Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Physics

Andreas Middelthon

Performance of neural networks in search for a new dark matter aware gauge boson Z' in final states with leptons and missing transverse energy at the LHC with the ATLAS detector

Master's thesis in Applied Physics and Mathematics
Supervisor: Farid Ould-Saada, Jon Andreas Støvneng
Co-supervisor: Eirik Gramstad
June 2023

NTNU
Norwegian University of
Science and Technology

Andreas Middelthon

Performance of neural networks in search for a new dark matter aware gauge boson Z' in final states with leptons and missing transverse energy at the LHC with the ATLAS detector

Master's thesis in Applied Physics and Mathematics
Supervisor: Farid Ould-Saada, Jon Andreas Støvneng
Co-supervisor: Eirik Gramstad
June 2023

Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Physics

NTNU
Norwegian University of
Science and Technology

Abstract

We perform a search for a new dark matter aware gauge boson Z' in final states with two leptons and missing transverse energy, predicted by dark Higgs and light vector models. Monte-Carlo simulated events are analysed, corresponding to data from the ATLAS detector taken during the full Run 2 at the LHC with $\sqrt{s} = 13$ TeV. Signal regions are constructed by the standard method of making cuts on kinematic variables as well as by use of neural networks. The neural networks are optimized in order to reach a high accuracy in classification of signal and background events. The methods are compared, and the neural networks are found to perform better than the standard approach. However, the highest expected significances are in the order of 10^{-1} for a coupling of Z' boson to leptons of $g_l = 0.01$, which is too low for a possible exclusion or discovery of the signals in the data from Run 2.

Sammendrag

Vi gjennomfører et søk etter et nytt gaugeboson Z' i sluttilstander med to leptoner og manglende transversal energi. Søket er basert på hypotetiske mørk Higgs-modeller og "light vector"-modeller. Data generert av Monte-Carlo-simuleringer analyseres. Disse korresponderer med data fra ATLAS-detektoren gjennom hele "Run 2" ved LHC med $\sqrt{s} = 13$ TeV. Signalområder konstrueres gjennom den tradisjonelle "cut and count"-metoden, basert på å gjøre kutt på bestemte kinematiske variabler, og ved bruk av nevrale nettverk. De nevrale nettverkene optimeres, med mål om å oppnå høyest mulig nøyaktighet i klassifiseringen av signal og bakgrunn. Metodene sammenlignes, og de nevrale nettverkene viser seg å prestere bedre enn den tradisjonelle fremgangsmåten. Likevel er den forventede signifikansen i størrelsesordenen 10^{-1} når koblingskonstanten til leptoner er $g_l = 0.01$, som er for lavt til at det er mulig å ekskludere eller oppdage signalene i data målt av detektoren gjennom Run 2.

Acknowledgements

First, I would like to thank my main supervisor at UiO, Farid Ould-Saada, for accepting to supervise me even though I did not officially belong to the University of Oslo. I also want to thank my co-supervisor at UiO, Eirik Gramstad. You have both been excellent supervisors in every way. I have always received the help I have needed and learned a lot from both you during the past year. Thank you for taking the time to review this thesis. Also, I want to thank you for making me feel like a part of the university and inviting me to the trip to CERN with the other students.

I want to thank my supervisor at NTNU, Jon Andreas Støvneng. You have provided me with all the help I have needed in order to write my master's thesis externally, and have always responded to my questions with detailed answers.

I would like to express my gratitude to the HEP group at UiO for welcoming me and including me in various social activities. You have made it very enjoyable to stay at UiO while writing my thesis. I also want to thank my fellow master's students Martin, Ruben, Mattias and Tobias. A special thanks to Ruben for interesting and helpful conversations about how to best use machine learning in particle physics.

Finally, I want to thank my family and friends for always supporting me.

Contents

†Adapted from [1].

†Adapted from [1].

†Adapted from [1]

[†]Adapted from [1]

List of Figures

List of Tables

Introduction[†]

Theoretical physicists have for a long time attempted to come up with solutions to the anomalies that arise in the Standard Model (SM) of particle physics, and a large amount of ideas and new physics models beyond the SM (BSM) have been proposed. However, one of the main difficulties over time has been the lack of new experimental results in order to be guided in a specific direction. A possible approach to solving the problem is to search for evidence for BSM models in data from the Large Hadron Collider (LHC). This may lead to a partial or full confirmation of the model, it may give inconclusive results or in some cases falsify it, all of which are useful results.

One of the main current pursuits in physics is to obtain a better understanding of dark matter. Astronomical observations have in several cases shown deviations from predictions based on currently accepted theories of gravity [2–4]. There are many possible explanations of this, including that current theories of gravity need modification or that the anomalies stem from the presence of large amounts of unknown matter in the universe, so called *dark matter* (DM). There are clear evidences of DM in the universe, but what it consists of is one of the mysteries in physics today. Several theories propose that DM may arise from primordial black holes or massive compact halo objects [5, 6]. Another popular theory is the existence of a new invisible, massive particle or a sector of them, which are called *weakly interacting massive particles* (WIMPs) [7, 8]. However, although it has been possible to observe the effects of DM at large scales, one has not yet succeeded in detecting any WIMPs in particle detectors. One of the main difficulties is that if WIMPs exist, they do not interact with light or bind to atoms, but only interact through the weak force (and gravity) [9, 10]. Their masses are also in most cases expected to be large compared to other SM particles, which makes them move slowly compared to the speed of light and thus fulfills the requirement of being cold dark matter, as opposed to e.g. neutrinos in the SM, which are light and represent what is known as hot dark matter. WIMPs may be detectable, and there are ongoing attempts to detect them indirectly, which will be discussed in this report.

In this thesis, we will consider two different models containing physics beyond the Standard Model and use these in order to search for a new dark matter aware

[†]Adapted from [1].

Z' boson. This will be done by analysing simulated Monte-Carlo samples, using the standard cut and count method as well as a machine learning method. The aim of the thesis is two-fold: Signal regions will be constructed using Monte-Carlo samples in order to determine whether it will be possible to discover new physics or exclude the models studied by later analysing real data. We also want to optimize both search methods and compare them in order to see which of them obtains the highest sensitivity.

The thesis begins by briefly introducing the Standard Model of particle physics as well as mentioning some of its shortcomings, particularly the evidence of dark matter. Chapter 2 provides information about the LHC and ATLAS. The search strategy is laid out in chapter 3, and the signal and background processes are introduced. In chapter 4 we introduce general machine learning concepts as well as the theory of neural networks. Chapter 5 describes the data preparation and considerations made before the search. In chapter 6, we perform a standard cut and count analysis by making cuts on specific variables and measuring the expected sensitivity to the signal models. In chapter 7, we optimize the neural networks before performing a corresponding machine learning based analysis. The thesis is concluded in chapter 8.

In some chapters, modified parts or full sections of an earlier work written by myself are used, reference [1]. These are mainly theory or preparatory sections and are marked by a † symbol and a footnote containing the reference.

1 The Standard Model[†]

In this chapter, we briefly describe the main elements of the Standard Model. In the first section, we introduce the different groups and types of particles and their characteristics. The types of interactions that are currently included in the Standard Model are described, as well as some phenomena and observations the SM has not yet been able to describe. In section 1.1 and 1.2, we use references [11–19] (except when other references are given). In the end, we provide some additional details about dark matter, which is of importance to the motivation of the search that is later performed.

The Standard Model is the theory describing the present-day understanding of the elementary particles. It includes all of the currently observed particles, as well as the interactions between them that we are aware of, except that it has not yet been able to include gravity. It is based on quantum field theory, and many parts of it have been experimentally verified with high accuracy.

1.1 The fundamental particles

The Standard Model includes 25 different elementary particles. These may be split into two main groups: The fermions which constitute matter, and bosons which act as force carriers (except in the case of the scalar Higgs boson). An overview of the Standard Model particles is shown in figure 1.1.

1.1.1 Fermions

The fermions have a spin quantum number of 1/2, and may be categorized as either leptons or quarks. The electron (e^-), muon (μ^-) and tau (τ^-) have integer electric charge, while the quarks have an electric charge of $-1/3$ or $+2/3$. The neutrinos have zero electric charge. The quarks are also subject to the quantum number color charge, while leptons are not. The fermions are divided into three generations. In the first generation, there are the up (u) quark and down (d) quark, as well as the electron (e^-) and electron neutrino (ν_e), while the second and third generations include copies of these that only differ by having a higher mass. The second generation include the charm (c) and strange (s) quarks, as well as the muon

[†]Adapted from [1].

(μ^-) and muon neutrino (ν_μ). The third generation contains the top (t) and bottom (b) quarks, the tau (τ^-) and tau neutrino (ν_τ). The mass of the particles increases significantly in each generation. While quarks in the first generation have a mass of a few MeV, the second generation contains masses of $\sim$ 100 Mev for the strange quark and $\sim$ 1 GeV for the charm, and in the third generation the masses of the top and bottom quark are 4.18 GeV and 173.1 GeV respectively. Each fermion particle has a corresponding anti-particle with the same properties, except having opposite electric charge, as well as other additive quantum numbers. Because fermions have half-integer spin, they obey the Pauli exclusion principle, which means that two or more identical particles cannot occupy the same quantum state at the same time. Due to quarks carrying color charge, they are not observed individually as will be discussed in section. Instead they are bound in colorless composite particles known as hadrons. These are divided into two groups called baryons and mesons. Baryons consist of an odd number of quarks. Examples of baryons include the proton (uud) and the neutron (udd). The other type of hadrons are called mesons. These are characterized by containing an equal number of quarks and antiquarks. Examples of mesons are the charged pions ($u\bar{d}$ and $\bar{u}d$).

1.1.2 Bosons

The bosons include the photon (γ), eight different gluons (g), the Z and $W^\pm$ bosons and the Higgs boson (H) [14]. These are characterized by having integer spin quantum number, with the Higgs boson having $s = 0$, making it a scalar boson. The rest have $s = 1$ and are called vector bosons. As bosons are not subject to the Pauli exclusion principle, they obey Bose-Einstein statistics. The vector bosons act as force mediators. The gluons couple to particles with color charge, and are massless and electrically neutral. However, they do carry color charge. Although gluons are massless, the range of the strong force is only $\sim 10^{-15}$m, as they self-interact. The photon is massless and electrically neutral. It couples to electrically charged particles through the electromagnetic interaction, which will be discussed in section 1.2.1. As it is massless and stable, the range of the electromagnetic interaction is infinite. The $W^\pm$ and Z bosons couple to weakly charged particles, as defined in section 1.2. These are massive particles, with masses $M_Z \approx 91.2$ GeV and $M_{W^\pm} \approx 80.4$ GeV. The Higgs boson, which was discovered at the LHC in 2012 [15,20], exists as a consequence of the Higgs field, which causes many of the Standard Model particles

to acquire mass due to spontaneous symmetry breaking. If gravity is mediated by a vector boson, the hypothetical graviton is expected to be the only boson with spin $s = 2$.

