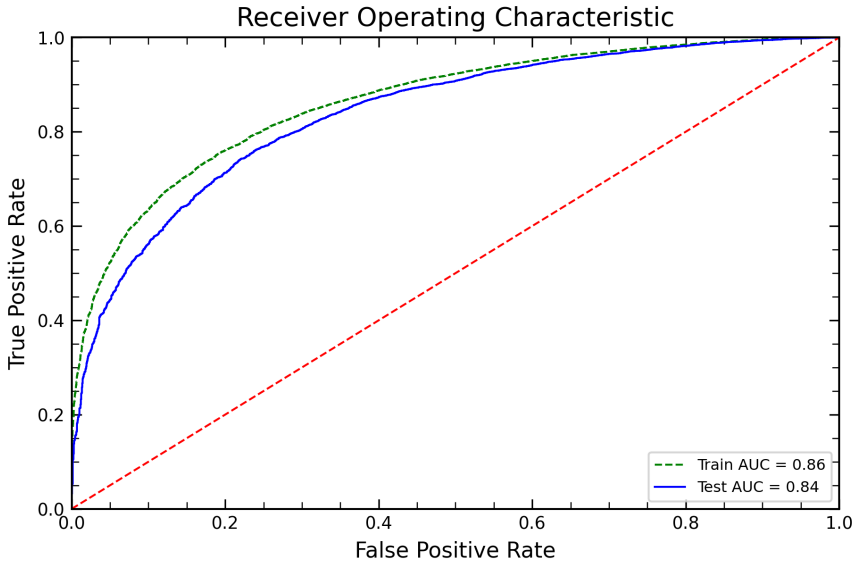


Figure 5.6: Example receiver operating characteristic curve with area under the curve values. True and false positive rates are evaluated for both training and test samples. Here, both are plotted for comparison and the curve's integral is evaluated to get the AUC value. A red, dashed line is drawn to denote what a random model would yield, for comparison.

An example ROC curve is shown in figure 5.6 with both samples' rates and an *Area Under the Curve* (AUC) value listed. A perfect classifier would have a maximal true positive rate and minimal false positives. This would draw a line that is, essentially, a square that covers the entire plot and its integral, the AUC, would be 1. One can see that if the ROC curve laid along the red line, the true and false positive rates would be identical. Meaning that the network is randomly classifying events. This random classification can be seen to yield an AUC value of 0.5.

To give the AUC some context, assume that all signal events are labeled as 1 and background events are 0. The AUC represents the probability that a signal-labeled event is given a larger estimated label, $\hat{y}$, than a background-labeled event. That is to say, if we placed all the events along a line between zero and one, a large value of AUC would mean that most of the signal events are towards 1. Note that this value does not represent the spread of signal and background labeled events but just if they are correctly ranked. This means that the AUC is scale invariant, which may be a desireable quality for some tasks.

5.2.4 Regularization

After discussing measurements of overfitting, it is worthwhile to understand potential sources of the behavior and techniques to overcome it. Earlier in section 5.2.3, it was vaguely mentioned that training for too long can lead to overfitting. This can be because the network begins to model the noise in the data as well. Another reason may be that the complexity of the network exceeds the complexity of the problem. On the other hand, the data itself may cause overtraining if there are insufficient events. In this case, the lack of representation of underlying distributions can make a network fail at generalizing. To avoid overfitting, one can employ *regularization* techniques in the training of a neural network.

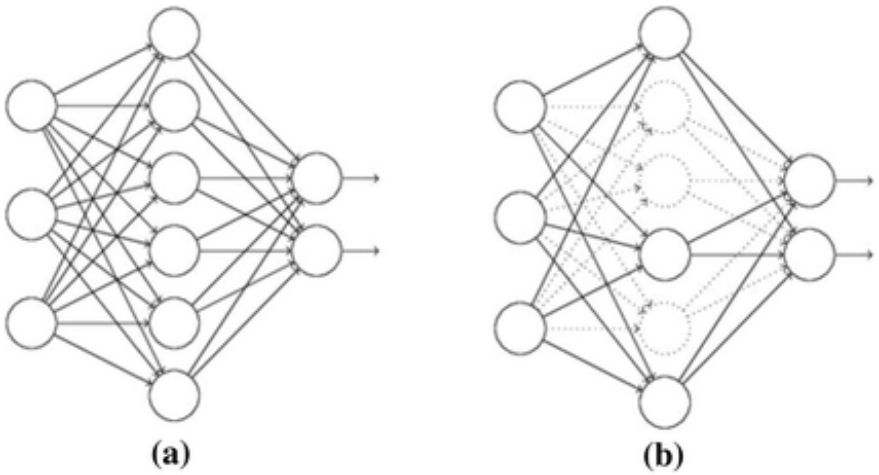


Figure 5.7: Schematic of a neural network with and without dropout nodes. The left figure shows a normal neural network with two output nodes. On the right hand side, three nodes in the hidden layer have been randomly chosen to be dropped out. [104]

A commonly used technique is *dropout*. Aptly named as it describes the procedure of randomly "dropping-out" connections between nodes from training each epoch. During each training step, each node has a user-defined probability of disconnecting for that step or remain on. When these nodes "drop out" of training, their weights are frozen for that iteration and their outputs are set to zero. A schematic of a neural network with dropout nodes can be seen in figure 5.7.

At first glance, such a procedure seems to be inefficient or ineffective. However, one potential reason for overfitting is that the network's nodes develop a co-dependence. This phenomenon occurs when the network's nodes are highly correlated. Such behavior can be seen in figure 5.8. This image was generated by using the tensorflow playground [105], an incredible learning tool that allows the user to visualize a NN as it trains. This example shows a clearly overtrained network that has made highly codependent nodes. Although this image has a lot of information, the focus should be on the nodes in the hidden layer and their relationships to each other. One can see that some nodes are blank and therefore unable to contribute, while others are almost repeating the pattern from the previous layer. Dropout may not be the solution to this example network, but it is useful as a way to visualize correlated nodes and their effect. Lastly, forcing nodes to become less reliant on each other allows the network to generalize better. This comes at a cost in terms of time as the network will likely take longer to converge.

In figure 5.8, it can be seen that nodes that are correlated are connected by thick lines, translating to large weights. Another way to combat weights becoming too large and causing nodes to become overly reliant on each other is to add penalty terms to the loss function. Depending on how the penalty term is added, this can be called a *L1* or *L2* regularization term. L2 regularization adds a scaled, squared norm of the weights to the loss function shown in equation 5.9.

$$\mathcal{L}_{\mathrm{L2}} = \mathcal{L}(y, \hat{y}) - \lambda \sum_{i=1}^{N_{\mathrm{batch}}} |\theta_i|^2, \tag{5.9}$$

where λ is a regularization parameter, or the scale by which this penalty term is added; and θ are

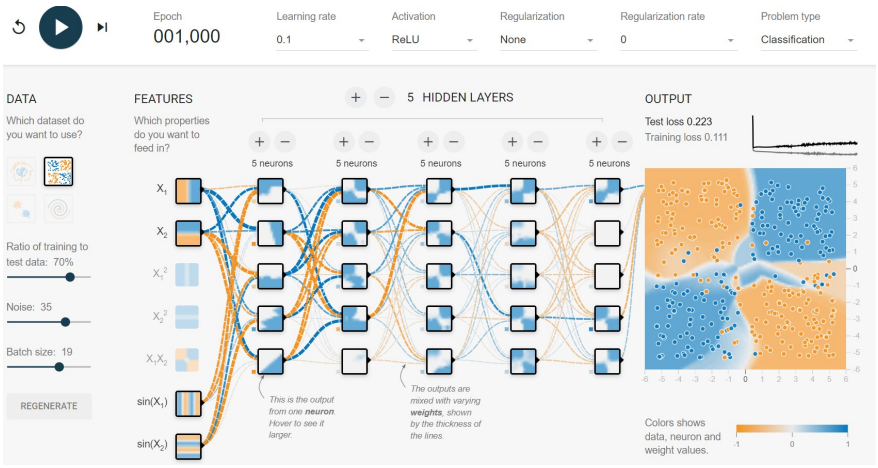


Figure 5.8: Image showing a network with highly correlated nodes and their effect on overfitting. Using the online tool, tensorflow playground [105], one can visualize a neural network as it trains. With the chosen hyperparameters, one can see some nodes in the middle layers become reliant on only a few nodes in the previous layer. This can be seen by the thickness of the lines, a visualization of weight magnitude, as well as the image inside the node being similar.

the parameters of the network, like the weights. It can be seen that this punishes the network for having large weights. Since the value of the weights is squared, the penalty term allows for small, non-zero weights. This allows for less significant features to still contribute and not be pruned out. The downside of L2 regularization is the inclusion of outliers in the data. This is because the squared terms blow up the loss function and the regularization will attempt to fix this by punishing the weights.

Similarly, L1 regularization adds a penalty term that is scaled but not squared shown in equation 5.10 with the same definitions as above.

$$\mathcal{L}_{\text{L1}} = \mathcal{L}(y, \hat{y}) - \lambda \sum_{i=1}^{N_{\text{batch}}} |\theta_i|. \tag{5.10}$$

The reason one might prefer L1 to L2 is for computation efficiency and if one has many features. When the weight is not squared, the small, non-zero weights are further pushed towards zero. If a feature contributes little to the loss function or has a small weight, it is pruned out. In a similar fashion when features or nodes are correlated, setting them to zero does not affect the classification component of the loss function but reduces the regularization term. In this way, the network complexity is reduced and overfitting is avoided.

5.2.5 Batch Sizes and Hardware

The task of training a network involves a considerable amount of number crunching done by computers. Should the network update parameters after every sample, the number of calculations would skyrocket. In order to optimize the training, a hyperparameter is created which controls how many samples are used before the model updates, called *batch size*. The batch size impacts not only computational performance but also aid in the accuracy of the model.

The network's batch size must be optimized to prevent the batches from being either too large or too

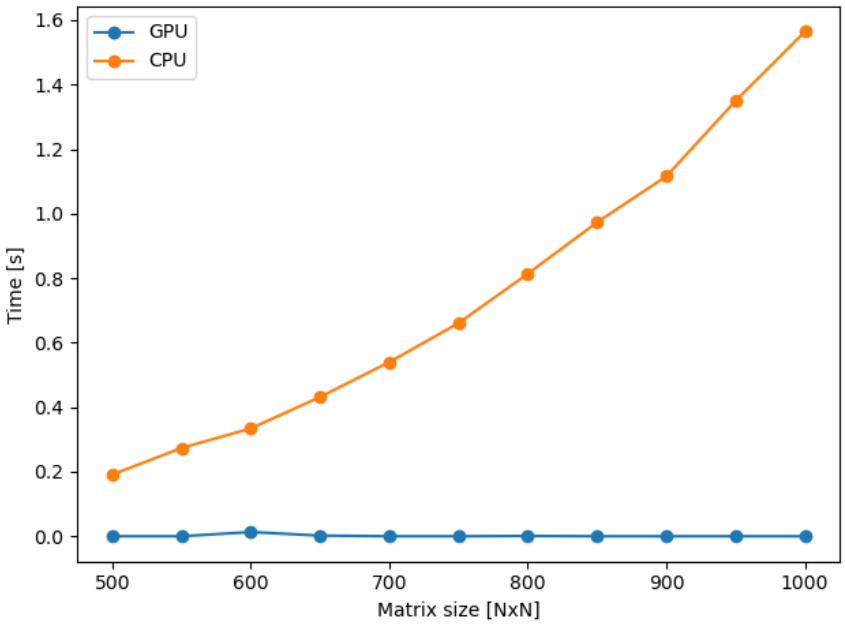


Figure 5.9: Matrix multiplication comparison between CPU and GPU. The CPU is an AMD Ryzen 7 1700X [106], an eight-core processor able to perform billions of simple commands per seconds. The GPU used is an Nvidia GeForce RTX 2070 SUPER [107], which is able to perform trillions of such operations.

small. Much like any other hyperparameter, there is no "one-size-fit-all" number to be noted down. For example, the batch size must be not too large such that the network is stuck in a local minima or overfit the data. A note: the number of samples in a batch should not be a large fraction of the total set, otherwise overfitting may occur. On the other hand, small batch sizes are less efficient in computational performance. In a small batch size, each individual sample has more power to influence network parameter updates. Therefore, a small batch may yield noisy gradients and prevent convergence of the network. This becomes a balancing act where the batch size allows for generalization and optimization.

In high energy particle physics, simulations may have several million events for any given process. To train on such quantities, large batch sizes may be appealing for the sake of performance. However, a central processing unit (CPU) can only perform so many operations at any given time. Even while threading is an option, the speed of training eventually plateaus regardless of batch size increases. For this reason, graphical processing units (GPU) are used in training. GPUs are optimized to do several simple operations simultaneously and are therefore ideal for large matrix multiplication.

For a comparison, figure 5.9 shows the performance of matrix multiplication when using either processor. This plot shows that matrix multiplication, essentially how neural networks function, are performed faster in GPUs. To be more precise, it shows the dimensions of matrices as a function of time spent in calculation. Effectively, the dimension of the matrix to be processed is related to the batch sizes in training and prediction.

5.3 Supervision

In the previous section, the network was tasked to map a set of inputs, $\vec{x}$, to a target y. If one has an accurate set of labels to target, then this is called *supervised learning*. This is because the user knows exactly what is the correct output for each input. However, there are cases where the labels are noisy, unreliable or non-existent. For these cases, one needs to employ *weak supervision learning* or *unsupervised learning* techniques. More accurately, one would employ weak supervision learning techniques in cases where labels are noisy or unreliable. Cases where labels are non-existent instead require unsupervised learning techniques.

5.4 Weak Supervision

In a fully supervised setting, nothing changes from the description before. Once labels have noise or are unreliable, one may need to modify their training procedure. A weakly supervised classifier that is trained on unreliable data is limited to the performance of a fully supervised classifier. Therefore, the task of a weakly supervised classifier is to perform as closely as possible to the supervised version. Although weakly supervised models are often not as accurate as fully supervised models, they are able to tackle tasks with real data.

5.4.1 Learning from Label Proportions

In the case where clear labels are unavailable but relative fractions of purity are known, one can use a technique called *Learning from Label Proportions* (LLP) [108]. As an example, consider two mixed samples of data M_1 and M_2 with some fraction of signal to background f_1 and f_2, respectively. These samples are constructed such that

$$\begin{aligned} M_1 &= f_1\mathcal{S} + (1 - f_1)\mathcal{B}, \text{ and} \\ M_2 &= f_2\mathcal{S} + (1 - f_2)\mathcal{B}, \end{aligned} \tag{5.11}$$

where the factions satisfy $0 \leq f_2 < f_1 \leq 1$ with $\mathcal{S}$ and $\mathcal{B}$ denote the signal and background samples, respectively. Giving such samples to the network, we move the classification task from signal and background to mixed samples M_1 and M_2. Since the fractions are known, one can include them in the training procedure.

One example loss function that encodes this information is given by

$$\mathcal{L}_{\text{LLP}} = \Big|\sum_{i=1}^{N_{M_1}}\Big(\frac{\hat{y}_i}{N_{M_1}}\Big) - f_1\Big| + \Big|\sum_{j=1}^{N_{M_2}}\Big(\frac{\hat{y}_j}{N_{M_2}}\Big) - f_2\Big|, \tag{5.12}$$

where $\hat{y}_i$ is the estimated label of event i and the subscripts i and j span only the mixed samples M_1 and M_2, respectively. Lastly, N_{M_1} and N_{M_2} are the number of events in the batches belonging to the mixed samples M_1 and M_2, respectively. One advantage of this approach is that it can be expanded to include more than two classes, as long as fractions are known. However, it is more likely that one does not know the fractions of purity of the classes.

5.4.2 Classification Without Labels

In the case where label fractions are unknown, one can use a technique called *Classification Without Labels* (CWoLa) [109]. This technique does not modify the loss or change the classification procedure. One simply trains the network to classify the mixed samples as if it were fully supervised rather than signal and background. In theorem 1 of [109], it is shown that the optimal classifier of the mixed samples is equivalent to the optimal classifier of the signal and background samples. This means that the network can be trained to classify the mixed samples as if they were fully supervised and retrieve the signal and background labels. However, the metrics described in section 5.2.3 are not directly applicable to this case. This is because, measures of accuracy and precision like the ROC curve rely on accurate label information.

It is therefore necessary to test the network on separable data sets or mixed sets with known proportions. For completely separable sets, like simulated samples, metrics are straightforward to compute. For mixed sets with known proportions, one can calculate the ROC curve by performing threshold cuts on test samples. Given two mixed test samples with known fractions, T_1 and T_2, one can count the number of events in each sample after each threshold cut:

$$\begin{aligned} P(f(\vec{x}) > c|T_1) &= f_1\epsilon_S + (1 - f_1)\epsilon_B \text{ and} \\ P(f(\vec{x}) > c|T_2) &= f_2\epsilon_S + (1 - f_2)\epsilon_B, \end{aligned} \tag{5.13}$$

where ϵ_S and ϵ_B are the efficiency of the signal and background samples, respectively. $f(\vec{x})|T_i$ is the output of the network on the test sample T_i and c is the threshold cut. After each cut and count, equation 5.13 can be used to calculate the efficiencies of the signal and background samples. From these efficiencies, one can calculate metrics as well as the true and false positive rates, therefore deriving the ROC curve.

With metrics, one can evaluate the performance of the network on the mixed samples. In figure 5.10, a comparison of LLP, CWoLa and a supervised network from [109] are shown. The figure shows the AUC value given some purity of the samples as well as number of training events. As expected, neither weak supervision technique outperforms a fully supervised network. However, these are able to reach its performance in some cases. In particular, it can be seen that with more events one has available, the closer the training becomes to the supervised model. It is worth noting, there is an asymptotic behavior about 0.5 in the x-axis, denoting equal parts signal and background for the mixed samples (as here $f_1 = 1 - f_2$). In this case, it can be seen that the network is attempting to classify identical samples and therefore fails.

5.5 No Supervision

Previously, classifiers have been discussed but this is not the limit of what neural networks can do. With or without labels, neural networks can be used to learn underlying distributions, likelihoods, or serve as event generators. Creating networks that map not just $\vec{x}$ to labels y but to other distributions or even themselves provide such uses. Networks that map input to itself or other distributions may provide discriminating power for anomalous events. Others that map complex to simple distributions may be used for sampling purposes. In most of these cases, one needs to modify the architecture of the network. *Autoencoders* and their variations are an example of such networks which require no labels.

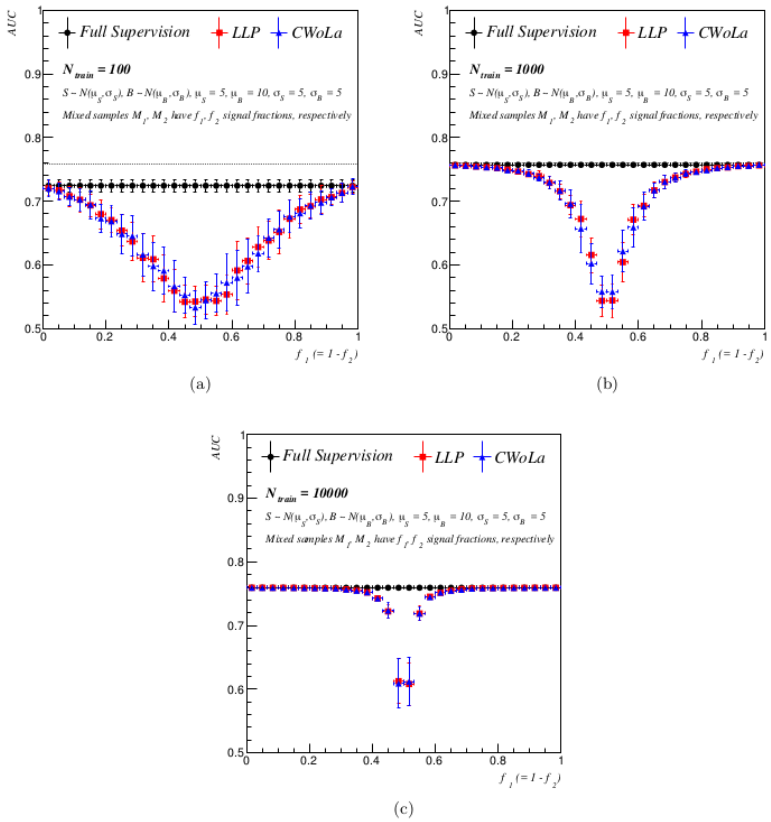


Figure 5.10: Comparative performance of CWoLa, LLP and a fully supervised model shown in [109]. This figure shows three plots of the AUC value as a function of the purity of the mixed samples assuming that $f_1 = 1 - f_2$. Each plot has increasing statistics starting with 100 events for (a), 1000 events for (b) and 10000 events for (c). It can be seen that statistics and purity of the samples have an impact on the performance of the network. Higher statistics allow for less pure samples to perform comparatively to a fully supervised model.

5.5.1 Autoencoders

Autoencoders are a type of neural network that map input to itself. At first glance, one could just copy the input to the output as the variables propagate through the network. However, this would not have any power and therefore the autoencoders must be more complex to achieve a useful purpose. In detail, an autoencoder is divided into three main components: the encoder, the hidden compressed representation and the decoder. A schematic of an autoencoder with labeled parts can be seen in figure 5.11. To some extent, the autoencoder can be thought of as a special case of a feed-forward neural network where the input is mapped to itself. This implies that an autoencoder can be trained similarly to a feed-forward network.

To build an autoencoder, one needs to build an encoder and decoder in a similar fashion to feed-forward networks and connect them. The encoder takes the input, $\vec{x}$, and maps it to a hidden representation, $\vec{h}$.

$$\vec{h}(\vec{x}) = f(\mathbf{W} \cdot \vec{x} + \vec{b}), \tag{5.14}$$

where, similar to equation 5.1, f is an arbitrary activation function. $\mathbf{W}$ is the matrix of weights from

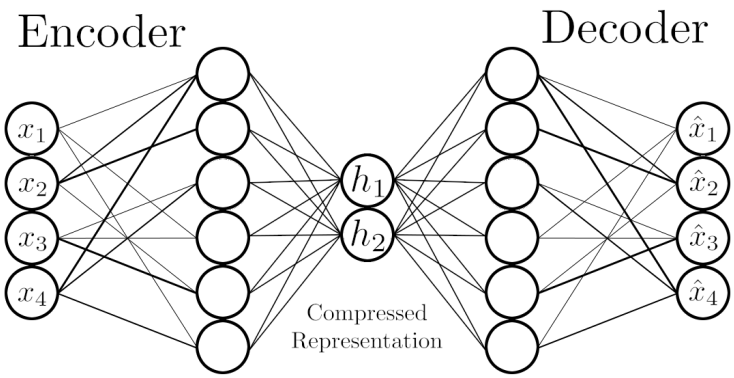


Figure 5.11: Schematic of an autoencoder network structure. The autoencoder is broken down into three parts: the encoder, the hidden compressed representation and the decoder. Input is mapped to itself and therefore the output of the decoder is an estimate of the input $\vec{x}$.

the compressed representation to the input layer, and $\vec{b}$ are the biases of the layers. The compressed representation, as its name suggests, is the input encoded in a lower dimensional space and is the most important feature of this architecture. This is done to restrict the flow of information from the encoder to the decoder. Lowering the dimension of the output forces the encoder to compress the most vital information of the input to the hidden representation. Lastly, the decoder takes the compressed representation and maps it back to the input.

$$\hat{x} = g(\mathbf{V} \cdot \vec{h}(\vec{x}) + \vec{c}), \tag{5.15}$$

where g, $\mathbf{V}$, and $\vec{c}$ are analogous to f, $\mathbf{W}$, and $\vec{b}$ from equation 5.14, respectively. It should be noted that although a smaller hidden representation yields a smaller risk of overfitting, it limits the amount of important information that may be encoded.

In section 5.2.1, loss functions were described as minimal when the true labels and the network estimate were close. Without labels, autoencoders must minimize the loss function differently. As an example, one can take the squared error between the input and the output:

$$\mathcal{L}(\vec{x}, g(f(\vec{x}))) = \sum_{i=1}^{N} (x_i - g(f(x_i)))^2, \tag{5.16}$$

where x_i is the input and $g(f(x_i))$ is the output of the network. Here, $f(\vec{x})$ is the output of the encoder and $g(f(\vec{x}))$ is the output of the decoder. When the network is able to encode and then reconstruct an event, the loss function reaches its minimum.

A network that reconstructs its input can be used to detect anomalies in other samples. The encoding of anomalous events would look different from the normal events. The decoder would then yield nonsense and the mean squared error of the event would be high.

5.5.2 Masked Autoregressive Density Estimators (MADE)

Beyond anomaly detection and classification, neural networks can be used for density estimation. This is a powerful technique as it allows one to generate new events once the network "understands" the input distributions. In general, many machine learning problems focus on understanding aspects of the joint distribution, $p(\vec{x})$, for classification. Learning the full joint probability density is a challenge in machine learning as it brings the *curse of dimensionality*. This means that as the dimensions of $\vec{x}$ increase, the volume of space a network must estimate increases exponentially.

In order to allow an autoencoder to estimate a probability, $p(\vec{x})$, it needs to be written in such a way that it can be computed. One can use the probability product rule to decompose a probability density function (pdf) as a product of conditionals:

$$p(\vec{x}) = \prod_{d=1}^{D} p(x_d|\vec{x}_{<d}), \tag{5.17}$$

where d denotes the current dimension in the product and D is the total dimensions of $\vec{x}$. Additionally, $p(x_d|\vec{x}_{<d})$ means the probability density of x's d-th dimension given all x's from previous dimensions. For example, if $d = 2$, then $p(x_2|\vec{x}_1, \vec{x}_0)$ is the distribution of x_2 given x_1 and x_0. Using this property, calculating the log-likelihood becomes

$$-\log p(\vec{x}) = \sum_{d=1}^{D} -\log p(x_d|\vec{x}_{<d}), \tag{5.18}$$

with the same definitions as equation 5.17. These equations can be expanded to account for more data events, as they only show the probability of one data point, $\vec{x}$. For N data points, the product of conditional probabilities can be constructed as

$$\prod_{d=1}^{D} p(x_d|\vec{x}_{<d}) = \prod_{i=1}^{N} p(x_d = i|\vec{x}_{<d})^{n_i}, \tag{5.19}$$

where i is the i-th event and n_i is the number of times the i-th outcome has happened. n_i can be rewritten as the multiplication of the total events, N, with the empirical probability of such outcome i, $q(x_d = i|\vec{x}_{<d})$ such that

$$\prod_{i=1}^{N} p(x_d = i|\vec{x}_{<d})^{n_i} = p(x_d = i|\vec{x}_{<d})^{N q(x_d=i|\vec{x}_{<d})}, \tag{5.20}$$

which, once one logs this quantity and divides by the number of events, the cross-entropy between the

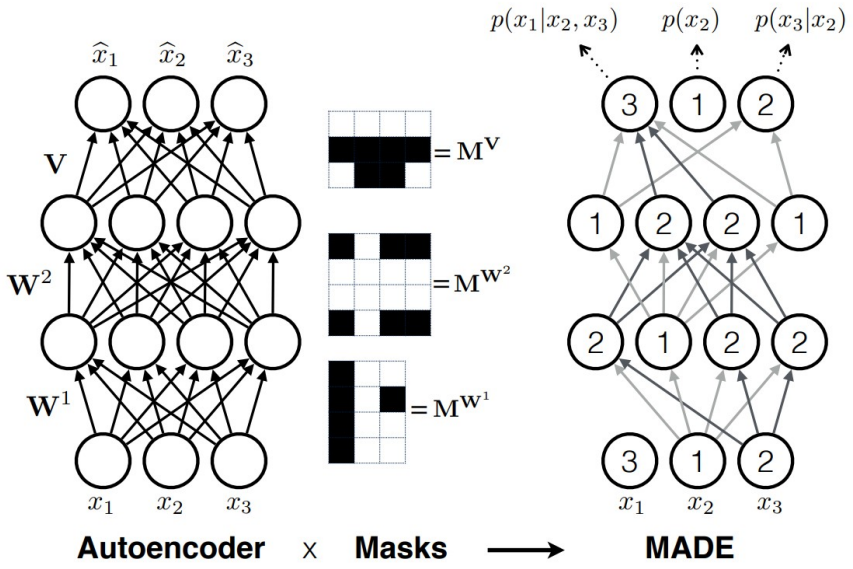


Figure 5.12: Schematic of an masked autoencoder network structure. [110] This autoencoder is built similarly to the autoencoder in figure 5.11, but with masked input. This masking ensures that the correctly labeled nodes only get information from previous dimensions only.

empirical and estimated distributions can be gotten,

$$\begin{aligned} -\log p(\vec{x}) &= -\sum_{i=1}^{N} q(x_d = i|\vec{x}_{<d}) \log p(x_d = i|\vec{x}_{<d}), \\ &= -\sum_{i=1}^{N} x_d \log p(x_d = i|\vec{x}_{<d}), \\ &= -\sum_{i=1}^{N} x_d \log \hat{x}_d. \end{aligned} \tag{5.21}$$

This can be used as a loss function to train a network to estimate the probability density of the data. The next step is to translate the conditional probabilities in a way that they can be translated in the context of an autoencoder.

Normally, an autoencoder already generates an $\hat{x}_d$ but it depends on all inputs from all other input. By masking each layer in the autoencoder, one can generate an $\hat{x}_d$ that is only dependent on the conditionals $x_{<d}$. This is referred as the *autoregressive property*, as the sequential prediction of each dimension of $\hat{x}$ is the same as computing the negative log-likelihood. Overall, a masked network that uses this property is called a *masked autoencoder* and a schematic can be found in figure 5.12. In this setup, the masked input ensures that the information from any particular dimension is only propagated to the correct conditional. This means that the output node labeled 3 receives the information from the nodes labeled 1 and 2. The output node labeled 2 does not receive information from any nodes but the ones labeled 1. Lastly, node labeled 1 receives no information from other dimensions as it has no conditionals.

One can achieve this by multiplying the weight matrix by a mask matrix which has binary values. This modifies equation 5.14 slightly

$$\vec{h}(\vec{x}) = f((\mathbf{W} \odot \mathbf{M}) \cdot \vec{x} + \vec{b}), \tag{5.22}$$

with a similar modification to equation 5.15

$$\hat{x} = g((\mathbf{V} \odot \mathbf{M}') \cdot \vec{h}(\vec{x}) + \vec{c}), \tag{5.23}$$

where M and M' are mask matrices for any particular layer. Lastly, this architecture can be made deep by just ensuring each layer is masked correctly. For a detailed explanation of the masking, see [110].

In figure 5.12, one can see a specific order of conditionals: $p(x_1|x_2, x_3)$, $p(x_2)$, and $p(x_3|x_2)$. In general, there is no reason why any one particular ordering should be the "natural" order. Additionally, conditionals of two different orders are not guaranteed to be the same or consistent. This would mean that different orders may effectively mean different potential networks. However, by randomizing the order of conditionals between batches, as presented in [110], one can achieve order-agnostic training. Similarly, the paper [110] shows algorithmically how to randomize the order of the connections between nodes in the autoencoder.

5.5.3 Normalizing Flows

Earlier in the chapter, event generation was briefly mentioned. The map presented by autoencoders was that of $X \to \hat{X}$, where $\vec{x} \in X$, which is less useful for event generation. One could generate a map of simple distributions to the X distribution such that $\mathcal{U} \to X$, where $\mathcal{U}$ could be some arbitrary distribution like a Gaussian. Alternatively, one could generate a map of the complex distribution to a simple one $X \to \mathcal{U}$, such that anything that does come from X cannot be mapped to $\mathcal{U}$. The method by which one can create such mappings is called *normalizing flows* (NF).

Normalizing flows are transformations of simple to complex distributions via invertible and differential variable transformations. To explain variable transformation, take some simple random variable $Z \in \mathbb{R}^D$ described by a probability distribution $p(\vec{z})$. The transformation $p(\vec{z})$ to $p(\vec{y})$, where $\vec{y} \in Y = g(Z)$ and g is an invertible function with $f = g^{-1}$, can be done by

$$\begin{aligned} p_Y(\vec{y}) &= p_Z(f(\vec{y}))|\det(Df(\vec{y}))|, \\ &= p_Z(f(\vec{y}))|\det(Dg(f(\vec{y})))|^{-1}, \end{aligned} \tag{5.24}$$

where D is the differential operator, $Df(\vec{y}) = \frac{\partial f}{\partial \vec{y}}$, known as the Jacobian of f. The new density $p_Y(\vec{y})$ is known as a *pushforward* of the density p_Z by the function g. The push of a simple function to a more complex distribution is often called a movement in the *generative direction*. On the other hand, the inverse function, f, moves the distributions into the *normalizing direction*.

Assuming that the distributions $p(\vec{z})$ and $p(\vec{y})$ are continuous and differentiable, such a transformation g can generate any distribution $p(\vec{y})$ from any distribution $p(\vec{z})$ [111–113]. The complexity of defining such a function which is also bijective (non-linear and invertible) should be noted. The difficulty is not just in the definition of g but in its calculation of the Jacobian and inversion. However, this difficulty can be mitigated by changing the definition of g as a series of transformations rather than just one. If one defines g as a composition of N bijective functions, written as $g \overset{\text{def}}{=} g_N \circ g_{N-1} \circ \cdots \circ g_1$,

then g is also bijective. If g is bijective, then it has an inverse function defined as $f \overset{\text{def}}{=} f_1 \circ f_2 \circ \cdots \circ f_N$. To calculate the determinant of the Jacobian, one only needs to multiply each inverse:

$$\det(Df(\vec{y})) = \prod_{i=1}^{N} \det(Df_i(\vec{y}_i)), \tag{5.25}$$

where the Df is the same as in equation 5.24 and i denotes the i-th intermediate flow defined as $\vec{y}_i = g_i \circ \cdots \circ g_1(\vec{z}) = f_{i+1} \circ \cdots \circ f_N(\vec{y})$. In this way, one can construct a complicated and bijective transformation g by using a series of simple transformations. Unfortunately, this does not fix the problem of calculating the Jacobian easily.

A clever transformation must be made in order to deal with the calculation of the Jacobian. One can use a *linear flow* which allows the learning of correlations between dimensions. These are written as

$$g(\vec{x}) = \mathbf{A} \cdot \vec{x} + \vec{b}, \tag{5.26}$$

where $\mathbf{A}$ is an invertible matrix of shape $\mathbb{R}^{D\times D}$ and $\vec{b}$ is a $\mathbb{R}^D$ vector. Both $\mathbf{A}$ and $\vec{b}$ are parameters of the flow. In this case, the determinant of the Jacobian is just $\det(\mathbf{A})$. The computation of this determinant and the inverse's is of $O(D^3)$ which is an example of the curse of dimensionality and how expensive such a calculation can be. However, one can restrict the form of $\mathbf{A}$ to be a triangular matrix. When $\mathbf{A}$ is triangular, its determinant is only the multiplication of the diagonal terms. This lowers the computational cost to be of $O(D^2)$ instead. One can stack these transformations as mentioned earlier to attain a more complex or general flow from simpler composites. However, the inverse of these matrices may be a combination of upper- and lower-triangular matrices which would make the product non-triangular.

Another complication arises in practice. Although calculating either g or f and then inverting to attain the other is possible, often times it is computationally expensive. For this reason, when one is using a NF, one should mind the direction in which the NF is being used. Commonly, when one desires to estimate the complex density, one should model the flow in the normalizing direction. In the case of event generation, one should model the flow in the generative direction. Although a NF is more of a mathematical model that can be "cranked out" by a likelihood fit, it can also be used in machine learning. In particular, [114] and [115] have used normalizing flows in the context of neural networks.

5.5.4 Masked Autoregressive Flows

Linear flows, even though they can express correlations in dimensions, are still limited. For example, the linear transformation as described in equation 5.26 can only take a normal Gaussian to another Gaussian. If z is described by the normal distribution $\mathcal{N}(z, \mu, \sigma)$, then its transformation remains Gaussian with $y = g(z) \sim \mathcal{N}(y, \mathbf{A}\mu + b, \mathbf{A}^{\mathrm{T}}\sigma\mathbf{A})$. However, using the triangular matrices described above in a clever way, one can still craft a powerful non-linear generalization. In the end, one can tie the previous sections about flows and autoregressive models together to form an autoregressive flow.