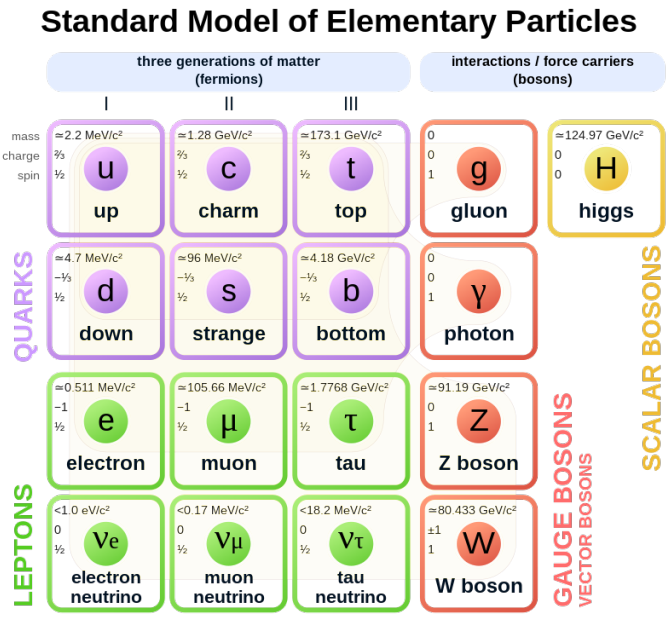


Figure 1.1: Overview of the particles of the Standard Model, and their characteristics. Figure taken from [21].

1.2 The fundamental interactions

The Standard Model describes the electromagnetic, weak and strong interactions. It is a quantum field theory and treats particles as excited states of quantum fields. The equations of motion for a field is found by solving the Euler-Lagrange equation

$$\partial_\mu \left(\frac{\partial \mathcal{L}}{\partial(\partial_\mu \phi)} \right) - \frac{\partial \mathcal{L}}{\partial \phi} = 0 \tag{1.1}$$

for a Lagrangian $\mathcal{L}$ that describes the system, and is a function of quantum fields $\phi(x)$ and their derivatives $\partial_\mu \phi$. The Standard Model is based on gauge symmetries that require the Lagrangian to be invariant under specific types of transformations. According to Noether's theorem, this leads to conservation laws. The Standard Model obeys the internal symmetries of the unitary product group

$$G_{SM} = SU(3)_c \times SU(2)_L \times U(1)_Y \tag{1.2}$$

where the $SU(3)_c$ group represents quantum chromodynamics (QCD), which is the theory of the strong interaction, and will be described further in section 1.2.2. $SU(2)_L \times U(1)_Y$ represents the electroweak symmetry [19]. The subscript in $SU(2)_L$, which will be described further in section 1.2.3, refers to the weak isospin transformations being restricted to left-handed particles (and right-handed anti-particles), thus incorporating parity violation in weak interactions. In addition to the continuous gauge symmetries, the fields obey some discrete symmetries. These are parity, referring to changing the sign of the spatial coordinates, charge conjugation, which is changing the sign of the charge of particles and time reversal, which changes the sign of the time coordinate.

1.2.1 Quantum electrodynamics

The electromagnetic interaction is described by quantum electrodynamics (QED) and involves interactions between electrically charged particles, with the photon as mediator. The starting point for arriving at the QED lagrangian is the lagrangian of a free fermion field ψ with mass m,

$$\mathcal{L} = \bar{\psi}(i\gamma^\mu \partial_\mu - m)\psi \tag{1.3}$$

where γ^μ are the four Dirac matrices and $\bar{\psi} = \psi\gamma^0$ is the conjugate of the fermion field. This lagrangian is invariant under global $U(1)$ transformations $\psi(x) \to \psi'(x) = e^{i\alpha}\psi(x)$. However, it is not invariant under local $U(1)$ transformations. In order to achieve local invariance, the derivative ∂_μ is replaced by a covariant derivative

$$D_\mu = \partial_\mu - ieA_\mu \tag{1.4}$$

where A_μ is a gauge field that we require to transform as $A_\mu(x) \to A'_\mu(x) = A_\mu(x) - \partial_\mu\alpha(x)$. The QED lagrangian then becomes

$$\mathcal{L}_{QED} = \bar{\psi}(i\gamma^\mu \partial_\mu - m)\psi + e\bar{\psi}\gamma^\mu\psi A_\mu - \frac{1}{4}F_{\mu\nu}F^{\mu\nu} \tag{1.5}$$

where $F_{\mu\nu} = \partial_\mu A_\nu - \partial_\nu A_\mu$. The term containing ψ, $\bar{\psi}$ and A_μ indicates an interaction between the fermion current and the gauge field. By interpreting the gauge field as a massless photon, QED allows for interactions between the fermion field and the photon field, as shown in figure 1.2.

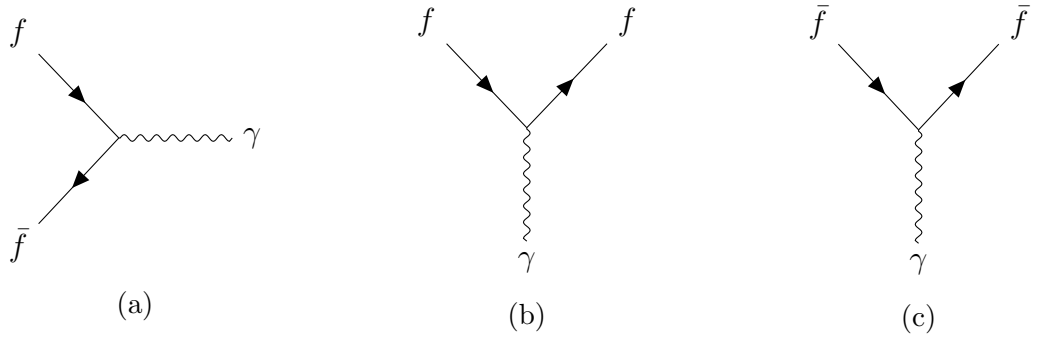


Figure 1.2: Basic QED vertices showing. a) shows a fermion-antifermion pair annihilating into a photon. b) and c) shows a fermion/antifermion interacting with the photon field, resulting in a change in its 4-momentum.

1.2.2 Quantum chromodynamics

The strong interaction is described in terms of quantum chromodynamics (QCD), which involves interactions between fermions with color charge (quarks) and gluons, of which there are 8 types. The three types of color charge are called *red*, *green* and *blue*, and the corresponding charges that neutralize them are *antired*, *antigreen* and *antiblue*. The mediating particle is a gluon, and the type of gluon depends on the colors of the interacting particles. The strong interaction is responsible for forming hadrons, such as the proton, neutron and different types of mesons.

QCD is symmetric under $SU(3)_c$ transformations, which is non-abelian and contain 8 generators. The generators are $\frac{1}{2}\lambda^a$, where λ^a are the 3×3 Gell-Mann matrices subject to the commutation relations

$$[\lambda_a, \lambda_b] = 2if_{abc}\lambda_c \tag{1.6}$$

for $a, b = 1, ..., 8$, where f_{abc} are the structure constants of $SU(3)$.

As the strong coupling constant diminishes as energy increases, the quarks interact weakly[1] at high energies, known as *asymptotic freedom*, and strongly when the energy is low, leading to confinement of quarks and gluons in composite hadrons. By replacing derivatives with QCD covariant derivatives and introducing gauge fields G^ρ_μ, the lagrangian becomes

[1] Quarks also interact through the weak interaction. Here we mean that the interaction strength is weak.

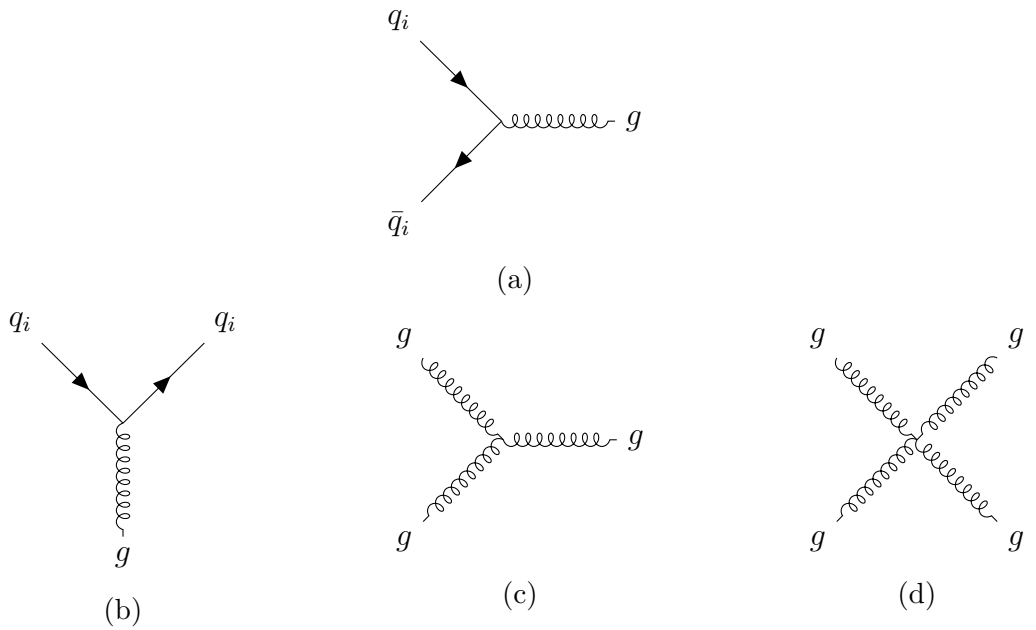


Figure 1.3: Basic QCD vertices. a) shows an incoming quark-antiquark pair interacting and becoming gluon, while b) shows a quark interacting with a gluon, resulting in a change in its 4-momentum. Here, $i = 1, 2, 3$, which corresponds with r, g, b. c) and d) show gluon self-interactions.

$$\mathcal{L}_{QCD} = \sum_q (\bar{q}(i\gamma^\mu \partial_\mu - m)q - g_s \bar{q}\frac{\lambda_\rho}{2} q G^\rho_\mu) - \frac{1}{4} G_{\mu\nu} G^{\mu\nu} \tag{1.7}$$

where $\bar{q}$ are the conjugates of the quark fields ($q = \begin{pmatrix} q_r \\ q_g \\ q_b \end{pmatrix}$ is a color triplet) and g_s is related to the strong coupling constant by $\alpha_s = \frac{g_s^2}{4\pi}$. By interpreting G^ρ_μ as the gluon fields, the interaction terms show that the possible QCD interactions are interactions between quarks, antiquarks and a gluon, as well as self-interactions between three or four gluons, as shown in figure 1.3.

1.2.3 The electroweak theory

The *weak interaction* affects left-handed fermions. It may be split into two types, called the *weak charged current interaction* and the *weak neutral current interaction*. The weak charged current interaction, mediated by the $W^\pm$ bosons, only applies to left-handed particles and right-handed anti-particles. The weak, neutral current interaction, mediated by the Z^0 boson, couple differently to left-handed and right-handed particles. The weak charged current interaction is the only interaction

that allows for change of flavour, and it is the only one that violates charge-parity symmetry. The weak and electromagnetic interaction were initially considered to be separate interactions. However, it was later discovered that they may be unified into a single type of interaction, called the electroweak interaction. Examples of electroweak interaction vertices are shown in figure 1.4.