In section 5.5.2, an autoregressive model was introduced where the probability density to be estimated was split into conditionals. The masking caused nodes to be estimates of the conditional probabilities $p(x_d|\vec{x}_{<d})$ of the d-th dimension. This probability can be parametrized by a Gaussian

whose arguments are functions of $\vec{x}_{<d}$ as they propagate through the network:

$$p(x_d|\vec{x}_{<d}) = \mathcal{N}(x_d|f_{\mu,d}(\vec{x}_{<d}), (\exp f_{\sigma,d}(\vec{x}_{<d}))^2), \tag{5.27}$$

where the functions $f_{\mu,d}$ and $f_{\sigma,d}$ are unconstrained scalar functions that compute the mean and log standard deviation of the d-th conditional. This model is generative as one can generate an x_d by using $x_d = u_d \exp(f_{\sigma,d}(\vec{x}_{<d})) + f_{\mu,d}(\vec{x}_{<d})$ where u_d is distributed as a normal $\mathcal{N}(0,1)$. In this regard, the model is similar to a normalizing flow set in the generative direction. The transformation f maps a space of random, normally distributed numbers $\vec{u}$ to the complex space of data $\vec{x}$. Additionally, the inverse of f exists as it is a bijection and therefore one can generate u_d from x_d by using $u_d = (x_d - f_{\mu,d}(\vec{x}_{<d})) \exp(-f_{\sigma,d}(\vec{x}_{<d}))$. Lastly, the transformation and its inverse are triangular by design as f at the d-th dimension only depends on $\vec{x}_{<d}$. This means that the Jacobian can be easily calculated as such:

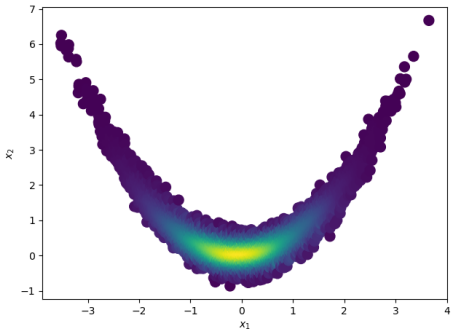
$$\det(Df(\vec{x})) = \exp(\sum_d -f_{\sigma,d}(\vec{x}_{<d})). \tag{5.28}$$

As mentioned in section 5.5.2, the order in which the conditional probability is defined may compose different mappings which may not converge. For example, the performance of this model can be tested by transforming the training data from the complex space to the simple space where all dimensions should become independent, normal distributions. This is exemplified in figure 5.13, where the order of the conditional probability can lead to a success or failure of the flow. It can be seen that every x_1 value can be mapped to a single x_2 value. In this case, the conditional probability $p(x_1|x_2)$ can be put into an autoregressive model used as a flow. On the other hand, each x_2 value can be mapped to one or more x_1 values; such a distribution is called multimodal. Putting a multimodal conditional probability $p(x_2|x_1)$ in such a model will cause the flow to fail as shown in figure 5.13(d). The problem of multimodal distributions can be solved by stacking such autoregressive models, like the MADE model introduced earlier, as a flow. This is called a *masked autoregressive flow* (MAF). By stacking MADE layers, one can achieve the same result as figure 5.13(b) regardless of input order. The most potent contribution made by MAFs is that by stacking layers, or transformations, one can transform arbitrarily complex distributions to a simpler one.

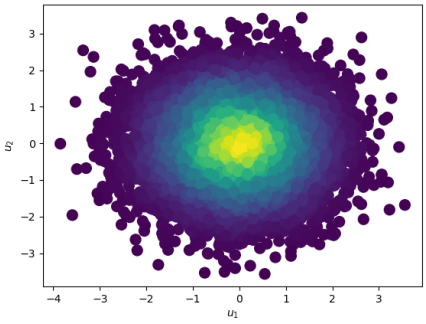
5.5.5 Anomaly Detection with Density Estimation

In section 5.5.2, autoregressive models were introduced as density estimators. Section 5.5.4 shows how stacking such autoregressive models can be used to form a normalizing flow. Such flows can be used to estimate the density of a distribution as well as to detect anomalies. Putting together all the tools explained thus far, one gets the *Anomaly Detection with Density Estimation* (ANODE) technique. This technique is used for "bump hunts" in data, for signals that are localized. In a nutshell, one estimates the density at the signal locale and where signal absent. By interpolating the density of the background into the signal region, one can construct a likelihood ratio that is highly sensitive to anomalies (i.e. signal).

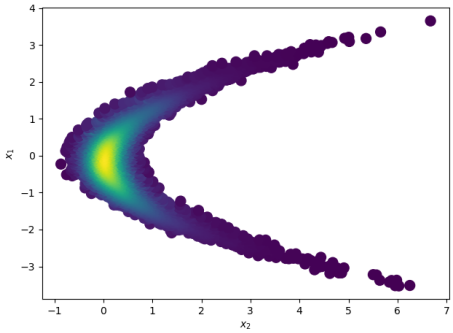
Normally, signal hunting is done by identifying key features and enhancing signal purity. Given some feature, m, where the anomaly is located around some m_0, one can define two orthogonal regions. A signal region (SR) where SR $\in [m_0 - \delta, m_0 + \delta]$ and a sideband region (SB) defined as not SR. Other discriminating features, $\vec{x}$, are typically chosen to cut and enhance the signal contribution in the m distribution. However, such a procedure requires a-priori knowledge of the distributions



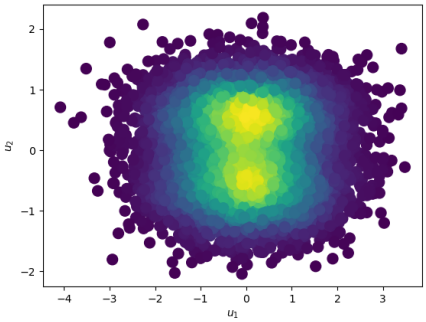
(a) Original density where x_2 is dependent on x_1



(b) Example of a resulting simply-distributed u_1 and u_2 if the input dependency is as figure 5.13(a)



(c) Original density where x_1 is dependent on x_2



(d) Example of a resulting simply-distributed u_1 and u_2 if the input dependency is as figure 5.13(c)

Figure 5.13: Example showing the importance of dependence order for a single MADE model. Figures 5.13(a) and 5.13(c) show the input distribution where the dependency between two features is switched. When x_1 depends on x_2, the conditional distribution of $p(x_1|x_2)$ is bimodal which causes the flow to fail and return a non-Gaussian distribution as shown in figure 5.13(d). Alternatively, when conditional distributions are not multimodal, the flow may succeed and yield the image in figure 5.13(b).

such that signal is correctly enhanced. The goal of ANODE is to perform such an estimation in a model-agnostic and data-driven way.

In ANODE, one estimates the densities of the SR and SB regions using flows of autoregressive models. From the SR, one gets the probability density of data, $p_{\text{data}}(\vec{x}|m)$, where the SB estimate yields the background only density, $p_{\text{background}}(\vec{x}|m)$. After extrapolating the SB density to the SR, these two densities can form a likelihood ratio which is sensitive to the difference between the two:

$$R(\vec{x}|m) = \frac{p_{\text{data}}(\vec{x}|m)}{p_{\text{background}}(\vec{x}|m)} \text{ for } m \in \text{SR}. \tag{5.29}$$

Assuming that $p_{\text{data}}(\vec{x}|m) = \alpha p_{\text{background}}(\vec{x}|m) + (1-\alpha)p_{\text{signal}}(\vec{x}|m)$ for $\alpha \in [0,1]$, equation 5.29 is the optimal statistic for signal presence [116]. If $R(\vec{x}|m) = 1$, then $p_{\text{data}}(\vec{x}|m) = p_{\text{background}}(\vec{x}|m)$ which indicates the absence of signal. Any other value of $R(\vec{x}|m)$ indicates the presence of signal as long as $p_{\text{signal}}(\vec{x}|m) \neq p_{\text{background}}(\vec{x}|m)$. However, given that both densities are estimates, it is not

guaranteed that the likelihood ratio should be equal to one in the absence of signal. If both densities are estimated well enough, the ratio should be localized around one in the SR with some events showing a higher ratio value. By applying a threshold cut on $R(\vec{x}|m)$ such that $R(\vec{x}|m) > R_c$, one can enhance signal purity. Additionally, the interpolated background density, $p_{\text{background}}(\vec{x}|m)$, can be used as a background estimate.

In [116], this method is compared to one called CWoLa hunting. Using the similar region definitions as above, one can train a CWoLa model to differentiate between the SR and SB regions. This classifier approaches the likelihood ratio and would have a similar form to equation 5.29 given some conditions. As long as there is no signal in the SB region and the features $\vec{x}$ and m are independent, then the ratios should be the same. However, none of these are guaranteed. Even if there is some signal contamination in the SB, CWoLa can still form a reasonable classifier but it would not be identical to ANODE's ratio. The problem rises in CWoLa's inherent assumption that $\vec{x}$ and m are independent for optimal signal extraction. Any dependence between these features would only lead CWoLa to discern the differences between SR and SB regions rather than signal and background. Given the same problem, ANODE succeeds where CWoLa fails as no relationship between $\vec{x}$ and m is required.

CHAPTER 6

Imprecise Modeled Processes

To summarize one of the problems, the $WWbb$ analysis is aimed to probe interference effects between tW and $t\bar{t}$. The variable $m_{b\ell}^{\text{minimax}}$ was introduced and shown to be highly sensitive to interference effects. This interference is shown as the difference between the DR and DS schemes. In section 6.1, strategies are presented with the purpose of generating a new variable with enhanced sensitivity to interference.

6.1 Interference Between tW and $t\bar{t}$

As mentioned in section 2.4, the tW and $t\bar{t}$ processes are not independent of each other. By interfering at higher orders, these two become intertwined in ways that are difficult to classify. Not just that but trying to measure the cross-section of one process means the other cross-section must change. After all, both are subject to the CKM element V_{tb} except one is singly resonant and the other doubly.

With all of these problems, physicists still use simulation to try and model these processes as if they were separate. The DS and DR schemes are used to estimate the tW contribution with and without interference, respectively. However, this interference is impossible to simply generate and instead is estimated in simulation. To improve simulation, experiments are made to measure the differential cross-section of these processes. Specifically, by probing interference-sensitive variables one can improve the modeling. Therefore, the goal explained in this section is the generation of a variable, or classifier, which is sensitive to the interference.

CWoLa

Section 5.4.2 outlined a model which compared two mixed samples in order to create a classifier which identifies the classes within them. The CWoLa technique is also useful for estimating outliers or bump hunting, often referred to as "CWoLa hunting". This technique is attractive for finding over-densities in a localized region of space. For this reason, the first strategy to create a classifier which can discern interference is inspired by CWoLa hunting.

Figure 4.2(b) showed the difference between the DR and DS schemes increase with higher values of the $m_{b\ell}^{\text{minimax}}$ distribution. This difference may be exploited such that events which are strictly tW are one class and not "well defined" events are another. To remind the reader, CWoLa relies on having enough statistics as the similarity between samples increases. Therefore, the advantage of looking

at MC is that a large amount of statistics are available and therefore the samples could have some similarity.

In the DS scheme, the total number of events which are not well-defined tW is not known. After all if it were possible to label them, then one would not need to create a DS scheme sample. The naïve strategy would be to train DR versus DS and let the network sort out the differences. However, since both schemes have events which are identical, the prudent choice would be to train on a region of enhanced difference. That way, the fraction of purity is increased and the likelihood of convergence increases.

A candidate for such a region would be to impose a criteria of two jets which must be b-tagged; called 2j2b. As was mentioned earlier, the 2j2b region is where the greatest interference effect can be witnessed. In accordance to the signal region defined in the $WWbb$ analysis, only opposite sign (OS) and opposite flavor (OF) leptons are considered. Furthermore, adding a cut on the $m_{b\ell}^{\text{minimax}}$ variable would increase the fraction of interference events in DS. There is a problem with cutting on $m_{b\ell}^{\text{minimax}}$ as this would make the network correlated to a variable which is desirable for differential measurement. Ideally, the classifier should not rely on variables of interest for the analysis. Therefore, a first attempt is made with DR versus DS and later tweaked as necessary.

A direct input of variables such as $m_{b\ell}^{\text{minimax}}$ may cause difficulties for an analysis use case. Therefore, the variables used are simply kinematics in order for the network to build a relationship similar to $m_{b\ell}^{\text{minimax}}$. Considering that interference terms should have an off-shell second top-quark, the decay products' kinematics should differ. Additionally, such differences are likely accentuated in the Lorentz invariant inner product.

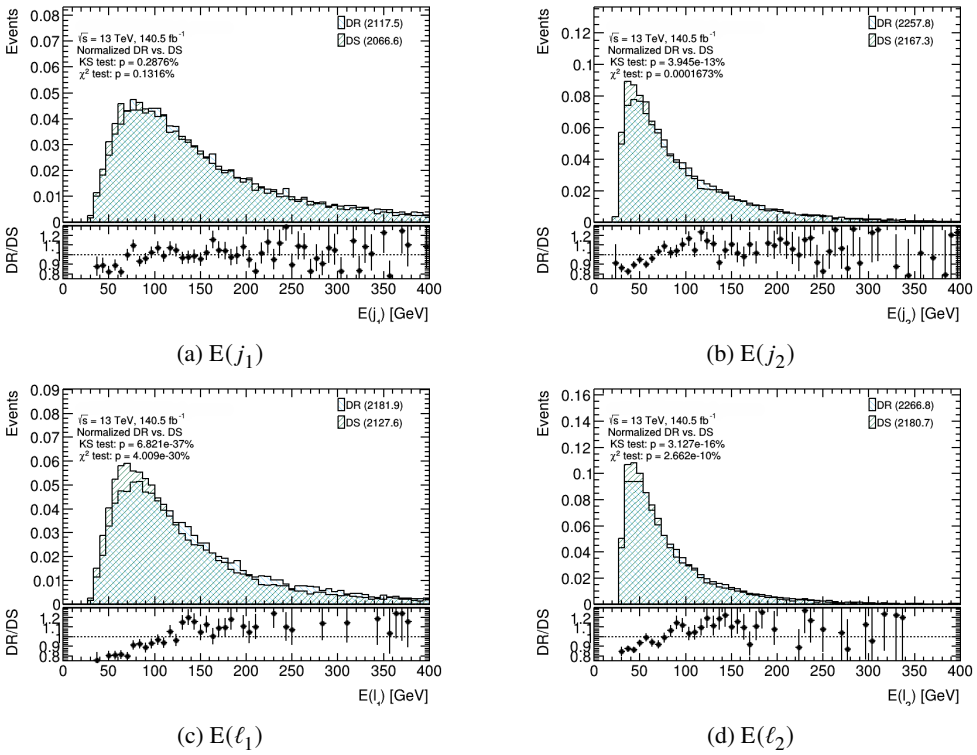
An example comparing the DR and DS schemes can be found in figure 6.1. It can be seen that the energies of particles have slight differences between the two schemes. Although nearly identical, the slight differences might be useful for a network. Specifically, the energy of the leading lepton and second jet differ the most. In the end, the variables used were:

- kinematics of both lepton and both jets (E, η, and ϕ),
- $E_{\text{T}}^{\text{miss}}$, and
- the Lorentz invariant inner product of any two of the four objects.

The network's hyperparameters are found by performing a grid search. A grid search is when one trains various models in order to find the optimal hyperparameters. Each model is different from the rest and later evaluated by the user to see which network performed best. Normally, one could try to look at a network on some metric like the AUC value. The problem with this one-dimensional approach is that a high AUC value may not always yield the best separation. Therefore, a holistic approach is better suited to evaluate various networks. One should consider every metric available and the output when choosing the appropriate model.

The best network was composed of 3 hidden layers with 32 nodes each with a dropout rate of 33 %. Each layer had a `ReLU` activation function with the exception of the last one which was sigmoid. In accordance with the CWoLa paper, a simple binary cross-entropy function was used as the loss optimized by the Adam algorithm. This network was trained for 200 epochs with a small batch size of 2 000.

A challenge seen during this training was how often models overtrained or failed to converge. Most of the time, the networks would fail to place the signal file and would just split it evenly across the



(a) E(j_1) (b) E(j_2)

(c) E(ℓ_1) (d) E(ℓ_2)

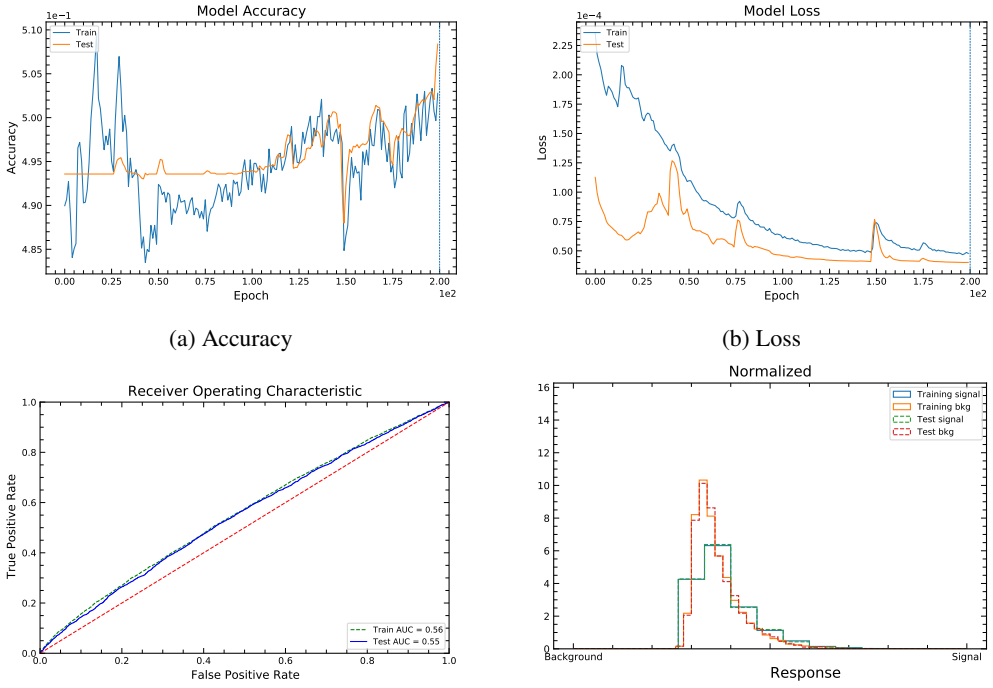
Figure 6.1: Measured energy of physics objects comparing the DR and DS schemes. A KS test score and χ^2 fit value are provided as metrics of how different these two distributions are.

response. In these cases, the DS had a surplus on one side which sounds expected but the training and test samples were too different to ignore. Even with adding normalization terms to the loss function and increasing dropout, networks failed to converge.

The performance of the best non-overtrained network can be seen in figure 6.2. Note that the accuracy depicted here is dependent on the assigned labels which are known to be incorrect for mixed samples. Furthermore, the network is tasked to classify two nearly identical samples so an accuracy barely above 50 % is expected if it worked perfectly. This is also corroborated by the ROC curve being near the middle "random assignment" threshold. The plot which is most telling is the response as it shows no clear separation. Additionally, the samples populate the middle rather than the extremes. This tells that the network is unsure of where to place events. Although the loss function performs somewhat as it should, all other metrics shown allow one to conclude this model is unsatisfactory.

In the end, there are many reasons why this method has failed. Primarily, the number of samples available for the network (about 35 000) were not enough to account for the small fraction of interference events. A large portion of the DS scheme is made up of tW events and therefore the fraction of impurity is too low for the network to account for. Second, as these two samples are nearly identical and interference events are not different enough, it is likely that this strategy is not sensitive enough for the desired task.

One could attempt to increase the fraction of interference events by performing a cut on $m_{b\ell}^{\text{minimax}}$. However, with the variables provided the network is likely to simply calculate the variable internally



(a) Accuracy (b) Loss

(c) (d)

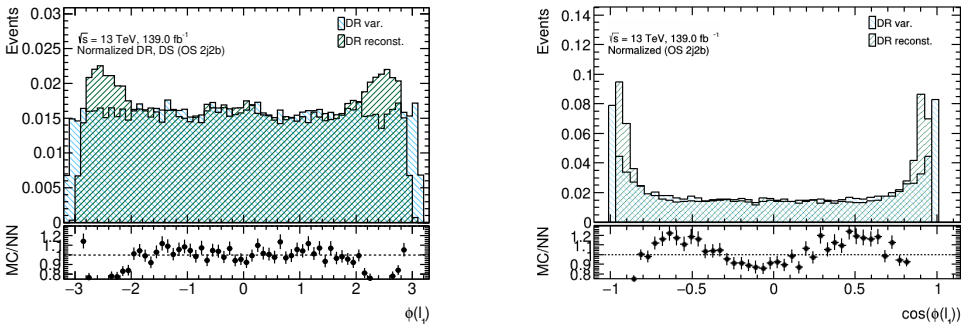
Figure 6.2: Metrics of the best performing NN tasked with identifying interference events in a $WWbb$ selection. The accuracy is included for completeness but it is not useful for weak classifiers.

and classify accordingly. After all, $m_{b\ell}^{\text{minimax}}$ is a measure of the off-shell-ness of a second top which is defined by the masses of the decay products. Another idea is to consider using LLP instead of CWoLa. By modifying the loss function to account for the fraction of impurity, the network may improve in performance. In this case, one would need to know the fraction of events that contaminate the mixed sample. This is not easily known but could be estimated and give the network a better chance at finding a classifying function. It should be noted that LLP would suffer similarly to this attempt due to the low statistics and purity. Therefore, one would reach the same conclusion of needing to increase the fraction of events and conclude in failure. For this reason, this strategy is no longer pursued and instead anomaly detection techniques are employed.

Autoencoders

Autoencoders are useful tools which compress input information into a hidden representation. The second part of the autoencoder then reconstructs the input features based on the encoded data. The ability to reconstruct input means that a well-tuned autoencoder can become sensitive to anomalous input. Events or features that an autoencoder is unfamiliar with yield a measurable reconstruction error which can be used to discern anomalous events in data. In this section, this approach is taken where an autoencoder is trained to identify tW events in the hopes of mis-reconstructing interference events.

In the previous strategy, a limiting factor was the unknown quantity of interference events which exist in the DS scheme. One could assume a contribution between 10 and 20 % but without proper



(a) $\phi(\ell_1)$ (b) $\cos\phi(\ell_1)$

Figure 6.3: Comparison of the $\phi(\ell_1)$ and $\cos\phi(\ell_1)$ distributions with their reconstruction. The autoencoder is unable to reconstruct these distributions and obvious artifacts can be seen.

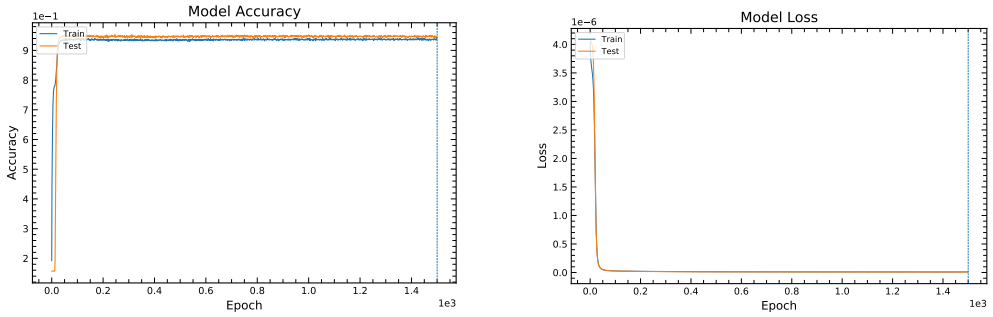
labels one cannot be sure. With certainty, it is known that the DR scheme contains only well-defined tW events. Therefore, one could attempt to treat the interference-like events as anomalies and employ an anomaly detection technique.

As autoencoders are easily accessible, they are the first choice to tackle this problem. The idea is to train an autoencoder to recognize tW events from the DR sample so it learns what well-defined tW events are. Of course, the selection of these events would be in the 2j2b region as before. If such a model is applied to the DS scheme, "anomalous" events where there is an off-shell top-quark should not be as well reconstructed. Such a reconstruction loss is measurable by comparing the initial variables and their reconstructed values. One advantage of this strategy over the previous is that no cut on interesting variables (such as $m_{b\ell}^{\text{minimax}}$) would be required. In fact, one can give this variable such that the network attempts to reconstruct it for both samples.

The next step is to consider the variables that the network is tasked to reconstruct. Initially, the same variables from section 6.1 were considered as these are the kinematics of all event objects. Distributions like the ϕ-direction of some object would seem easy to model; as it is a flat distribution. However, all networks with differing architectures failed to reconstruct the ϕ distribution of all objects. This can be seen in figure 6.3(a) where the reconstructed distribution given by the autoencoder has "Batman ears". As the network is unable to reconstruct distributions such as these, the function $\cos\phi$ is used instead to give the network better input. This is done with the assumption that the reason for failure is the lack of distinct features in the distribution (i.e. a peak).

This new distribution and its reconstruction are shown in figure 6.3(b). Even with a different distribution, the network was unable to correctly reconstruct the input. Although technically a failure, this shows that the network is unable to learn sharp cut-offs. This is seen most easily in the right-hand side of the $\cos\phi$ distribution as the peak is followed by a sharp fall. This feature implies the network has approximated the input by continuous functions with differentiable edges. An obvious conclusion in hindsight. In this case, the inability for the network to reconstruct the initial flat distribution was not the lack of features but lack of differentiable edges. For this reason, features which had sharp cut-offs are omitted. The variables included in the network were:

- the energy and η of all four final state particles,
- $E_{\text{T}}^{\text{miss}}$,



(a) Accuracy

(b) Loss

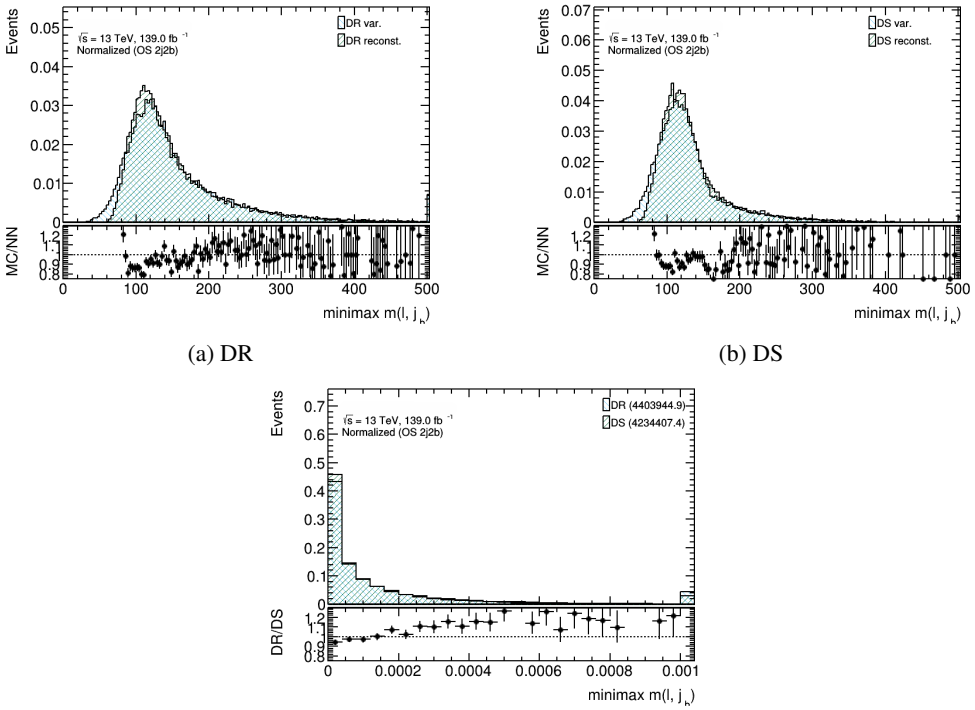
Figure 6.4: Metrics of the autoencoder trained on the DR sample with 2j2b selection.

- the LIIP of any two objects,
- $m_{b\ell}^{\text{minimax}}$,
- and the radial distance between any two objects.

After tweaking several parameters, the best autoencoder was composed of a 4×64 encoder and decoder with a latent dimension of 12. As total there were 22 input variables, the latent dimension was made to be smaller to function as a normal autoencoder. The encoder had the the same activation function, `eLU` throughout until the last layer where it used a tanh function. The decoder used a sigmoid function in the input layer, followed by `eLU` for all internal layers, until the output layer which used another sigmoid function. The loss function used to train was the MSE as it compares the decoder's output with the input directly. As input variables had differing ranges, these were standardized to be within 0 and 1 while retaining their shapes by the `MinMaxScaler`. The network was optimized using the Adam algorithm with Tensorflow's default parameters. The training was performed over 1 500 epochs with a small batch size of 750 as the network was given only about 33 000 events. This is the network which performed the reconstructions shown in figure 6.3(b).

The metrics of this network can be seen in figure 6.4. It is shown that the network achieves a high accuracy in reconstruction. However, this metric is not incredibly helpful as Tensorflow uses a binary accuracy. Therefore, one should take this metric with a grain of salt. The loss, MSE, calculated shows how close the average estimate is to the original input. It should be noted that the low value can be attributed to the weight of the events; often in the order of 1×10^{-6}. Additionally, the values being compared in the loss calculation have been constrained to a unit range and therefore expected to be small. Given these points, the loss is expected to be orders of magnitude smaller than 1. It is then concluded that this model learned something but a full assessment can only be made after checking the results.

Figure 6.5 shows the reconstructed $m_{b\ell}^{\text{minimax}}$ values for both DR and DS samples. Also, the comparison of the reconstruction error is shown in figure 6.5(c). From this variable, it can be seen that the reconstruction is not entirely accurate. Furthermore, the reconstruction of both DR and DS is reasonably good, at least for this variable. However, the comparison of the error for both distributions show the network does not perform the desired task. It is shown in this plot that the reconstruction error for this variable is better for the DS sample than DR. In which case, the network cannot be used to discriminate against interference events.



(a) DR

(b) DS

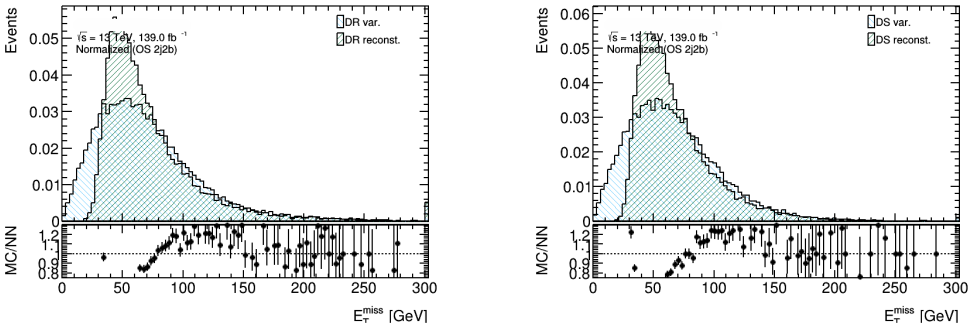
(c) Reconstruction Error Comparison

Figure 6.5: Reconstructed $m_{b\ell}^{\text{minimax}}$ variable compared to the original in both DR (a) and DS (b) samples. The reconstructed error shown in (c) compares both samples.

It should be noted that not all variables were well reconstructed. An example of a badly reconstructed variable is shown in figure 6.6. Here, the reconstructed $E_{\text{T}}^{\text{miss}}$ for both samples is not as sharp as the original distribution. Therefore, the network has failed to learn these distributions. However, the bad reconstruction of the variable is relatively similar for both samples as the comparative error seems to be around 1 throughout. All reconstructed variables and similar comparison plots can be found in appendix F.

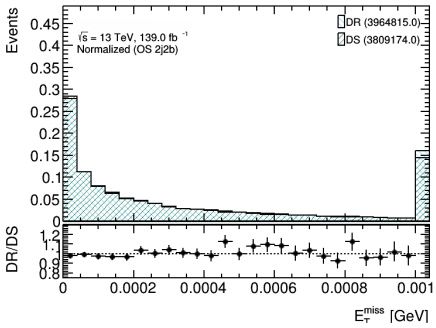
Finally, the errors can be aggregated to calculate the average reconstruction error. This plot is shown in figure 6.7. With all 22 variables reconstructed and their error calculated, it becomes obvious that this network has failed at its task. The reconstruction error for the DR sample is greater than the DS sample and therefore no classification can be easily made.

The autoencoder strategy comes short as no explored setup was able to estimate the correct densities. It is likely the error lies in the construction of such a network. After all, most of the uses of autoencoders have been in image recognition or de-noising. In this case, it was used to attempt and estimate probability distributions. It is also possible that the tool just does not fit the job and therefore it was unable to succeed. No claim is made on whether an autoencoder could not be built for this purpose. But the complexity of the problem and the similarity between interference and well-defined events may prove too difficult regardless. Rather a simple autoencoder, even if perfectly built, may not be sensitive enough to discern the differences in DR and DS more than the $m_{b\ell}^{\text{minimax}}$ distribution already does.



(a) DR

(b) DS



(c) Reconstruction Error Comparison

Figure 6.6: Reconstructed $E_{\mathrm{T}}^{\mathrm{miss}}$ variable compared to the original in both DR (a) and DS (b) samples. The reconstructed error shown in (c) compares both samples.

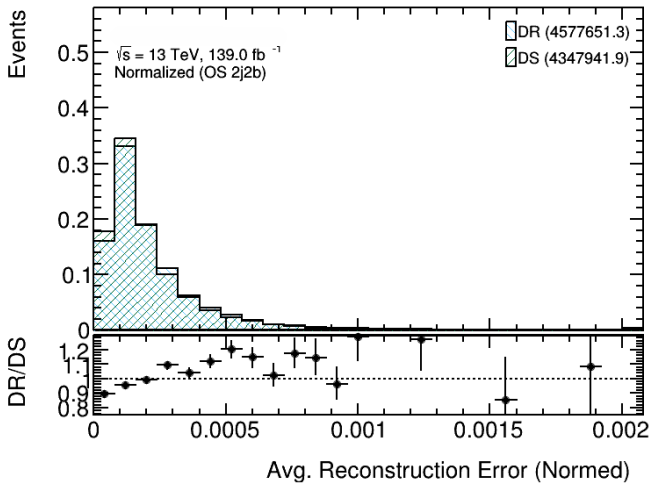


Figure 6.7: Average reconstruction error of the autoencoder trained on the *tW* DR sample.

Masked Autoregressive Flows

With autoencoders failing to estimate the distributions of well-defined tW events, a new tool is chosen. The core strategy remains the same: train some network on tW DR and apply it to DS. The resulting application should show some discernable feature which can be used as an interference-sensitive variable.

Normalizing flows were introduced in section 5.5.3 as functions which map complex to simple distributions, or vise-versa. Additionally, these transformations can be stacked to, in theory, model an arbitrarily complex distribution. Using normalizing flows, the goal is to map the complex distributions from the DR sample to a multi-dimensional Gaussian. From this mapping, any events in another sample which are not in the trained set would not map accordingly. Therefore, the event and sample selection remains the same as before.

Although any given variable should be usable for this application, only a few were chosen. The selection of variables was similar to that of the previous sections but with some exclusions; for example the LIIPs. The reason was that scaling these variables which had a huge range of values caused instabilities during training. Ideally, the variables should have distributions that can be easily fit in the scaled range such that the network can learn. Otherwise, the network would see sharp (delta-distribution-like) peaks which are then propagated throughout the network. Similarly, adding variables like the ϕ difference between two objects was problematic as these are multimodal distributions. By removing variables such as these, the network was allowed to converge more reliably.

Furthermore, density estimators such as MADE, MAFs, etc. struggle with sharp edges or hard boundaries in distributions. For this reason, the input data is transformed by a different procedure than using a scaler from the `scikit-learn` package. First, all features are scaled to be bound between 0 and 1. Then the features are put through a logit transformation which changes the range to $(-\infty, \infty)$. The logit transformation is defined as:

$$y = \log\left[\frac{x}{1-x}\right]. \tag{6.1}$$

Although this distribution is better for density estimators, the boundaries are still somewhat difficult for the network. For that reason, the scaled distribution is limited between some range; for example between 0.01 and 0.99. A slight loss in training events is to be expected but given the large data size, it is not worrisome. More specifically, out of about 33 000 events less than 150 events are lost to this cut.

Setting up a MADE network is possible with the use of the package `Tensorflow-probability` [117]. This was integrated in the NN package mentioned in appendix A for future use. However, work was already done to provide a package but in PyTorch [118] by Luka Vomberg during his masters thesis. Using his code found in [119], I was able to create a MADE model which was trained on the tW DR set.

The architecture chosen was four hidden layers of 30 nodes each with the `ReLU` activation function. This architecture was chosen for speed as the intent later on is to stack such networks. Additionally, increasing the complexity of the network was not likely to improve the flow as it was only one MADE layer. To foreshadow later discussion points, multiple MADE networks interconnected improve the transformation from complex to simple distributions. Therefore, simple networks are desirable to keep the overall training short. The minimization of the network is done by the Adam optimizer using the negative log likelihood as its loss function. The Adam optimizer had a custom learning rate of 0.05. The network was trained for 600 epochs with a batch size of 5 000.

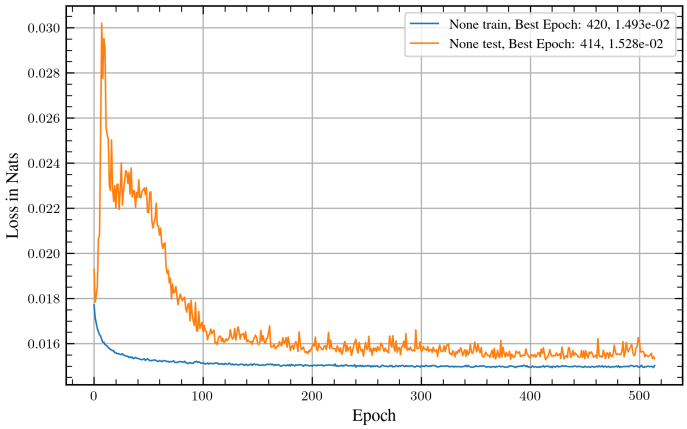


Figure 6.8: Loss of the single MADE layer network. The loss measures the negative log likelihood of input data.

As accuracy and ROC do not make sense for this type of network, the only metric is the loss of the MADE network which can be seen in figure 6.8. One visible feature of this loss curve is that it ends shortly after the 500 mark. This is because the network did not improve in over 100 epochs and therefore it stopped. At the beginning of training, the test curve fluctuates rapidly and is sporadic. However, as the training improves, this curve converges with the training loss curve; implying that the network has reached a minimum which treats both sets similarly.

Figure 6.9 shows four selected variables and their distributions after being put through the MADE network. It can be clearly seen that these distributions have been changed as they look nothing like the original. Some of these are more Gaussian-like than before, in particular figures 6.9(a), 6.9(b) and 6.9(c). However, figure 6.9(d) has become similar to a mixture of two Gaussian distributions. Clearly, this MADE network has reached a conclusion that is unsatisfactory. Furthermore, the loss curve implies that this is as far as this network can go as a bijector.

One could try to expand the architecture of the network in the hopes that an arbitrarily complex network can perform the desired transformation. After seeing a multimodal distribution in the output, the simpler course of action is to stack a second MADE network after this one. This second network would take the output of this network and attempt to map it to Gaussian distributions; the same task as before. Each of these networks is a bijector and a composition of bijectors is still a bijector. Therefore, this process can be repeated N number of times until the map is correctly performed, expecting each stack to get closer to the desired goal.

Figure 6.10 shows the loss curve for a network composed of four MADE layers. Similar to the previous one-layer network, the initial losses are disparate but they trend similarly as the training improves. This loss shows a more erratic test curve but the last 50 epochs show a convergent point where the network stops. Using this network, the resulting bijector can be checked and evaluated.

Figure 6.11 shows the same four selected variables and their distributions after being put through the MADE network. In this way, it is easy to compare to figure 6.9 and see the difference between one and four layers. By comparing figures 6.11(d) and 6.9(d), one can see the improvement clearly. The four-layer network creates a bijector that has answered the multimodal problem of the previous "shallow" network. It is not completely Gaussian and still has some artifacts but it is an improvement on the previous distribution. Unfortunately, the other three distributions do not show much of a change.

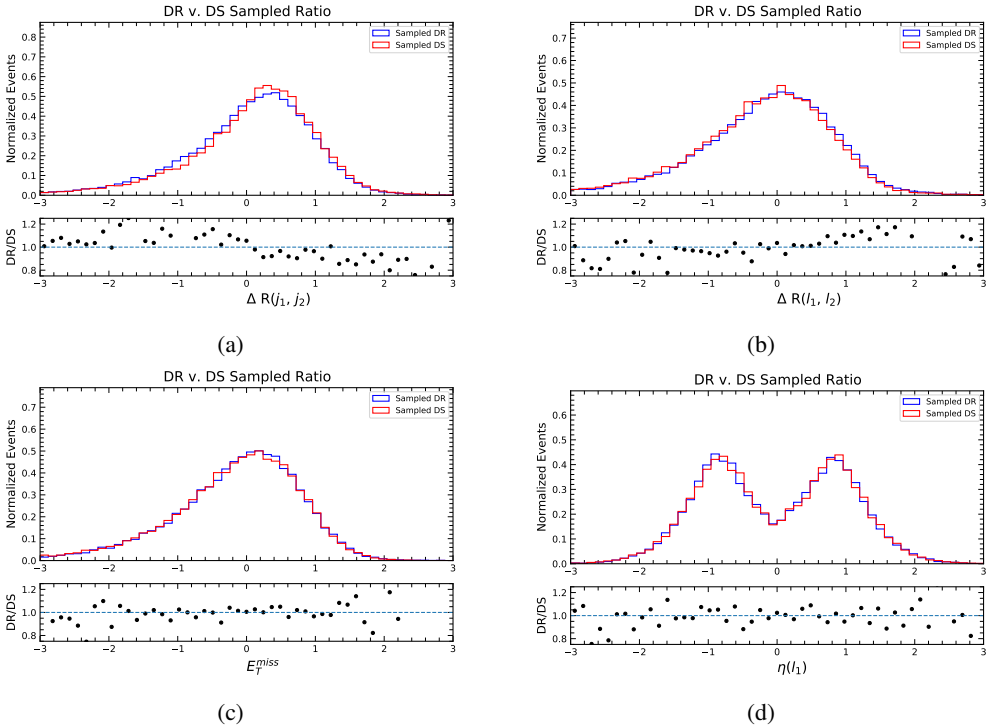


Figure 6.9: Select variables after a one-MADE-layer transformation. Both the DR and DS samples have been put through the same transformation trained on the DR sample. The network is considered successful if the output distribution shown is Gaussian in shape.

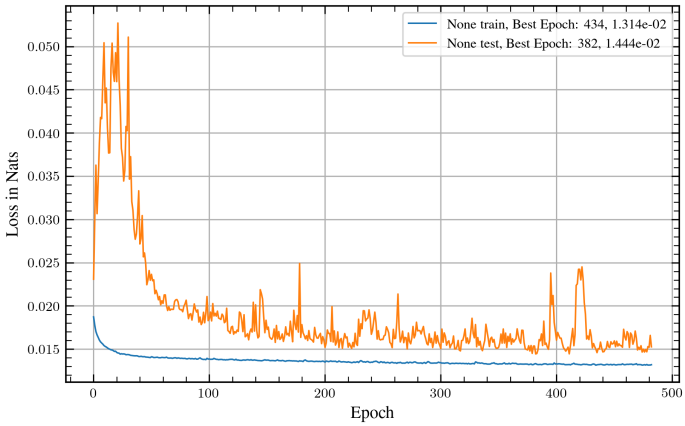


Figure 6.10: Loss of the quadruple MADE layer network. The loss measures the negative log likelihood of input data.

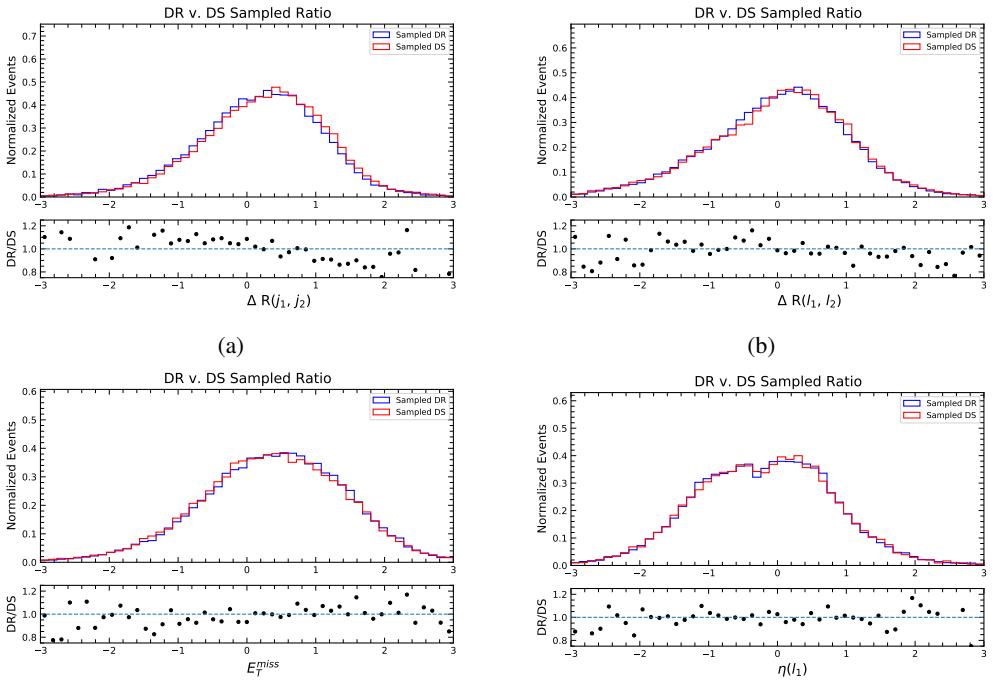


Figure 6.11: Select variables after a four-MADE-layer transformation. Both the DR and DS samples have been put through the same transformation trained on the DR sample. The network is considered successful if the output distribution shown is Gaussian in shape.

To improve the transformations, a hyperparameter tuning can be done including varying the number of layers each MADE network has. However, the differences between DR and DS are not obvious which is the goal of this network. Another improvement is the inclusion of a discriminating feature as part of the conditional of every density.

The output of this network is a series of conditional probabilities in the shape of $p(x_i|x_{<i})$. As was shown in [120], adding a conditional distribution to the flow improves performance of the MAF. The inclusion is easily implemented as by adding the distribution to each layer. In particular, these values need not be masked as each node must receive the information this input. In the end, the new conditional probabilities are in the form of $p(x_i|x_{<i}, m)$ where m is the chosen conditional. Therefore, if the aim of this network is to become sensitive to events with off-shell top-quarks, then the conditional distribution m can be the $m_{b\ell}^{\text{minimax}}$ variable.

An eight-MADE-layer network was trained, each MADE layer had a 6×64 architecture. This model increased in complexity compared to previous models to match the problem's complexity. That is to say, the addition of the conditional distribution caused simpler models to fail. Additionally, the optimizer was changed to Adagrad as the adaptive learning rate helped the network converge.

The performance of the conditional network can be seen in figure 6.12. The loss curve quickly converges for both training and test set and appears to be stable. Figure 6.13 shows the distributions after the transformation. It can be seen that this complex model performs better at normalizing the distributions. However, the DS sample does not differ much from the DR sample regardless of the

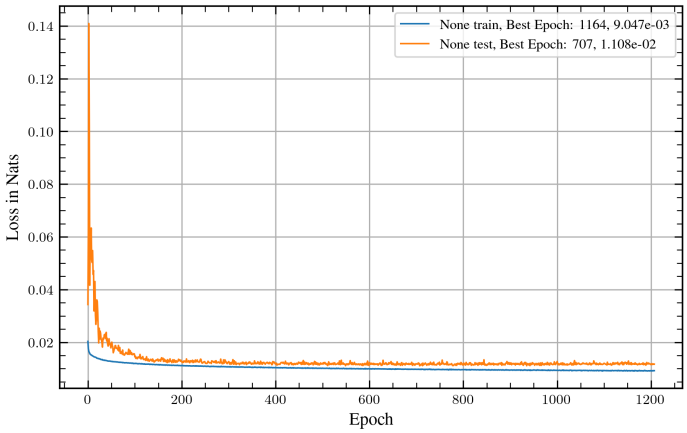


Figure 6.12: Loss of the eight-MADE-layer network with the $m_{b\ell}^{\text{minimax}}$ distribution as conditional. The loss measures the negative log likelihood of input data.

addition of $m_{b\ell}^{\text{minimax}}$ as conditional.

Even by looking at the log likelihood of the events, shown in figure 6.14, one cannot conclude any sensitivity to off-shell events. In this case, one can conclude that this method is not sensitive enough to the differences between these two samples. This can be because the method is not sensitive enough to the subtle differences; in which case, a more sophisticated approach may be necessary.

For such complex networks, two different training methods were tested. Normally, one could train all networks simultaneously but training them in sequence was also possible. In this way, the first block reaches a minimum before being *frozen* and letting the following MADE layer train. A frozen layer means that the weights associated with the layer do not update during training. When training this way, the time needed is significantly longer and therefore limited tests were made. However, in those tests no significant improvement was seen.

ANODE

In section 5.5.5, a technique sensitive to rare anomalies was discussed. As demonstrated in [116], this process is sensitive to anomalies even if they are less than 1 % of the total events. The procedure is to train on two regions; one containing the anomaly, named signal region (SR), and its complement denoted as the sideband (SB). Then, one uses the SB to interpolate into the SR which estimates the background contribution. Specifically, the network learns what should be in the SR from the estimated background and therefore any anomalies should stands out.

The definition of signal and sideband regions must be made such that the desired background is completely encompassed in the sideband. Ideally, one would choose the mass of the most off-shell top-quark. This variable would likely be the most sensitive to interference effects. However, a dilepton final state makes this impossible with two emitted neutrinos. The next possible variable would be $m_{b\ell}^{\text{minimax}}$ as it is an indirect measure of the "off-shell-ness" of a top-quark.

As discussed in section 4.3, higher values of $m_{b\ell}^{\text{minimax}}$ show an increase in interference effect. This is shown in figure 6.15 where the interference effects appear after about 150 GeV; seen from the divergence of red (DR) and green (DS) markers. Therefore, such a value would be the obvious choice

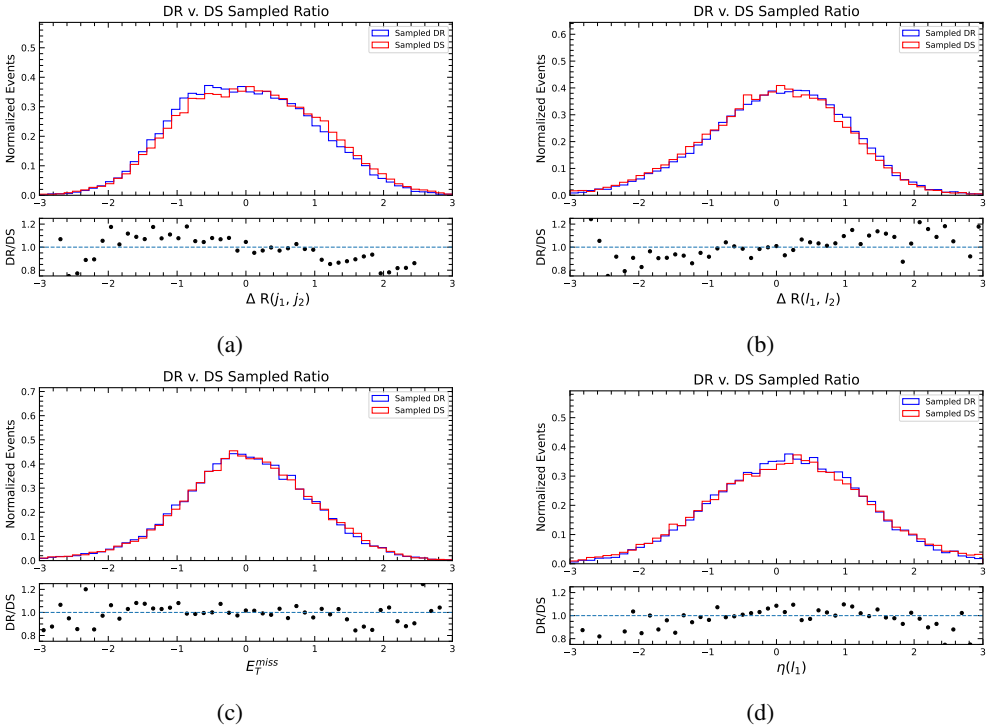


Figure 6.13: Select variables transformed after the eight-MADE-layer network with $m_{b\ell}^{\text{minimax}}$ as conditional distribution m. Both the DR and DS samples have been put through the same transformation trained on the DR sample. The network is considered successful if the output distribution shown is Gaussian in shape.

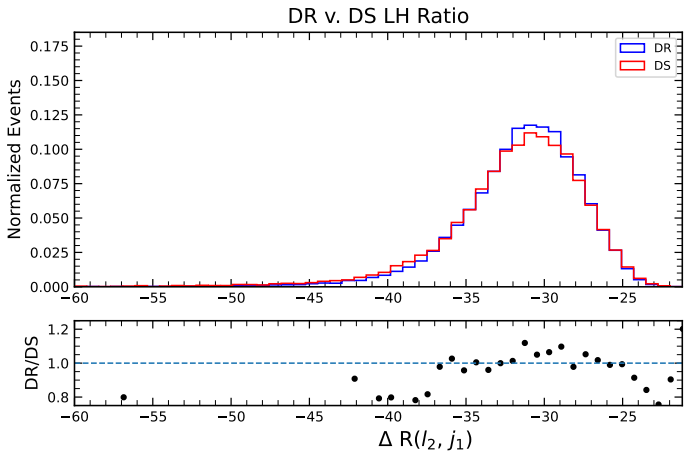


Figure 6.14: Log likelihood of each event in the eight-MADE-layer network with $m_{b\ell}^{\text{minimax}}$ as conditional distribution m.

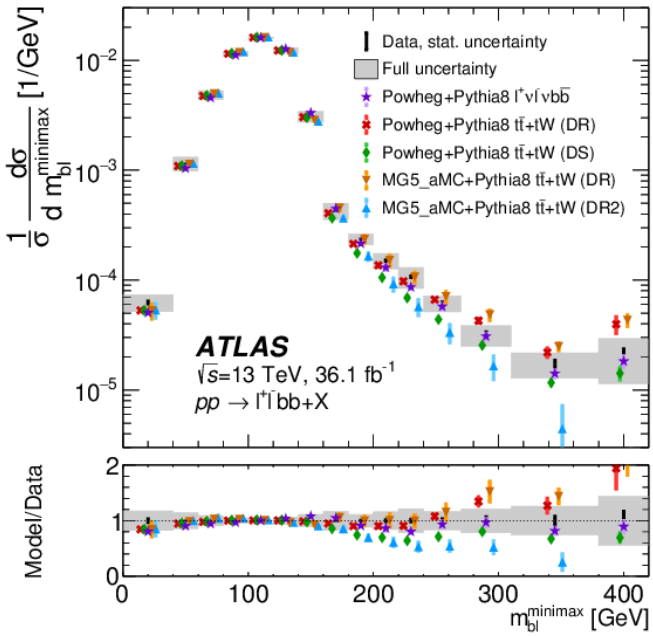


Figure 6.15: Unfolded and normalized differential $m_{b\ell}^{\text{minimax}}$ cross-section [121] with theoretical models for comparison. Different event and parton shower generators are color coded for comparison. The uncertainty includes statistical and systematic sources with more detail found in [121].

for region definition. The next step would be to choose a training set.

Unfortunately, after a cut on 150 GeV on $m_{b\ell}^{\text{minimax}}$ the number of events in the SR would be a small fraction of the total. There would be too few events for the network to correctly estimate the probability density accurately. For this reason, one can train on both the tW DR scheme and $t\bar{t}$ together. This choice comes with two benefits. First, the sample size of both regions increase which help the network performance. Second, the two samples model what are practically the only processes in this region. This means that if this method is satisfactory for MC, it can be directly applicable to data.

Using the same density estimation techniques described in section 5.5.4, two MAFs are chosen as the estimators of each region; similar to [116]. Training and evaluating the MAFs are done in the same way as described in section 6.1. Additionally, the input data goes through the same preparation before being split into the SR and SB regions. Similarly to the previous section, some variables were not easy to estimate and were dropped. Thus, the following is the list of variables used in training:

- ΔR of any two leptons or jets with the exception of the combination of the second lepton and second jet,
- $m_{\text{T}}(j_1, E_{\text{T}}^{\text{miss}})$ and $m_{\text{T}}(j_2, E_{\text{T}}^{\text{miss}})$,
- the ratio of p_{T} and energy of the two lepton system (denoted as *centrality*),
- and the Δp_{T} of the $E_{\text{T}}^{\text{miss}}$ and a system composed of the two leptons and the leading jet.

It is important to remember that these networks are not trying to reconstruct the $m_{b\ell}^{\text{minimax}}$ or any input distributions. These networks are tasked with estimating densities in both regions by performing

a normalizing flow. Therefore, when the output of these MAFs are Gaussian then one can conclude it has succeded in its estimation. For clarity, the extrapolation is done by introducing the SR events to the network which is trained in the SB.

Previous studies, performed by a Masters student in [**Luka2022**], were made using this same technique for this particular problem. The procedure he followed included training on the DS sample instead as it contains the anomaly. One of the issues faced in that work was the fact that the interference is destructive. Therefore, the SR had fewer events interference examples and the SR was mainly $t\bar{t}$ events. In [**Luka2022**], it was claimed that the SR definition was at fault. This work explores the alternative of using DR to validate the region definitions and attempt to see the destructive interference in the LH ratio.

By training on DR, one can check if the extrapolation is comparable. Equation 5.29 shows the ratio of these probabilities which should be equal to one if the extrapolation is sound. Applying the same transformation on the DS sample allows for the comparison of both distributions. It is expected that probability, when DS events are given, should differ in the SR. This difference should still be visible in the ratio of probabilities. By the nature of the interference, the probability of the SR should change such that:

$$p_{\text{data}}(\vec{x}|m) = \alpha p_{\text{background}}(\vec{x}|m) - (1-\alpha)p_{\text{signal}}(\vec{x}|m), \tag{6.2}$$

for $\vec{x}, m \in \text{SR}$. Therefore, the ratio should be less than one for interference events.

Given that there is no reason for both MAFs to be identical, the SR and SB networks are allowed to have differing architectures and hyperparameters. Several degrees of complexity were tested for each model including the amount of MADE layers and the architecture of all MADE layers. Additionally, different $m_{b\ell}^{\text{minimax}}$ cuts were tested as 150 GeV had non-converging SR models. The highest value of this cut where networks were stable was 140 GeV. In the end, the best SB network was composed of 4 MADE layers with a 3×256 architecture each. The SR network was a larger stack of MADE layers which were shallower but denser. This network was made of 6 MADE layers with a 2×720 architecture each. All of the above used `elu` as the activation function. The batch sizes of the SR and SB were 500 and 10 000, respectively. The difference in batch sizes is expected as the SR has few events by comparison to the SB. Although the SB had enough events to justify increasing the batch size, the value was limited by the available memory of the GPU. Both networks' training was optimized using Adam with different learning rates; 5×10^{-4} for the SR and 1×10^{-3} for the SB. Lastly, an L2 regularization term was added with a parameter of 1×10^{-6}.

The loss of both networks can be found in figure 6.16. This plot shows the log likelihood of both regions as well as the training and test samples for both, color coded. Early stopping is implemented which discontinues training if the test loss does not change after 25 epochs; similarly if the loss gets worse. When this stop is triggered, the network goes back to the previous best checkpoint. This accounts for the SB taking fewer epochs to train than the SR.

As this type of network is a normalizing flow, similar plots to section 6.1 are made. These are shown in figures 6.18 and 6.17 for SR and SB, respectively. As before, when the output is Gaussian in shape then the network has learned the conditional distribution. For comparison, the plots contain a green line which is a normal distribution. It can be seen that the SB network was able to transform the input distributions into Gaussian in shape. However as shown in figure 6.17(a), there is a slight offset in the mean of the distribution. This small offset was seen in every network and the shown result is from the best performing network. Figure 6.18 shows that the SR transformation performs in a comparable manner; with a small offset visible.

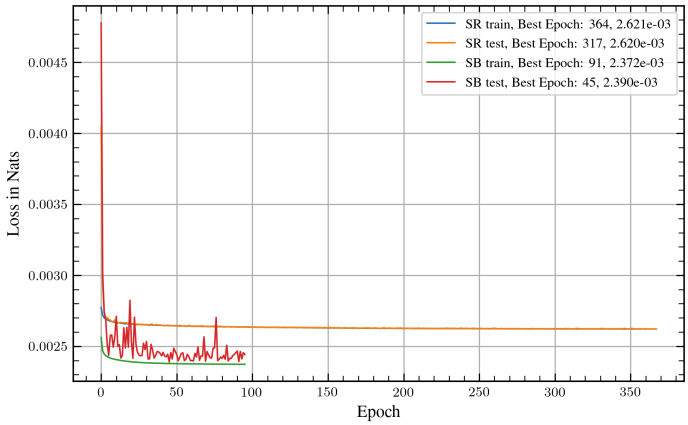
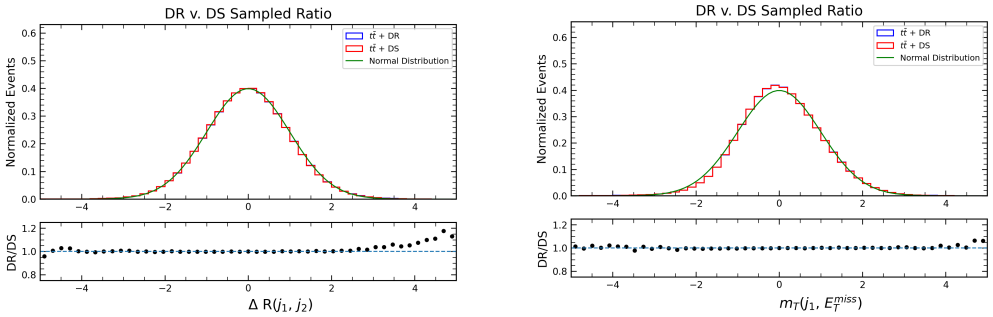
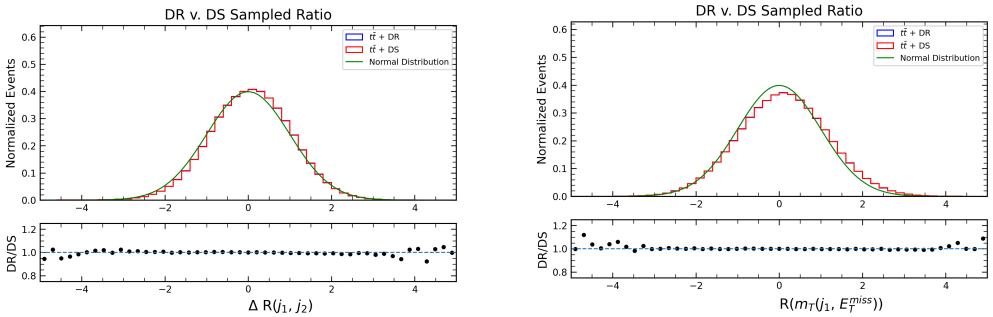


Figure 6.16: Loss curve of both train and test samples for the SR and SB regions. The losses are color coded and their difference in length is attributed to early stopping.



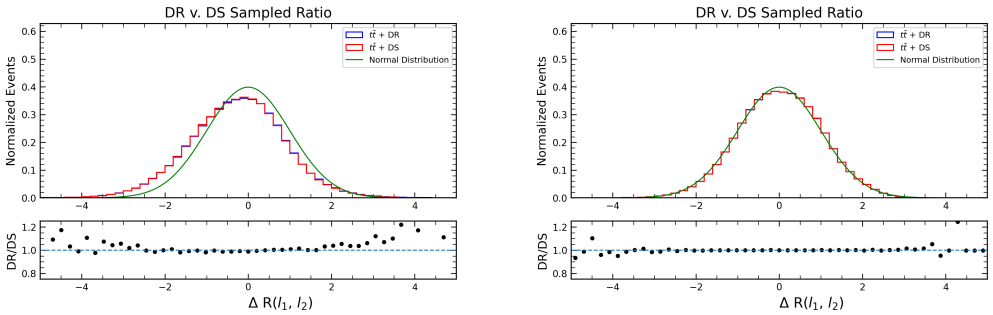
(a) (b)

Figure 6.17: Select SB variables transformed after the ANODE network with $m_{b\ell}^{\text{minimax}}$ as conditional distribution m. The DR and DS distributions include $t\bar{t}$ as this is consistent with the training sample. Included in each figure is a normal Gaussian to aid the reader in comparing shapes. The ratio plot is labeled as DR/DS as the same $t\bar{t}$ sample is added to both.



(a) (b)

Figure 6.18: Select SR variables transformed after the ANODE network with $m_{b\ell}^{\text{minimax}}$ as conditional distribution m. The DR and DS distributions include $t\bar{t}$ as this is consistent with the training sample. Included in each figure is a normal Gaussian to aid the reader in comparing shapes. The ratio plot is labeled as DR/DS as the same $t\bar{t}$ sample is added to both.



(a) (b)

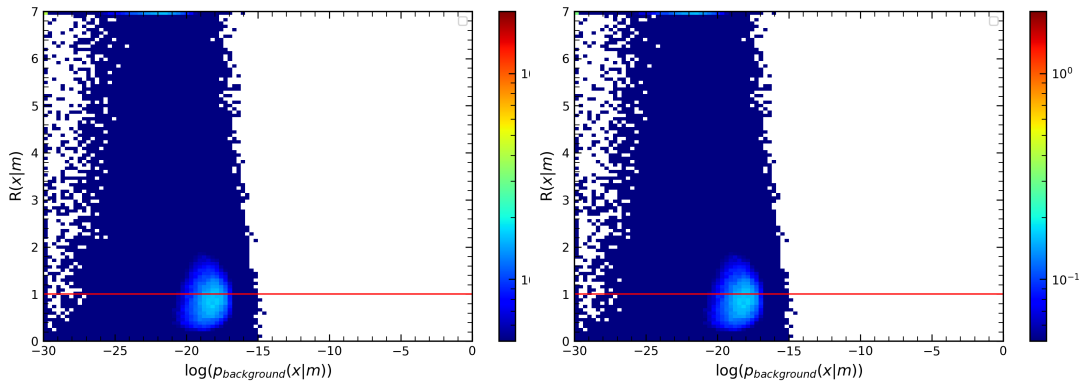
Figure 6.19: Example of the extrapolated distributions in comparison to the estimated SR distribution. The extrapolated distribution (a) is measured by giving the SB network variables in the SR. If the extrapolation is done correctly, then the shape should be similar to the SR estimated distribution (b).

DR and DS do not differ by much in both figures 6.17 and 6.18 with only the extreme fringes deviating from a ratio of one. A conclusion cannot be reached about this difference as it is apparent where statistical effects would dominate. One can say that both the SB and SR networks transform both samples in a practically identical manner. In [**Luka2022**], the student mentioned that potentially the existence of interference events in the SB may be one reason why this method fails. Instead, here the networks are trained in DR yet they both transform both samples equally. Therefore, the conclusion of interference events in the SB is incorrect. Both works likely show that DR and DS are not different enough.

Figure 6.19 compares the normalized distribution given by the SR network with the extrapolated distribution from the SB model. As expected, the similarities between the DR and DS sample are still visible in the extrapolated model. The noteworthy feature is the difference between both models's output shape. Even if the distributions of the signal region differ from that of the sideband, the network should ideally compensate. The sideband network is trained to estimate $p_{\text{SB}}(\vec{x}_{\text{SB}}|m_{\text{SB}})$ and when given SR variables, it should yield $p_{\text{background}}(\vec{x}_{\text{SR}}|m_{\text{SR}})$. However, that is not what is seen.

Since the SR model is also trained in DR, the extrapolated model should output the same (or at least similar) distribution to the SR network. In this way, one can see that the extrapolation is not good enough for comparison. One reason why this extrapolation fails, concluded in [**Luka2022**], is the definition of the regions. Moreover, it is likely the root cause is the one-sided choice of region. In the paper [116], the signal region is a localized region in some distribution. More specifically, if the signal region is designed such that $x \in [m_0 - \delta, m_0 + \delta]$ then the sidebands are the two regions around this bump. Although not explicitly negated, the use of a one-sided SB deprives the network of examples with a complementary set of densities to learn. In simpler terms, it is easier to estimate a small middle range than to estimate a distribution that extends indefinitely with only half of the information. Therefore, two sidebands may be needed for a more robust extrapolation.

With this network, one can construct the ratio of the two probabilities as prescribed. The ratio as a function of log likelihood is shown in figure 6.20. Shown is a comparison between $t\bar{t}$ in addition to DR and DS in figures 6.20(a) and 6.20(b), respectively. It can be seen that the plots are practically identical. One should expect a straight line with some width in the control sample, figure 6.20(a). Instead one sees a blob loosely around $R = 1$ and many errant events at various R values, in both images. This is consistent with the previous statements about the estimation not being good and the



(a) $t\bar{t}$ + DR (b) $t\bar{t}$ + DS

Figure 6.20: Scatter plots of the likelihood ratio R as a function of negative log likelihood in the SR. $t\bar{t}$ + DR (a) is shown as the control sample which should have most events populating the $R = 1$ line. In contrast, $t\bar{t}$ + DS (b) should have most events along $R = 1$ with anomalous events straying away from this line. A red line along $R = 1$ is drawn for visibility and the overflow is plotted at the limits of the plot.

extrapolation being too different from the estimated SR.

Even after realizing the problems with the initial assumptions, one could conclude the approach may not be right for this problem. After all, the task of the network is to essentially classify events which are nearly identical to the training set. That is to say, the kinematic difference between a tW and $t\bar{t}$ event and their interference may not be enough for this type of network. Furthermore, the interference is destructive. This method was meant to be used in bump hunts and in this case the bump disappears. One could remove the weights of the MC when training but this means the network is no longer directly applicable to data. After all, weights of MC are introduced to correct the shapes of distributions generated to match data. If this case worked, the network could learn this distinction only in simulation.

Given the previous reasons, this trying to create a variable sensitive to the interference between tW and $t\bar{t}$ was no longer pursued. With the benefit of hindsight, the reasons of failure could have been visible from the beginning. However, valuable experience was gained. This conclusion should not discourage a reader from attempting such networks in their own projects. After all, density estimators like ANODE has been used in [122], mixture density models have been used in [123, 124], and CATHODE was used in [125].

CHAPTER 7

Poorly Modeled Backgrounds

On another topic, hadronically decaying tau (τ_{had}) leptons are difficult to reconstruct and identify. A tool exists [30, 79] which identifies τ_{had} leptons by feeding various variables to an RNN. This tool is general and applicable to many different regions. However, by focusing on the kinematics of a specific region, it may be possible to gain separation power. The strategies and procedures employed for this purpose are detailed in section 7.1.

7.1 Hadronic τ Leptons

The RNN used in ATLAS for τ_{had} identification is a general network which focuses on signal ID with no training of background sources. It is a robust strategy which boasts precision in the range of 2 to 20 % depending on the kinematics of the tau candidate. The medium and tight working point, shown in table 7.1, are shown to have a background rejection rate which improved upon the BDT [78] previously used. This table also shows that as working points increase in tightness, the rejection rate increases at a greater rate than the loss of signal efficiency. Even with such a rejection rate, this comes at the cost of 25 and 40 % of the signal for medium and tight, respectively.

	Signal efficiency [%]		Background rejection	
Working Point	1-prong	3-prong	1-prong	3-prong
Tight	60	45	70	700
Medium	75	60	35	240
Loose	85	75	21	90
Very loose	95	95	9.9	16

Table 7.1: Defined working points for the τ_{had} ID RNN with corresponding efficiencies and background rejection. [30] It can be seen that tighter working points yield a better background rejection.

In the following sections, a few strategies are explored with the purpose of improving signal efficiency while retaining background rejection. The models employed are specific in kinematics as they are intended to be trained in a tHq analysis selection or analogous. Three strategies are attempted with different architectures and training set definitions. The first strategy was similar in strategy to the RNN but employing autoencoders. The second strategy uses analogous regions to train data against

data in a weak supervision model, CWoLa. Lastly, a region of data with low quantities of real τ_{had} is trained against MC where the τ_{had} is known to be true.

All strategies employed for the purpose of identifying τ_{had} leptons used a custom package wrapping around Tensorflow [126]. The package was written to streamline analyses such that they do not always need to write their own neural networks. For this reason the package is controlled by a configuration file which creates different types of networks and handles different tasks. More information on this package can be found in appendix A.

Autoencoders

The strategy of using an autoencoder on true τ_{had} is similar to that of the RNN. The purpose of this method is to focus from the general training to the specific kinematic region of the analysis. However, the training sample contains only true τ_{had} from simulation in the $2\ell + 1\tau_{\text{had}}$ OS region of the tHq analysis. As the autoencoder compresses and decompresses information, events which are anomalous should stand out from the norm. In this case, a loss function such as the Mean Square Error (MSE) is used to show the sensitivity to anomalous events and to train the network. Additionally, comparing the reconstructed variables can be used to determine the performance of the network.

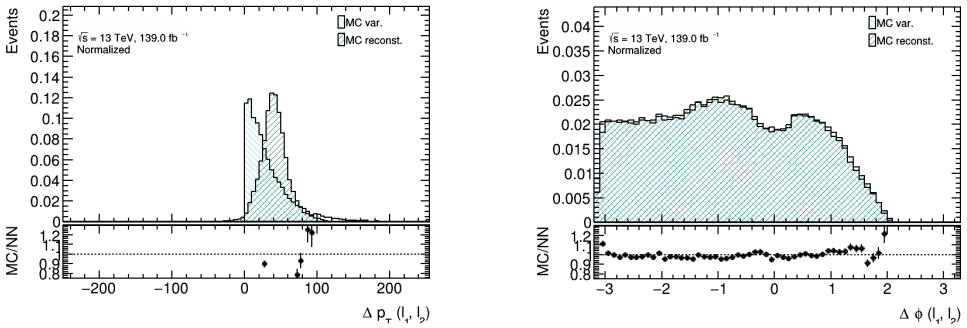
To begin, the goal of this network is simple: learn what a true τ_{had} event looks like; imposing the requirements that the existing RNN identification is not included in the training. Variables one could use for training are of different order of magnitudes which implies the requirement of scaling. However, the autoencoder is desirable as it is intended to learn the distributions of its input. Scaling is still done but with `scikit-learn`'s [127, 128] `MinMaxScaler` module. This scaler transforms each feature to fit in the range between 0 and 1 but retaining its shape.