As the weak interaction only applies to left-handed particles and right-handed antiparticles, it is necessary to decompose the fermion field by applying the chiral projection operators $P_R = \frac{1}{2}(1+\gamma^5)\psi$ and $P_L = \frac{1}{2}(1-\gamma^5)\psi$ so that $P_R\psi = \psi_R$ and $P_L\psi = \psi_L$. The gauge symmetry in the weak charged interaction is the $SU(2)_L$ group, which transforms the left-handed fermion field as

$$\psi_L(x) \to \psi'_L(x) = e^{iga(x)\cdot\vec{\tau}}\psi_L(x), \tag{1.8}$$

where $\vec{\tau} = \frac{1}{2}\vec{\sigma}$ are the Pauli matrices that generate the $SU(2)_L$ group and g is the weak coupling. The unification of the weak and electromagnetic interaction assumes that the lagrangian is symmetric under $SU(2)_L \times U(1)_Y$ where Y is the *weak hypercharge*, which is related to the electric charge Q and the third component of the *weak isospin* (where the weak isospin is the generator of $SU(2)_L$), T_3, by $Y = 2(Q - T_3)$. By applying a similar procedure as in QED and QCD, replacing derivatives with covariant derivatives and introducing gauge fields, we get three gauge fields W^i_μ related to $SU(2)_L$ and one gauge field B_μ from the hypercharge group $U(1)_Y$. Left-handed fields are represented by weak isospin doublets with $T_3 = \pm 1/2$,

$$\begin{pmatrix}\nu_e\\ e^-\end{pmatrix}_L \quad \begin{pmatrix}\nu_\mu\\ \mu^-\end{pmatrix}_L \begin{pmatrix}\nu_\tau\\ \tau^-\end{pmatrix}_L \quad \begin{pmatrix}u\\ d'\end{pmatrix}_L \begin{pmatrix}c\\ s'\end{pmatrix}_L \quad \begin{pmatrix}t\\ b'\end{pmatrix}_L$$

while right-handed fields are represented as weak isospin singlets with $T_3 = 0$, (u_R,d_R,...,e^-_R,ν_{eR}). The weak interaction also includes flavor mixing between d, s and b. The probability of mixing between different states are described by the Cabibo-Kobayashi–Maskawa (CKM) matrix

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

Where d, s, b are mass eigenstates and d', s', b' are weak eigenstates. The prob-

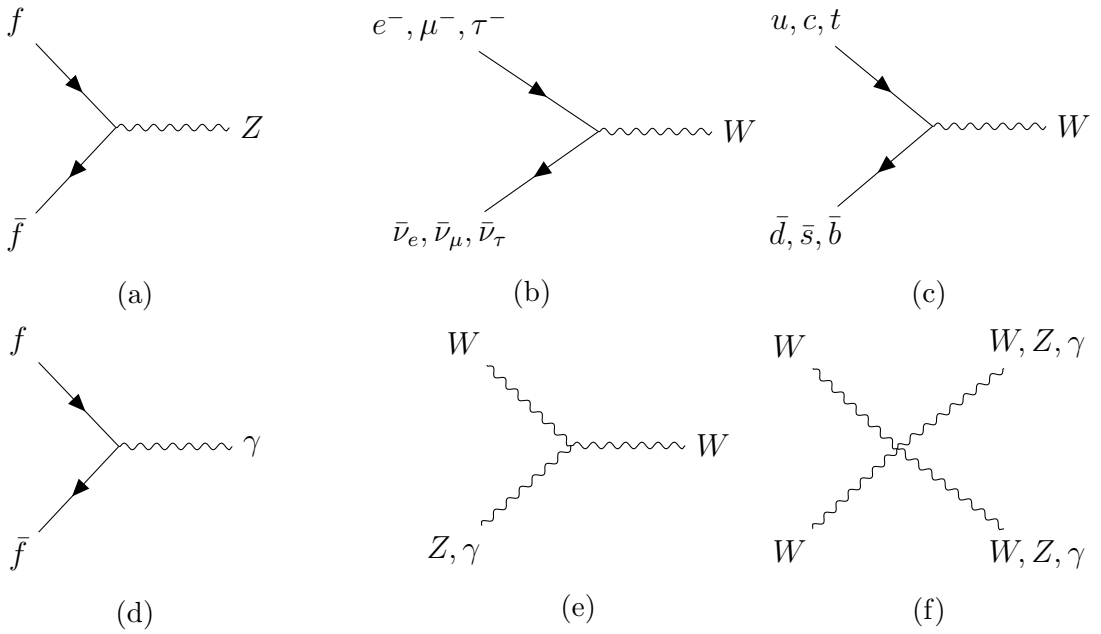


Figure 1.4: Basic electroweak vertices. a), b) and c) are weak vertices showing interactions between fermions and Z and W bosons. d) shows an electromagnetic vertex. e) and f) show electroweak interactions between γ, Z and W bosons. The vertices may be rotated in order to find other possible interactions.

ability of transition between the flavours i and j is proportional to $|V_{ij}|^2$. Writing left-handed and right-handed terms separately, the electroweak lagrangian becomes

$$\mathcal{L}_{EWK} = \sum_f \bar{\psi}_L^f \gamma^\mu \Big[i\partial_\mu + g\vec{\tau} \cdot \vec{W}_\mu + \frac{g'}{2} Y B_\mu \Big] \psi_L^f + \sum_f \bar{\psi}_R^f \gamma^\mu \Big[i\partial_\mu + \frac{g'}{2} Y B_\mu \Big] \psi_R^f \\ - \frac{1}{4} \vec{W}_{\mu\nu} \cdot \vec{W}^{\mu\nu} - \frac{1}{4} B_{\mu\nu} \cdot B^{\mu\nu}, \tag{1.9}$$

where g and g' are the coupling constants of $SU(2)_L$ and $U(1)_Y$, respectively.

1.3 Shortcomings of the standard model

The Standard Model is a successful and precise theory. The lack of anomalies is one of the reasons why it has been difficult to develop the theory further. However, there are a number of phenomena the SM has not yet been able to explain or that are not incorporated into the current theory.

1.3.1 Gravity

The Standard Model includes three of the four known forces of nature, but it does not incorporate gravity [11]. After the success of the other gauge theories, similar methods were attempted in order to explain the gravitational force with a "graviton" as force mediator. However, these theories turned out not to be renormalizable [22]. This has led to the question of whether gravity is fundamentally different from the other forces or if the framework of the Standard Model is flawed or incomplete. One of the reasons it has been difficult to develop a quantum theory of gravity is that the force is much weaker than the other interactions at the microscopic scale, and the effects of it are not detectable at the LHC.

1.3.2 Dark matter

Several experiments and observations suggest that there is more matter in the universe than what is possible to optically and electromagnetically observe. One of the evidences of this is the difference between the observed rotation curves of galaxies and the theoretical predictions of how they should behave [2]. Other evidences of dark matter include observations from gravitational lensing, the power spectrum of the Cosmic Microwave Background, as well as dark matter providing an explanation for the structure formation of the universe [3, 4]. The main hypothesis explaining these observations proposes that there is a particle or a number of particles with specific properties that are responsible for them. Another possible explanation is that there is a flaw in the present theory of gravity over long distances, known as modified gravity. A dark matter particle must have specific properties, including zero electric charge and color charge and a non-zero mass, among other things [9]. In order to detect it, one searches for specific candidates, such as weakly interacting massive particles (WIMPs) [7, 8]. The evidence of dark matter and its candidate particles will be discussed further in section 1.4.

1.3.3 Neutrino oscillations and masses

In the SM, neutrinos are assumed to be massless [11]. However, in a multitude of experiments it has been shown that neutrinos can oscillate, which means they undergo a change of flavour [23, 24]. Although experiments have shown that the neutrino mass must be very small, the oscillations require the neutrinos to have mass and mix such that the weak eigenstates are combinations of the mass eigenstates. Thus, some modification is necessary in the Standard Model to account for this.

1.3.4 The hierarchy problem

In the calculation of the Higgs Boson mass, divergences appear [25]. To cancel out these divergences, one needs to make a cut-off at the order of the Planck scale. However, in order for the divergences to cancel, it is also necessary that the mass of the Higgs boson is in the order of 10^{19} GeV, which is far from the observed value of ~ 125 GeV [26]. This discrepancy is known as the hierarchy problem.

1.3.5 Accelerated expansion of the universe

From observations, it has been established that the universe is expanding, and that the expansion is accelerating, which means that the velocity with which a distant object moves away from an observer is increasing with time [27, 28]. The introduction of a cosmological constant accounts for this, which is equivalent to the presence of dark energy [29]. This causes a repulsive force, resulting in the accelerated expansion. However, although it is included in cosmological models by hand, it is not accounted for in the Standard Model and it is unknown where it comes from.

1.4 Dark matter

We will now elaborate on some of the evidences for dark matter mentioned in section 1.3.2 and present some of the candidates for dark matter.

1.4.1 Evidence

By studying velocity curves of stars orbiting galaxies, significant deviations from expectations have been found [2]. If spherical symmetry is assumed, the orbital average velocity of the stars may be written as a function of the distance from the galaxy center by [30, 31]

$$u(r) = \frac{G_N M(r)}{r}, M(r) = 4\pi \int \rho(r) r^2 dr \tag{1.10}$$

where G_N is the gravitational constant. This means that the velocity should decrease as $\propto \frac{1}{\sqrt{r}}$. However, the observed data show a distribution that is flat when the radius increases, as seen in figure 1.5. Before accepting this as evidence of dark matter, attempts were made to see if there were other possible reasons for the flat curve. This was done by assuming that the stars in the galactic discs had as much mass as would still be consistent with current theories [32]. However, contributions from a dark matter halo turned out to be necessary at large radii, as shown in figure

1.5.

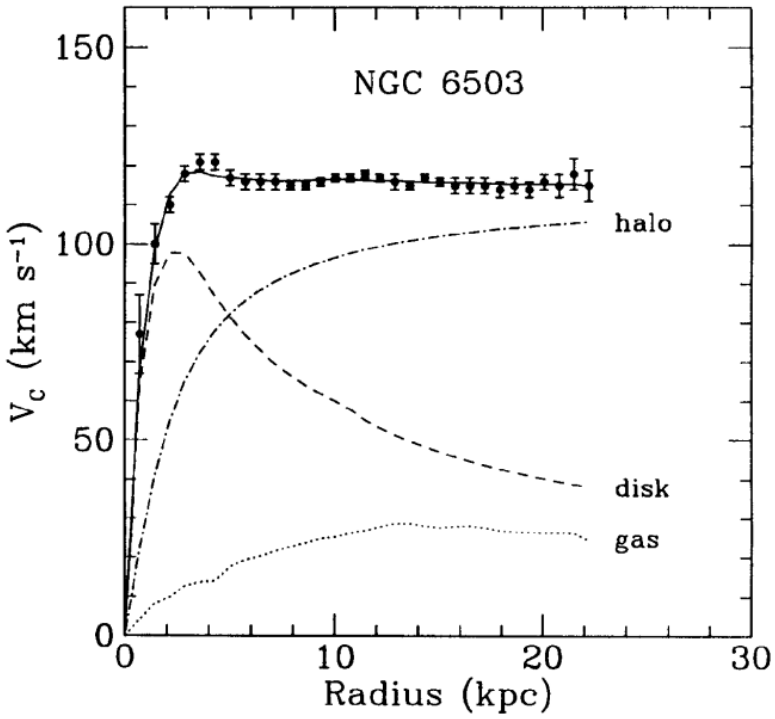


Figure 1.5: Rotation curve velocity for the dwarf spiral galaxy NGC 6503. The dots represent the observed data, while the dashed lines show the expected disk and gas contributions in addition to the dark matter halo needed to obtain similar values to the data. Figure taken from [33].