Training events must fulfill certain criteria:

- $p_{\text{T}}(\ell_3) > 14\,\text{GeV}$,
- light leptons must be of opposite sign and tight,
- $N_{\text{jets}} > 2$,
- $N_{b-\text{jets}} = 1$,
- $N_{\text{prong}} = 1$,
- event must have a true tau.

To ensure that the tau candidate is a true tau, the *PDG ID* of the object is checked. This is a number assigned by the PDG [7] to form a common numeration of MC particles. The number assigned to τ leptons is 17 with the negative implying the antiparticle. The selection of b-jets is exclusive as different multiplicities have other background sources and should be treated accordingly. Lastly, this selection is further split by the number of prongs of the tau candidate. Similar to b-jet multiplicity, different backgrounds mimic a τ_{had} lepton signature depending on this condition. A one-prong selection is chosen and discussed in this section as it was used to test the strategy before expanding to other kinematic regions.

The variables used in the network must be chosen such that they can be reconstructed well by an autoencoder in addition to being different for real and fake τ_{had}. From experimentation, distributions which had a hard limit were difficult for the network to recreate. This is shown in figure 7.1(a) where



(a) $\Delta p_{\mathrm{T}}(\ell_1, \ell_2)$ (b) $\Delta\phi(\ell_1, \ell_2)$

Figure 7.1: Distributions of kinematic variables and their reconstructed distribution for comparison. These distributions involving the two light leptons show how good the reconstruction is by the autoencoder. It can be seen that the $\Delta p_{\mathrm{T}}(\ell_1, \ell_2)$ (a) is not a well reconstructed as it has a hard limit. However, $\Delta\phi(\ell_1, \ell_2)$ (b) has no such limits and is reconstructed with acceptable success.

the $\Delta p_{\mathrm{T}}(\ell_1, \ell_2)$ fails to be reconstructed. For comparison, figure 7.1(b) shows that the $\Delta\phi(\ell_1, \ell_2)$ distribution was well reconstructed. One possible way to deal with hard limit variables is to encode or include information which changes the distribution but retains information. For example, the ϕ distribution of the missing transverse energy is isotropic but has negative and positive values. In practice, all tested architectures of the autoencoder failed to reconstruct composite variables like these. After training, one can calculate the reconstruction error by using the MSE of output variables against their original distribution. In this way, anomalous events should have a higher reconstruction loss and be identified.

The following variables were used in the autoencoder:

- $\Delta\phi(\ell_{\mathrm{light1}}, \ell_{\mathrm{light2}})$, difference in the ϕ direction between the two light leptons.
- $\Delta R(\ell_{\mathrm{light1}}, \ell_{\mathrm{light2}})$, $\phi - \eta$ distance between the leading lepton and the sub-leading lepton.
- $\Delta R_{\mathrm{max.}(\tau_{\mathrm{had}}, j)}$, angular distance between the τ_{had} and the furthest jet.
- $\eta(\tau_{\mathrm{had}})$.
- JetCaloWidth(τ_{had}), the width of the jet associated with the τ_{had} as deposited in the calorimeter.
- $\Delta\phi(\tau_{\mathrm{had}}, \ell_{\mathrm{light1}})$, angular difference in the transverse plane between the τ_{had} and the leading light lepton.
- $\Delta\phi(\tau_{\mathrm{had}}, \ell_{\mathrm{light2}})$, angular difference in the transverse plane between the τ_{had} and the sub-leading light lepton.
- $\Delta\eta(\tau_{\mathrm{had}}, \ell_{\mathrm{light1}})$, difference in the pseudorapidity between the τ_{had} and the leading light lepton.
- $\Delta\eta(\tau_{\mathrm{had}}, \ell_{\mathrm{light2}})$, difference in the pseudorapidity between the τ_{had} and the sub-leading light lepton.

More distributions can be found in appendix B. The formula used to gauge the reconstruction error is as follows:

$$\text{Err.(total)} \stackrel{\text{def}}{=} \frac{1}{N_{\text{feats.}}} \sqrt{\sum_{i}^{\text{feats.}} (X_i - \hat{X}_i)^2}, \tag{7.1}$$

where X_i denote a feature and $\hat{X}_i$ is the estimate given by the autoencoder. The squared difference of each feature (contracted to feats.) is added together and contributes to the total error. Other variables were tested as well, but no autoencoder configuration was able to correctly reconstruct them; thus they were dropped from the training set. Examples of such variables were the Lorentz invariant inner product of two objects in the event, the minimum distance between a jet and each of the leptons, and kinematics of other objects.

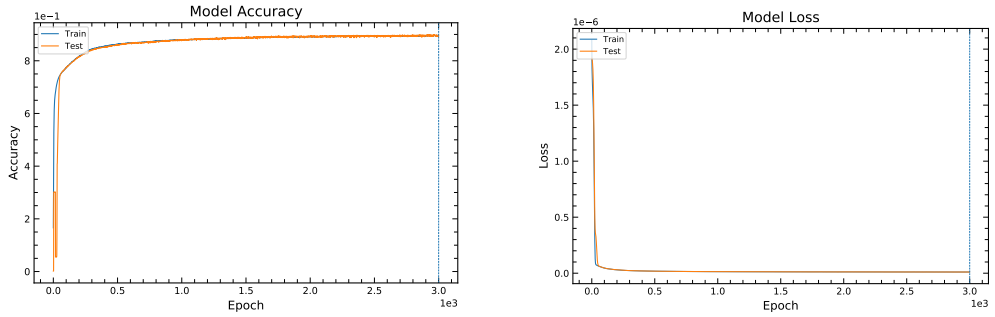
The network's hyperparameters are found by performing a grid search. The depth of the network is varied between two to five hidden layers in both encoder and decoder; independent of each other. The density of each layer varied between 16 to 64 nodes per layer increasing by 8 nodes. Additionally, the number of nodes was allowed to vary between hidden layers with the closest to the hidden representation having the fewest nodes. The activation functions used also varied with different degrees of failure between them. These functions were also allowed to differ between the layers such that any two layers of the encoder may not have the same function. Similarly, the architecture of the decoder was allowed to vary independently of the encoder instead of always being the reverse. Lastly, the number of elements in the hidden representation was allowed to vary between 4 to 8 elements.

Hyperparameters like the dropout, optimizer, and L1/2 regularizers were untouched. Dropout and L1/2 regularization were not used in the autoencoder as they held little meaning for this task. Overtraining was not seen, requiring no introduction of dropout layers. An autoencoder with a hidden dimension larger or equal to the number of inputs runs the risk the network learning to copy the input directly. In such a network, a regularizer is useful to turn the network into a sparse autoencoder that learns features rather than copying. As the number of hidden elements is lower by design, the sparsity is achieved without the use of regularization. The optimizer chosen was `Adam` with default values as it performed adequately in the chosen optimized network.

After varying the hyperparameters, the optimal network was found. The encoder is composed of three hidden layers containing 48, 32, 32, 32 and 32 nodes per layer. Each of these nodes has the `ReLU` activation function. The encoder maps the input to a latent dimension containing 8 elements as anything under that impacted the accuracy significantly. The decoder takes the latent dimension as input and propagates it through three layers with 32, 32, 32 and 48 nodes per layer. These layers have the same activation functions as the encoder (`ReLU`) except for the last layer which has a linear activation function. Both the decoder and encoder have batch normalization layers in between each dense layer. The autoencoder is trained on 70 % (84 078) of events from the tZq MC sample which pass the selection for 5 000 epochs. The network is tested on the remaining 30 % to check for overtraining.

A tZq sample is used here instead of tHq as it contains more events and is full sim. Furthermore, the both tZq and tHq have similar kinematics. As this network is intended to correctly identify true τ_{had} events, it should perform similarly regardless of the input sample. Therefore, training this network and applying it on other samples like tHq should yield similar results.

Metrics such as the loss and accuracy of the network can be seen in figure 7.2. It can be seen that the network converges early and consistently improves. Figure 7.2(b) shows the loss of the training and test sample remain consistent throughout, implying no overtraining. As the loss is the MSE,



(a) Accuracy

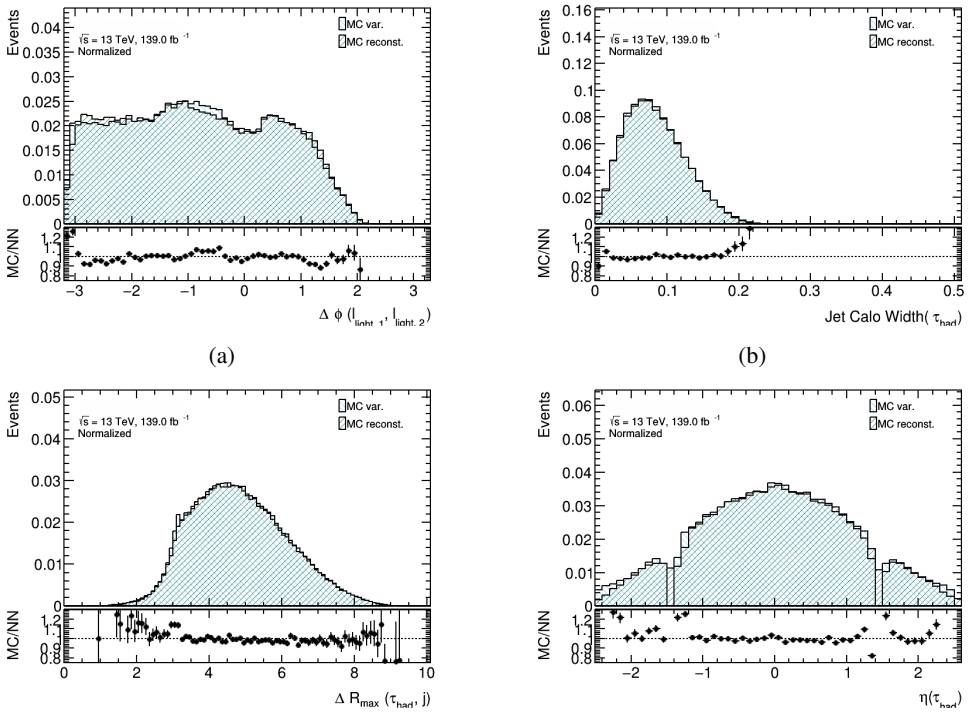
(b) Loss (MSE)

Figure 7.2: Metrics of the autoencoder tasked with identifying τ_{had}. The loss and accuracy are calculated for the autoencoder every epoch as it trains. The accuracy is not a well-defined value in an autoencoder but is shown for completeness.

lower values imply a more accurate reconstruction and therefore are desired. However, Keras also returns an accuracy value shown in figure 7.2(a). The final accuracy estimated by the network reached nearly 89.5 %. It should be noted that for regression tasks such as this, the accuracy metric is not well-defined. For this reason, the accuracy were not considered as much as the loss and response plots for evaluating the performance of the network. Using the network described, figures 7.1 and 7.3 are created to compare the input with its reconstructed counterpart. The variables shown in figure 7.3 were among the six best reconstructed variables.

To test the sensitivity of the reconstruction error, the network is applied to a different process and predictions are generated. As $t\bar{t}$ is a source of both true and many fake τ_{had}, it is the chosen candidate for testing. Additionally, the fact that it is simulation allows one to split the sample into true tau leptons and fakes. True tau leptons are selected with the same signal cut chosen for the training. By flipping the identification requirement, fake tau leptons can also be correctly selected from the MC sample. One can calculate the mean squared error for both regions, shown in figure 7.4, and compare the reconstruction for both. The average error calculated with all of the input of the network is shown in figure 7.4(a). It can be seen that the two reconstruction error curves differ in the expected behavior but a clean distinction is not present. Since the distribution is considered insufficiently sensitive, the number of variables used for the calculation was restricted. The variables used to calculate the error shown in figure 7.4(b) were the top six: $\Delta\phi(\ell_{\text{light1}}, \ell_{\text{light2}})$, $\Delta R_{\text{max.}(\tau_{\text{had}},j)}$, $\eta(\tau_{\text{had}})$, JetCaloWidth(τ_{had}), $\Delta\phi(\tau_{\text{had}}, \ell_{\text{light1}})$, $\Delta\phi(\tau_{\text{had}}, \ell_{\text{light2}})$. This plot shows a similar trend but with overall lower reconstruction loss which is expected.

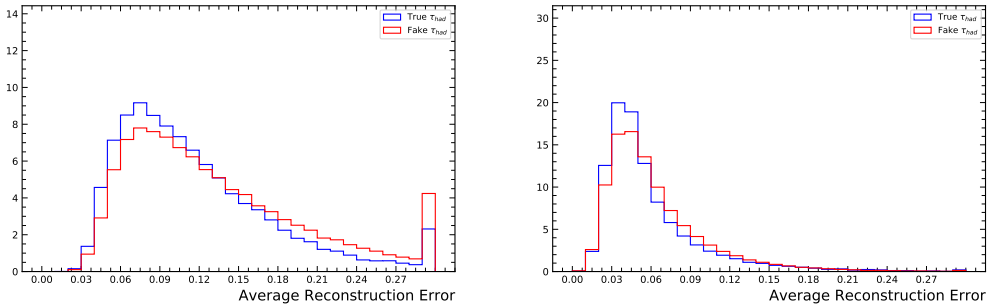
By cutting on the reconstruction error from figure 7.4(a) or figure 7.4(b), one can calculate signal efficiencies and background rejection values. This is done to compare to table 7.1 using the same calculation for background rejection outlined in [30]. Given the similarities in the reconstruction error of both fake and real τ_{had}, any cut on this value would yield similar efficiencies for both classes. Table 7.2 shows the calculated background rejection from the same signal efficiency points given in table 7.1. It can be seen that the rejection from the RNN is at least nine times higher. Even by only selecting the best reconstructed variables for the calculation, the performance remains under the RNN. Similarly, the 3-prong autoencoder does not provide enough background rejection to compete. Therefore, one can conclude that this method does not provide a competing classification power. Given the difficulties



(a) (b)

(c) (d)

Figure 7.3: Well reconstructed distributions by the autoencoder designed to identify τ_{had}. All these plots are of the *tZq* MC; the training sample. For the complete set of reconstructed *tZq* variables, see appendix B.



(a) All Reconstructed Variables (b) Best Six Reconstructed Variables

Figure 7.4: Average reconstruction error of the $t\bar{t}$ MC sample with true and fake τ_{had} selected. The plots shown are normalized such that their distributions are directly comparable. This is done because the amount of true τ_{had} in this sample are a small fraction of the whole.

and the low sensitivity achieved by this method, the strategy was dropped for a weak classifier.

Signal efficiency [%]	Background rejection (RNN)	Background rejection (Autoencoder)
60	70	1.9
75	35	1.5
85	21	1.3
95	9.9	1.1

Table 7.2: Comparison of defined working with corresponding efficiencies and background rejection in the 1-prong selection. RNN background rejection values are given in [30]. The values are derived from the total reconstruction error from all reconstructed features.

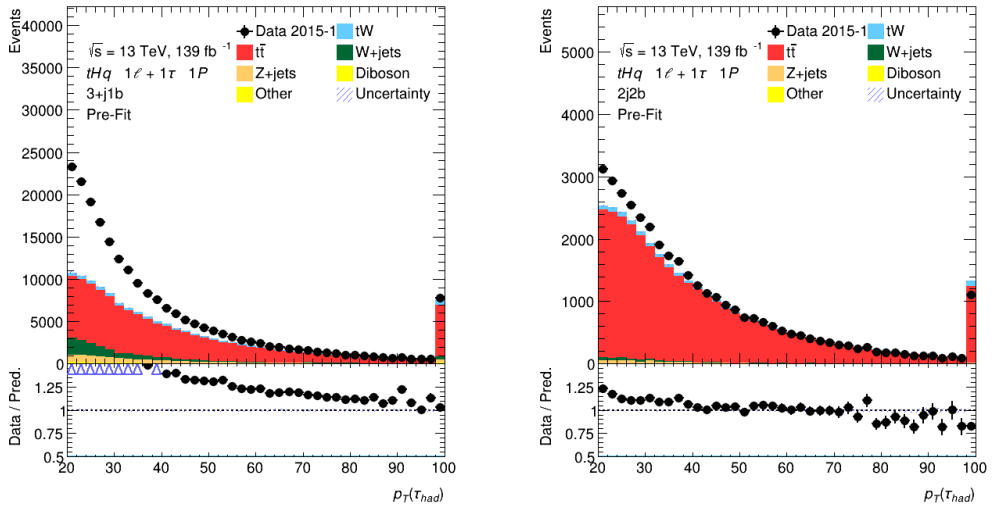
A Weak Classifier Using Surrogate Regions

The strategy tested in this section involves training a classifier in a kinematic region similar to the signal region but orthogonal to it. Additionally, the network is given information of both signal and backgrounds rather than only signal. As data contains the desired background, fully supervised techniques are unlikely to be used. However, strategies discussed in 5.4, such as CWoLa or LLP, can be used to train on two, non-simulation sets with mixed purities. In particular, the CWoLa technique requires no prior knowledge of purity fractions in the data. Therefore, the choice of training set would not be constrained by knowledge of its composition. Either technique comes with the caveat that higher statistics improve the performance of the model.

The definition of training regions poses the first and most difficult challenge. Because although the tau lepton remains itself regardless of where it originates, its kinematics are not guaranteed to be the same in all regions. Therefore, the chosen training set should be a surrogate of the desired signal region, $2\ell + 1\tau_{\text{had}}$. Should a trained network be applied to a region where input variables differ, it would not correctly classify events. Additionally, it should provide enough separation power for a network to succeed in its task. The background composition should be similar such that the network learns what the correct background is. Lastly to employ a technique like CWoLa, the training set should be divisible such that one can give it labels.

With these points in mind, the chosen candidate as an analogue for a $2\ell+1\tau_{\text{had}}$ region is $1\ell+1\tau_{\text{had}}$. In both selections, one has a hadronic tau lepton which is the desired particle to identify. Both selections also have a large $t\bar{t}$ contribution. However, the $2\ell + 1\tau_{\text{had}}$ region has a significant contribution from Z+jets events; about half that of $t\bar{t}$. In the $1\ell + 1\tau_{\text{had}}$ region, this process is suppressed as the lepton flavor differs. It is possible that the Z boson decays into two tau leptons where one decays leptonically and the other hadronically. Though this is less likely than a Z boson decaying into two light leptons of the same flavor. Even though the modeled background ratios may differ, the processes which fake the τ_{had} should be the same between these two regions.

The $1\ell+1\tau_{\text{had}}$ selection can be further divided into jet multiplicities which have different kinematics and background compositions. In particular, regions with adjacent or similar jet multiplicities to the $2\ell + 1\tau_{\text{had}}$ signal region are considered. The estimated event yields of various regions of $1\ell + 1\tau_{\text{had}}$ can be seen in table 7.3. Yields are given without error as no declaration of true yields is made with this table. The main purpose of this table is to show an estimate of the ratio of simulation to data, interpreted as the poorly modeled background. An estimate of error on the yields could be made



(a) 3+j1b (b) 2j2b

Figure 7.5: Analogue regions containing different ratios of real and fake τ_{had}. It can be seen that the agreement between data and MC is greater for the `2j2b` region. The gap in both images can be safely assumed to be composed primarily of fake τ_{had}.

by the inclusion of systematic uncertainties and performing a fit but that is beyond the scope of the current task.

To give more detail, the table has labeled regions in the format "XjYb". This format implies that the event has a total of "X" jets of which "Y" are b-tagged. Therefore the 2j2b region signifies events where there are only two jets which are both b-tagged. Additionally, a plus sign signifies an inclusive selection. The region 3+j1b has at least three jets of which one must be b-tagged.

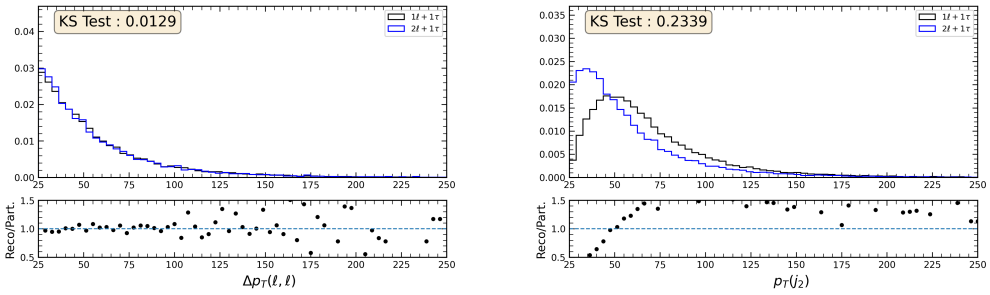
From these regions, two can be picked such that their ratio of fake to real τ_{had} differs enough for a NN to learn. Using the region named 2j2b would be ideal for true τ_{had} as over 90 % of events are modeled. However, the source of background events is harder to choose from as the lack of modeling is not as pure as desired. It may still be usable as, according to [109], higher available statistics can account for lower purity fractions. This region has over 215 000 events which may be sufficient for such a low purity sample. The possibility still exists as the lack of purity is offset by the availability of statistics, shown in figure 5.10. Therefore, the choice of fake τ_{had} can be the region noted as 3+j1b. This selection on jet multiplicity is identical to that of the $2\ell + 1\tau_{\text{had}}$ signal region with at least three jets and one b-jet. Both of these regions are shown in figure 7.5.

As mentioned earlier, the composition of these regions is not identical to the $2\ell + 1\tau_{\text{had}}$ signal region. After all, there is a lack of Z + jets events. However, both have a large fraction of $t\bar{t}$ events which may contribute significantly to fake τ_{had} events. And both selections are still susceptible to QCD processes which can fake τ_{had} leptons. With highly populated regions chosen, the next step is considering which variables one can use for training. In brief, the training variables must be similar to the SR or the network would not be applicable.

In order to choose variables to be used in training, a comparison to the $2\ell + 1\tau_{\text{had}}$ region (SR) must be made. The main comparison is made between the data with similar selection to the SR. This means that the impure sample of 3+j1b is compared to the same selection in the $2\ell + 1\tau_{\text{had}}$ region. The

	4j2b	3j2b	2j2b	4j1b	3j1b	2j1b	3+j1b
tW	798.0	1 256.7	1 153.0	1 823.9	4 119.5	7 014.6	6 877.8
$t\bar{t}$	24 494.6	36 592.7	32 178.6	32 372.3	53 408.6	58 249.0	107 759.0
W + jets	277.3	467.7	354.5	3 365.1	7 465.6	12 638.7	12 935.8
Z + jets	363.8	609.4	544.3	2 536.6	5 339.5	9 797.5	9 343.5
Diboson	32.5	52.5	42.4	351.1	564.0	826.2	1 161.7
Other	141.7	89.4	33.7	229.3	207.2	160.7	781.6
Total	26 107.8	39 068.4	34 306.5	40 678.3	71 104.3	88 686.6	138 860
Data 2015-18	39 687	49 906	36 794	66 431	100 889	112 722	217 855
MC/Data	0.658	0.783	0.932	0.612	0.705	0.787	0.637

Table 7.3: Estimated event yields of the $1\ell + 1\tau_{\text{had}}$ channel. The common selection applied to each of these is that the two leptons (τ_{had} and light lepton) are of opposite charge, the light lepton is tight, and the τ_{had} is one-pronged. The only difference between each region is the jet and b-jet multiplicity. It should be noted that "3+j" implies at least three jets in the event and therefore inclusive in higher jet multiplicity. The category "other" implies rare processes like tZq, tHq, $t\bar{t}$ plus a vector boson, etc. Errors in yields are omitted as these values are not meant to make any claim other than loosely estimate the MC-to-data ratio.



(a) $\Delta p_{\mathrm{T}}(\ell_1, \ell_2)$ (b) $p_{\mathrm{T}}(j_2)$

Figure 7.6: Kinematic comparison from data between $2\ell + 1\tau_{\mathrm{had}}$ and $1\ell + 1\tau_{\mathrm{had}}$ with similar jet multiplicity requirements. The images show two variables, $\Delta\eta$ of the most forward light jet and its closest lepton (a) and the p_{T} of the second jet (b). A ratio plot is provided under each main plot to enhance the similarities or differences in the images. Furthermore, a Kolmogorov-Smirnov (KS) test score is given which is used as a metric of difference. A lower score in the KS test implies similarity between two distributions.

comparison can be seen in figure 7.6 where two distributions are shown. The distribution shown in 7.6(a) is an example of a variable which both regions have nearly identical distributions. Not only are the ratio plots available for comparison, but a Kolmogorov-Smirnov (KS) test score is used as a metric of similarity. Comparing the two datasets in figure 7.6(b) show that this variable is not usable in training. One should note, a lower score in the KS test implies similarity between the two distributions and is therefore desirable.

The KS test is used here as it is sensitive to distributions which may have similar sample mean and standard deviation but are inherently different. From the KS test, an easy to digest P-value can be calculated which is related to the confidence level of similarity. The metric used is an intuitive value with higher values implying increasingly identical distributions. In familiar terms, a P-value greater than 0.15 implies less than 1σ significance in difference. Similarly, a P-value of 0.05 is the 2σ limit and 1.3×10^{-3} is the 3σ point.

Table 7.4 shows the most similar distributions. One thing which can be easily seen is the abundance of ϕ distributions. This is unsurprising as the processes measured tend to be isotropic in ϕ. For that same reason, these distributions provide no separation power and should be ignored in training. Furthermore, some of these variables are reliable on multiple jets which would be unavailable in the 2j2b sample. Therefore, events which include the third or higher jet should be ignored. In the end, the list of potentially useful variables drop to only seven. This list is composed of the pseudorapidity of both (leading) leptons and angular distances between leptons and forward jets or b-jets.

As the samples have unreliable labels, metrics such as accuracy and the ROC are not useful. For this method, one should look at the response plot and predict on simulation. Should the strategy succeed, the MC with true τ_{had} should be in one side of the distribution and poorly modeled background in the other. With the abundance of true τ_{had} seen in the 2j2b region, it makes sense to set this sample to be background. This makes the less pure fake τ_{had} sample of 3+j1b the signal sample. The choice of signal and background is almost irrelevant as the reverse would be the reverse classifier.

Using a similar grid search method as explained in section 7.1, several architectures were tried. The grid search done for this strategy included the modification of dropout, which was allowed to vary between 0 and 50 % in steps of 5 %. Regardless of architecture, the networks failed to converge

Variable	KS score	P-value
$\phi(j_4)$	4.31×10^{-3}	0.92
$\phi(\tau_{\text{had}})$	4.69×10^{-3}	0.86
$\phi(E_{\text{T}}^{\text{miss}})$	5.53×10^{-3}	0.69
$\phi(j_1)$	6.23×10^{-3}	0.54
$\eta(\ell_1)$	6.34×10^{-3}	0.52
$\phi(\ell_2)$	6.48×10^{-3}	0.50
$\phi(\ell_1)$	7.01×10^{-3}	0.39
$\phi(j_3)$	7.85×10^{-3}	0.26
$\phi(j_2)$	9.82×10^{-3}	0.084
$\Delta\phi(\ell_2, j_b)_{\text{min}}$	1.14×10^{-2}	0.029
$\Delta\phi(j_f, \ell)_{\text{min}}$	1.15×10^{-2}	0.027
$\phi(j_5)$	1.17×10^{-2}	0.022
$\eta(\ell_2)$	1.31×10^{-2}	7.13×10^{-3}
$\Delta\phi(\ell_1, j)_{\text{min}}$	1.32×10^{-2}	6.10×10^{-3}
$\Delta\eta(\ell_1, j)_{\text{min}}$	1.37×10^{-2}	4.07×10^{-3}
$\Delta p_{\text{T}}(\ell, \ell)$	1.50×10^{-2}	1.27×10^{-3}

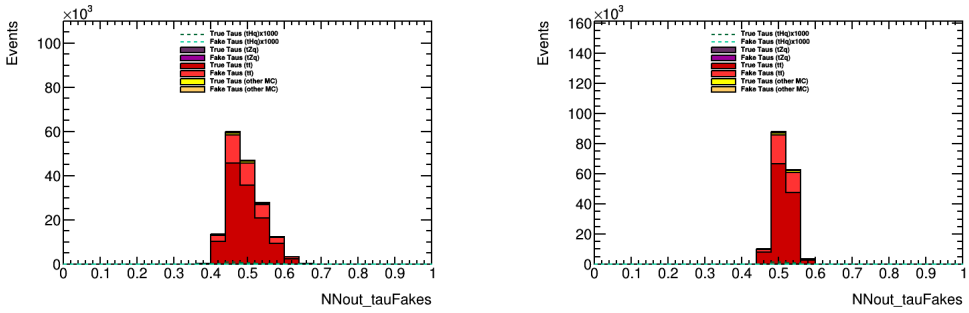
Table 7.4: Variables which have similar distributions in $2\ell + 1\tau_{\text{had}}$ and $1\ell + 1\tau_{\text{had}}$. The KS score and P-value are provided as metrics of similarity between these two regions.

to a reasonable classifier. Notably, using the `Adam` optimizer lead the network to instantly learn the differences between kinematic regions. This is implied by the perfect separation of signal and background in the response plot. If the network had perfectly learned to classify τ_{had}, the samples would have some fraction distributed at both ends; an unrealistic scenario regardless. A switch to using a stochastic gradient descent was made which failed to converge after 5 000 epochs.

Given that using CWoLa had failed, an attempt was made using the LLP technique. As a refresher, this means the samples and training are treated identically but the loss function is modified. Shown in equation 5.12, the used loss function can be seen which includes the information about the fraction of purity. This fractions used were from the estimate given in table 7.3.

Unfortunately, the performance of both techniques was poor. As no network performed the task, this strategy was not pursued further. For completeness, two networks were chosen to demonstrate the inability to classify. One network used the CWoLa technique and the other used LLP. The response on the $1\ell + 1\tau_{\text{had}}$3+j1b region is shown in figure 7.7. Here, it can be seen that neither network was able to separate signal from background. Both fake and real τ_{had} are classified similarly and nothing is put in either extreme. Given this result, application to the $2\ell + 1\tau_{\text{had}}$ region is meaningless. However, there is always something to learn from failure.

With the hindsight of experience, it is easy to see why this approach failed. Naïvely, the assumption that the two regions chosen could correctly exemplify the problem was not right. Additionally, the information one could gain from the chosen seven variables was severely limiting. Even if the previous problems were not enough, there was no guarantee that the fake τ_{had} background had the same profile in $1\ell + 1\tau_{\text{had}}$ and $2\ell + 1\tau_{\text{had}}$. Likely, the contribution of fake τ_{had} in the two regions differ as much as



(a) CWoLa (b) LLP

Figure 7.7: NN response on the 3+j1b region of the $1\ell + 1\tau_{\text{had}}$. The tHq contribution is increased one-thousand-fold for visibility.

the modeled backgrounds. Without proper samples and enough information, the networks would not be able to succeed. With the knowledge of why this failed, a new strategy is developed to address these issues.

Weak Supervision on Loose Data

Techniques like CWoLa and LLP are limited by the statistics and samples one uses to train. That is to say, the training samples must be well populated and as close to the desired application as possible. In the previous section, the statistics were high but the regions chosen were too distant to the intended usage. Therefore, one reassesses the assets one has and develops a new strategy.

One of the failures of the previous strategy was the usage of a different kinematic region for pure τ_{had}. This meant that the network had a chance to learn the difference between 2j2b and 3+j1b instead of what is a true and fake τ_{had}. A plentiful source of true τ_{had} are available in the same region as the $2\ell + 1\tau_{\text{had}}$ by use of MC simulation. As τ_{had} were well-simulated, these could be used as a signal sample. For the purpose of testing the robustness of the model, only one process is input as the training set. The tHq MC sample is the only one which the network will draw examples of true τ_{had}. The intent is to check the response of other MC such that any bias towards this particular process can show itself.

The background sample cannot be simulation as it is unavailable. Additionally, the other failure in the previous strategy came from the unreliable fake τ_{had} sample. Without knowledge of its fake τ_{had} composition, the similarity to fake τ_{had} in the $2\ell + 1\tau_{\text{had}}$ region could not be shown. Given that the SR defined by the tHq analysis requires a "medium" working point for tau identification, the "looser" data is discarded. This means that flipping the requirement to have loose-not-medium (called loose from now on) yields real data which is orthogonal to the measurement. Also, this loose region of data has a higher population of fake τ_{had} and forms a good set for training.

With that in mind, the selection is as shown in table 7.5. The selection is within the SR for the desired analysis which means that kinematics would match. This overcomes the previous problem of limiting variables and information to distributions which match. Therefore, all variables are available to use with such a selection. It should be noted that prong-ness of τ_{had} is split as they are subject to different backgrounds. This reduces the statistics available but enhances the learning potential of two individual networks. Similarly, the b-jet multiplicity is divided. In total, there are four networks to be trained with this method with the combination of prong-ness and b-jet multiplicity.

Signal	Background
$p_{\mathrm{T}}(l_3) > 14\,\mathrm{GeV}$	
$N_{\mathrm{jets}} > 2$	
$N_{b-\mathrm{jets}} = 1/2$	
$N_{\mathrm{prong}} = 1/3$	
OS light leptons	
tight light leptons	
τ_{had} is tight	τ_{had} is loose-not-tight
PDG ID must match tau	

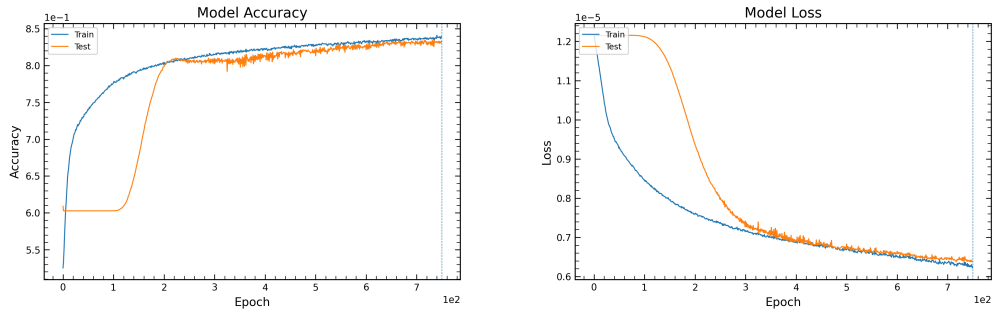
Table 7.5: Selection of signal and background samples for the MC vs. data strategy in τ_{had} identification. Both selections are identical with the exception of tau lepton tightness and true τ_{had} matching with truth information. The l_3 in the first criteria defines the third lepton in the event, including the τ_{had}.

The problem this selection raises is that regions or sets to be fed into a neural network are defined by the underlying RNN classifier. For this reason, variables chosen must not reveal information about tightness such that a classifier is orthogonal to the existing RNN. Additionally, checks must be made to ensure that the network has learned to identify objects rather than the working point. In the first step, variables chosen are purely kinematics or angles between reconstructed objects in the event. As these two sample selections are similar kinematically, the differences in these variables should exemplify τ_{had} identification. The variables used were:

- $E_{\mathrm{T}}^{\mathrm{miss}}$,
- $\Delta\phi(\ell_1, \ell_2)$,
- $\Delta R(\ell_1, \ell_2)$,
- $m(\ell_1, \ell_2)$,
- $\Delta p_{\mathrm{T}}(\ell_1, \ell_2)$,
- Lorentz invariant inner product (LIIP) between any two objects (light leptons and first three jets),
- τ_{had} kinematics (p_{T}, E, η, ϕ)
- $\Delta\phi(\tau_{\mathrm{had}}, \ell_1)$,
- $\Delta\phi(\tau_{\mathrm{had}}, \ell_2)$,
- $\Delta\eta(\tau_{\mathrm{had}}, \ell_1)$,
- $\Delta\eta(\tau_{\mathrm{had}}, \ell_2)$.

where ℓ signifies light leptons only and the Lorentz invariant inner product is defined as:

$$\mathrm{LIIP}(\mathbf{x}, \mathbf{y}) \stackrel{\mathrm{def}}{=} p_{\mathrm{T}}(\mathbf{x}) p_{\mathrm{T}}(\mathbf{y}) \left[\cosh(\eta(\mathbf{x}) - \eta(\mathbf{y})) - \cos(\phi(\mathbf{x}) - \phi(\mathbf{y}))\right].$$



(a) Accuracy (b) Loss

Figure 7.8: Metrics of the network trained with loose data and truth-matched MC for τ_{had} identification. The accuracy (a) and loss (b) are separate in the beginning but the training quickly converges.

As a proof of concept, this strategy with these variables was employed in the $1\ell + 1\tau_{\text{had}}$ region as it contains more events than $2\ell + 1\tau_{\text{had}}$. Like before, a grid search is conducted to find the best hyperparameters in the network. The best architecture to come out was a network with a 4×72 structure; four hidden layers of 72 nodes each. Each layer had the `ReLU` activation function with the exception of the output layer which had a sigmoid function. The loss function was a binary cross-entropy, typical of such classification tasks. The optimization algorithm which performed best was Adam. A 25 % dropout rate was added to every layer in order to counteract overtraining. The network was trained for 750 epochs with a large batch size of 40 000.

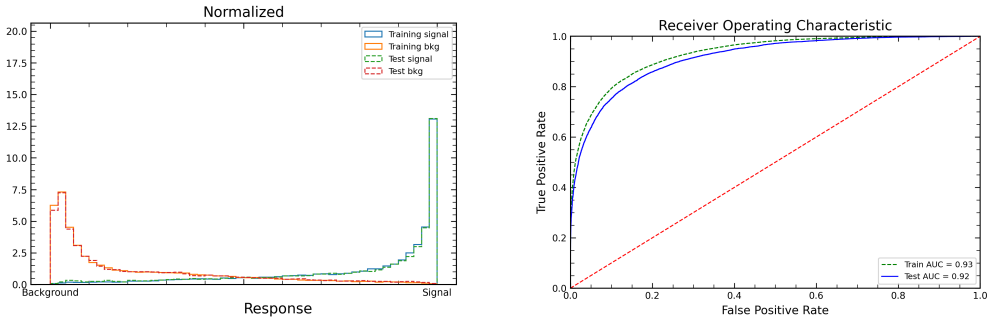
The performance of this network is shown in figure 7.8. An interesting feature of these plots is the divergence at the beginning. Before the 300th epoch, the network had learned something which was not in the testing sample. However, this minima was not desired and over time the network went to a more general minimum. The converged lines continue as expected until the end of training where no divergence can be seen.

Naturally, the response is the most important check to make; shown in figure 7.9. The training response, figure 7.9(a), shows a clear separation between signal and background. The separation power can be seen in the ROC curve, or more accurately the AUC which is above 0.90. It can be seen that the network has learned something to separate from these two samples well. As mentioned before, some checks need to be made in order to ensure that the learned pattern is τ_{had} identification.

As the signal sample contains only tHq events, the behavior of the network on other processes yields the existence of a bias. Figure 7.10 shows a comparison of tHq with separate MC processes. Furthermore, each process is divided by true or fake τ_{had} from truth information. The desired behavior is that distributions of fake and true τ_{had} are similar between processes.

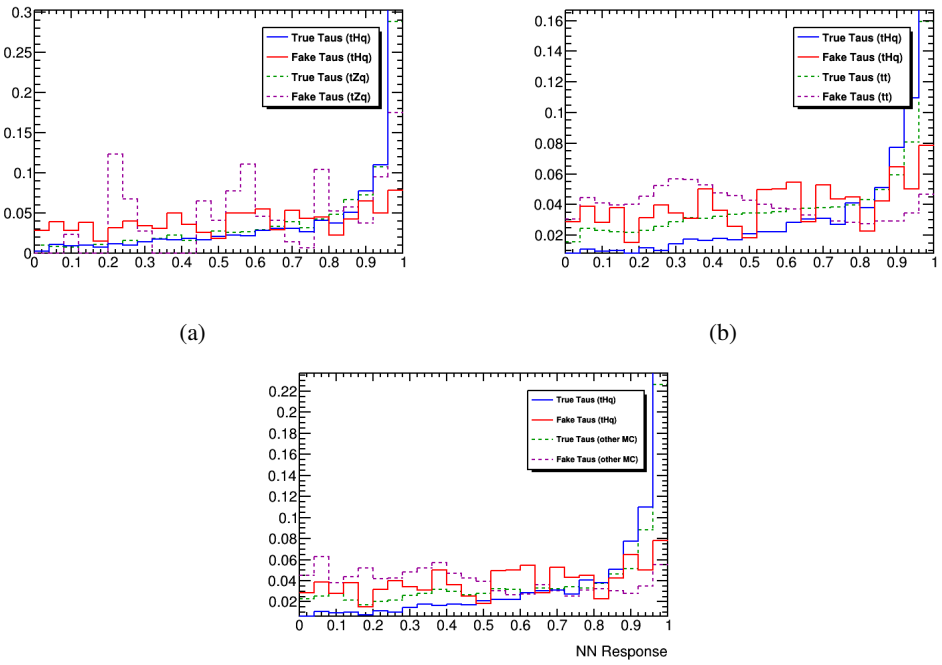
Figure 7.10(a) shows the tZq response due to its similarity to tHq. As tZq is closest in terms of kinematics, any bias in this particular plot would imply a faulty classifier. It can be seen that the general trend is that true τ_{had} events mostly populate the right side of the plot. In contrast to figure 7.9(a), the fake τ_{had} from these processes is almost flat in the response of the network. This is because the training background set includes loose-only τ_{had} where as the events shown in 7.10 are tight. It would be unlikely that the background behavior remain identical but it is positive that the network does not assign it as signal. This behavior is more obvious in the other plots as the tZq sample has so few fake events that one cannot clearly see a trend.

Next, figure 7.10(b) includes $t\bar{t}$ where a large quantity of fake τ_{had} leptons come from. The



(a) NN response (b) ROC

Figure 7.9: NN response and the separation power shown by the ROC curve. This response shows the training samples and is subject to change once applied to the SR.



(a) (b)

(c)

Figure 7.10: NN response of true and fake τ_{had} in MC simulation in the $1\ell + 1\tau_{\text{had}}$ signal region. The response plots compare the signal (tHq) sample shape after training against other processes. This comparison is made against tZq (a), $t\bar{t}$ (b), and other MC (c).

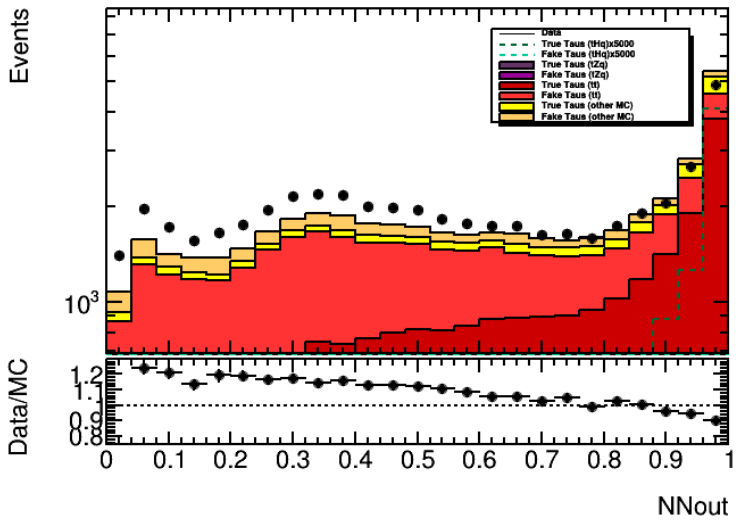


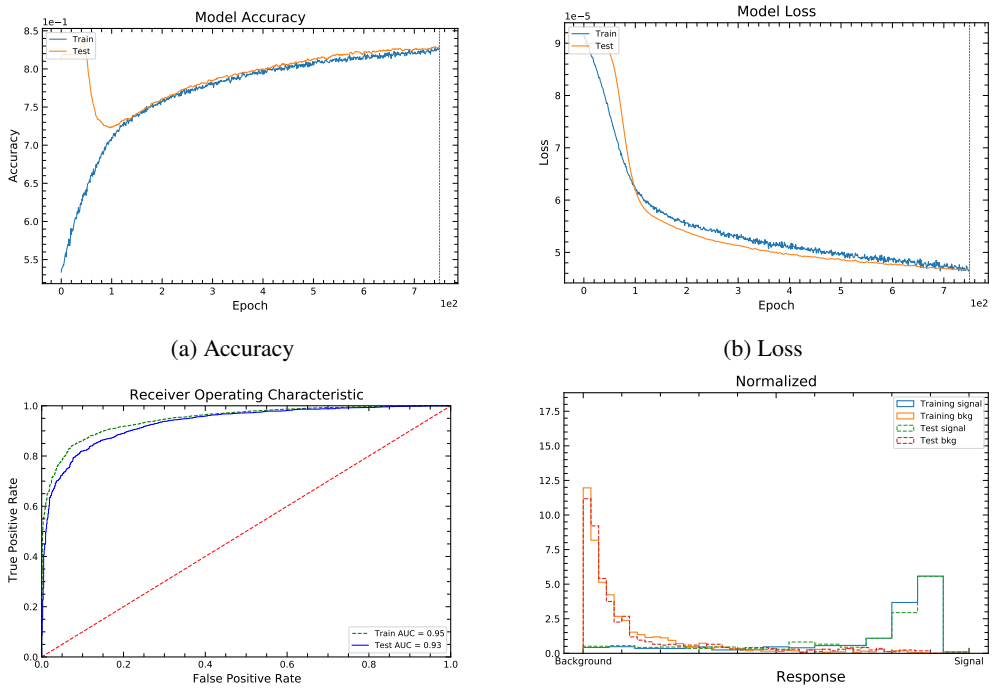
Figure 7.11: NN response stack plot in the $1\ell + 1\tau_{\text{had}}$ signal region. MC is further split by true and fake τ_{had}. tHq yields are increased 5 000-fold for visibility.

kinematics of $t\bar{t}$ differ from tHq greatly and therefore any bias should easily show in this plot. However, the trend of true τ_{had} is the same for both; pushed to the right side. Given the contribution of $t\bar{t}$ to fake τ_{had} leptons, a clear trend can be seen in the fake line of this sample. There is a maximum towards the left side, a desired trait, and fewer events populate the right side. The ideal scenario is as the training response but a flat line or a bump to the left is a reasonably positive outcome.

Lastly, other MC samples are shown in 7.10(c). Other MC includes rare top-quark processes, single top-quark processes, diboson, and Z/W plus jets events. Similarly to $t\bar{t}$, the kinematics of most of these processes differ from tHq and therefore bias could be expected if the model has failed. What can be seen is the same trend as $t\bar{t}$ with fake τ_{had} showing a relatively flat line with a maxima near the left side. On the other side, the events containing true τ_{had} are pushed to the right with the same shape to the signal sample. From these three plots, it is reasonable to conclude that the network has generalized τ_{had} identity. This is because it can apply this knowledge and correctly tag events regardless of the kinematics of the source.

Once applied to all MC and data, the distribution can be plotted and data can be included for comparison. Shown in figure 7.11, one can see the previously discussed trend but with scaling to luminosity. Here it can be seen that true τ_{had} events populate the right side. One positive trend is the increase of data and MC agreement towards signal-like values. This implies that fakes seen in data but poorly modeled are still being pushed towards the left-hand side of the plot. With the shown potential, this strategy is carried on to the $2\ell + 1\tau_{\text{had}}$ signal region.

Given that the $2\ell + 1\tau_{\text{had}}$ region has a small fraction of events that $1\ell + 1\tau_{\text{had}}$ has, a modification to the event selection is made. The signal sample, tHq simulation, is not given a cut on the τ_{had} identification. This modification improves upon the previous selection in two ways. True τ_{had} which did not succeed the RNN identification could be useful in this classifier, giving a second chance. But more importantly, events where a true τ_{had} exists but is not selected as the τ candidate are also considered as signal. In this way, the inclusion of true τ_{had} in the loose working point from the data should end up correctly identified as impurities.



(a) Accuracy (b) Loss

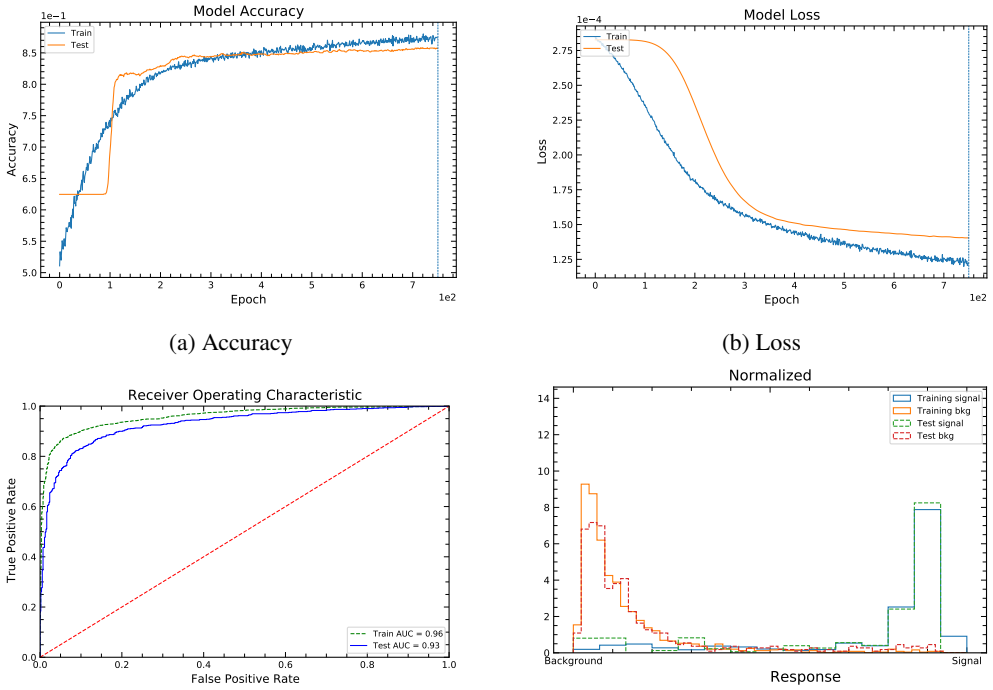
(c) ROC (d) NN output

Figure 7.12: Metrics of the network trained with loose data and truth-matched MC for 1-prong τ_{had} identification in the $2\ell + 1\tau_{\text{had}}$ region.

After performing the same grid search, the architecture of this network is derived. It is important to note that although the problem is similar the kinematics are different enough to require a new architecture. The setup of the network contains three hidden layers of 48 nodes each. The first activation function was tanh followed by `ReLU` functions in the hidden layers leading to a final sigmoid function in the output. A binary cross-entropy loss function was used with an Adam optimizer using custom parameters. The optimizer had a learning rate of 2.5×10^{-5}, 5 % momentum without decay and Nesterov momentum. A 25 % dropout rate was added to every layer in order to counteract overtraining. Similarly to before, the network was trained for 750 epochs but with a smaller batch size of 2 000. The reason for the smaller batch size was the total size of the training data was significantly smaller. A batch size above 10 000 would be a significant fraction of all training events and therefore undesirable.

As before, the metrics of the 1-prong training are shown in figure 7.12. The accuracy has a strange disparity at the beginning but the feature quickly converges to expected behavior. Although unlikely, it is possible that the beginning parameters randomly assigned accurate labels to the test sample. After all, the initialization is random as well as the assignment of training and test events. Other than that, the loss curve shows reasonable behavior and no divergence towards the end of training. Along with the loss, the ROC curve shows no overtraining and a well-performing classifier. Of course, that is cemented by the response of the network in both training samples. Here it can be seen that the network has effectively classified signal and background.

As one- and three-prong τ_{had} have different backgrounds, the networks are trained separately. Figure



(a) Accuracy

(b) Loss

(c) ROC

(d) NN output

Figure 7.13: Metrics of the network trained with loose data and truth-matched MC for 3-prong τ_{had} identification in the $2\ell + 1\tau_{\text{had}}$ region.

7.13 shows the same metrics as before but on the 3-prong network. There is a strange behavior at the beginning as other models had with similar convergences. The loss in this model seems to diverge a bit towards the end which can mean overtraining. However, the ROC curve shows that the difference in AUC is small which is deemed acceptable after looking at the response. One important point with regards to the 3-prong selection is that the total number of events is about a fourth of 1-prong. This means that the 3-prong model may find difficulties generalizing when given fewer examples to learn. With the high purity of real and fake τ_{had}, the low statistics does not prevent learning.

As these networks were trained on the tHq MC sample, there might be some bias still. For this reason, the same checks are made against other MC samples, shown in figure 7.14 for the 1-prong model. The first thing to be noticed is in figure 7.14(a) where the fakes of tHq and tZq are not well separated. Although there is a small bump on the left hand side for fakes, there is a significant pollution on the right-hand side. Normally, this could be considered an issue but these two processes contribute a small fraction to the overall event yields. Therefore, fake τ_{had} from these processes would not contribute to a great amount and could be ignored. On the other hand, $t\bar{t}$ shows the same behavior as the $1\ell + 1\tau_{\text{had}}$ training or better. As $t\bar{t}$ contributes a significant fraction of fakes, seeing a reasonable separation is a positive. Other MC, such as Z + jets, are assigned an ideal classification. Processes unlike the training sample in figure 7.14(c) show that true τ_{had} are adequately classified. Moreover, the fake τ_{had} from these contributions are largely shoved to the left-hand side of the response range.

Similar behavior can be seen in figure 7.15 for the 3-prong model. Although the other MC are not as drastically separate as the 1-prong model, the separation is comparable to that of $t\bar{t}$. Either way, the

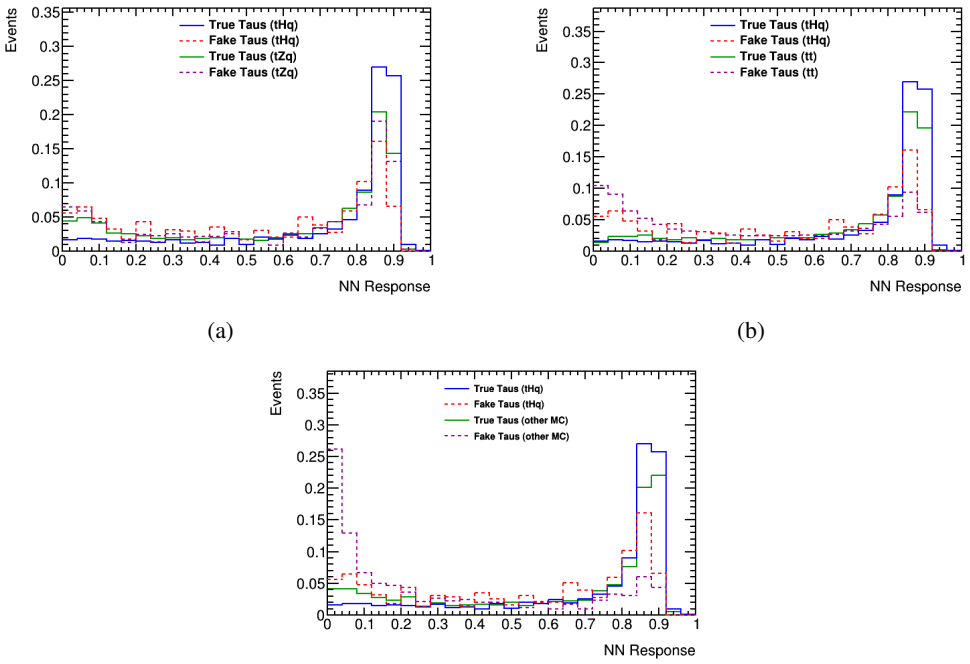
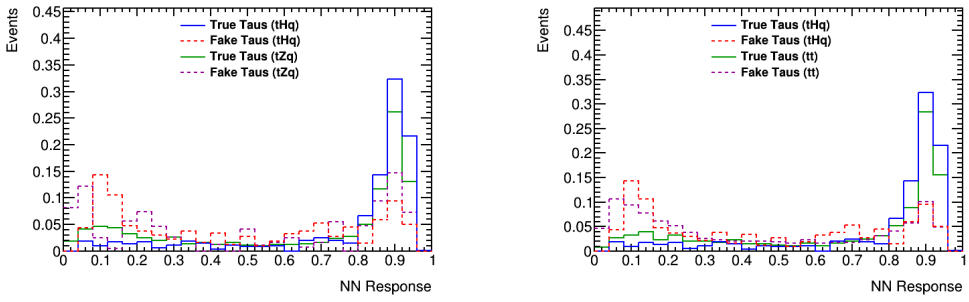


Figure 7.14: NN response of true and fake 1-pronged τ_{had} in MC simulation in the $2\ell + 1\tau_{\text{had}}$ signal region. The response plots compare the signal (tHq) sample shape after training against other processes. This comparison is made against tZq (a), $t\bar{t}$ (b), and other MC (c).

trend of true τ_{had} to be correctly classified remains for all of the three sets. In contrast to the training on $1\ell + 1\tau_{\text{had}}$, some separation can be seen in these samples. Where before the background had a flat distribution, the separation in the $2\ell + 1\tau_{\text{had}}$ region appears to be better.

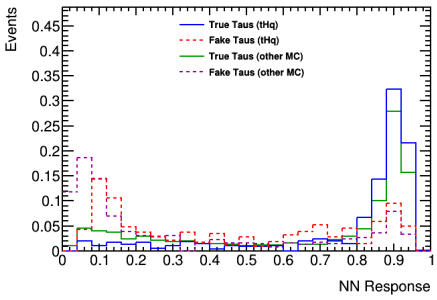
Figure 7.16 shows the response of all MC stacked in the $2\ell + 1\tau_{\text{had}}$ SR. As these figures have a further division between their fake and true τ_{had}, a more comprehensive conclusion can be drawn. Although the right side sees still some contamination from fakes, the majority are contained in the left-hand side of the response. As mentioned earlier, the contribution of fakes from tZq and tHq is negligible as it cannot be seen in any plot. With the signal region of the response appearing so pure, an aggressive cut could diminish fake τ_{had} contribution in the SR without removing much signal. Additionally, performance plots of the $2b$ network can be found in appendix C. These were not included in the main text as their performance was similar to that of $1b$.

As mentioned earlier, different networks were created to account for one- and three-pronged τ_{had} as well as b-jet multiplicity. After all the checks, the application of these networks are performed in two regions: a fakes control region (CR) and the signal region. This CR is defined by flipping the τ_{had} ID used for the signal region; effectively, it is nearly identical to the background selection from training. The distributions can be seen in figures 7.17 (CR) and 7.18 (SR). All of the images show the NN response of all four networks. The control region shows that the lower values of the response (fake-like) are populated by what is expected. Likewise, the signal regions show the expected true and fake τ_{had} distributions. Although the distributions in the SR have a decent separation, the contribution in the signal-like bins contain a significant background contribution.



(a)

(b)



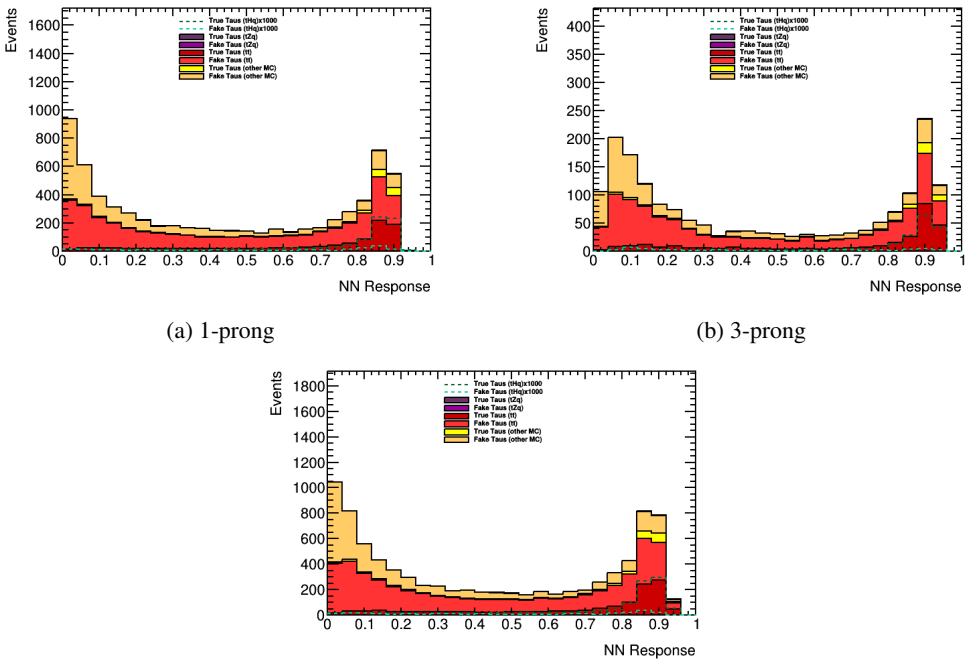
(c)

Figure 7.15: NN response of true and fake 3-pronged τ_{had} in MC simulation in the $2\ell + 1\tau_{\text{had}}$ signal region. The response plots compare the signal (tHq) sample shape after training against other processes. This comparison is made against tZq (a), $t\bar{t}$ (b), and other MC (c).

To improve the network, one could include variables such as the decay jet's width of the calorimeter deposit and track width. These variables are typically used to estimate the contribution of fake τ_{had} as they are sensitive to these processes. After training new networks, the separation did not improve; plots can be found in appendix D. Either way, the network is still useful as the distribution of quark- and gluon-initiated jets which fake τ_{had} have different distributions. This feature implies that the network could be used for a template fit and estimate backgrounds. As the works described in this thesis do not include any fits, a brief description of how general fits are performed can be found in appendix E. By using this fit, an estimate on the contribution of fakes of different types can be determined.

Figures 7.19 and 7.20 show the same network response with the output of the template fit. It should be noted that the fit was not performed on the NN response but the sensitive variables mentioned earlier. From these distributions, it can be seen that the network's classification mostly agrees with the template fit; validating the network. That is to say, should the estimate of fakes from the fit been thrown off by the network response then one could have concluded the network did not learn what was desired. Therefore, one can conclude that the network is performing as intended and classifying τ_{had} rather than some other process.

A second way to validate the network can be from defining regions orthogonal to the RNN. In the region of $2\ell + 1\tau_{\text{had}}$, one of the light leptons comes from the other τ which is emitted by the Higgs boson in the event. As these bosons decay, the angle at which the two τ leptons are emitted should be relatively close; or at least closer to each other than the other light lepton. With this knowledge, two regions can be created where the candidate pair are either of the same sign or opposite sign. Same



(a) 1-prong

(b) 3-prong

(c) Both

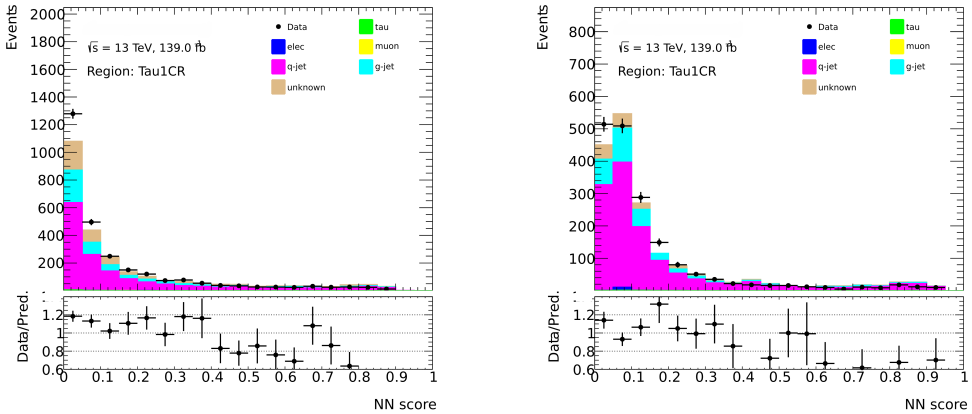
Figure 7.16: NN response stack plot in the $2\ell + 1\tau_{\text{had}}$ signal region. MC is further split by true and fake τ_{had}. tHq yields are increased 1 000-fold for visibility.

sign leptons who make a Higgs candidate should be mainly fake τ_{had} while opposite sign leptons are likely to come from Higgs decay. These checks are done in the tHq MC as it is the training signal and the $t\bar{t}$ MC since it provides many fakes to the SR. This charge pair gives an orthogonal selection to the existing τ identification and allows for validation of input kinematics. Furthermore, by validating on MC it is possible to check how often this assignment of H candidate is wrong. Unfortunately, due to time constraints this validation method was not performed.

From the plots of the SR shown in figure 7.20, one can better gauge the performance of the network with respect for the backgrounds. In particular, the network performs reasonably well at mitigating the "unknown" fakes. Gluon-initiated jets, for the most part are also nicely but not completely separated. The signal region is highly populated by quark-initiated jets which fake τ_{had}. Unfortunately, although a large majority of the fakes are correctly classified, a significant fraction pollutes the signal-like region of the response.

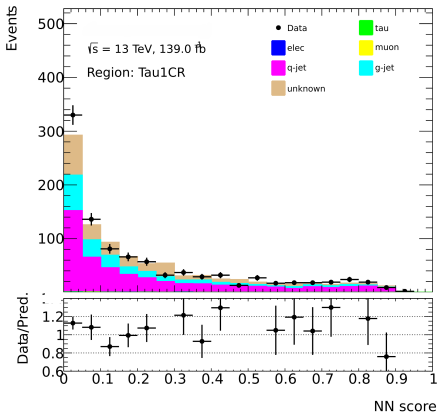
As the intended use of this network was cutting, the task is done and signal purity is extracted. From figure 7.21, one can see that achieving a high signal significance is possible with a high cut at about 0.8. In these plots, signal purity implies the population of tHq compared to all other processes. Additionally, the signal significance is calculated as the total signal events divided by the square root of all events. This cut would remove not just the fake background but also a portion of modeled backgrounds increasing signal purity. From the plot, one can see that a cut of 0.8 on the response corresponds to about 0.6 % significance. This value is competitive with the NN used for the specific purpose of identifying tHq.

Although the signal is enhanced with this network, it is not specifically designed for such a purpose;

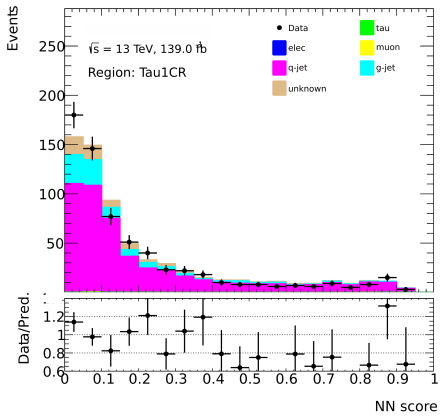


(a) 1-Prong, 1b

(b) 3-Prong, 1b

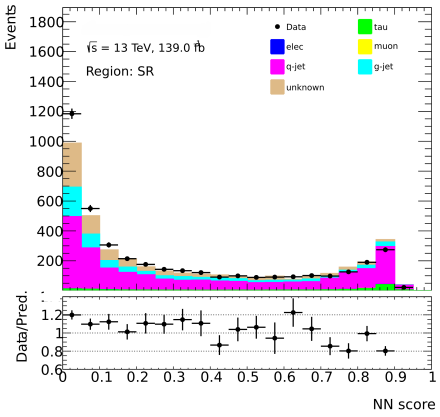


(c) 1-Prong, 2b

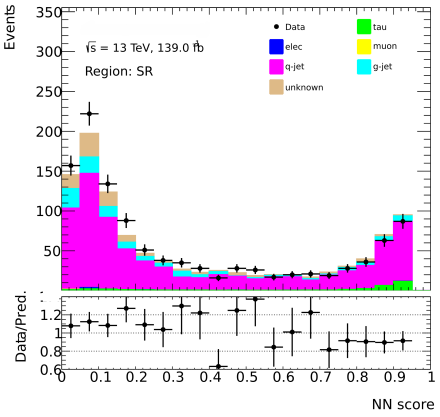


(d) 3-Prong, 2b

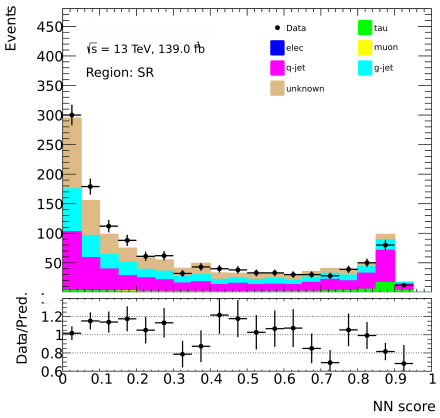
Figure 7.17: Fake τ_{had} control region plots of the NN response. True τ_{had} are included in the plots as well as different sources of fakes. The plots shown are do not have the corrected distributions of fakes from the template fit.



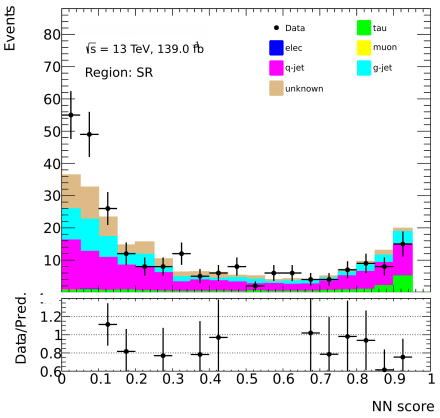
(a) 1-Prong, 1b



(b) 3-Prong, 1b

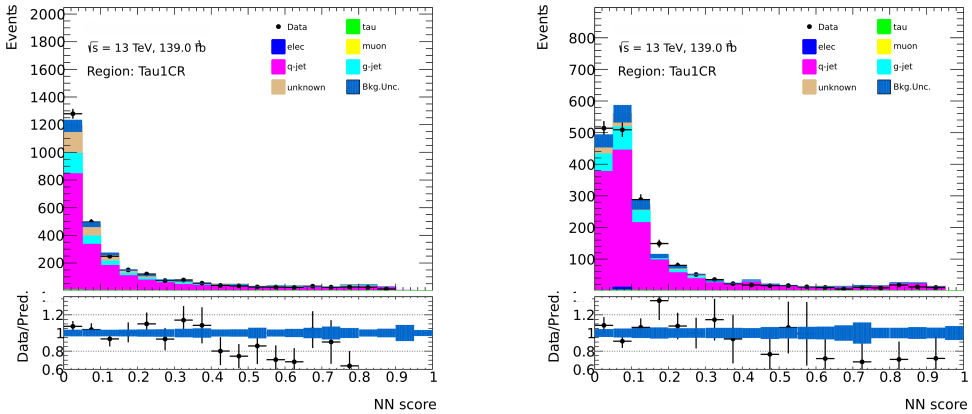


(c) 1-Prong, 2b



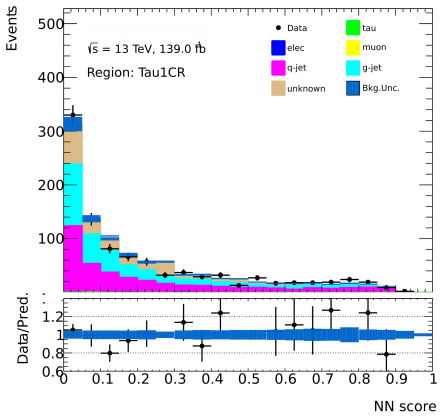
(d) 3-Prong, 2b

Figure 7.18: Fake τ_{had} signal region plots of the NN response. True τ_{had} are included in the plots as well as different sources of fakes. The plots shown are do not have the corrected distributions of fakes from the template fit.

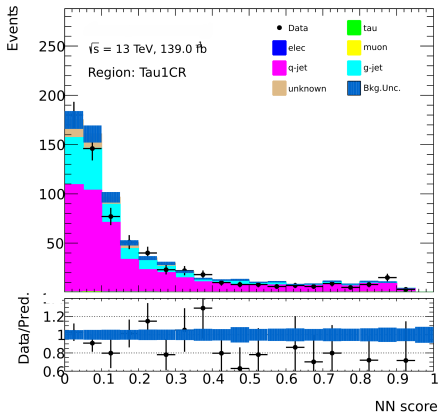


(a) 1-Prong, 1b

(b) 3-Prong, 1b

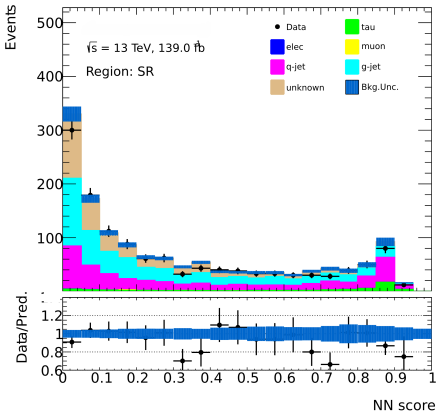


(c) 1-Prong, 2b

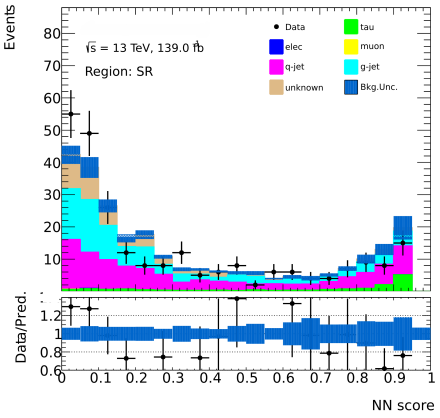


(d) 3-Prong, 2b

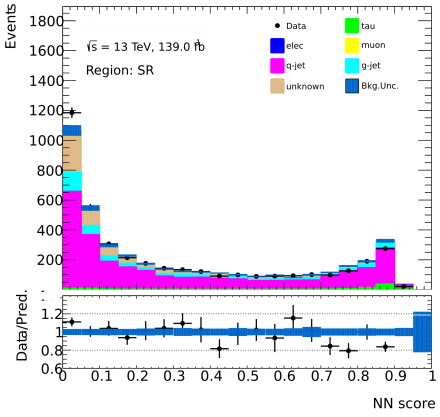
Figure 7.19: Fake τ_{had} control region plots of the NN response. True τ_{had} are included in the plots as well as different sources of fakes. Plots shown have the corrections applied from the template fit performed on the `JetCaloWidth` variable.