Another source of evidence of dark matter is the *Cosmic Microwave Background* (CMB). The CMB was produced by photon freeze-out, which refers to the photons generally persisting over time, rather than continously being created and annihilated, which was the case in the early universe due to its higher density [29]. These photons travel through space-time and have reached microwave frequency. They appear as black-body radiation with a temperature of ~ 2.755 K [34]. The CMB is isotropic except from small temperature fluctuations. The angular scale and height of the peaks of these fluctuations determine several cosmological parameters, including the dark matter component. The power spectrum of the temperature fluctuation is shown in figure 1.6. We will not go into the details of this measurement as it is based on multipole expansions [29], but the oscillations are affected by gravity and photon radiative pressure. The effect of dark matter increases the height of the third peak. The results turn out to be in agreement with the lambda-CDM model [35], while

being hard to reproduce with modified Newtonian gravitational models [36].

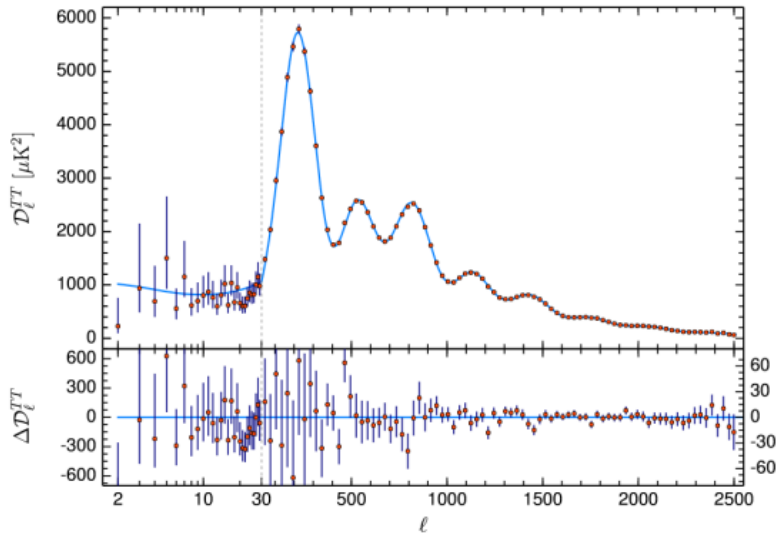


Figure 1.6: Cosmic microwave background temperature power spectrum showing the temperature fluctuations as a function of the multipole moment l. Dark matter effects are expected to increase the height of the third peak. Figure taken from [37].

1.4.2 DM candidates

Many types of dark matter particles have been proposed. However, although some of its characteristics if it exists are known, there is too little experimental evidence to have specific information about many of its properties. For example, it is not known whether it is likely to be one type of particle or a sector of particles, as well as what their masses are. However, some of the properties of DM can be inferred from the observations discussed above [9]. First, as it is not visible, it does not interact with light, which means that it is electrically neutral. As it does not bind to the normal atoms, it does not have color charge, but it is possible that dark matter particles can bind to each other. It must be massive and interact gravitationally, as the evidence for it include the observation of a larger gravitational effect than expected. Although it does not have electric charge or color charge, it may interact with normal matter through the weak interaction, but it may also not interact at all, except through gravitation. A large amount of hot dark matter in the universe has also been shown to prevent the formation of galaxies, and therefore it is thought to be cold.

One of the popular candidates for dark matter are known as *weakly interacting massive particles* (WIMPS) [7, 8]. These are massive, electrically neutral and color neutral particles, meaning that they only interact through gravity and the weak interaction. They may also interact through currently unknown interactions weaker than the weak interaction. The exact properties of WIMPs are unknown, and there does not exist a formal definition of them. They are however thought to have large masses compared to SM particles, which would cause them to be slow moving. When assuming a WIMP particle with a mass within the allowed range, it results in the relic density that is necessary for dark matter. This is known as the *WIMP miracle* [9].

Another possible candidate is the *axion* [38, 39]. It was postulated in 1977 as a solution to the strong CP problem in QCD [40, 41]. This requires adding a new global symmetry to the SM that is spontaneously broken, known as the Peccei-Quinn symmetry. The spontaneously broken symmetry along with QCD effects produce a cosmological population of cold axions. The axion is a boson with spin-0, and is expected to have very low interaction cross sections for strong and weak forces.

These are only some of the many hypothetical particles with characteristics similar to those that dark matter particles are required to possess. Some other candidates are *sterile neutrinos* [42], *strongly interacting massive particles* [43] and *self-interacting dark matter* [44].

In this chapter, we have introduced the particles and interactions included in the Standard Model of particle physics, as well as highlighting some of the shortcomings of the theory. We will now move on to explain how the SM is tested at the LHC.

2 LHC and ATLAS[†]

2.1 The LHC

The Large Hadron Collider (LHC) is the largest particle accelerator in the world, with a circumference of 27 km. In the LHC, two particle beams travel in opposite directions, resulting in proton-proton collisions with center-of-mass energy of up to 13.6 TeV. This energy was recently reached in the Run 3 period, which started in 2022 [45, 46]. Particle beams are guided by superconducting electromagnets, with four crossing points where the particles collide. These represent the four large experiments, which include ATLAS, LHCb, ALICE and CMS. ATLAS and CMS study a wide range of phenomena in proton-proton collisions, while LHCb specializes in the study of the bottom quark through B-hadron interactions, and ALICE studies heavy-ion physics. There are also several smaller experiments. For the Run 2, which lasted from 2015 to 2018, the center-of-mass energy in the collisions was 13 TeV [47].

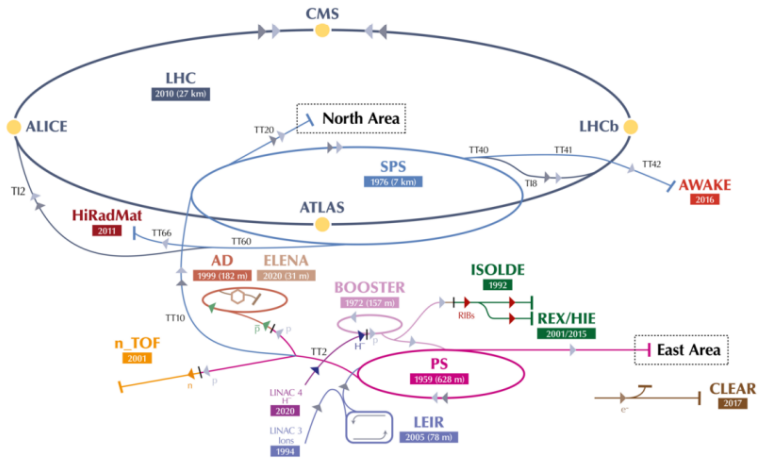


Figure 2.1: An illustration showing CERN's accelerator complex. Figure taken from [48].

2.2 The ATLAS detector

The *ATLAS* (A Toroidal LHC ApparatuS) detector is the largest detector constructed for a particle collider [49]. It has a cylindrical shape, with a diameter of 25m and a length of 46m. It consists of different layers and sub-detector systems, the main ones being the inner detector, the calorimeters, the muon spectrometer,

[†]Adapted from [1].

as well as the magnet system. An overview of the detector is shown in figure 2.2.

The inner detector is the innermost layer [49]. It consists of three different systems that measure the direction, momentum and charge of electrically charged particles. This is done by a solenoid creating a magnetic field of 2T, which bends charged particles. The three systems are the pixel detector, the semiconductor tracker and the transition radiation tracker. The pixel detector is the innermost of them, and is located 3.3 cm from the beam line. It is made up of four layers of silicon pixels. The semiconductor tracker surrounds the pixel detector and consists of silicon micro-strip trackers. In combination, the pixel detector and the semiconductor layer are used to detect and reconstruct charged particle tracks, which is used to determine the momentum of the particles. The transition radiation tracker consists of straw tubes filled with a gas mixture. When charged particles pass through the tubes, they ionise the gas, creating an electric signal. This detector therefore helps identifying electrons.

Outside the inner detector are the calorimeters [50]. There are two different calorimeters. These are the electromagnetic calorimeter also known as the *Liquid Argon* (LAr) *Calorimeter*, and the *Tile Hadronic Calorimeter*. The LAr calorimeter consists of layers of metal (tungsten, copper and lead). Electrically charged particles are absorbed by the layers, causing them to emit electromagnetic showers of new, lower energy $e^{\pm}$ and γ. These showers ionise liquid argon, which is between the layers, producing an electric current. The Tile Calorimeter works in a similar way, but consists of steel and plastic scintillating tiles. When hadrons pass through the steel layers they produce showers of hadrons. The plastic scintillators then produce photons which are converted into electric signals.

The *Muon Spectrometer* consists of precision detectors and fast-response detectors [49,51]. The precision detectors use Monitored Drift Tube detectors, consisting of aluminium tubes filled with a gas mixture. Muons passing through the tubes interact with electrons in the gas, causing them to drift to a wire in the centre of the tube, creating a signal. The fast-response detectors consist of Resistive Plate Chambers and Thin-Gap Chambers, which both detect muons from ionisation of gas and are used to trigger on muons.

ATLAS also has a trigger and data acquisition system which decides which

events possess the characteristics that make them interesting for physics analysis [52]. The first-level trigger system is hardware-based and works on information from the calorimeters and the muon spectrometer, while keeping the event data in storage buffers. The second-level trigger system is software-based and selects around 1000 events per second.

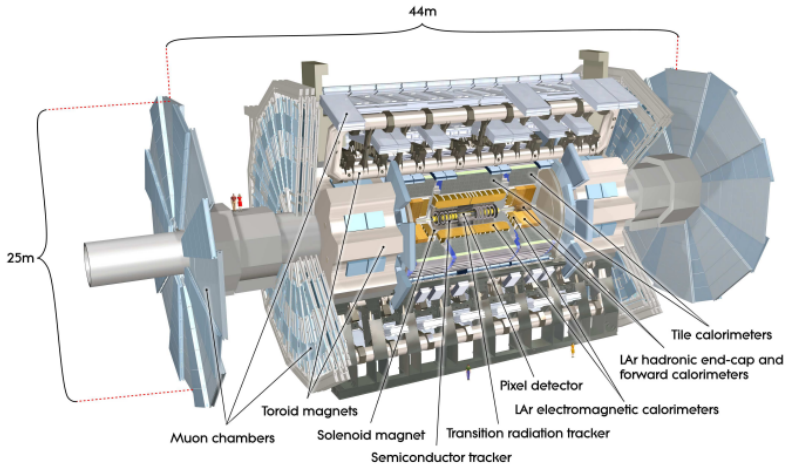


Figure 2.2: An illustration showing the main components of the ATLAS detector. Figure taken from [53].

2.2.1 Particle detection

The characteristics of the detected particles are found from the way in which they interact with the different layers [49,50]. When viewing a cross section of the detector, as shown in figure 2.3, the electric charge of a particle is measured by the curve of its trajectory, the sign of the charge being identified by the direction in which the curve is bent. The inner detector only detects electrically charged particles, like electrons and protons. As a result, neutral particles like photons and neutrons do not leave any trace.

The *electromagnetic* (EM) *calorimeter* measures all particles subject to the electromagnetic interaction [54]. Electrons and photons stop in this layer and produce EM showers of particles. The profile of their energy deposits in the EM calorimeter therefore determine the direction of the particles, as well as their energy. The hadronic calorimeter works in a similar way, but measures hadrons subject to the strong interaction, like protons and neutrons, and these are identified in this layer.

The hadrons stop in this layer due to hadronic showers, and are completely absorbed, which makes it possible to measure their energy. Although muons interact with the EM calorimeter, they only loose a small amount of their energy, and pass through. These are measured in the muon spectrometer, which is the outermost layer of the detector [51]. The muon spectrometer does not stop the muons, but instead measures their momenta from the curvature of their tracks. There are also some particles that are not detected. Particles such as the electroweak (EW) bosons and the Higgs boson decay before they reach the detector layers. The neutrinos are considered stable particles, which means they reach through all of the layers, but are not detected as they only interact through the weak interaction. This leads to some expected missing energy and momentum.