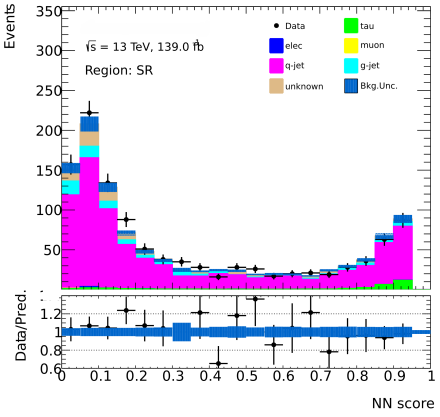
(a) 1-Prong, 1b



(b) 3-Prong, 1b

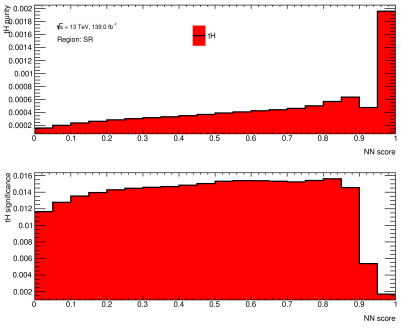


(c) 1-Prong, 2b

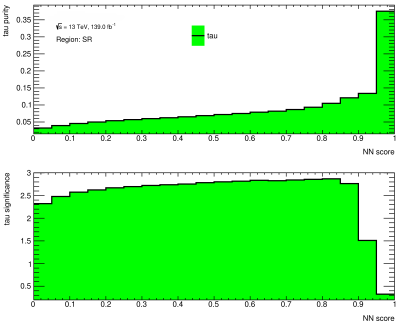


(d) 3-Prong, 2b

Figure 7.20: Fake τ_{had} signal region plots of the NN response. True τ_{had} are included in the plots as well as different sources of fakes. Plots shown have the corrections applied from the template fit performed on the `JetCaloWidth` variable.



(a) Signal tHq



(b) τ_{had}

Figure 7.21: Fake τ_{had} signal region plots of the NN response.

it is just a useful side effect. A network designed to specifically classify tHq versus its other modeled backgrounds add an edge to what this network does. Or, more accurately, by including the data-driven classification knowledge, one could improve the signal purity further. After all, there is no guarantee that some background events this network considers signal would also be assigned the same value by the competing network. Therefore, it would be interesting to see the combination of both networks in use as they may improve the analysis. At the time of the writing of this document, the analysis has not yet done any studies on the idea.

After all the attempts to create a NN which can classify poorly modeled backgrounds, it is nice to see one succeed. It should be noted that the strategies described here were not the only ones attempted but instead were the ones which reached failure or success at a point worth discussing. In the end, by applying a weak supervision technique a classifier was created capable of identifying true τ_{had} and give some discrimination against poorly modeled backgrounds. Although the classification was not perfect, it is still useful as it competes with a network that specifically tries to identify the signal sample. Furthermore, the information this network provides could also be used to improve the analysis by mitigating the poorly modeled background.

CHAPTER 8

Summary and Conclusion

The goal of this work was to explore options provided by advanced machine learning techniques to treat known issues in physics analyses. In particular, the exploration of background estimation and mitigation was explored. Additionally, generating a useful variable to enhance difficult to simulate effects for measurement was attempted. Many tools used were derived from different machine learning techniques with varying degrees of supervision. Some of which were autoencoders which require no labels, models with nearly fully labeled sets, or density estimators. Overall, it was interesting to explore new avenues to overcome problems as new tools are developed.

Most of the strategies presented (and some which were omitted) were failures which became learning opportunities. Each loss leading to the forging of a new course of action which sought to correct the mistakes of what came before. If anything, the reader should take what is written here as experience and the process of self-improvement.

To tie it all together, different models were employed with two particular tasks. These were to create a classifier for fake τ_{had} objects and generate a variable sensitive to the tW and $t\bar{t}$ interference. For analyses of rare processes like tHq with τ leptons in the final state, identification is paramount. As backgrounds need to be estimated, an poorly modeled background poses another difficulty in measurement of desired properties. Also, as processes like tW and $t\bar{t}$ are not well-defined, being able to probe the problematic regions is desirable. By constructing ways to better view interference effects, one may construct better MC models.

While exploring the interference between tW and $t\bar{t}$, classifiers were insufficient and density estimators were explored. First, normalizing flows were used with a similar goal to autoencoders. By normalizing LO tW events to some Gaussian distribution, any interference-like events would form under- or over-densities in the transformed space. As this sensitivity was not as useful, creating a density estimator using the ANODE technique was pursued. Similarly to normalizing flows, ANODE estimates an underlying distribution but is highly sensitive to anomalies in a dataset. For this reason, the ratio of probabilities is constructed where higher values denote anomalous (interference) events. Unfortunately, the results of this were inconclusive as tW and the interference were too similar for the network to separate. Additionally, the fact that the anomaly was destructive may have caused the network to not consider the DS sample as anomalous as it should.

First, autoencoders were used as density estimators in both in order to use the reconstruction error as an anomaly detector. Should an autoencoder be trained on true τ_{had}, anything which looks differently would be wrongly reconstructed. Similarly, an autoencoder expecting LO tW events would ideally

wrongly reconstruct events where an second, off-shell top-quark exists. Neither of these first attempts succeeded in their task and were both replaced.

Following the autoencoders for τ_{had} identification, a weak classifier was constructed using neighboring regions. A test was made to find variables which were close enough to the desired signal region. What was found was a limited set of variables which provided no discriminating power. Similarly, a strategy using the CWoLa paradigm was used in an attempt to test two regions with and without interference. This, too, did not give a more sensitive variable than the existing $m_{b\ell}^{\text{minimax}}$.

The difficulties found in the neighboring regions strategy of τ_{had} were addressed in a new classifier using SR events. This strategy used the well-defined τ_{had} simulation against data which was populated by fake τ_{had}. Therefore, it addressed the kinematic differences and purity issues of the previous strategy. Finally, this strategy found some success in creating an identifier for events with real τ_{had}.

Although time was limited, the classifier for τ_{had} which was derived could be expanded to a two τ_{had} system. By using a technique called Tag N' Train (TNT) [129], one could take two classifiers dedicated to identifying a leading and sub-leading tau. These two models could then be used to further solidify each other and therefore create a stronger model which identifies two τ_{had} events.

If there was more time for projects like this I would pursue a different architecture, namely Graph Neural Networks. In brief, each graph gives three levels of information: meaning of a node, meaning in connection between nodes, and overall representation. By constructing graphs relating physics objects and training a network on simulation, one could tackle both problems described in this work. In essence, building graphs representing the decay objects of τ_{had} or the kinematics of $WWbb$ processes.

In hindsight, there were moments when writing where the problems became more obvious as they were being written. But on the same vein, new ideas on how to correct these strategies also arose. In the end, the exploration pursued contributed to the tHq analysis in the form of scale factor measurements for fake τ_{had} estimation. Additionally, a framework was developed to be used by students such that they can dive straight into physics without having to mess around with TensorFlow. Lastly, it is the hope of the author that someone may continue to explore and take some of the experience written here to avoid the same pitfalls.

APPENDIX A

Machine Learning Package

One of the projects undertaken during my work was the creation of a NN package found in [130]. The goal is to have one package for many analyses which cover many use cases. By doing this, future students do not have to sit through hours of Tensorflow tutorials to set up a NN. After which they will make the same errors we all made at the beginning and spend days or weeks debugging. This package offers a "black box" package where all the user needs is a config file and the `README`.

The package offers more than a typical feed-forward classifier. It offers autoencoders, adversarial neural networks, parallel learners, evolutionary networks, categorical classifiers, and masked autoregressive flows. Ideally, I would have added normalizing flows and graph networks but the time was limited. To set up any of these types of networks, the user only need to set a variable called `dnn_method` to the appropriate setting. Then one can construct any network architecture; even with variable number of nodes per layer. One has control over the optimizers' parameters, normalization, and losses. A savvy user also has the resources to craft their own custom losses and layers. Examples are given in the appropriate files `Networks/Losses.py` and `Networks/Layers.py`, respectively.

The network uses a controller file called `DNN.py` which calls the other modules to set up the appropriate network and load the files. Input is handled by the `Uproot` [131] package which is fast and returns either Pandas data frames or NumPy arrays. Both of which should be fairly familiar to any Python user. Additionally, the network creates new files with the NN response in addition for later use, also with `Uproot`. It also provides the user with many plots for metric assessment and includes the ability for a deeper analysis by providing Tensorboard logs, if requested.

The input is always normalized by use of the scaler functions provided by `scikit-learn` [127, 128]. The default is the `StandardScaler` but the user has the ability to use any of the following: `MinMaxScaler`, `MaxAbsScaler`, `RobustScaler`, `QuantileTransformer`, and `PowerTransformer`. Additionally, one can use Principal Component Analysis to whiten/decorrelate data before training the network. The package also has a ranking function where it returns a list ranking the significance of the given training features. Lastly, a folder containing scripts are provided which include examples on how to run the network on BAF. Other scripts included are analysis specific plotting code which may be amended to function in other works.

APPENDIX B

Autoencoder Distributions

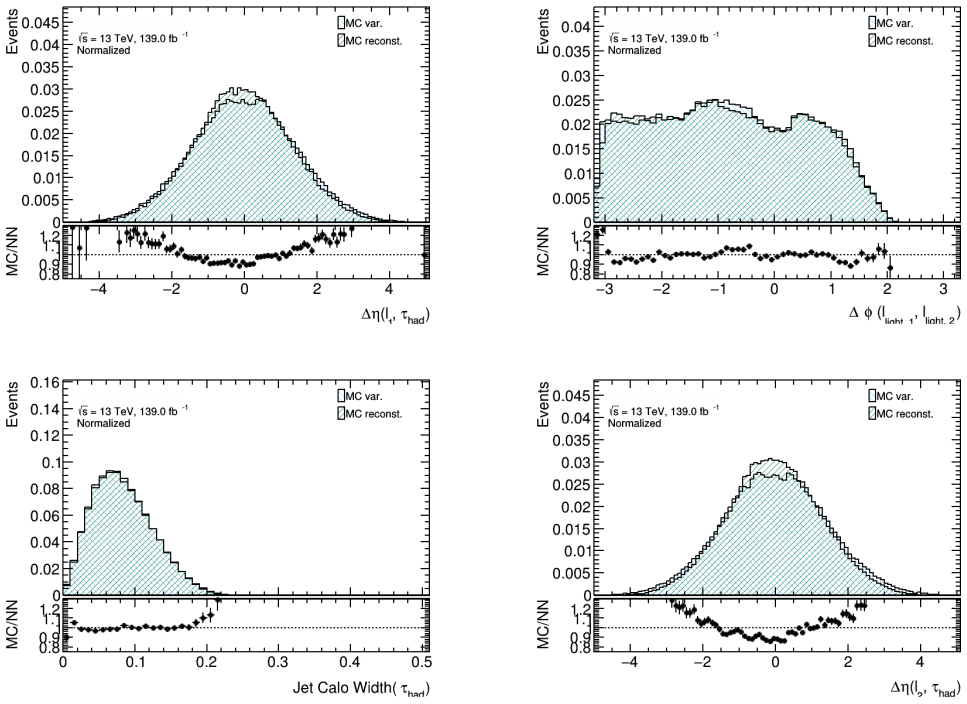


Figure B.1: Input variables for the autoencoder designed to reconstruct fake τ_{had}. Included in the plot is the reconstructed variable along with a ratio plot to help in the comparison.

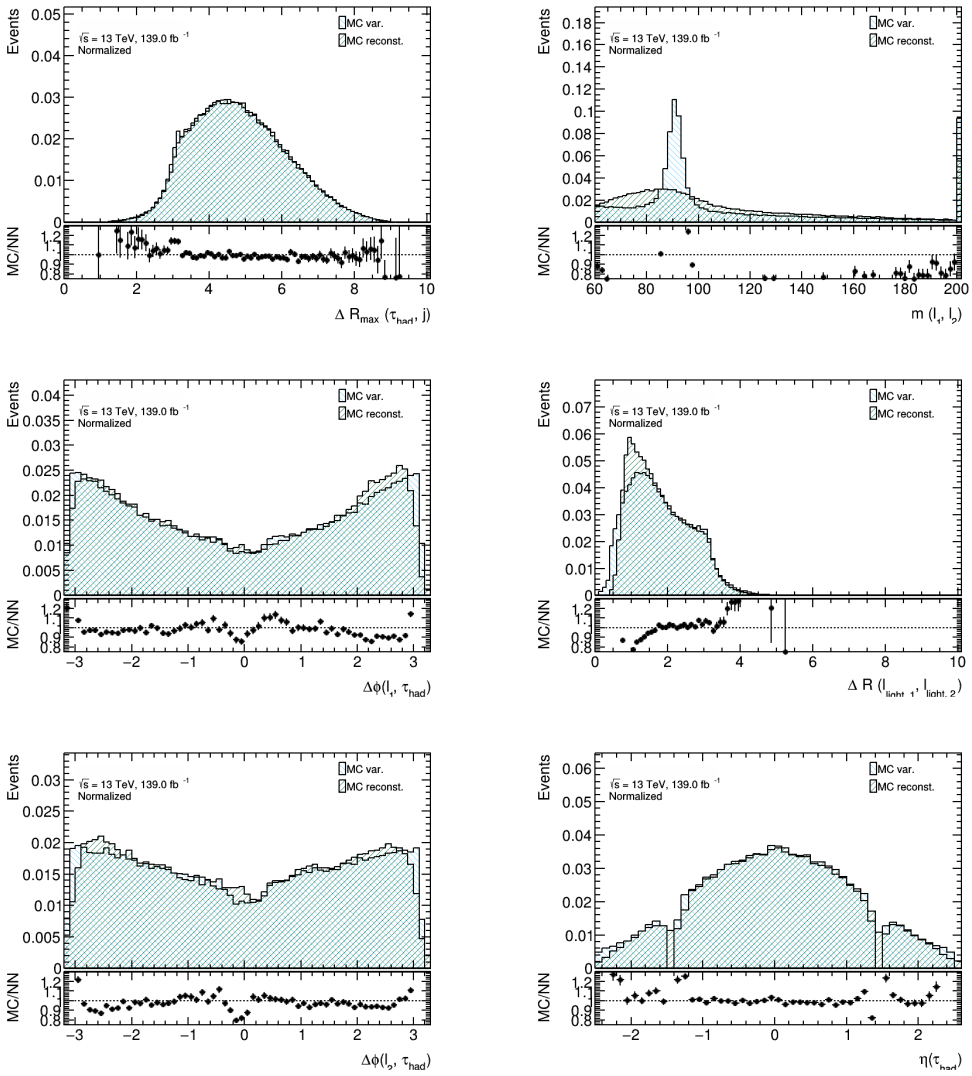
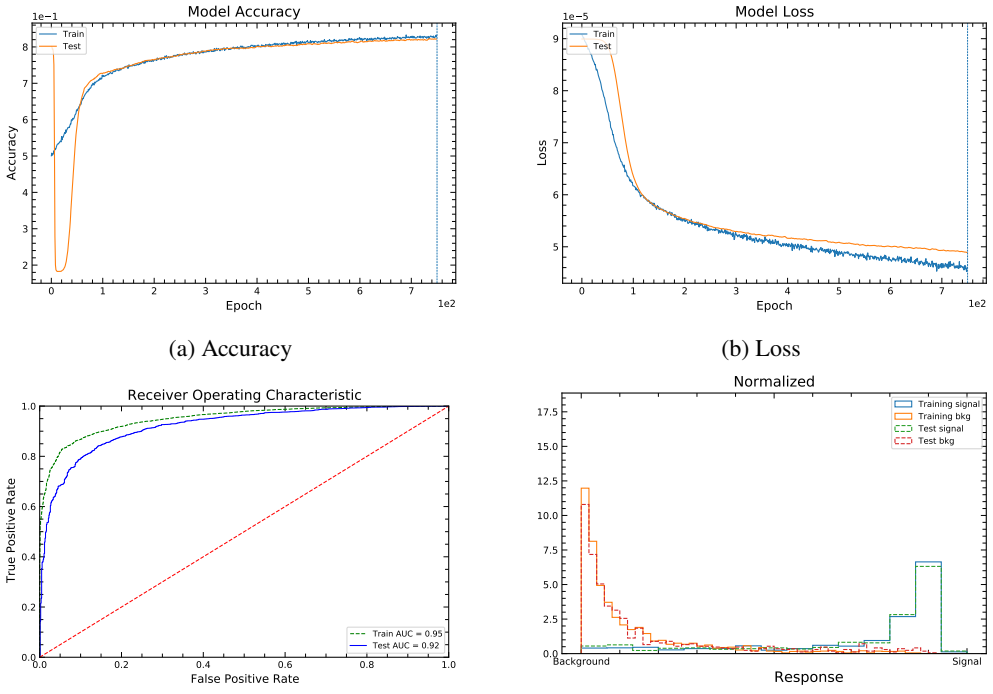


Figure B.2: Supplemental input variables for the autoencoder designed to reconstruct fake τ_{had}. Included in the plot is the reconstructed variable along with a ratio plot to help in the comparison.

APPENDIX C

Data vs. MC Classifier for τ_{had} with a two b-jet selection



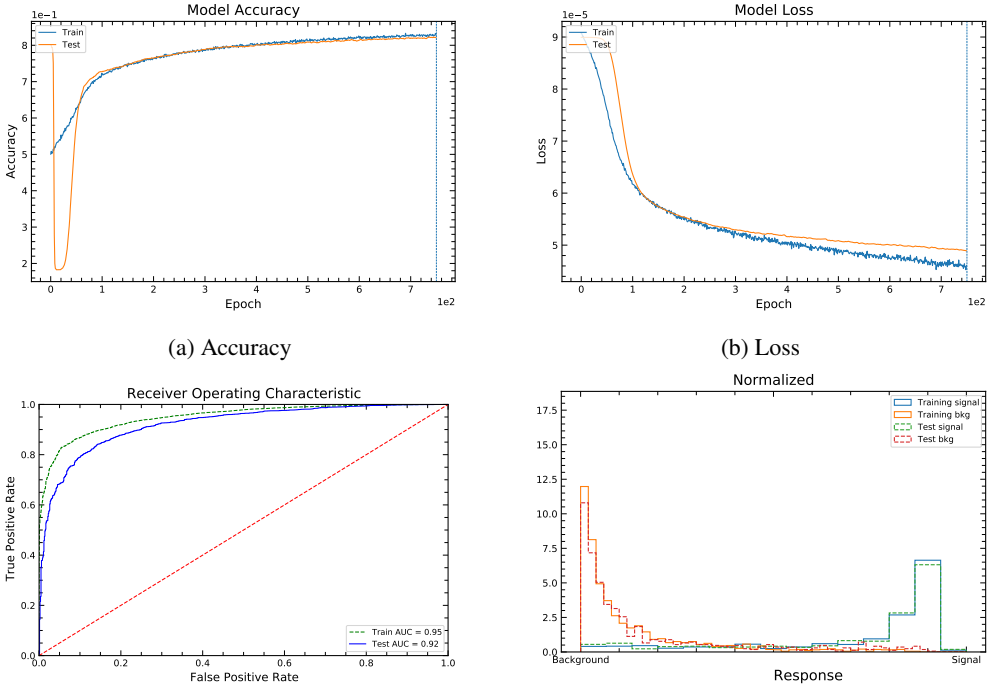
(a) Accuracy

(b) Loss

(c) ROC

(d) NN output

Figure C.1: Metrics of the network trained with loose data and truth-matched MC for 1-prong τ_{had} identification in the $2\ell + 1\tau_{\text{had}}$ region.



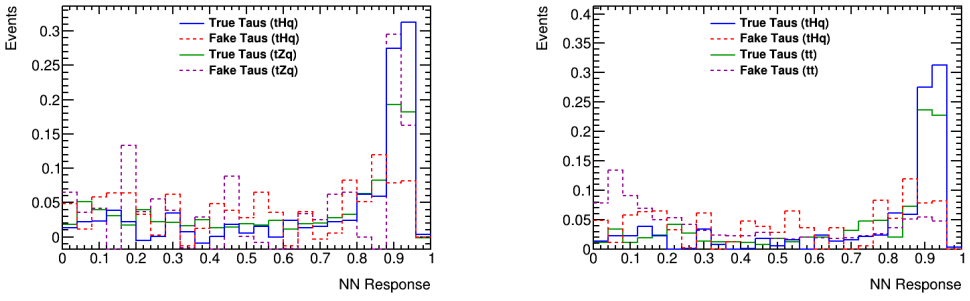
(a) Accuracy

(b) Loss

(c) ROC

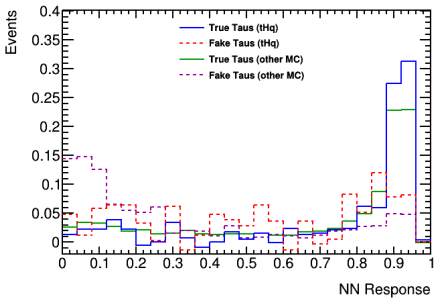
(d) NN output

Figure C.2: Metrics of the network trained with loose data and truth-matched MC for 3-prong τ_{had} identification in the $2\ell + 1\tau_{\text{had}}$ region.



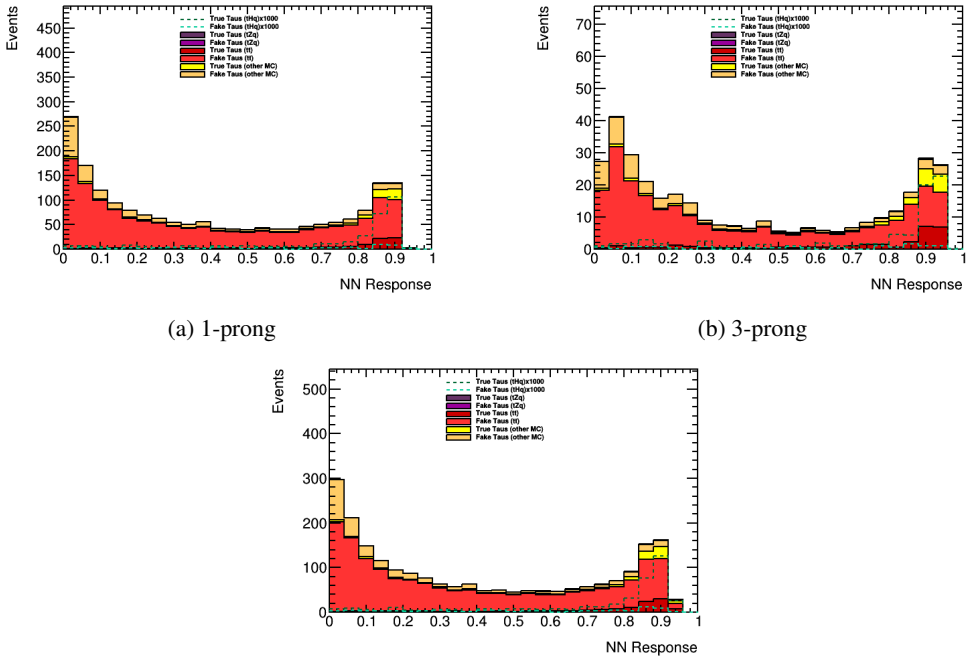
(a)

(b)



(c)

Figure C.3: NN response of true and fake 3-pronged τ_{had} in MC simulation in the $2\ell + 1\tau_{\text{had}}$ signal region. The response plots compare the signal (tHq) sample shape after training against other processes. This comparison is made against tZq (a), $t\bar{t}$ (b), and other MC (c).



(a) 1-prong

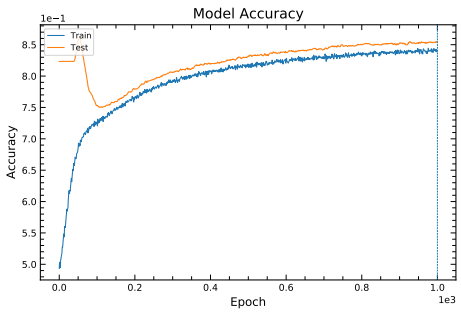
(b) 3-prong

(c) Both

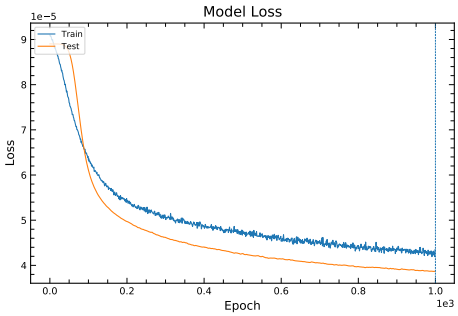
Figure C.4: NN response stack plot in the $2\ell + 1\tau_{\text{had}}$ signal region. MC is further split by true and fake τ_{had}. tHq yields are increased 1 000-fold for visibility.

APPENDIX D

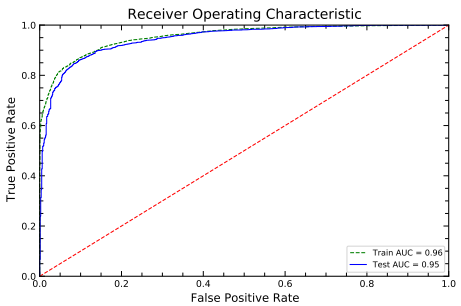
Networks which Included the Jet and Track Calorimeter Widths as Training Variables



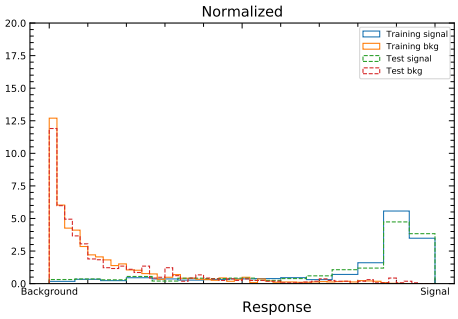
(a) Accuracy



(b) Loss

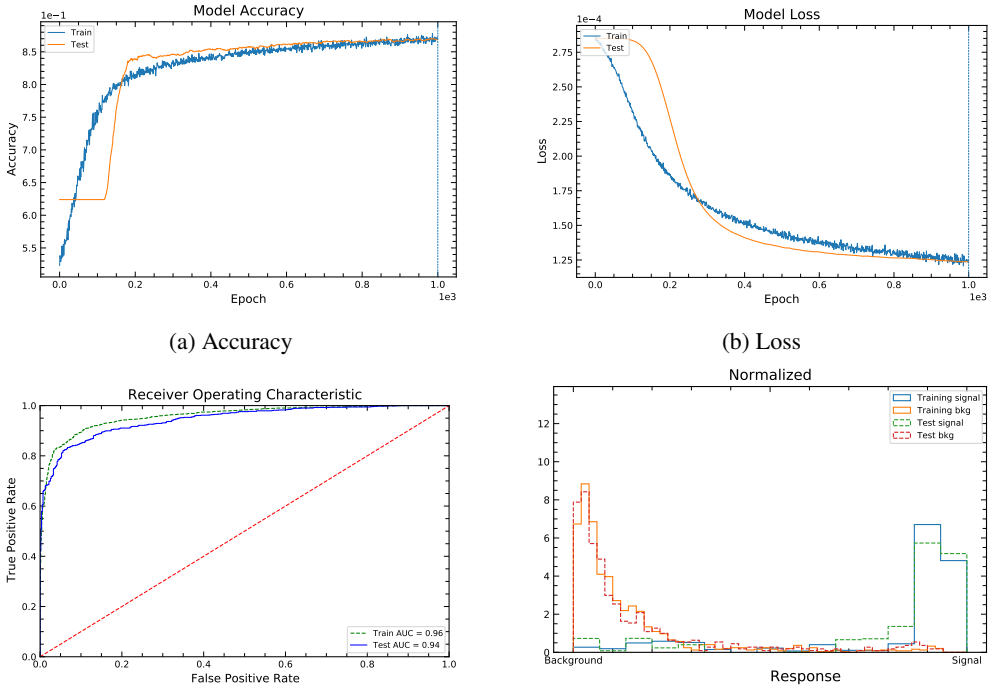


(c) ROC



(d) NN output

Figure D.1: Metrics of the network trained with loose data and truth-matched MC for 1-prong τ_{had} identification in the $2\ell + 1\tau_{\text{had}}$ region.



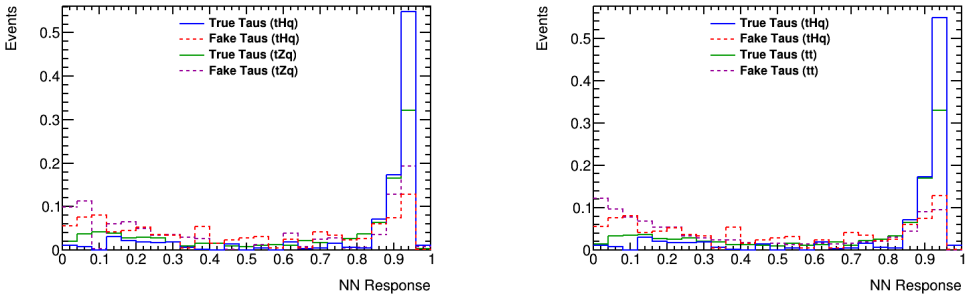
(a) Accuracy

(b) Loss

(c) ROC

(d) NN output

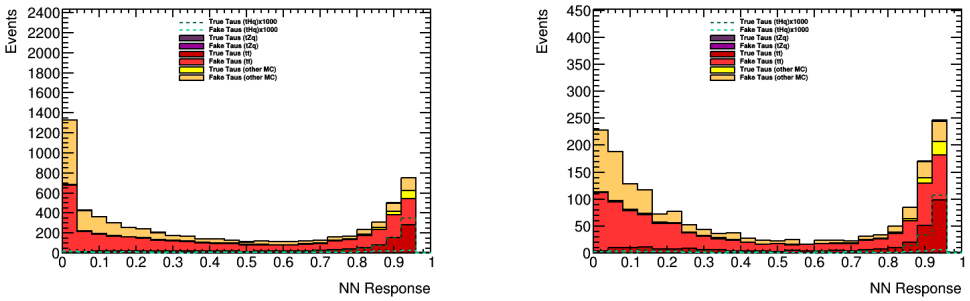
Figure D.2: Metrics of the network trained with loose data and truth-matched MC for 3-prong τ_{had} identification in the $2\ell + 1\tau_{\text{had}}$ region.



(a) (b)

(c)

Figure D.3: NN response of true and fake 3-pronged τ_{had} in MC simulation in the $2\ell + 1\tau_{\text{had}}$ signal region. The response plots compare the signal (tHq) sample shape after training against other processes. This comparison is made against tZq (a), $t\bar{t}$ (b), and other MC (c).



(a) 1-prong (b) 3-prong

Figure D.4: NN response stack plot in the $2\ell + 1\tau_{\text{had}}$ signal region. MC is further split by true and fake τ_{had}. tHq yields are increased 1 000-fold for visibility.

APPENDIX E

Likelihood Fit

This section of the appendix is paraphrased from [132]. Here, the description of what a likelihood function and fit can be found.

The likelihood function is a tool used to maximize the probability that the model describes the data. It takes the parametrized model with the best fit one provides and gives the representation of how likely different parameters for the distribution are. In this analysis, the cross-section is measured using the principle of *maximum likelihood*. This is the procedure of finding the parameters of a model in order to maximize a known likelihood function.

One can define the total events in the signal region, n, as a sum over signal and background processes:

$$n(\mu) = \mathcal{L}\epsilon_0\sigma_0\mu + \mathcal{L}\sum_{j}^{\text{bkg.}}\epsilon_j\sigma_j, \tag{E.1}$$

where $\mathcal{L}$ denotes the integrated luminosity, ϵ_j is the efficiency of reconstruction and selection of events of background process j, and σ_j is the cross-section for each background process. ϵ_0 and σ_0 are the efficiency and cross-section for the tW process, respectively. The theoretical cross-section is used multiplied with a scaling parameter, μ, in order to extract the tW cross-section.

The probability of observing N events (without accounting for uncertainties) in an interval follows a Poisson distribution:

$$P(N) = e^{-r}\frac{r^N}{N!}, \tag{E.2}$$

where r is the mean number of events per interval. With the total number of events previously defined, one can then write the likelihood function for one bin as:

$$P(N, n) = e^{-n(\mu)}\frac{n(\mu)^N}{N!}. \tag{E.3}$$

Thus the probabilities can be multiplied, defining the likelihood function for all bins:

$$L(\boldsymbol{N}, \boldsymbol{n}) = \prod_{i}^{\text{bins}} P(N_i, n_i), \tag{E.4}$$

$$= \prod_{i}^{\text{bins}} e^{-n_i(\mu)} \frac{n_i(\mu)^{N_i}}{N_i!}, \tag{E.5}$$

where $\boldsymbol{N}$ and $\boldsymbol{n}$ are now vectors and the i subscript denotes the bin number.

Commonly, the natural logarithm of the likelihood function, known as *log-likelihood*, is used instead for convenience. This logarithmic function is strictly increasing, it achieves its maximum value at same points as the function itself, and in this case turns multiplication over bins into addition. The equation is as follows:

$$\Lambda(\boldsymbol{N}, \boldsymbol{n}) = -2 \ln(L(\boldsymbol{N}, \boldsymbol{n})), \tag{E.6}$$

where the factor of -2 is introduced purely for convention. The most likely value of the parameter of interest, $\hat{\mu}$, is the closest to μ and maximizes the likelihood function, L, or minimizes the log-likelihood function, Λ. The error for the nuisance parameter is calculated by shifting $\hat{\mu}$ until the log-likelihood function increases by one unit.

With one source of uncertainty included, the nuisance parameter, θ, is added to account for the influence that this systematic uncertainty has on the total number of events. θ has an influence on the total number of events, denoted by δ, that is estimated by varying the nuisance parameter by one standard deviation ($\theta = \pm 1$). This is shown in the following equations:

$$n_i(\mu, \theta) = n_i(\mu)(1 + \delta \cdot \theta), \tag{E.7}$$

$$L(\boldsymbol{N}, \boldsymbol{n}(\mu, \theta)) = \prod_{i}^{\text{bins}} [P(N_i n_i)] f_{\mathcal{N}}(\theta), \tag{E.8}$$

$$= L_{\text{nom.}}(\boldsymbol{N}, \boldsymbol{n}(\mu)) f_{\mathcal{N}}(\theta), \tag{E.9}$$

where the nom. subscript denotes a θ-less likelihood function, and $f_{\mathcal{N}}(\theta)$ is a normal Gaussian distribution describing the probability density distribution of θ. The calculation of δ is generally done by modifying the parameters in the simulation or in reconstruction objects. For example, consider some systematic k that can be varied such that the number of events is either increased or decreased in a given bin, denoted as N_k^+ and N_k^- respectively. Then delta is calculated as:

$$\delta_k = \frac{N_k^+ - N_k^-}{2N}, \tag{E.10}$$

where N is the total number of events in that bin. More systematic uncertainties have an additive effect to the total number of events which translates to a multiplicative effect on the likelihood function. This can then be written as follows:

$$n_i(\mu, \theta) = n_i(\mu)(1 + \sum_k^{\text{unc.}} \delta_k \cdot \theta_k), \tag{E.11}$$

$$L(\boldsymbol{N}, \boldsymbol{n}(\mu, \theta)) = L_{\text{nom.}}(\boldsymbol{N}, \boldsymbol{n}(\mu)) \prod_k^{\text{unc.}} f_{\mathcal{N}}(\theta_k), \tag{E.12}$$

$$\Lambda(\boldsymbol{N}, \boldsymbol{n}) = \Lambda_{\text{nom.}}(\boldsymbol{N}, \boldsymbol{n}) + \sum_k^{\text{unc.}} \ln(f_{\mathcal{N}}(\theta_k)), \tag{E.13}$$

where k denotes the systematic associated with the nuisance parameter, θ, and error, δ. When there are multiple nuisance parameters, the log-likelihood function is minimized not just in μ but also in θ. Generally, the most likely nuisance parameter associated with an uncertainty, $\hat{\theta}$, can minimize the log-likelihood function for $\mu \neq \hat{\mu}$ thus broadening the parabolic shape. This broadened log-likelihood increases the uncertainty associated with $\hat{\mu}$.

Furthermore, the error intervals and contours can be approximated using a covariance matrix of the parameter estimates. The matrix is defined as follows:

$$\hat{V}_{ab}^{-1} = -\frac{\partial^2 L}{\partial \hat{\theta}_a \partial \hat{\theta}_b}, \tag{E.14}$$

where the subscripts a and b denote the uncertainty associated to the estimator $\hat{\theta}$. The μ parameter can be easily included by defining $\hat{\theta}_0 \stackrel{\text{def}}{=} \hat{\mu}$.

The *impact* of a systematic uncertainty on the measurement of the cross-section is to be calculated. This is defined as the shift of the maximum-likelihood estimate for μ when the nuisance parameter θ_k is shifted by $\pm\Delta\theta_k$. Similarly to the estimation of error intervals, this can be approximated using a covariance matrix:

$$\frac{\text{cov}[\hat{\mu}, \hat{\theta}_k]}{\sqrt{\text{cov}[\hat{\theta}_k, \hat{\theta}_k]}}. \tag{E.15}$$

Nuisance parameters are estimated from the MC simulation but are actually measured from, or constrained by, the data. This measurement is done by constraining processes as they have different event yields per bin. In more detail, bins that contain different signal to background ratios can be used to constrain the uncertainties in these processes. Furthermore, bins with no signal events can limit the uncertainties of background processes to the statistical uncertainty. Therefore, sufficient amount of data can reduce uncertainties to great effect.

The likelihood fit is performed by the TRExFitter program. It is capable of performing a *profile likelihood* fit, or a likelihood fit for models with more than one unknown parameter. The program builds a global likelihood function for all the bins and includes all parameters. One parameter in particular, denoted as the parameter of interest (POI), can be measured by performing a log-likelihood minimization on the global likelihood function.

APPENDIX F

DR/DS Autoencoder Distributions

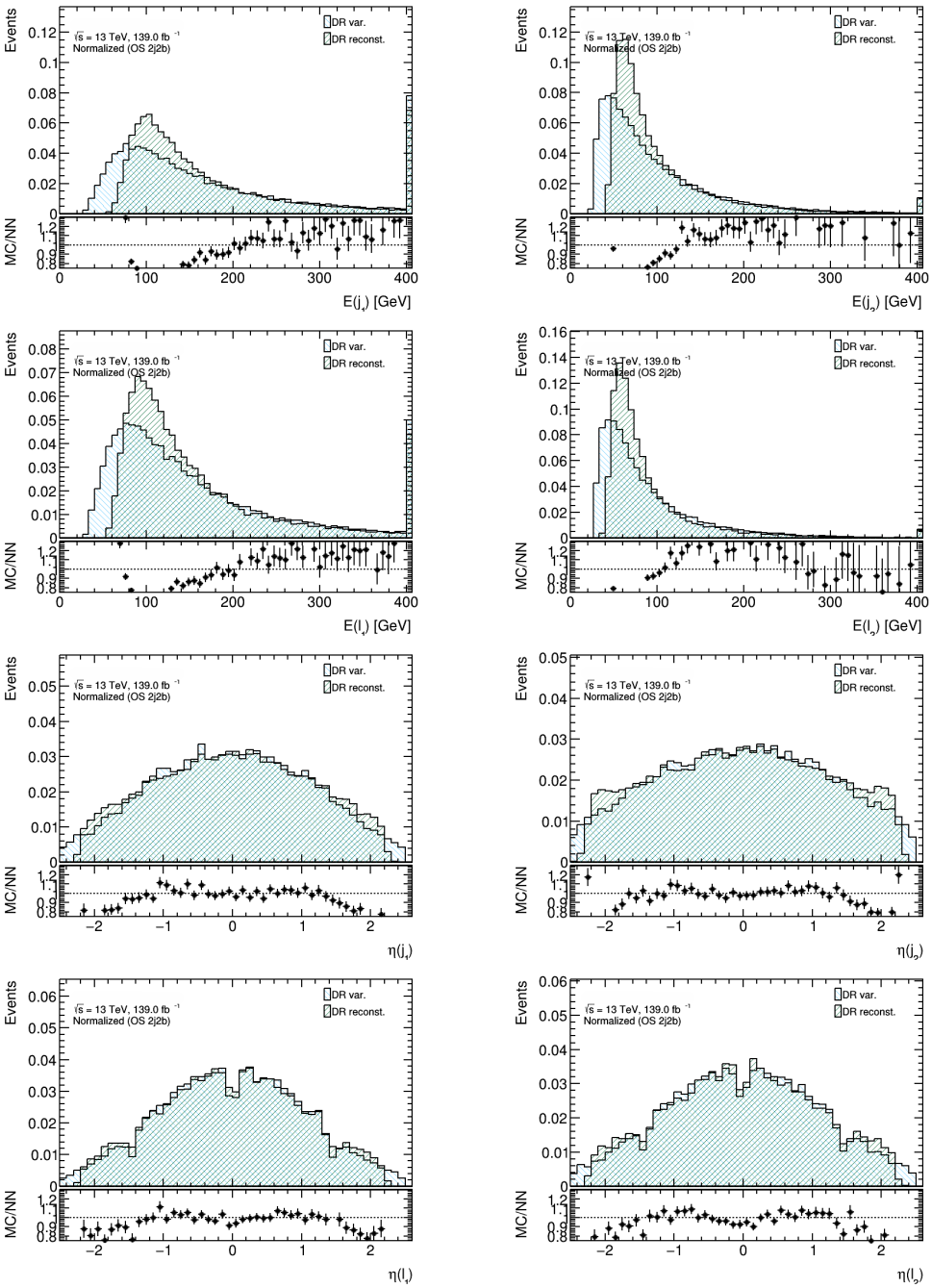


Figure F.1: Variables and their reconstructed values for the DR sample.

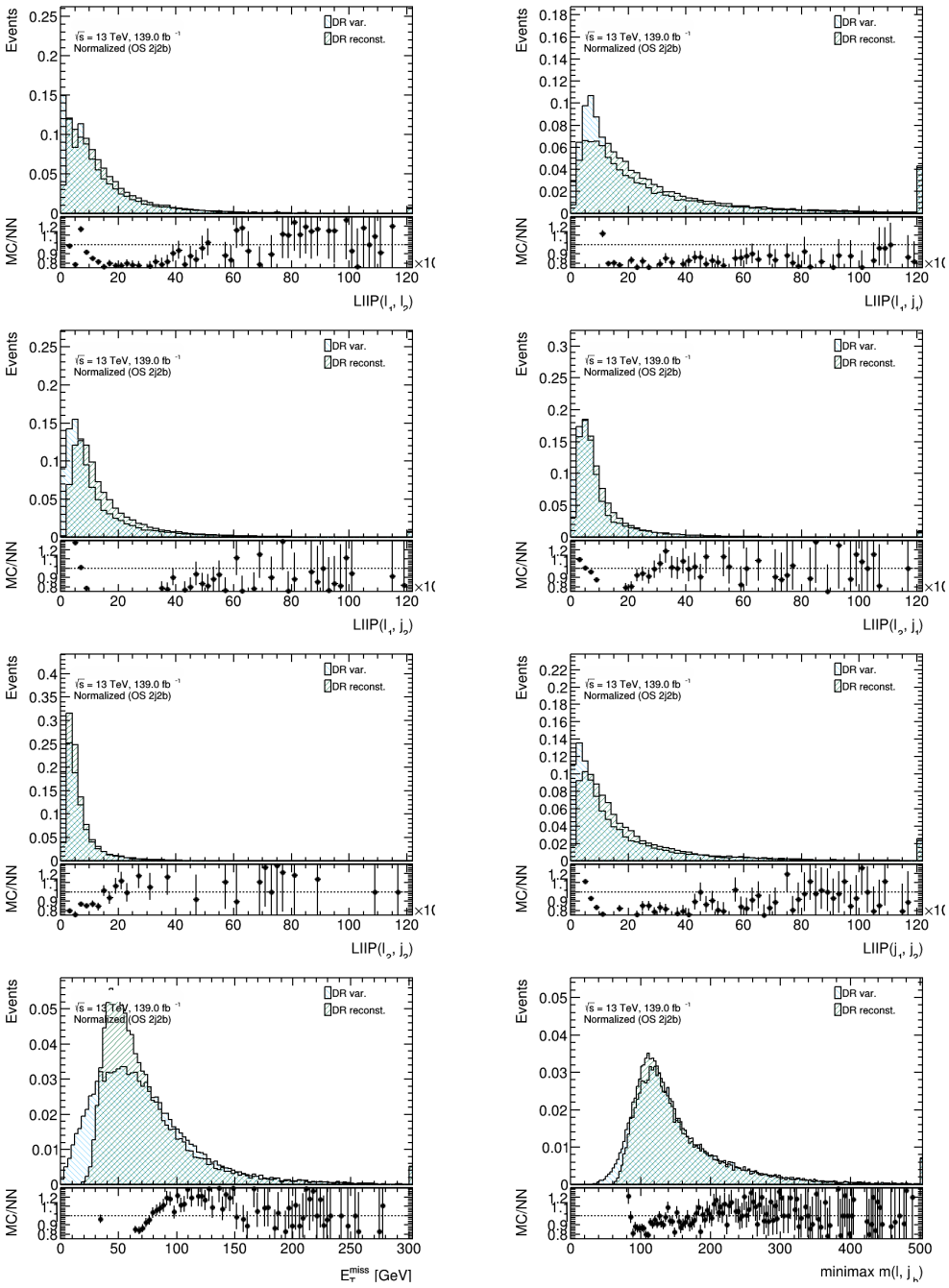


Figure F.2: Variables and their reconstructed values for the DR sample.

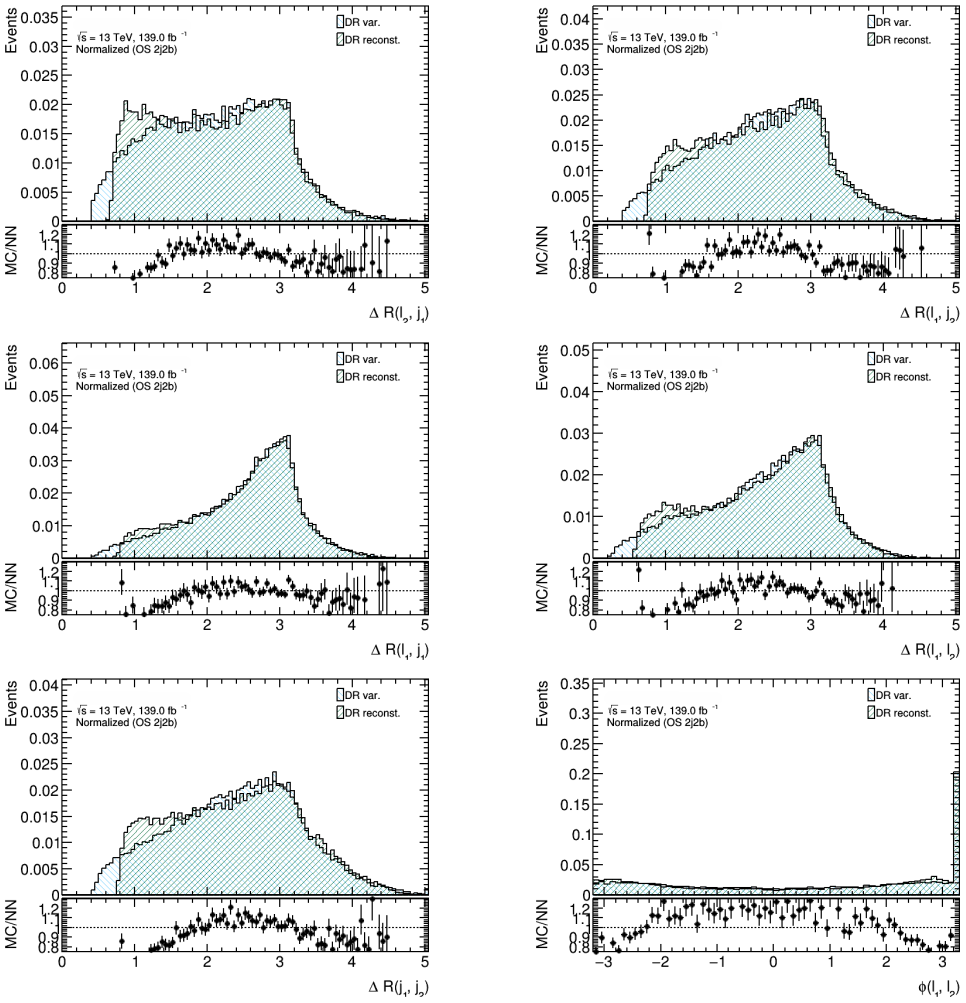


Figure F.3: Variables and their reconstructed values for the DR sample.

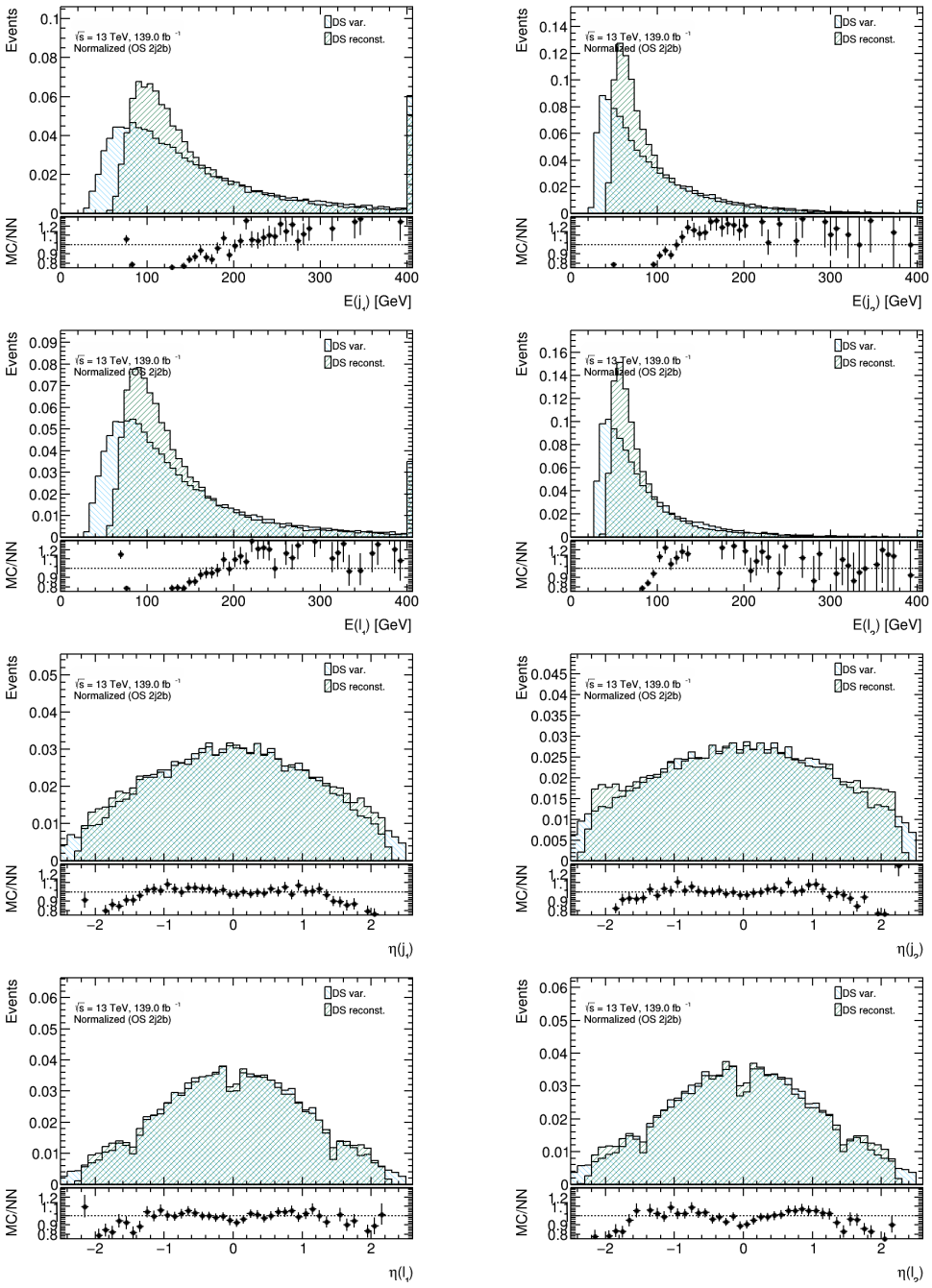


Figure F.4: Variables and their reconstructed values for the DS sample.

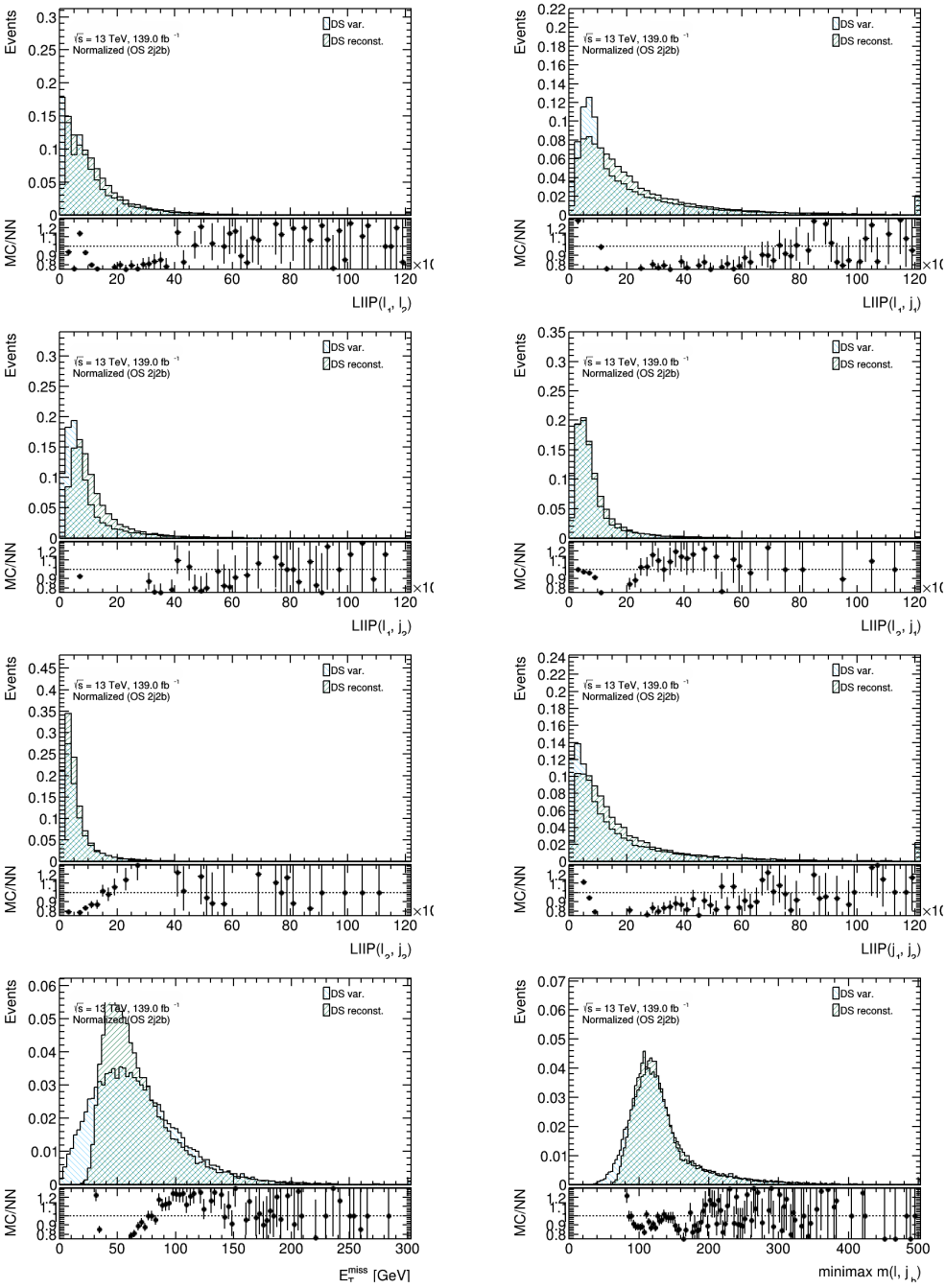


Figure F.5: Variables and their reconstructed values for the DS sample.

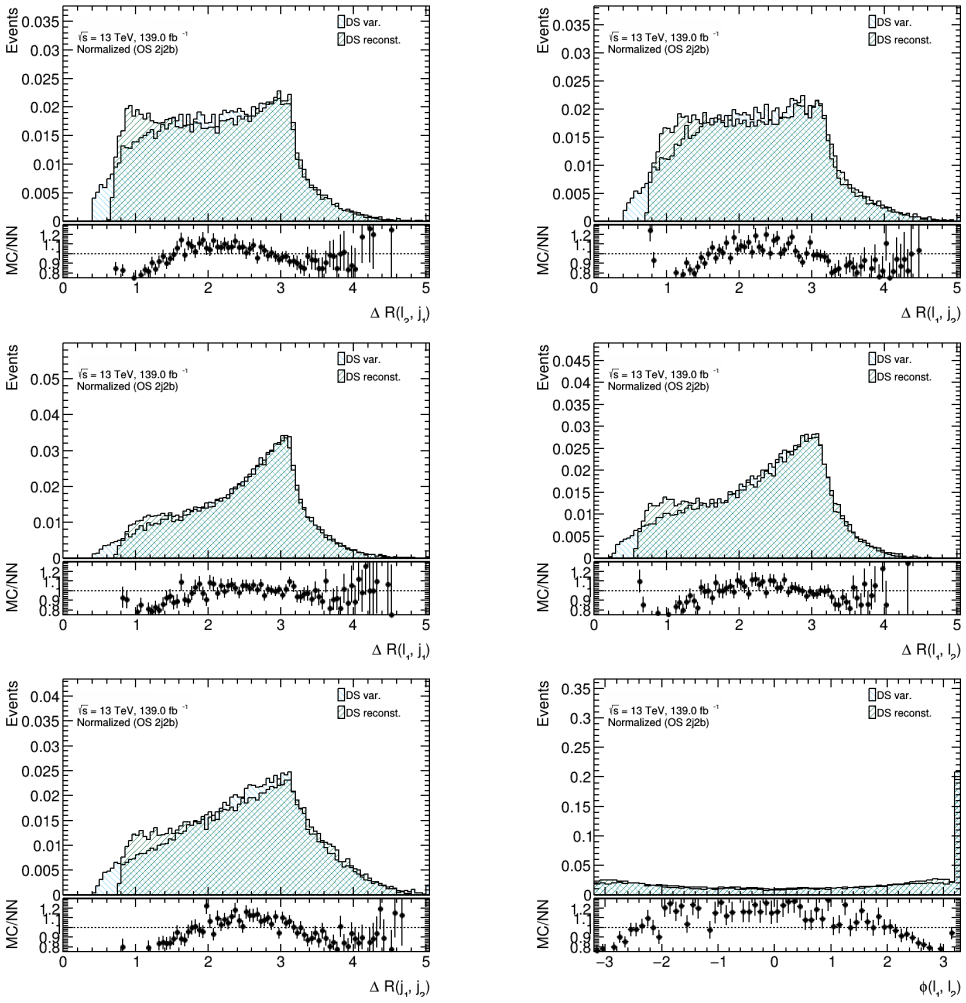


Figure F.6: Variables and their reconstructed values for the DS sample.

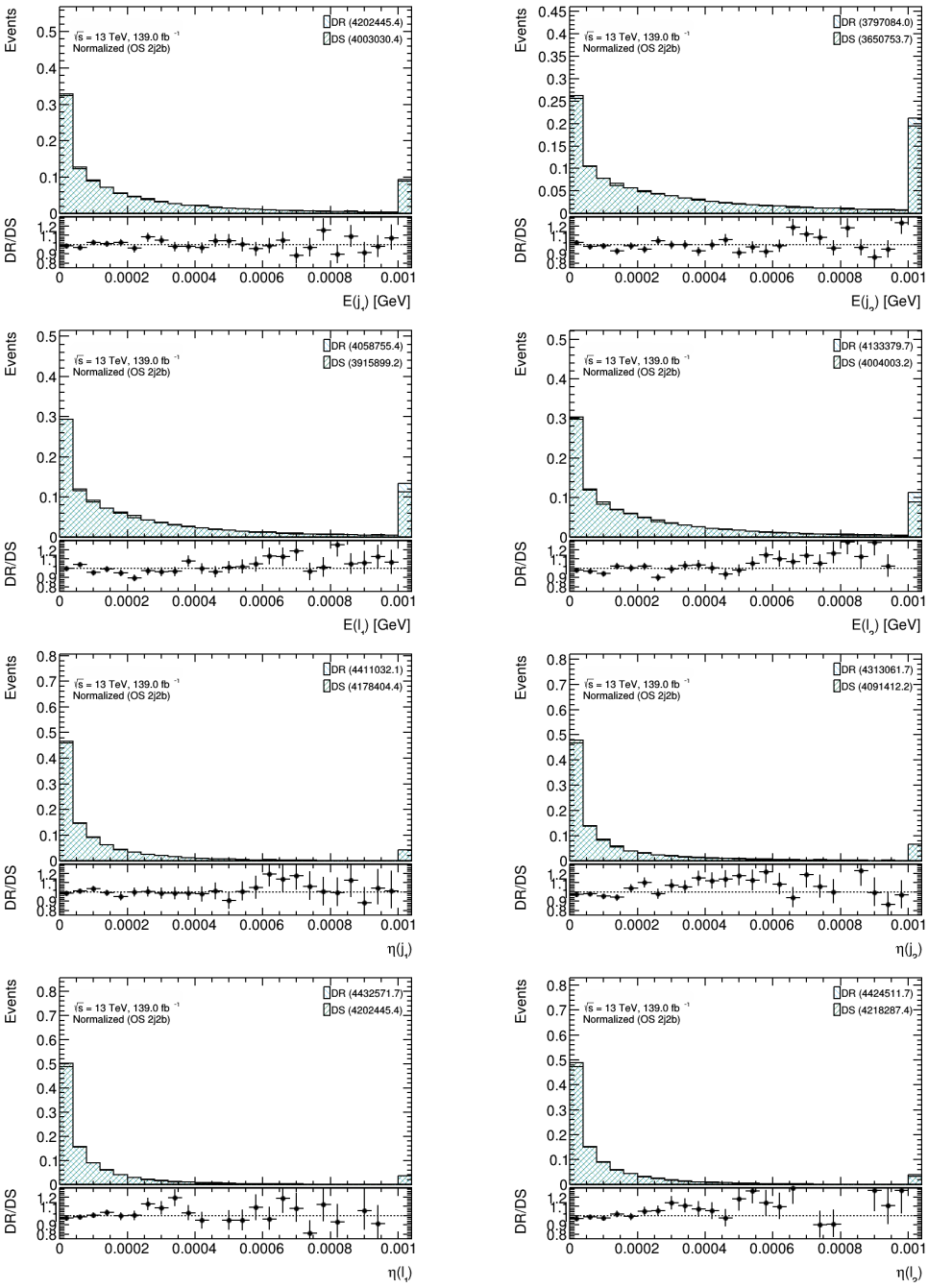


Figure F.7: Reconstruction error comparisons between the DR and DS samples

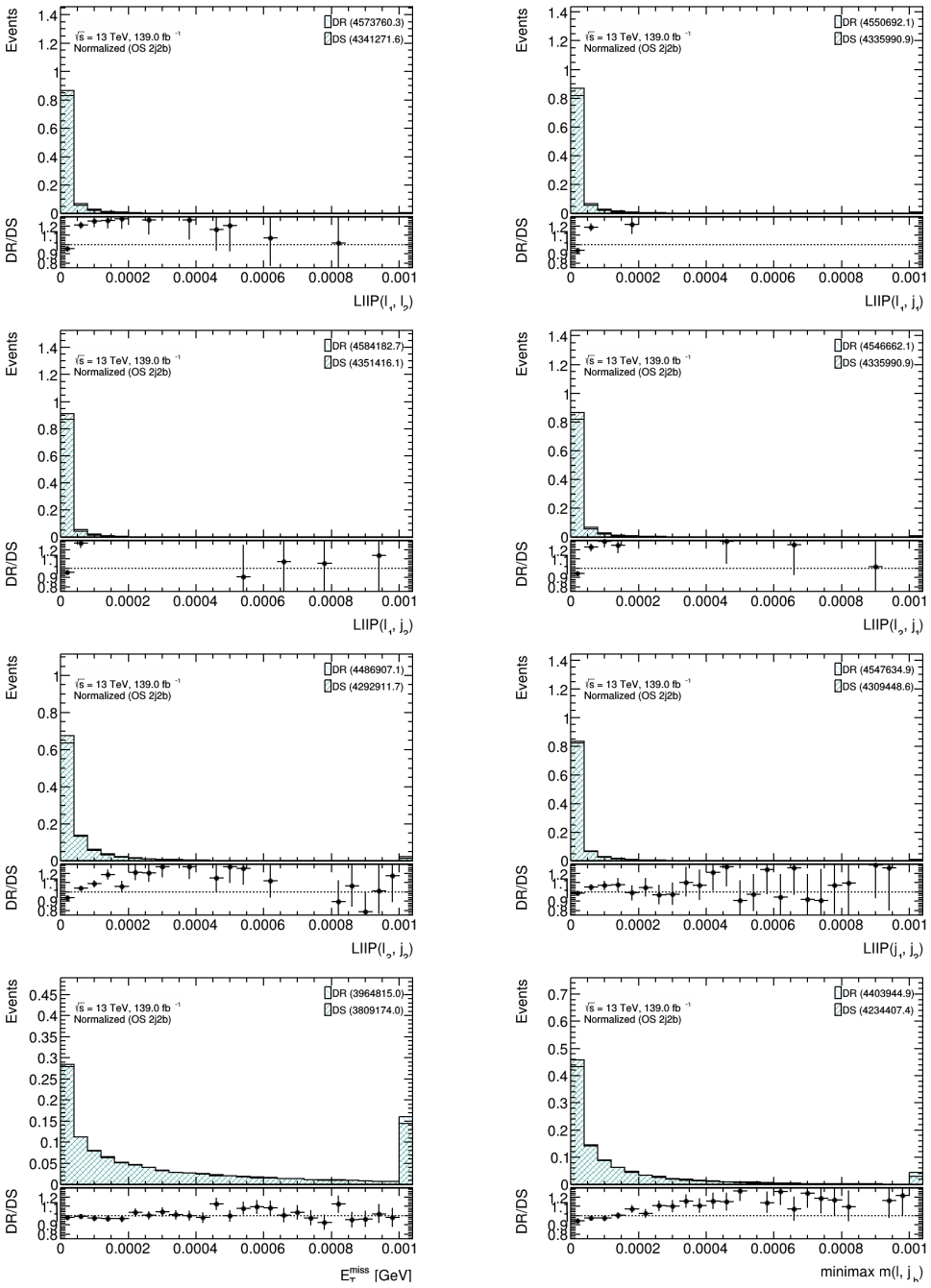


Figure F.8: Reconstruction error comparisons between the DR and DS samples

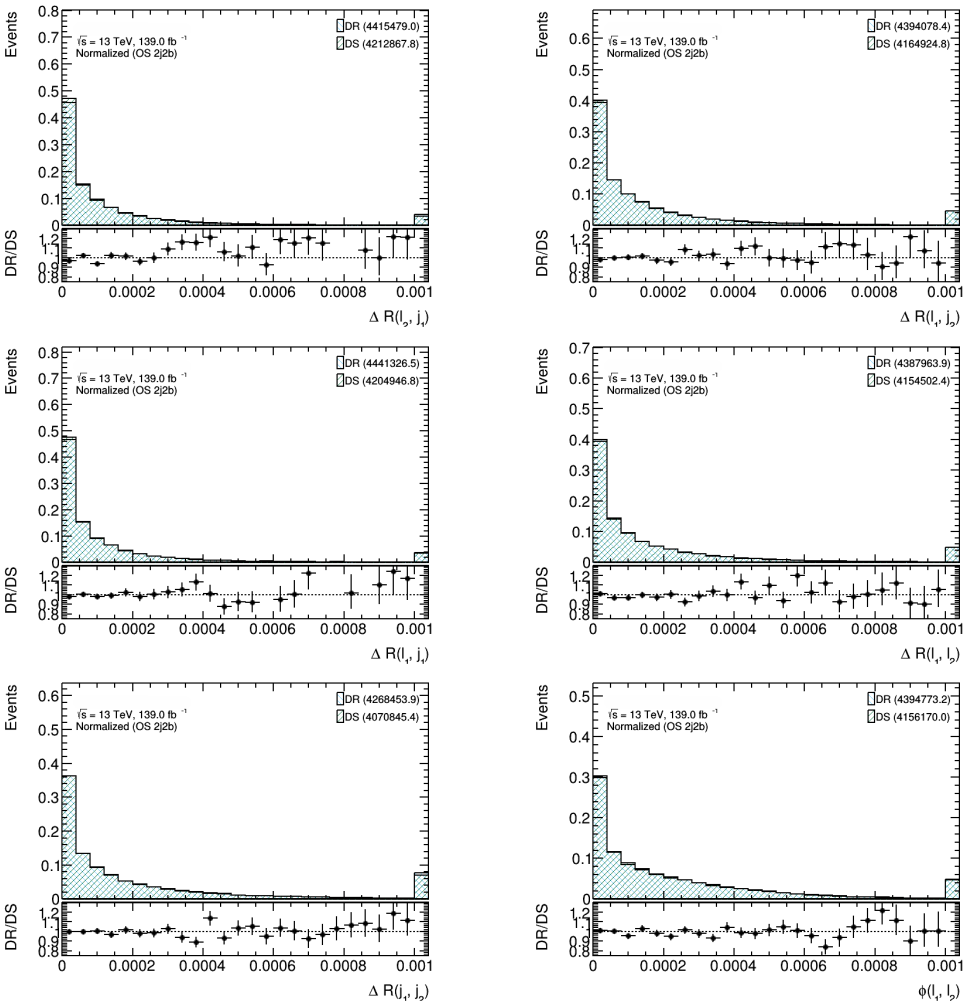


Figure F.9: Reconstruction error comparisons between the DR and DS samples

Bibliography

[1] The ATLAS Collaboration, "Standard Model Total Production Cross Section Measurements", 2017, URL: https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/CombinedSummaryPlots/SM/ (cit. on p. 4).

[2] Wikipedia, *The Standard Model — Wikipedia, The Free Encyclopedia*, [Online; accessed 08-01-2018], 2004, URL: https://en.wikipedia.org/wiki/Standard_Model (cit. on p. 5).

[3] R. Aaij et al., *Observation of the Resonant Character of the $Z(4430)^-$ State*, Phys. Rev. Lett. **112** (22 2014) 222002, URL: https://link.aps.org/doi/10.1103/PhysRevLett.112.222002 (cit. on p. 6).

[4] R. Aaij et al., *Observation of $J/\psi p$ Resonances Consistent with Pentaquark States in $\Lambda_b^0 \to J/\psi K^- p$ Decays*, Phys. Rev. Lett. **115** (7 2015) 072001, URL: https://link.aps.org/doi/10.1103/PhysRevLett.115.072001 (cit. on p. 6).

[5] D. J. Griffiths, *Introduction to elementary particles; 2nd rev. version*, Physics textbook, New York, NY: Wiley, 2008, URL: https://cds.cern.ch/record/111880 (cit. on pp. 6–8).

[6] B. Odom, D. Hanneke, B. D'Urso and G. Gabrielse, *New Measurement of the Electron Magnetic Moment Using a One-Electron Quantum Cyclotron*, Phys. Rev. Lett. **97** (3 2006) 030801, URL: https://link.aps.org/doi/10.1103/PhysRevLett.97.030801 (cit. on p. 9).

[7] R. L. Workman et al., *Review of Particle Physics*, PTEP **2022** (2022) 083C01 (cit. on pp. 10, 11, 22, 24, 37, 42, 46, 92).

[8] H1 and ZEUS Collaborations, *Combination of Measurements of Inclusive Deep Inelastic $e^{\pm}p$ Scattering Cross Sections and QCD Analysis of HERA Data*, ArXiv e-prints (2015), arXiv: 1506.06042 [hep-ex] (cit. on p. 14).

[9] *Luminosity determination in pp collisions at $\sqrt{s}$ = 8 TeV using the ATLAS detector at the LHC*, The European Physical Journal C **76** (2016), URL: https://doi.org/10.1140%2Fepjc%2Fs10052-016-4466-1 (cit. on p. 16).

[10] *Luminosity determination in pp collisions at $\sqrt{s}$ = 13 TeV using the ATLAS detector at the LHC*, 2022, URL: https://arxiv.org/abs/2212.09379 (cit. on p. 16).

[11] M. Cacciari, G. P. Salam and G. Soyez, *The anti-k_t jet clustering algorithm*, Journal of High Energy Physics **4**, 063 (2008) 063, arXiv: 0802.1189 [hep-ph] (cit. on pp. 17, 34, 36).

[12] M. Czakon and A. Mitov,
Top++: A program for the calculation of the top-pair cross-section at hadron colliders,
Computer Physics Communications **185** (2014) 2930,
URL: https://doi.org/10.1016%2Fj.cpc.2014.06.021 (cit. on p. 18).

[13] M. Czakon, P. Fiedler and A. Mitov,
Total Top-Quark Pair-Production Cross Section at Hadron Colliders Through $O(\alpha_S^4)$,
Phys. Rev. Lett. **110** (25 2013) 252004,
URL: https://link.aps.org/doi/10.1103/PhysRevLett.110.252004 (cit. on p. 18).

[14] M. Czakon, M. L. Mangano, A. Mitov and J. Rojo,
Constraints on the gluon PDF from top quark pair production at hadron colliders,
Journal of High Energy Physics **2013** (2013),
URL: https://doi.org/10.1007%2Fjhep07%282013%29167 (cit. on p. 18).

[15] W. Wagner,
NNLO+NNLL top-quark-pair cross sections. ATLAS-CMS recommended predictions for top-quark-pair cross sections using the top++v2.0 program (M. Czakon, A. Mitov, 2013),
2022, URL: https://twiki.cern.ch/twiki/bin/view/LHCPhysics/TtbarNNLO
(visited on 01/09/2022) (cit. on p. 18).