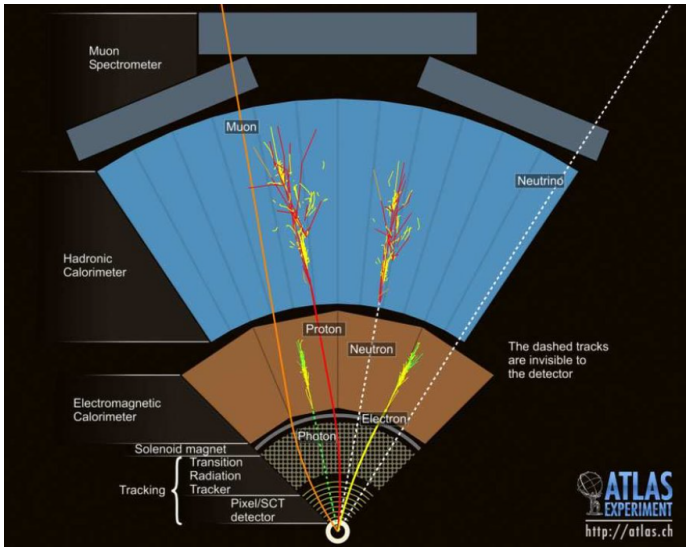


Figure 2.3: An illustration of a cross section of the ATLAS detector, showing the different layers and how different types of particles interact with them. Figure taken from [53].

2.3 Physics at the LHC

2.3.1 Controllable and known parameters

Because protons are composite particles, one cannot control the energy of every single quark. However, one can control the energy of the protons as a whole. This leads to a number of parameters that may be controlled with a high degree of precision for the incoming protons.

The energy of a collision is usually considered in the center-of-mass frame, leading to a *center-of-mass energy*, denoted by $\sqrt{s}$ [13]. This may be found from the energy and momentum of the particles by taking the square-root of the Lorentz-invariant quantity

$$s = \left(\sum_i E_i\right)^2 - \left(\sum_i \mathbf{p}_i\right)^2. \tag{2.1}$$

The center-of-mass energy during the full Run 2 data-taking period at the LHC was $\sqrt{s} = 13$ TeV [55].

The *interaction cross section* σ measures the probability that two incoming particles will interact with each other and lead to some new particles being produced, and is a parameter which may be calculated for the specific process one is interested in [56]. The *instantaneous luminosity*, $\mathcal{L}$, measures the ability of a particle collider to produce a required number of interactions [57] per second, and is defined as

$$\mathcal{L} = \frac{1}{\sigma}\frac{dN}{dt} \tag{2.2}$$

where σ is the cross section and $\frac{dN}{dt}$ is the number of events per time unit [58]. The unit of luminosity is $\mathrm{cm}^{-2}\mathrm{s}^{-1}$. For two parallel, approximately Gaussian proton beams colliding, the luminosity is

$$\mathcal{L} = \frac{N_1 N_2 f N_b}{4\pi\sigma_x\sigma_y}, \tag{2.3}$$

where N_1 and N_2 are the number of protons in the two colliding bunches, f is the revolution frequency and N_b is the number of bunches in one beam. σ_x and σ_y are the root-mean-square horizontal and vertical beam sizes.

The *integrated luminosity* L is the instantaneous luminosity integrated over a time period,

$$L = \int \mathcal{L}(t)dt. \tag{2.4}$$

This quantity is of importance as it is related to the number of expected events for some process with cross section σ,

$$L \cdot \sigma = \text{number of events}. \tag{2.5}$$

The unit of integrated luminosity is m^{-2}, and is usually given in the inverse unit of barns, where 1 barn = 10^{-28} m^2 [57].

2.3.2 Kinematic variables

As the protons are composite particles and only one of its partons typically takes part in an interaction, the longitudal component of its momentum is not known. However, because the protons travel in parallell with the beam line, the vector sum of the transverse components of the momentum of the partons in the initial state is zero. This is also true for the final state particles because of momentum and energy conservation. Therefore, in the analysis of the kinematics of an event, the transverse components of the variables are usually considered.

The 4-momentum of a particle may be reconstructed from variables that the detector measures. These are usually collected in a 4-vector of the form

$$p^{\mu}_{spherical} = (E, p_T, \eta, \phi), \tag{2.6}$$

where E is the energy and p_T is the transverse component of the momentum. η is the *pseudo-rapidity*, defined by $\eta = -\frac{1}{2}\ln\tan(\theta/2)$, where θ is the polar angle and ϕ the azimuthal angle [13]. An illustration of the coordinate system in the detector is shown in figure 2.4.

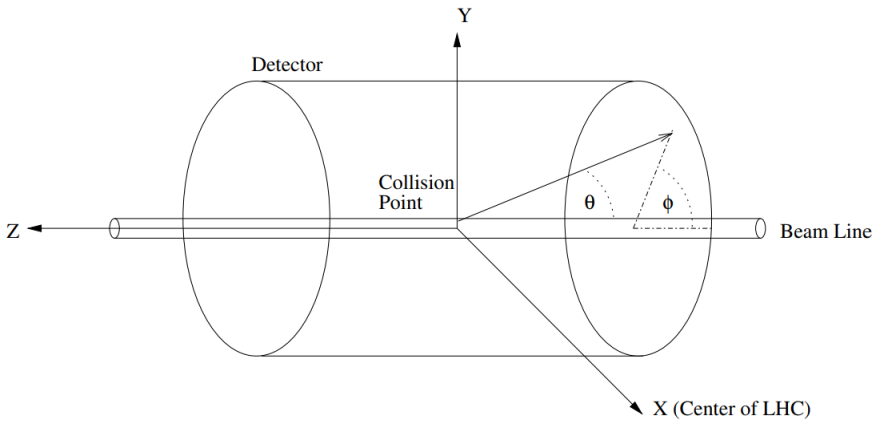


Figure 2.4: An illustration showing the coordinate system used in the detector for calculating properties of the observed particles. Figure taken from [59].

The *missing transverse energy*. E_T^{miss}. is the transverse energy that is expected

to be detected due to conservation of energy, but that is not detected. It is therefore the magnitude of the vector momentum imbalance in the plane perpendicular to the beam [60]. This quantity is determined by

$$E_T^{miss} = \|\mathbf{p}_T^{miss}\|, \tag{2.7}$$

where

$$\mathbf{p}_T^{miss} = -\sum_i \mathbf{p}_{T,i}. \tag{2.8}$$

is the *missing transverse momentum*, and the sum runs over all particles that are detected, as well as good tracks which are not associated with any particle, called *soft terms*.

It is often useful to distinguish between energy that is genuinely missing, and that which is missing due to sources like object misreconstruction, finite detector resolution or detector noise. This issue is mitigated by defining the E_T^{miss} *significance*, $E_T^{miss,sig}$, which is obtained from a likelihood formalism by

$$E_T^{miss,sig} \equiv 2\ln\left(\frac{\mathcal{L}(\vec{\varepsilon} = \sum \vec{\varepsilon}_i)}{\mathcal{L}(\vec{\varepsilon} = 0)}\right), \tag{2.9}$$

where $\vec{\varepsilon}$ is the true E_T^{miss}, $\sum \vec{\varepsilon}_i$ is the observed E_T^{miss} [61]. However, it may be thought of in a more simplified way as E_T^{miss} in units of its experimental uncertainty, $E_T^{miss,sig} = E_T^{miss}/\sigma(E_T^{miss})$.

The *invariant mass* of a system of particles is a characteristic of the system's momentum and energy, and is invariant under Lorentz transformations [13]. It is an important property in experimental particle physics because it tells you which particle a number of final state particles may stem from. The square of the invariant mass of a system of two particles is

$$\begin{aligned} m^2 &= (E_1 + E_2)^2 - \|\mathbf{p}_1 + \mathbf{p}_2\|^2 \\ &= m_1^2 + m_2^2 + 2(E_1 E_2 - \mathbf{p}_1 \cdot \mathbf{p}_2). \end{aligned} \tag{2.10}$$

If one considers a particle with mass m decaying to two particles where one of them is invisible, the *transverse mass* is sometimes a useful quantity. In this case, the invisible particle only shows itself through the missing energy. The definition of transverse mass used by particle physicists is somewhat different from the standard definition, and is of the form

$$\begin{aligned} m_T^2 &= (E_{T,1} + E_{T,2})^2 - (\vec{p}_{T,1} + \vec{p}_{T,2})^2 \\ &= m_1^2 + m_2^2 + 2(E_{T,1} + E_{T,2} - \vec{p}_{T,1} \cdot \vec{p}_{T,2}), \end{aligned} \tag{2.11}$$

where E_T is the transverse energy of each of the daughter particles [62]. However, in reality only one of the particles is detected, while the other is replaced by $\mathbf{p}_T^{miss}$. One may therefore rewrite the formula as

$$m_T = \sqrt{2(p_T^l p_T^{miss} - \mathbf{p}_T^l \mathbf{p}_T^{miss})}, \tag{2.12}$$

where $\mathbf{p}_T^l$ represents the measured particle and p_T^{miss} is the magnitude of $\mathbf{p}_T^{miss}$, as earlier defined. Also it is sometimes useful to measure the *hadronic activity*, H_T, which is defined as the scalar sum of the momenta of the hadronic jets in an event

$$H_T = \sum_i \|\mathbf{p}_{T,i}\|. \tag{2.13}$$

In this chapter, the main components of the ATLAS detector have been explained, as well as how the particles are detected and identified. The parameters and kinematic variables that will be used later in our analysis have also been introduced. We will now lay out the search strategy and introduce the signals that will be searched for.

3 Search strategy[†]

In this chapter, we will begin by outlining the general search methods. The signal models for the search are then introduced, as well as the SM processes that contribute to the background. In chapter 6 and 7, the ML method will be compared with the cut and count method, which both will be introduced in this chapter, in order to study its performance relative to a standard method. This chapter forms the basis of both analyses.

3.1 General outline of search methods

In this thesis, we will search for a new gauge boson Z', decaying to a lepton-antilepton pair, and dark matter through the process $pp \rightarrow Z' + \chi\chi \rightarrow l^+l^- + E_T^{miss}$, which means that the final state collected by the detector will be a lepton-antilepton pair and missing transverse energy (E_T^{miss}). The models which form the basis of this process are discussed in detail in section 3.2.

In order to perform a search for hypothetical, new particles or processes in proton-proton collisions with the ATLAS detector, one needs to know how such processes would behave or look in the data. In classical physics, it is often possible to perform calculations which yield a result that one may either verify or falsify. Particle physics is subject to quantum uncertainty, which means that although it is possible to calculate certain characteristics, one can often only calculate how often the process will happen, given the initial conditions. One may also calculate the characteristics of the final states when such a process occurs, which gives information about what the detector will measure, although this is subject to quantum uncertainty as well.

Because of the reasons stated above, it is usually not possible to identify the process that has taken place from a single event, as each event of the same type of process lead to different final characteristics. For example, the transverse momentum and the angle of particles produced in specific processes will not be the same every time, although they stem from the same process and similar initial conditions. However, the statistical distribution of these variables may be estimated. Often, there are other processes that can lead to the same final state being detected and

[†]Adapted from [1]

with some of the same characteristics, which is referred to as *background*.

The strategy for searches for new particles has to take into account that the *signal*, which is the process that is searched for, may be hidden behind large amounts of background. In addition, if the signal is there, it may only account for a small fraction of the observed events. The first preparatory step in a search is to estimate the amount of events that is expected from all of the processes in the Standard Model that lead to the same final state particles, as well as their characteristics, meaning the distributions of the kinematic variables the detector is expected to measure. This must also be done for the signal that is searched for. The standard way to do this is to perform Monte-Carlo simulations of each type of process, which takes into account the randomness of quantum behaviour. This will be discussed further in section 5.1.

Then, one may compare the data recorded by the detector with the expected data produced by the simulations. If the simulations are performed correctly, and assuming the Standard Model in general is an accurate representation of particle physics, the difference between the recorded data and the simulations is expected to be small in most regions, especially in those where a large number of events are recorded.

However, if there is a flaw or something missing in the theory, there may be detected an excess or lack of events, compared to simulations, in certain regions. The region where a large amount of signal events are expected, is called the *signal region*. If a larger amount of events are observed in the signal region than is expected from the Standard Model, and the excess of events also is of a similar magnitude to what is expected from the signal model, it will act as evidence for the signal model. In other cases, the excess of events expected from the signal is not observed, and this will act as evidence against the signal model describing reality. If the signal is large enough, it can in principle be visible in the real data. However, as mentioned earlier, the signal is often several orders of magnitude smaller than the background and therefore not visible. Therefore, one needs to use a strategy that increases the expected amount of signal in relation to the expected background. After this is done, one may calculate the expected significance Z, which tells us how likely we are to be able to claim a discovery if the signal model exists in nature, or if it may be falsified if it does not exist. If the expected significance is too low, it is likely

not possible to verify or falsify the theory. How the expected significance is measured will be explained in section 5.9. One may also use the results to set limits on the parameters of the model. First, we will introduce the standard cut and count method.

3.1.1 Cut and count method

One of the most common ways to increase the expected amount of signal relative to background, is by making cuts on some kinematic variables, so that all events outside a certain interval are excluded. This is called the *cut and count* (C&C) method. For example, if a model predicts the signal events to have a peak in the invariant mass at 300 GeV, with few signal events having an invariant mass below 200 GeV or above 400 GeV, one may exclude all events outside this region. The same may be done for other suitable variables, which in total will reduce the background more than it reduces the signal. There are also methods for deciding how to make the most effective cuts, which we will not go into detail about here. The region that is left after a set of cuts are made, is called the signal region, as discussed above. Other cuts are often also made in order to construct control and validation regions. The control regions are constructed in order to verify that the simulations represent the data in a satisfactory way. The validation regions are used in order to confirm that the background modelling inferred from the control regions is representative of the background expected in regions closer to the signal regions.

3.1.2 ML classification

Another way of maximizing the expected amount of signal compared to background, is by using machine learning (ML) methods. The most intuitive way to do this is by using a binary classification method. While the cut and count method attempts to find a region in the parameter space where it is expected to be a high amount of signal in comparison to background, an ML classification method analyses every single event separately in order to decide whether it is likely to be a signal or background event. A signal region is then constructed by taking the set of events receiving a probability above some threshold of being a signal event. Classification methods will be discussed in more detail in section 4.1.1. We will now introduce the signal models for the search, as well as the different types of background processes

that lead to the same final state particles.

3.2 Signal models

We will search for evidence of new phenomena predicted by two different groups of models, called *Dark Higgs Models* and *Light Vector Models*. They are different in important ways, but also share some characteristics.

Both of the models have a mechanism of dark matter production, where the dark matter particles are mediated by a new gauge boson, called Z' [63]. The models thus produce a signal $Z' + E_T^{miss}$, where the missing energy is due to dark matter particles that are not detected. The Z' boson can decay to a pair of leptons (l^+l^-), where $l = e^-, \mu^-$, (which are the final state particles used in this search) or to a quark-antiquark pair which may then lead to 2-jets (jj). It may therefore be possible to observe a resonance in the dilepton mass spectrum, as well as elevated levels of missing transverse energy. However, one should be aware that a dilepton resonance combined with missing transverse energy does not necessarily require a Z', as other processes could cause similar results, such as new scalar resonances and colored resonances.

In both models, we assume that the Z' corresponds to a new $U(1)'$ symmetry group and couples to quarks by the interaction term

$$\mathcal{L} \supset -\sum_q g_q \bar{q} \gamma^\mu q Z'^\mu \tag{3.1}$$

where g_q is the coupling strength between Z' and quarks. The couplings to leptons are also vector couplings of the form

$$\mathcal{L} \supset -g_l \bar{\psi} \gamma^\mu \psi Z'^\mu, \tag{3.2}$$

where $\bar{\psi}$ and ψ represent the antilepton and lepton states.

3.2.1 Dark Higgs model

The Dark Higgs model [63] assumes there is a new massive scalar particle that couples to the Z', called the *dark Higgs boson*. As this particle couples to invisible states (dark matter particles), one of the main strategies when searching for it could be to search for missing transverse energy (E_T^{miss}). The processes that are studied in our

case have three free mass parameters. These are the mass of the Z' boson, the mass of the darks sector fermion (χ) and the mass of the dark Higgs boson (h_d). There are also three coupling parameters, g_D, g_q and g_l, which represent the coupling of the Z' boson to the dark Higgs boson, the coupling of Z' to quarks, and of Z' to leptons, respectively. In figure 3.1, a Feynman diagram representing a possible dark Higgs process leading to two leptons is shown.

The model [63] is implemented by assuming a new $U(1)'$ symmetry with a charged scalar field Φ_D representing the dark Higgs field and a singlet scalar ϕ_X representing dark matter states. These fields give the following contributions to the lagrangian

$$\begin{aligned}\mathcal{L} \supset\ & |D_\mu \Phi_D|^2 + \mu_D^2 |\Phi_D|^2 - \lambda_D |\Phi_D|^4 - \frac{1}{4}(F'_{\mu\nu})^2 \\ & + \frac{1}{2}(\partial_\mu \phi_X)^2 - \lambda_X |\Phi_D|^2 \phi_X^2 - V(\phi_X),\end{aligned} \tag{3.3}$$

where V is the scalar potential and $\Phi_D = \frac{1}{2}(v_D + h_D)$, where v_D is the vacuum expectation value for the dark Higgs field. λ_D and μ_D are constants analogous to λ and μ in the Higgs potential. The coupling of h_D to Z' is given by

$$Q_h g_z M_{Z'} h_D Z'_\mu Z'^\mu \equiv g_{h_D} M_{Z'} h_D Z'_\mu Z'^\mu \tag{3.4}$$

where Q_h is the charge of Φ_D, which is absorbed when defining the effective coupling g_{h_D}. The dark Higgs boson may decay to the ϕ_X states through the λ_X coupling. The mass of the dark Higgs boson is unknown, but it cannot be much heavier than the Z' or lighter than the χ. We consider the two scenarios which are within these ranges, which are labelled the *light dark sector* and the *heavy dark sector*. The masses in the different scenarios considered, are listed in table 3.1.

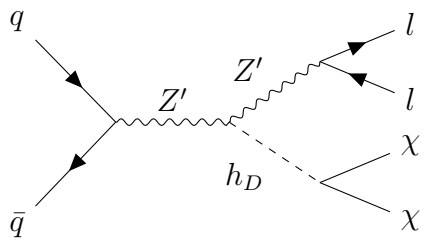


Figure 3.1: An example of a process in the dark Higgs model leading to two final state leptons and a pair of dark matter particles.

	Dark Higgs	Light Vector
Light sector	$m_\chi = 5$ GeV	$m_{\chi_1} = 5$ GeV
	$m_{h_D} = 125$ GeV	$m_{\chi_2} = m_{\chi_1} + m_{Z'} + 25$ GeV
Heavy sector	$m_\chi = 5$ GeV	$m_{\chi_1} = m_{Z'}/2$
	$m_{h_D} = m_{Z'}$	$m_{\chi_2} = 2m_{Z'}$

Table 3.1: Overview of the masses of particles in the light dark sector and heavy dark sector in the dark Higgs and light vector model [63].

3.2.2 Light vector model

When the Z' is light, it may be produced from decays of a heavy dark sector particle χ_2. This scenario is called the light vector model [63]. An example of such a process leading to a lepton-antilepton pair and two dark states χ_1 and χ_2 is shown in figure 3.2. In this case, the χ_2 represents a heavier state decaying to a Z' and a lighter χ_1 dark sector particle, which is a stable dark matter candidate. The processes that are studied in our case have three free mass parameters, which are the mass of the Z' boson and the masses of the two dark sector fermions, χ_1 and χ_2. This model does not include a dark Higgs boson. There are also three coupling parameters, g_D, g_q and g_l, which in a similar way to the dark Higgs model represent the coupling of the Z' boson to the dark sector fermions, the coupling of Z' to the quarks, and of Z' to leptons, respectively.

In the mathematical description of the model, the Z' couples to a fermion χ, which has both Dirac and Majorana mass. It initially has Dirac mass M_d. A Majoana mass M_m may then be generated from the vacuum expectation value of a $U(1)'$ Higgs through an interaction $y_\chi \Phi_\chi \bar{\chi} \chi^c$, so that

$$\mathcal{L} \supset \bar{\chi}(i\not{D} - M_d)\chi - \frac{M_m}{2}(\bar{\chi}\chi^c + h.c.). \tag{3.5}$$

This leads to two Majorana states χ_1 and χ_2 with masses $M_{1,2} = |M_m \pm M_d|$. The interaction between these states and the Z' are off-diagonal and is given by

$$\frac{g_\chi}{2} Z'_\mu (\bar{\chi_2}\gamma^\mu\gamma^5\chi_1 + \bar{\chi_1}\gamma^\mu\gamma^5\chi_2) \tag{3.6}$$

In this model, the cross section increases with lower χ_1 mass. Therefore, the light dark sector is an optimistic case with a light χ_1, while in the heavy sector the dark fermion masses scale with $m_{Z'}$. The masses in these scenarios are listed in table 3.1.

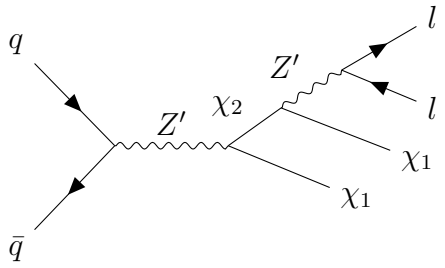


Figure 3.2: An example of a process in the light vector model leading to two final state leptons and dark matter particles.

3.3 Background contributions

The Standard Model includes many processes that may result in the detector measuring two leptons and E_T^{miss}. We will now present the different types of processes that contribute to the background and that will be modelled with Monte-Carlo (MC) simulations.

3.3.1 Drell-Yan and Z+jets

A *Drell-Yan* (DY) process takes place in hadron-hadron collisions when a quark from one hadron and an anti-quark from another annihilate in order to produce a virtual photon γ^* or a real or virtual Z/Z^* boson which then decay to a lepton-antilepton pair. This process is shown in figure 3.3 a). In our search for a new Z' boson, the DY background is irreducible, as the Z' is produced through a similar interaction $q\bar{q} \to Z' \to l\bar{l}$, resulting in similar topology and kinematics. However, it does not necessarily produce a similar distribution for all kinematic variables. For example, the resonance in the invariant mass (m_{ll}) distribution will depend on the Z' mass, which will likely deviate from the Z/Z^* mass. The m_{ll} distribution for the DY background instead has a large amount of events close to the Z/Z^* mass, but also a long tail due to virtual Z^*. When the process also results in jets from radiation of the initial partons in the collision, it is usually called *Z+jets*. The DY/Z+jets is the dominant background type in the processes we are searching for.