[16] C. Escobar, *NLO single-top channel cross sections. ATLAS-CMS recommended predictions for single-top cross sections using the Hathor v2.1 program*, 2017,
URL: https://twiki.cern.ch/twiki/bin/view/LHCPhysics/SingleTopRefXsec
(visited on 01/09/2022) (cit. on p. 18).

[17] P. Kant et al., *HatHor for single top-quark production: Updated predictions and uncertainty estimates for single top-quark production in hadronic collisions*,
Computer Physics Communications **191** (2015) 74,
URL: https://doi.org/10.1016%2Fj.cpc.2015.02.001 (cit. on p. 18).

[18] M. Aliev et al., *HATHOR – HAdronic Top and Heavy quarks crOss section calculatoR*,
Computer Physics Communications **182** (2011) 1034,
URL: https://doi.org/10.1016%2Fj.cpc.2010.12.040 (cit. on p. 18).

[19] N. Kidonakis, *Top Quark Production*, 2013, URL: https://arxiv.org/abs/1311.0283
(cit. on p. 18).

[20] N. Kidonakis, *Two-loop soft anomalous dimensions for single top quark associated production with a W- or H-*, Physical Review D **82** (2010),
URL: https://doi.org/10.1103%2Fphysrevd.82.054018 (cit. on p. 18).

[21] R. Zhang, *Measurement of tW differential cross-sections with ATLAS at 13 TeV*, 2018,
arXiv: 1809.01433, URL: https://cds.cern.ch/record/2637450 (cit. on pp. 19, 42).

[22] S. Frixione, E. Laenen, P. Motylinski, C. White and B. R. Webber,
Single-top hadroproduction in association with a W boson,
Journal of High Energy Physics **2008** (2008) 029,
URL: https://doi.org/10.1088%2F1126-6708%2F2008%2F07%2F029 (cit. on p. 21).

[23] T. Loddenkötter, *Implementation of a kinematic fit of single top-quark production in association with a W boson and its application in a neural-network-based analysis in ATLAS*, BONN-IR-2012-06, PhD Thesis: University of Bonn, 2012, URL: http://hss.ulb.uni-bonn.de/diss_online (cit. on p. 21).

[24] F. Demartin, B. Maier, F. Maltoni, K. Mawatari and M. Zaro, *tWH associated production at the LHC*, The European Physical Journal C **77** (2017), URL: https://doi.org/10.1140%2Fepjc%2Fs10052-017-4601-7 (cit. on p. 21).

[25] T. M. P. Tait, *tW^- mode of single top quark production*, Physical Review D **61** (1999), URL: https://doi.org/10.1103%2Fphysrevd.61.034001 (cit. on p. 21).

[26] D. Pagani, I. Tsinikos and E. Vryonidou, *NLO QCD+EW predictions for tHj and tZj production at the LHC*, 2020, URL: https://arxiv.org/abs/2006.10086 (cit. on p. 22).

[27] F. Demartin, F. Maltoni, K. Mawatari and M. Zaro, *Higgs production in association with a single top quark at the LHC*, 2015, URL: https://arxiv.org/abs/1504.00611 (cit. on pp. 22, 44).

[28] D. de Florian et al., *Handbook of LHC Higgs Cross Sections: 4. Deciphering the Nature of the Higgs Sector*, CERN Yellow Reports: Monographs, 869 pages, 295 figures, 248 tables and 1645 citations. Working Group web page: https://twiki.cern.ch/twiki/bin/view/LHCPhysics/LHCHXSWG, Geneva: CERN, 2017, URL: https://cds.cern.ch/record/2227475 (cit. on p. 23).

[29] *Reconstruction, Identification, and Calibration of hadronically decaying tau leptons with the ATLAS detector for the LHC Run 3 and reprocessed Run 2 data*, tech. rep., All figures including auxiliary figures are available at https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-PHYS-PUB-2022-044: CERN, 2022, URL: https://cds.cern.ch/record/2827111 (cit. on p. 24).

[30] *Identification of hadronic tau lepton decays using neural networks in the ATLAS experiment*, tech. rep., All figures including auxiliary figures are available at https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-PHYS-PUB-2019-033: CERN, 2019, URL: http://cds.cern.ch/record/2688062 (cit. on pp. 24, 37, 91, 95, 97).

[31] *Cryogenics: Low temperatures, high performance*, URL: https://home.cern/science/engineering/cryogenics-low-temperatures-high-performance (cit. on p. 25).

[32] L. Evans and L. Linssen, *The Super-LHC is on the starting blocks*, CERN Courier (2008) (cit. on p. 26).

[33] *The ATLAS Experiment at the CERN Large Hadron Collider*, Journal of Instrumentation **3** (2008) S08003, URL: https://doi.org/10.1088/1748-0221/3/08/s08003 (cit. on p. 25).

[34] T. C. Collaboration et al., *The CMS experiment at the CERN LHC*, Journal of Instrumentation **3** (2008) S08004, URL: https://doi.org/10.1088/1748-0221/3/08/s08004 (cit. on p. 25).

[35] T. L. Collaboration et al., *The LHCb Detector at the LHC*, Journal of Instrumentation **3** (2008) S08005, URL: https://doi.org/10.1088/1748-0221/3/08/s08005 (cit. on p. 25).

[36] T. A. Collaboration et al., *The ALICE experiment at the CERN LHC*, Journal of Instrumentation **3** (2008) S08002, URL: https://doi.org/10.1088/1748-0221/3/08/s08002 (cit. on p. 25).

[37] M. Lamont, *Status of the LHC*, Journal of Physics: Conference Series **455** (2013) 012001, URL: https://doi.org/10.1088/1742-6596/455/1/012001 (cit. on p. 26).

[38] *CERN's large hadron collider gears up for run 2*, URL: https://home.cern/news/news/accelerators/cerns-large-hadron-collider-gears-run-2 (cit. on p. 26).

[39] *The third run of the Large Hadron Collider has successfully started*, URL: https://home.cern/news/news/cern/third-run-large-hadron-collider-has-successfully-started (cit. on p. 27).

[40] *Public atlas luminosity results for run-2 of the LHC*, URL: https://twiki.cern.ch/twiki/bin/view/AtlasPublic/LuminosityPublicResultsRun2 (cit. on pp. 27, 40).

[41] J. JPequenao, "Computer generated image of the whole ATLAS detector", 2008, URL: https://cds.cern.ch/record/1095924 (cit. on p. 28).

[42] The ATLAS Collaboration, *Expected Performance of the ATLAS Experiment - Detector, Trigger and Physics*, ArXiv e-prints (2009), arXiv: 0901.0512 [hep-ex] (cit. on p. 28).

[43] *Long shutdown 2*, URL: https://atlas.cern/Discover/Detector/Long-Shutdown-2 (cit. on pp. 28, 32).

[44] M. Capeans et al., *ATLAS Insertable B-Layer Technical Design Report*, tech. rep. CERN-LHCC-2010-013. ATLAS-TDR-19, 2010, URL: https://cds.cern.ch/record/1291633 (cit. on p. 29).

[45] *ATLAS magnet system: Technical Design Report, 1*, tech. rep., 1997, URL: https://cds.cern.ch/record/338080 (cit. on p. 31).

[46] ATLAS Collaboration, *The Run-2 ATLAS Trigger System*, 2016, URL: https://cds.cern.ch/record/2133909 (cit. on p. 31).

[47] J. Pequenao and P. Schaffner, "An computer generated image representing how ATLAS detects particles", 2013, URL: https://cds.cern.ch/record/1505342 (cit. on p. 33).

[48] K. Grimm et al., *Primary vertex reconstruction at the ATLAS experiment*, tech. rep., CERN, 2017, URL: https://cds.cern.ch/record/2253428 (cit. on p. 34).

[49] T. Cornelissen et al., *Concepts, Design and Implementation of the ATLAS New Tracking (NEWT)*, tech. rep., All figures including auxiliary figures are available at https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-SOFT-PUB-2007-007: CERN, 2007, URL: https://cds.cern.ch/record/1020106 (cit. on p. 34).

[50] ATLAS Collaboration, *Performance of the ATLAS track reconstruction algorithms in dense environments in LHC Run 2*, The European Physical Journal C **77** (2017), URL: https://doi.org/10.1140%2Fepjc%2Fs10052-017-5225-7 (cit. on p. 34).

[51] *Performance of the ATLAS Silicon Pattern Recognition Algorithm in Data and Simulation at* $\sqrt{s} = 7$ *TeV*, tech. rep., All figures including auxiliary figures are available at https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/CONFNOTES/ATLAS-CONF-2010-072: CERN, 2010, URL: https://cds.cern.ch/record/1281363 (cit. on p. 34).

[52] M. Elsing, *ATLAS Track and Vertex Reconstruction for Run-3 and High-Luminosity LHC*, (2021), URL: https://cds.cern.ch/record/2780986 (cit. on p. 34).

[53] N. Calace, *Track and Vertex reconstruction in ATLAS for LHC Run-3 and High-Luminosity phases*, (2021), URL: https://cds.cern.ch/record/2777660 (cit. on p. 34).

[54] S. D. Ellis and D. E. Soper, *Successive combination jet algorithm for hadron collisions*, Physical Review D **48** (1993) 3160, URL: https://doi.org/10.1103%2Fphysrevd.48.3160 (cit. on p. 34).

[55] M. Wobisch and T. Wengler, *Hadronization Corrections to Jet Cross Sections in Deep-Inelastic Scattering*, 1999, URL: https://arxiv.org/abs/hep-ph/9907280 (cit. on p. 34).

[56] W. Lampl et al., *Calorimeter Clustering Algorithms: Description and Performance*, tech. rep., All figures including auxiliary figures are available at https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-LARG-PUB-2008-002: CERN, 2008, URL: https://cds.cern.ch/record/1099735 (cit. on p. 35).

[57] B. Pereira, *Performance of the ATLAS/LHC Tile Calorimeter Plastic Scintillators*, (2022), URL: https://cds.cern.ch/record/2835898 (cit. on p. 35).

[58] R. Bouquet, *Jets and missing transverse energy reconstruction and calibration in ATLAS*, (2022), URL: https://cds.cern.ch/record/2826560 (cit. on pp. 35, 36).

[59] A. Ahmad, *The ATLAS Tile Calorimeter Performance and Its Upgrade towards the High-Luminosity LHC*, Moscow Univ. Phys. Bull. **77** (2022) 156, URL: https://cds.cern.ch/record/2836170 (cit. on p. 35).

[60] N. Fritzsche, *Machine Learning for Real-Time Processing of ATLAS Liquid Argon Calorimeter Signals with FPGAs*, (2022), URL: https://cds.cern.ch/record/2826542 (cit. on p. 35).

[61] ATLAS Collaboration, *Electron efficiency measurements with the ATLAS detector using the 2015 LHC proton–proton collision data*, ATLAS-CONF-2016-024, 2016, URL: https://cds.cern.ch/record/2157687 (cit. on p. 35).

[62] *Electron and photon performance measurements with the ATLAS detector using the 2015–2017 LHC proton-proton collision data*, Journal of Instrumentation **14** (2019) P12006, URL: https://doi.org/10.1088%2F1748-0221%2F14%2F12%2Fp12006 (cit. on p. 35).

[63] ATLAS Collaboration, *ATLAS electron, photon and muon isolation in Run 2*, This note contains the Moriond 2017 recommendations. It will be updated when new recommendations become available., 2017, URL: https://cds.cern.ch/record/2256658 (cit. on p. 35).

[64] R. Ospanov, R. T. Roberts and T. R. Wyatt, *Tagging non-prompt electrons and muons*, tech. rep., CERN, 2016, URL: https://cds.cern.ch/record/2220954 (cit. on p. 35).

[65] *Electron and photon energy calibration with the ATLAS detector using data collected in 2015 at* $\sqrt{s}$ = 13TeV, tech. rep., All figures including auxiliary figures are available at https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-PHYS-PUB-2016-015: CERN, 2016, URL: https://cds.cern.ch/record/2203514 (cit. on p. 35).

[66] J. Illingworth and J. Kittler, *A survey of the hough transform*, Computer Vision, Graphics, and Image Processing **44** (1988) 87, ISSN: 0734-189X, URL: https://www.sciencedirect.com/science/article/pii/S0734189X88800331 (cit. on p. 36).

[67] ATLAS Collaboration, *Muon reconstruction and identification efficiency in ATLAS using the full Run 2 pp collision data set at* $\sqrt{s}$ = 13 *TeV*, Eur. Phys. J., C **81** (2021) 578. 44 p, 64 pages in total, author list starting page 42, auxiliary material starting at page 59, 34 figures, 3 tables. All figures including auxiliary figures are available at https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PAPERS/MUON-2018-03/, arXiv: 2012.00578, URL: https://cds.cern.ch/record/2746302 (cit. on p. 36).

[68] *Jet reconstruction and performance using particle flow with the ATLAS Detector*, The European Physical Journal C **77** (2017), URL: https://doi.org/10.1140%2Fepjc%2Fs10052-017-5031-2 (cit. on p. 36).

[69] J. Roloff and A. Collaboration, *Pileup Mitigation*, (2017), URL: https://cds.cern.ch/record/2276691 (cit. on p. 36).

[70] T. Ingebretsen Carlson, *Optimisation of the ATLAS jet vertex tagger for particle flow jets through track-to-vertex-association improvements.*, tech. rep., CERN, 2021, URL: https://cds.cern.ch/record/2789416 (cit. on pp. 36, 42).

[71] A. Collaboration, *Jet energy scale and resolution measured in proton–proton collisions at* $\sqrt{s}$ = 13 *TeV with the ATLAS detector*, Eur. Phys. J. C **81** (2021) 689. 73 p, 73 pages in total, author list starting page 57, 31 figures, 2 tables, submitted to Eur. Phys. J. C. All figures including auxiliary figures are available at http://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PAPERS/JETM-2018-05, arXiv: 2007.02645, URL: https://cds.cern.ch/record/2722869 (cit. on p. 36).

[72] ATLAS Collaboration, *Performance of b-jet identification in the ATLAS experiment*, Journal of Instrumentation **11** (2016) P04008, arXiv: 1512.01094 [hep-ex] (cit. on p. 36).

[73] *Optimisation of the ATLAS b-tagging performance for the 2016 LHC Run*, 2016, URL: https://cds.cern.ch/record/2160731 (cit. on p. 36).

[74] *Optimisation and performance studies of the ATLAS b-tagging algorithms for the 2017-18 LHC run*, tech. rep., All figures including auxiliary figures are available at https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-PHYS-PUB-2017-013: CERN, 2017, URL: https://cds.cern.ch/record/2273281 (cit. on p. 36).

[75] *Identification of Jets Containing b-Hadrons with Recurrent Neural Networks at the ATLAS Experiment*, tech. rep., All figures including auxiliary figures are available at https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-PHYS-PUB-2017-003: CERN, 2017, URL: https://cds.cern.ch/record/2255226 (cit. on p. 36).

[76] L. Pereira Sanchez, *Calibration of flavour tagging algorithms in ATLAS on $t\bar{t}$ and Z+jets final states*, tech. rep., CERN, 2022, URL: https://cds.cern.ch/record/2784271 (cit. on p. 37).

[77] *Simulation-based extrapolation of b-tagging calibrations towards high transverse momenta in the ATLAS experiment*, tech. rep., All figures including auxiliary figures are available at https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-PHYS-PUB-2021-003: CERN, 2021, URL: https://cds.cern.ch/record/2753444 (cit. on p. 37).

[78] *Identification and energy calibration of hadronically decaying tau leptons with the ATLAS experiment in pp collisions at $\sqrt{s}$ = 8 TeV*, The European Physical Journal C **75** (2015), URL: https://doi.org/10.1140%2Fepjc%2Fs10052-015-3500-z (cit. on pp. 37, 91).

[79] *Measurement of the tau lepton reconstruction and identification performance in the ATLAS experiment using pp collisions at $\sqrt{s}$ = 13 TeV*, tech. rep., All figures including auxiliary figures are available at https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/CONFNOTES/ATLAS-CONF-2017-029: CERN, 2017, URL: https://cds.cern.ch/record/2261772 (cit. on pp. 37, 91).

[80] *Topological cell clustering in the ATLAS calorimeters and its performance in LHC Run 1*, The European Physical Journal C **77** (2017), URL: https://doi.org/10.1140%2Fepjc%2Fs10052-017-5004-5 (cit. on p. 37).

[81] A. Buckley et al., *General-purpose event generators for LHC physics*, Physics Reports **504** (2011) 145, ISSN: 0370-1573, URL: https://www.sciencedirect.com/science/article/pii/S0370157311000846 (cit. on p. 40).

[82] P. Nason, *A New Method for Combining NLO QCD with Shower Monte Carlo Algorithms*, Journal of High Energy Physics **2004** (2004) 040, URL: https://doi.org/10.1088%2F1126-6708%2F2004%2F11%2F040 (cit. on p. 40).

[83] S. Alioli, P. Nason, C. Oleari and E. Re, *NLO single-top production matched with shower in POWHEG: s- and t-channel contributions*, JHEP **09** (2009) 111, [Erratum: JHEP 02, 011 (2010)], arXiv: 0907.4076 [hep-ph] (cit. on p. 40).

[84] T. Sjöstrand et al., *An introduction to PYTHIA 8.2*, Computer Physics Communications **191** (2015) 159, URL: https://doi.org/10.1016%2Fj.cpc.2015.01.024 (cit. on p. 40).

[85] S. Agostinelli et al., *Geant4—a simulation toolkit*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **506** (2003) 250, ISSN: 0168-9002, URL: https://www.sciencedirect.com/science/article/pii/S0168900203013688 (cit. on p. 40).

[86] R. Harrington, "ATLAS fast simulation and digitisation/reconstruction", Talk, 2014, URL: %7Bhttps://indico.cern.ch/event/279530/contributions/634994/attachments/511923/706532/LPCC_fastsim_robert.pdf%7D (cit. on p. 40).

[87] M. Chadeeva and S. Korpachev, *Machine-learning-based prediction of parameters of secondaries in hadronic showers using calorimetric observables*, 2022, URL: https://arxiv.org/abs/2205.12534 (cit. on p. 41).

[88] P. Nason, *A New Method for Combining NLO QCD with Shower Monte Carlo Algorithms*, Journal of High Energy Physics **11**, 040 (2004) 040, eprint: hep-ph/0409146 (cit. on p. 41).

[89] T. Sjöstrand, S. Mrenna and P. Skands, *A brief introduction to PYTHIA 8.1*, Computer Physics Communications **178** (2008) 852, arXiv: 0710.3820 [hep-ph] (cit. on p. 41).

[90] I. Connelly, *Simulation of Top Quark Production for the ATLAS Experiment*, tech. rep., CERN, 2016, URL: https://cds.cern.ch/record/2231527 (cit. on p. 41).

[91] 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 **7**, 79 (2014) 79, arXiv: 1405.0301 [hep-ph] (cit. on p. 41).

[92] J. Bellm, S. Gieseke and S. Plätzer, *Merging NLO Multi-jet Calculations with Improved Unitarization*, Eur. Phys. J. C **78** (2018) 244, arXiv: 1705.06700 [hep-ph] (cit. on p. 41).

[93] J. Bellm, G. Nail, S. Plätzer, P. Schichtel and A. Siódmok, *Parton Shower Uncertainties with Herwig 7: Benchmarks at Leading Order*, Eur. Phys. J. C **76** (2016) 665, arXiv: 1605.01338 [hep-ph] (cit. on p. 41).

[94] T. Gleisberg et al., *Event generation with SHERPA 1.1*, Journal of High Energy Physics **2**, 007 (2009) 007, arXiv: 0811.4622 [hep-ph] (cit. on p. 41).

[95] H. Schulz, *Sherpa 2.2.2 manual*, URL: https://sherpa.hepforge.org/doc/SHERPA-MC-2.2.2.html (cit. on p. 41).

[96] S. Amoroso, F. Siegert, J. Kretzschmar and C. Gutschow, *PMG references document*, tech. rep., CERN, 2019, URL: https://cds.cern.ch/record/2678867 (cit. on p. 41).

[97] S. Amoroso et al., *Recommendations on the treatment of theoretical systematic uncertainties in statistical analysis of ATLAS data*, tech. rep., CERN, 2020, URL: https://cds.cern.ch/record/2715689 (cit. on p. 41).

[98] *Background studies for top-pair production in lepton plus jets final states in* $\sqrt{s} = 7TeV$ *ATLAS data*, tech. rep., All figures including auxiliary figures are available at https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/CONFNOTES/ATLAS-CONF-2010-087: CERN, 2010, URL: https://cds.cern.ch/record/1298967 (cit. on p. 43).

[99] T. P. S. Gillam and C. G. Lester, *Improving estimates of the number of fake leptons and other mis-reconstructed objects in hadron collider events: BoB's your UNCLE. (Previously "The Matrix Method Reloaded")*, (2014), URL: https://arxiv.org/abs/1407.5624 (cit. on p. 43).

[100] I. Brock, T. Holm, F. Kirfel, C. Kirfel and O. Kivernyk, Private Communication, 2022 (cit. on pp. 46, 47).

[101] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*, http://www.deeplearningbook.org, MIT Press, 2016 (cit. on p. 51).

[102] I. Sutskever, J. Martens, G. Dahl and G. Hinton, "On the Importance of Initialization and Momentum in Deep Learning", *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, Atlanta, GA, USA: JMLR.org, 2013 III-1139-III (cit. on p. 52).

[103] S. Ruder, *An overview of gradient descent optimization algorithms*, 2016, URL: https://arxiv.org/abs/1609.04747 (cit. on p. 52).

[104] T. Tang and P. Hu, *Quantitative standard of promotion strategy and analysis on the influence of consumer purchase behavior*, Cluster Computing **22** (2019) (cit. on p. 57).

[105] *TensorFlow Playground*, URL: http://playground.tensorflow.org/ (visited on 01/07/2021) (cit. on pp. 57, 58).

[106] URL: https://www.amd.com/en/product/1226&lang=en (cit. on p. 59).

[107] *GeForce RTX 2070 Super*, URL: https://www.msi.com/Graphics-Card/GeForce-RTX-2070-SUPER-GAMING/Specification (cit. on p. 59).

[108] L. M. Dery, B. Nachman, F. Rubbo and A. Schwartzman, *Weakly supervised classification in high energy physics*, Journal of High Energy Physics **2017** (2017), URL: https://doi.org/10.1007%2Fjhep05%282017%29145 (cit. on p. 60).

[109] E. M. Metodiev, B. Nachman and J. Thaler, *Classification without labels: learning from mixed samples in high energy physics*, Journal of High Energy Physics **2017** (2017), URL: https://doi.org/10.1007%2Fjhep10%282017%29174 (cit. on pp. 61, 62, 98).

[110] M. Germain, K. Gregor, I. Murray and H. Larochelle, *MADE: Masked Autoencoder for Distribution Estimation*, (2015), URL: https://arxiv.org/abs/1502.03509 (cit. on pp. 65, 66).

[111] C. Villani, *Topics in Optimal Transportation*, Graduate studies in mathematics, American Mathematical Society, 2003, ISBN: 9780821833124, URL: https://books.google.de/books?id=idyFAwAAQBAJ (cit. on p. 66).

[112] K. Medvedev, *CERTAIN PROPERTIES OF TRIANGULAR TRANSFORMATIONS OF MEASURES*, Theory of Stochastic Processes **1** (2008) (cit. on p. 66).

[113] V. I. Bogachev, A. V. Kolesnikov and K. V. Medvedev, *Triangular transformations of measures*, Sbornik: Mathematics **196** (2005) 309 (cit. on p. 66).

[114] A. Grover, M. Dhar and S. Ermon, *Flow-GAN: Combining Maximum Likelihood and Adversarial Learning in Generative Models*, 2017, URL: https://arxiv.org/abs/1705.08868 (cit. on p. 67).

[115] J. Su and G. Wu, *f-VAEs: Improve VAEs with Conditional Flows*, 2018, URL: https://arxiv.org/abs/1809.05861 (cit. on p. 67).

[116] B. Nachman and D. Shih, *Anomaly detection with density estimation*, Physical Review D **101** (2020), URL: https://doi.org/10.1103%2Fphysrevd.101.075042 (cit. on pp. 69, 70, 83, 85, 88).

[117] J. V. Dillon et al., *TensorFlow Distributions*, 2017, URL: https://arxiv.org/abs/1711.10604 (cit. on p. 79).

[118] A. Paszke et al., "Automatic differentiation in PyTorch", *NIPS-W*, 2017 (cit. on p. 79).

[119] L. Vomberg, *myANODE*, 2022, URL: https://github.com/lukavom/myANODE (cit. on p. 79).

[120] G. Papamakarios, T. Pavlakou and I. Murray, *Masked Autoregressive Flow for Density Estimation*, 2017, URL: https://arxiv.org/abs/1705.07057 (cit. on p. 82).

[121] M. Aaboud et al., *Probing the Quantum Interference between Singly and Doubly Resonant Top-Quark Production in pp Collisions at $\sqrt{s} = 13$ TeV with the ATLAS Detector*, Phys. Rev. Lett. **121** (15 2018) 152002, URL: https://link.aps.org/doi/10.1103/PhysRevLett.121.152002 (cit. on p. 85).

[122] G. Aad et al., *Dijet Resonance Search with Weak Supervision Using $\sqrt{s} = 13$ TeV pp Collisions in the ATLAS Detector*, Phys. Rev. Lett. **125** (13 2020) 131801, URL: https://link.aps.org/doi/10.1103/PhysRevLett.125.131801 (cit. on p. 89).

[123] D. Munoz Perez, *Improved track reconstruction for prompt and long-lived particles in ATLAS for the LHC Run 3*, tech. rep., CERN, 2022, URL: https://cds.cern.ch/record/2840786 (cit. on p. 89).

[124] E. E. Khoda, *Mixture Density Networks for tracking in dense environments on ATLAS*, (2020), URL: https://cds.cern.ch/record/2707229 (cit. on p. 89).

[125] A. Hallin et al., *Classifying anomalies through outer density estimation*, Phys. Rev. D **106** (5 2022) 055006, URL: https://link.aps.org/doi/10.1103/PhysRevD.106.055006 (cit. on p. 89).

[126] Martín Abadi et al., *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, Software available from tensorflow.org, 2015, URL: https://www.tensorflow.org/ (cit. on p. 92).

[127] F. Pedregosa et al., *Scikit-learn: Machine Learning in Python*, Journal of Machine Learning Research **12** (2011) 2825 (cit. on pp. 92, 119).

[128] L. Buitinck et al., "API design for machine learning software: experiences from the scikit-learn project", *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013 108 (cit. on pp. 92, 119).

[129] O. Amram and C. M. Suarez,
Tag N' Train: a technique to train improved classifiers on unlabeled data,
Journal of High Energy Physics **2021** (2021),
URL: https://doi.org/10.1007%2Fjhep01%282021%29153 (cit. on p. 118).

[130] F. G. Diaz Capriles and C. Kirfel, *CWoLa-Fakes*, 2022,
URL: https://gitlab.cern.ch/fdiazcap/cwola-fakes (cit. on p. 119).

[131] J. Pivarski et al., *Uproot*, 2017 (cit. on p. 119).

[132] F. G. D. Capriles, *Measurement of the Single Top tW-Channel Inclusive Cross Section in the Single Lepton Final State at 13 TeV with ATLAS*, BONN-IB-2014-08, Universität Bonn, 2014 (cit. on p. 131).

List of Figures

2.1 Public cross-section measurements from the ATLAS collaboration. . . . 4
2.2 Schematic of the SM of particle physics. . . . 5
2.3 Feynman diagram depicting a charged lepton and its antiparticle annihilating into a photon. . . . 9
2.4 Sketch of the Higgs potential. . . . 12
2.5 PDF plot of a proton from a combination of measurements at HERA. . . . 14
2.6 Top-antitop-quark pair production Feynman diagrams. . . . 18
2.7 Single top-quark production channel Feynman diagrams. . . . 19
2.8 Decay signatures of the top-quark. . . . 20
2.9 Decay signatures of tW and $t\bar{t}$ with examples of tW and $t\bar{t}$ interference. . . . 20
2.10 Feynman diagram depicting one of many ways to produce tZq (a) and tHq (b) final states. . . . 22
2.11 Higgs boson branching ratio and uncertainties near the measured Higgs boson mass. 23
2.12 Illustration of tau lepton decay modes. . . . 24

3.1 Schematic of the LHC and all its sequential accelerators. . . . 26
3.2 Drawing of the LHC and its four different detectors. . . . 27
3.3 Plot showing the delivered integrated luminosity per year at the LHC. . . . 27
3.4 Detailed depiction of the ATLAS detector with labeled components. . . . 28
3.5 Cross sectional image of the ATLAS detector showing how different types of particles are detected and measured. . . . 33

4.1 Total number of intersections per crossing for Run-2 recorded by ATLAS. . . . 40
4.2 $m_{b\ell}^{\text{minimax}}$ distribution in the $WWbb$ region with highest interference. . . . 43
4.3 Feynman diagrams depicting various tHq production modes. . . . 44
4.4 Estimated composition of tau candidates in the $2\ell + 1\tau_{\text{had}}$ region. . . . 47

5.1 Illustration of a NN's architecture. . . . 50
5.2 Example two-parameter loss function . . . 51
5.3 Example comparison of an optimizer with and without momentum. . . . 53
5.4 Examples of over- and underfitting and comparison to a well-fitted model. . . . 54
5.5 Example metrics from a network in training. . . . 55
5.6 Example receiver operating characteristic curve with area under the curve values. . . . 56
5.7 Schematic of a neural network with and without dropout nodes. . . . 57
5.8 Image showing a network with highly correlated nodes and their effect on overfitting. 58
5.9 Matrix multiplication comparison between CPU and GPU. . . . 59
5.10 Comparative performance of CWoLa, LLP and a fully supervised model. . . . 62

5.11 Schematic of an autoencoder network structure. . . . 63
5.12 Schematic of an masked autoencoder network structure. . . . 65
5.13 Example showing the importance of dependence order for a single MADE model. . . 69

6.1 Measured energy of physics objects comparing the DR and DS schemes. . . . 73
6.2 Metrics of the best performing NN tasked with identifying interference events in a $WWbb$ selection. . . . 74
6.3 Comparison of the $\phi(\ell_1)$ and $\cos\phi(\ell_1)$ distributions with their reconstruction. . . . 75
6.4 Metrics of the autoencoder trained on the DR sample with 2j2b selection. . . . 76
6.5 Reconstructed $m_{b\ell}^{\text{minimax}}$ variable compared to the original in both DR (a) and DS (b) samples. . . . 77
6.6 Reconstructed $E_{\text{T}}^{\text{miss}}$ variable compared to the original in both DR (a) and DS (b) samples. . . . 78
6.7 Average reconstruction error of the autoencoder trained on the tW DR sample. . . . 78
6.8 Loss of the single MADE layer network; measures negative log likelihood. . . . 80
6.9 Select variables after a one-MADE-layer transformation. . . . 81
6.10 Loss of the quadruple MADE layer network. . . . 81
6.11 Select variables after a four-MADE-layer transformation. . . . 82
6.12 Loss of the eight-MADE-layer network with the $m_{b\ell}^{\text{minimax}}$ distribution as conditional. 83
6.13 Select variables transformed after the eight-MADE-layer network with $m_{b\ell}^{\text{minimax}}$ as conditional distribution m. . . . 84
6.14 Log likelihood of each event in the eight-MADE-layer network with $m_{b\ell}^{\text{minimax}}$ as conditional distribution m. . . . 84
6.15 Unfolded and normalized differential $m_{b\ell}^{\text{minimax}}$ cross-section with theoretical models for comparison. . . . 85
6.16 Loss curve of both train and test samples for the SR and SB regions. . . . 87
6.17 Select SB variables transformed after the ANODE network with $m_{b\ell}^{\text{minimax}}$ as conditional distribution m. . . . 87
6.18 Select SR variables transformed after the ANODE network with $m_{b\ell}^{\text{minimax}}$ as conditional distribution m. . . . 87
6.19 Example of the extrapolated distributions in comparison to the estimated SR distribution. 88
6.20 Scatter plots of the likelihood ratio R as a function of negative log likelihood in the SR. 89

7.1 Distributions of kinematic variables and their reconstructed distribution for comparison. 93
7.2 Metrics of the autoencoder tasked with identifying τ_{had}. . . . 95
7.3 Well reconstructed distributions by the autoencoder designed to identify τ_{had}. . . . 96
7.4 Average reconstruction error of the $t\bar{t}$ MC sample with true and fake τ_{had} selected. . . 96
7.5 Analogue regions containing different ratios of real and fake τ_{had}. . . . 98
7.6 Kinematic comparison from data between $2\ell + 1\tau_{\text{had}}$ and $1\ell + 1\tau_{\text{had}}$ with similar jet multiplicity requirements. . . . 100
7.7 NN response on the 3+j1b region of the $1\ell + 1\tau_{\text{had}}$. . . . 102
7.8 Metrics of the network trained with loose data and truth-matched MC for τ_{had} identification. . . . 104
7.9 NN response and the separation power shown by the ROC curve. . . . 105
7.10 NN response of true and fake τ_{had} in MC simulation in the $1\ell + 1\tau_{\text{had}}$ signal region. . 105

7.11 NN response stack plot in the $1\ell + 1\tau_{\text{had}}$ signal region. . . . 106
7.12 Metrics of the network trained with loose data and truth-matched MC for 1-prong τ_{had} identification in the $2\ell + 1\tau_{\text{had}}$ region. . . . 107
7.13 Metrics of the network trained with loose data and truth-matched MC for 3-prong τ_{had} identification in the $2\ell + 1\tau_{\text{had}}$ region. . . . 108
7.14 NN response of true and fake 1-pronged τ_{had} in MC simulation in the $2\ell + 1\tau_{\text{had}}$ signal region. . . . 109
7.15 NN response of true and fake 3-pronged τ_{had} in MC simulation in the $2\ell + 1\tau_{\text{had}}$ signal region. . . . 110
7.16 NN response stack plot in the $2\ell + 1\tau_{\text{had}}$ signal region. . . . 111
7.17 Fake τ_{had} control region plots of the NN response. . . . 112
7.18 Fake τ_{had} signal region plots of the NN response. . . . 113
7.19 Fake τ_{had} control region plots of the NN response. . . . 114
7.20 Fake τ_{had} signal region plots of the NN response. . . . 115
7.21 Fake τ_{had} signal region plots of the NN response. . . . 115

B.1 Input variables for the autoencoder designed to reconstruct fake τ_{had}. . . . 121
B.2 Supplemental input variables for the autoencoder designed to reconstruct fake τ_{had}. . . . 122

C.1 Metrics of the network trained with loose data and truth-matched MC for 1-prong τ_{had} identification in the $2\ell + 1\tau_{\text{had}}$ region. . . . 123
C.2 Metrics of the network trained with loose data and truth-matched MC for 3-prong τ_{had} identification in the $2\ell + 1\tau_{\text{had}}$ region. . . . 124
C.3 NN response of true and fake 3-pronged τ_{had} in MC simulation in the $2\ell + 1\tau_{\text{had}}$ signal region. . . . 125
C.4 NN response stack plot in the $2\ell + 1\tau_{\text{had}}$ signal region. . . . 126

D.1 Metrics of the network trained with loose data and truth-matched MC for 1-prong τ_{had} identification in the $2\ell + 1\tau_{\text{had}}$ region. . . . 127
D.2 Metrics of the network trained with loose data and truth-matched MC for 3-prong τ_{had} identification in the $2\ell + 1\tau_{\text{had}}$ region. . . . 128
D.3 NN response of true and fake 3-pronged τ_{had} in MC simulation in the $2\ell + 1\tau_{\text{had}}$ signal region. . . . 129
D.4 NN response stack plot in the $2\ell + 1\tau_{\text{had}}$ signal region. . . . 129

F.1 Variables and their reconstructed values for the DR sample. . . . 136
F.2 Variables and their reconstructed values for the DR sample. . . . 137
F.3 Variables and their reconstructed values for the DR sample. . . . 138
F.4 Variables and their reconstructed values for the DS sample. . . . 139
F.5 Variables and their reconstructed values for the DS sample. . . . 140
F.6 Variables and their reconstructed values for the DS sample. . . . 141
F.7 Reconstruction error comparisons between the DR and DS samples 142
F.8 Reconstruction error comparisons between the DR and DS samples 143
F.9 Reconstruction error comparisons between the DR and DS samples 144

List of Tables

2.1 The four fundamental forces of nature with their mediating particle and relative strength to gravity. . . . 6
2.2 Branching ratio of different hadronic tau lepton decay modes in percentages with respect to total tau lepton decay modes. . . . 24

4.1 Event yields in the $WWbb$ high interference region after cuts. . . . 44
4.2 Event yields of the $2\ell + 1\tau_{\text{had}}$ channel. . . . 46

7.1 Defined working points for the τ_{had} ID RNN with corresponding efficiencies and background rejection. . . . 91
7.2 Comparison of defined working with corresponding efficiencies and background rejection in the 1-prong selection. . . . 97
7.3 Estimated event yields of the $1\ell + 1\tau_{\text{had}}$ channel. . . . 99
7.4 Variables which have similar distributions in $2\ell + 1\tau_{\text{had}}$ and $1\ell + 1\tau_{\text{had}}$. . . . 101
7.5 Selection of signal and background samples for the MC vs. data strategy in τ_{had} identification. . . . 103