3.3.2 Top-antitop production

Another major background contribution is *top-antitop production* ($t\bar{t}$). The $t\bar{t}$ pairs are produced from various gluon-gluon interactions and q-$\bar{q}$ annihilation, before de-

caying quickly due to the large mass of the top quark. Each top quark decays through the weak interaction to a W boson and a bottom quark. The W may then decay to a lepton-neutrino pair, where the presence of the neutrino can only be inferred from the measured E_T^{miss} in the event. Since each top-quark will produce one W each, these events may have two leptons and E_T^{miss} in their final states. An example of a top-antitop production process is shown in figure 3.3 b).

3.3.3 Single top

Single top (ST) quarks may be produced through a variety of processes. An example is shown in figure 3.3 c). As in the $t\bar{t}$ background, the top quark may decay to a W boson and a bottom quark where the W then decays to a lepton-neutrino pair (or antilepton-neutrino pair). If the single top is produced in combination with a W boson, it may therefore lead to a lepton-antilepton pair.

3.3.4 Diboson

Diboson processes are processes where two gauge bosons are produced in pairs (WW, WZ, ZZ or γZ) from two incoming quarks. Examples of such processes are shown in figure 3.3 d) and e). If a W^+W^- pair is produced, they may both decay to a lepton-neutrino pair. If a Z boson is produced, it may decay to a lepton-antilepton pair or a neutrino-antineutrino pair. This background is small compared to the ones mentioned above, but it is irreducible.

3.3.5 W+jets

W+jets background refers to the scenario where a single W boson is produced in combination with jets. The W may decay to a lepton and a neutrino. Although it will not result in a lepton-antilepton pair, the jet is sometimes misidentified as a lepton, or a conversion of a radiated photon to an e^+e^- pair may occur, where one of these are reconstructed as an electron or positron. Leptons may also come from semileptonic decays within B-hadrons. The W+jets background is small, but still part of the MC simulated background. An example of a W+jets process is shown in figure 3.3 f).

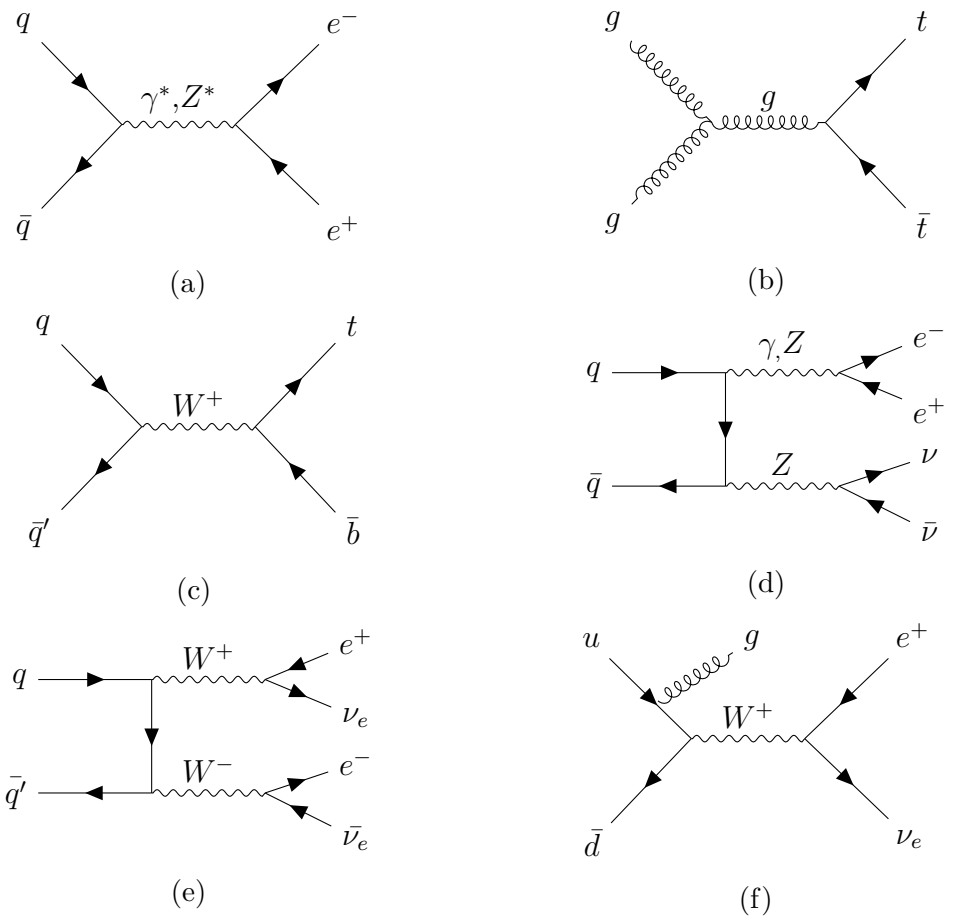


Figure 3.3: Examples of Feynman diagrams for some of the processes contributing to the background: a) Drell-Yan Z^*/γ^* production, b) top-antitop production, c) single top production, d) ZZ or Z,γ diboson production, e) WW diboson production and f) $W+$ jets. Final states with $\mu^+\mu^-$ are also considered.

Data Sample	2015+2016	2017	2018	Comb.
Integrated luminosity (fb^{-1})	36.2	44.3	58.5	139.0
Total uncertainty (fb^{-1})	0.8	1.0	1.2	2.4

Table 3.2: Integrated luminosity that was labelled *good for physics* during different time periods of Run 2, as well as its total uncertainty [64].

3.4 Data sets

The data studied in this thesis are from proton-proton collisions recorded by the ATLAS detector during the Run 2 at the LHC. The center-of-mass energy of the collisions was $\sqrt{s} = 13$ TeV, corresponding to a beam energy of 6.5 TeV, with an average of 1.1×10^{11} protons per bunch and a bunch-spacing of 25 ns [47, 55]. Run 2 took place from 2015 to 2018, and was divided into three periods, the first one in 2015-2016, the second in 2017 and the third in 2018. Figure 3.4 shows the integrated luminosity recorded during Run 2, as a function of time. The green part is the total integrated luminosity delivered by the LHC, and the yellow part shows the part of it that was recorded by the ATLAS detector. The blue part is the integrated luminosity of the events that passed all data quality requirements. This is the part that is used in our analysis, and it amounts to 139 fb^{-1} [64]. The integrated luminosity and its uncertainty for each period is listed in table 3.2. The real data will be compared with Monte-Carlo (MC) simulated events. This is done in order to compare the data with predictions from the Standard Model, and to search for deviations to new physics phenomena. This will be discussed further in section 5.1. As a machine learning based analysis will be performed in chapter 7, we will now introduce the theory of neural networks, which will be used to perform in the analysis.

In this chapter, we laid out the two different search strategies that will be used in the analyses in chapter 6 and 7. The signal models that will be searched for were introduced, as well as the background contributions. In the following chapter, machine learning and neural networks will be introduced, as these will be used in the analysis in chapter 7.

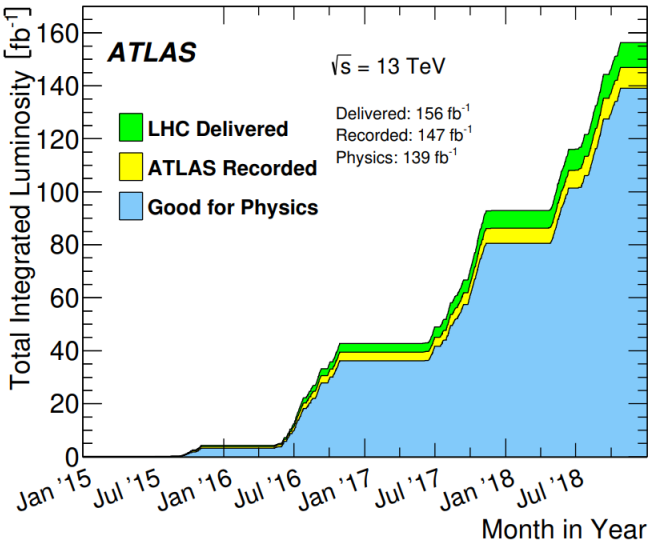


Figure 3.4: The integrated luminosity at the LHC during Run 2 as a function of time. The green (including the area below it) is the amount delivered by the LHC, while the yellow area is how much was recorded by the ATLAS detector. The blue area is the amount that passed data quality requirements in order to be considered *good for physics*. Figure taken from [53].

4 Neural networks

Although the standard cut and count method is effective in many cases, there are other methods that may sometimes yield better results. Machine learning (ML) methods such as neural networks and XGBOOST (eXtreme Gradient Boosting) have become increasingly popular alternatives. These possess several possible advantages. One of these is their ability to find correlations between variables for each specific event. In the cut and count method, the same cuts are performed for all of the events without knowing how they will affect the distributions of other variables. Also, the cuts are usually made on a relatively small number of variables, which means that possible information from other variables is not used. A machine learning model may analyze a large amount of parameters and find connections between them.

There are however some disadvantages of using machine learning models. One of them is the possibility of overfitting, which will be discussed in more detail in section 4.1.3. This means that the ML model finds spurious relationships in the training data, which will not be present when tested on a new data set. Another problem, which applies to the cut and count method, is if the MC simulation of the signal model does not exactly represent the way the real signal would look if it exists in reality, as there are usually uncertainties in the free parameters. An example of this would be if the simulations produce correlations between specific features that are not present in the real signal if a version of it exists. This may lead the ML model to classify the signal as background. Although this would also be an issue when using the cut and count method, it often has a larger margin of error as the cuts are usually relatively broad, while the ML model may have more specific criteria to classify an event as signal. In order to reduce this problem, one should attempt to avoid overfitting. We will now introduce some of the theory behind neural networks, which is the search method used in this thesis (in addition to the cut and count method), as well as explaining some of the choices that have been made when developing and training the neural networks. Specific optimization choices are discussed later in chapter 7. This chapter mainly follows references [65, 66].

4.1 General ML concepts

4.1.1 Classification problems

Machine learning may be used for different kinds of tasks that would require different machine learning methods. The task it will be used for in this thesis is called *classification*, specifically *binary classification*. Classification problems are problems where each data point belongs to a category, and the task of the ML model is to decide which group it belongs to. The most common ML output for k different categories is a vector $\vec{\hat{y}} = (t_1, .., t_i, ..., t_k)$, where the value of each coordinate i represent the probability of the data point belonging to category i. This is known as one-hot encoding. However, if there are only two categories ($k = 2$), one may instead use only a single *target variable* $y = 0, 1$, where $y = 0$ and $y = 1$ respectively represent the two different categories C_1 and C_2, and the ML output $\hat{y}$ will be somewhere in between these values, $\hat{y}$ representing the model's assigned probability that the data point belongs to C_1 and $1 - \hat{y}$ the probability that it belongs to C_2. It should be noted that this kind of approach is a form of *supervised learning*, where each data point is associated with a label or target, and these are known in the data set we are using for training the model. In cases where we have data set without labels, *unsupervised algorithms* are needed instead.

The problem of separating signal events from background events may be considered a binary classification problem, where signal events are assigned with a target variable $y = 1$ and background events are assigned $y = 0$. The input for such a model is an $N \times M$ matrix where each row (N) represents an event and the columns (M) represent the different *features*, which may be kinematic variables or other characteristic, such as flavor. The model then produces an output value y for each event, representing the probability of it being a signal event and $1 - y$ represents the probability of it being a background event.

4.1.2 Evaluation metrics

In order to optimize an ML model, a method for measuring the quality of its output is necessary. As ML methods are used for many different types of problems, the suitable evaluation metric will depend on the problem type. Here we will focus on classification problems, specifically binary classification.

As there are only two target labels 0 and 1, the ideal output is one that is exactly

the correct label, either 0 or 1. As the output will usually be somewhere in between these values, the closer it is to the correct label, the better. A simple and often effective method is to measure the *accuracy* of the model. The accuracy is defined by

$$accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} = \frac{TP + TN}{TP + FP + TN + FN} \tag{4.1}$$

where TP, FP, TN and FN respectively are *true positives, false positives, true negative* and *false negatives*. A true positive is a data point that is correctly classified as the $y = 1$. FP, TN and FN are defined using the same reasoning. Whether a data point is classified as positive or negative, in our case signal or background, depends on the threshold required. A standard threshold is to use 0.5, meaning outputs above 0.5 is classified as positives and outputs below 0.5 as negatives. It should then be noted that although the accuracy will be between 0 and 1 (representing 0% and 100% accuracy respectively), an ML model with no predictive power will be expected to have an accuracy close to 50% in the case of binary classification, as there is a 50% chance of guessing the right label without any information.

A different way of testing performance, which is somewhat similar to accuracy, is measuring the area under the *Receiver Operating Characteristic* (ROC) *curve*, also known as *Area Under Curve* (AUC). A typical ROC curve is shown in figure 4.1. The ROC curve is the True Positive Rate (TPR) plotted against the False Positive Rate (FPR) at different thresholds between 0 and 1. The TPR and FPR are defined as

$$TPR = \frac{TP}{TP + FN},\ FPR = \frac{FP}{FP + TN}. \tag{4.2}$$

The AUC is the area under this curve, which is a number between 0 and 1. When the AUC is 1, the model correctly distinguishes between all the positives and negatives at different thresholds. If the AUC is 0, it predicts that the positives are negatives and negatives are positives. As the AUC tests the model at different thresholds, it provides more nuanced information about its output than the accuracy does. It can often be useful to use these methods in combination as they provide complementary information.

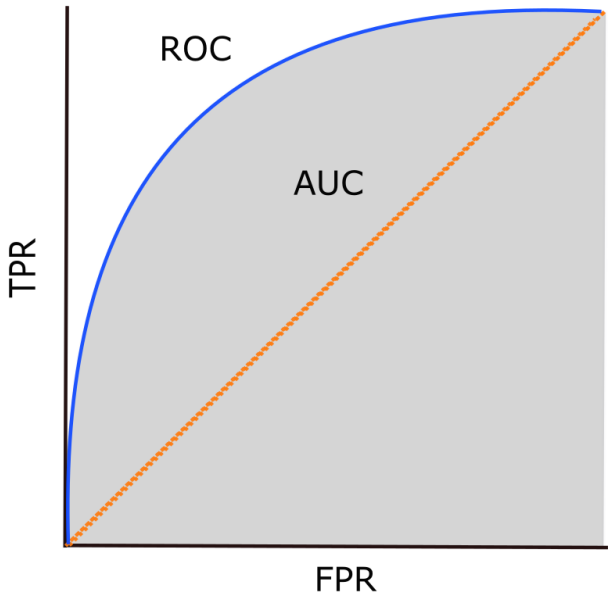


Figure 4.1: A typical ROC curve (the blue line) for an efficient classifier. The shaded area is the AUC. The orange line corresponds to a random classifier with no predictive power. The figure is my own work.

As will be discussed in more detail in section 4.2, the neural network assesses the error of its output on the training data for every epoch in order to optimize its weight parameters. This is done by using the maximum likelihood principle, which involves minimizing a *loss function*. The main role of the loss function is to determine the error of the output. Which loss function to use depends on the type of problem. For a linear regression problem, one would minimize the mean-squared error, while for classification one uses the *cross-entropy loss*, in our case the *binary cross-entropy loss*, as the task of separating signal and background is a binary problem. In the case where we have a true probability or target variable y, a value $\hat{y}$ predicted by the ML model, the dissimilarity between y and $\hat{y}$ is measured by the binary cross-entropy loss is

$$H(y, \hat{y}) = -\sum_i y_i \log \hat{y}_i = -y \log \hat{y} - (1 - y) \log(1 - \hat{y}). \tag{4.3}$$

However, in the optimization stage, the loss is calculated for a large number of events, so the average loss is needed. This is given by

$$J(\mathbf{w}) = -\frac{1}{N}\sum_{i=1}^{N} H(y_i, \hat{y}_i) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log \hat{y}_i - (1 - y_i)\log(1 - \hat{y}_i)]. \tag{4.4}$$

where $\mathbf{w}$ represents the vector of optimized weights (which will be discussed in section 4.2) and N is the number of observations or events. $J(\mathbf{w})$ is usually called the *cost function*.

4.1.3 Training and testing ML models

In order to develop an optimal ML model, it is usually necessary to train and test the model many times and test different features, such as *hyperparameters* (which will be introduced in section 4.6) and *network architectures* (which will be described in section 4.2) in order to achieve the best result, using the evaluation metrics defined in section 4.1.2. In the process of training and testing, there are some practises that should be followed in order to optimize efficiently and to make sure the results from the model may be trusted.

The data set that will be analyzed should first be split into a *training set* and *test set*. The main reason for this is that it is not necessarily difficult for an ML model to learn the set it is training on and reach a high accuracy when tested on the same set. The reason for this is that it can learn every single data point and relate it to its target variable. The problem usually arises when it is tested on a new data set (the test set), as the data points will be different, and small deviations in the characteristics of each data point may lead to a large difference in output. The difference in error between the output for the training and test set, is called the *generalization error*. A large generalization error is a sign of overfitting. One should instead train the model on a training set and then check its performance on a test set to see whether it has captured the broad trends and characteristics that exists in the data. One may then make changes to the ML model and re-train it in order to optimize its performance on the test set. Usually, the test set does not need to be as large as the training set, although there are no specific rules for this. The advantage of having a larger training set, is that one has more training data which is expected to lead to better performance. However, it is important to keep a large enough test set so that the performance metrics can be trusted. In our analysis, we will be using 50% of the data for testing, is more than what is

necessary in order to check its performance. However, this is the data set that is used to perform the search, and therefore it should be as large as possible in order to have enough statistics, without significantly affecting the ML model performance.

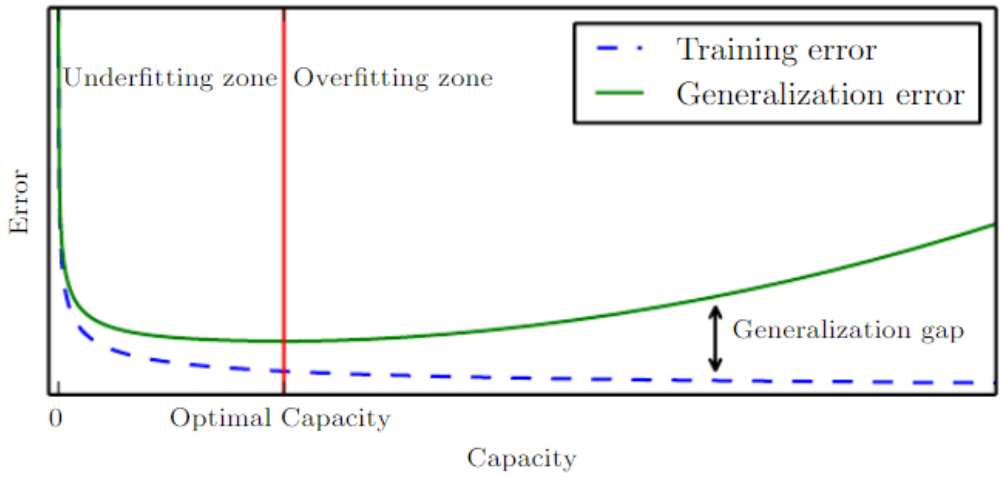


Figure 4.2: An illustration showing a typical loss or error curve as a function of training epochs. Although the training loss (the blue, dashed line) keeps decreasing, the generalization error (the green line) starts increasing after a while due to overfitting. Figure taken from [65].

A problem that arises when optimizing for the test, is that one is still fine-tuning the ML model in order to reach a high performance on that specific set. While the negative effects of this often are small, it may be considered an indirect form of overfitting. The problem may be solved by introducing a *validation set*. This is a subset of the training set that the model will not train on, but that will be used for testing the model during training. Because of the reason mentioned above, one should therefore not use the test set for making changes to the network, but instead use the validation set. After the model has been optimized for the validation set, it is tested on the test set. The validation set is also useful for preventing overfitting while training. Figure 4.2 shows a typical situation where the training loss keeps decreasing while the validation loss (the blue, dashed line) decreases for some time, before reverting and increasing due to overfitting (causing the generalization error, which is the green line, to increase). It is useful to check the loss and and accuracy for the training and validation set periodically during training. If the performance on the validation set does not improve after a number of training epochs (which is discussed further in section 4.6), while the performance on the training set keeps improving, the network is likely overfitting. In this case, it is useful to initialize *early*

stopping, which prevents the model from training further. However, this does not protect the model from overfitting if it is designed in a way that causes it to overfit easily.

4.2 Feed-forward neural networks

Artificial neural networks, from now on referred to as *neural networks* (NN), generally refer to a model that consists of *nodes* that are assembled in layers, and interact with each other in order to perform a task. They loosely resemble the way neurons function in a brain. Its general structure is shown in figure 4.3. The way the neural network is structured, such as the number of layers and neurons per layer is referred to as the *network architecture*.

We will focus on what is known as *feed-forward neural networks*, specifically the *multilayer perceptron*, that attempts to approximate a multi- or single-variable function. It is called feed-forward because information from the input $\mathbf{x}$ flows through the different layers in the network in order to produce an output. The network consists of a number of input nodes, which equals the number of features in the data samples, as well as a number of output nodes, depending on the desired type of output. In the case of binary classification there is one output node, which produces an output between 0 and 1. Between the input and output layer, there are a number of *hidden layers* which determine the *depth* of the network, and a number of nodes in each layer which determine the *width*. Each node in a layer interacts with each node in the next layer.

In the multilayer perceptron, each layer, except the input layer, consists of an *activation function*, taking a linear function $\mathbf{w}^T\mathbf{x}_{nodes}+\mathbf{b}$ as input, where $\mathbf{x}_{nodes} \in \mathbb{R}^p$ is the input vector from nodes in the previous layer (p is the number of nodes in the previous layer), $\mathbf{w}$ is a $p \times q$ matrix (q is the number of nodes in the next layer) where its elements are called *weights* that the network attempts to optimize and $\mathbf{b} \in \mathbb{R}^q$ is a vector called a *bias*. The output from a layer is therefore of the form

$$\mathbf{h} = g(\mathbf{w}^T\mathbf{x}_{nodes} + \mathbf{b}) \tag{4.5}$$

where g is an activation function and $\mathbf{h} \in \mathbb{R}^q$ is a vector which becomes the input of the next layer. Different types of activation functions are described in section 4.4. In the multilayer perceptron, the layers are stacked in the following way:

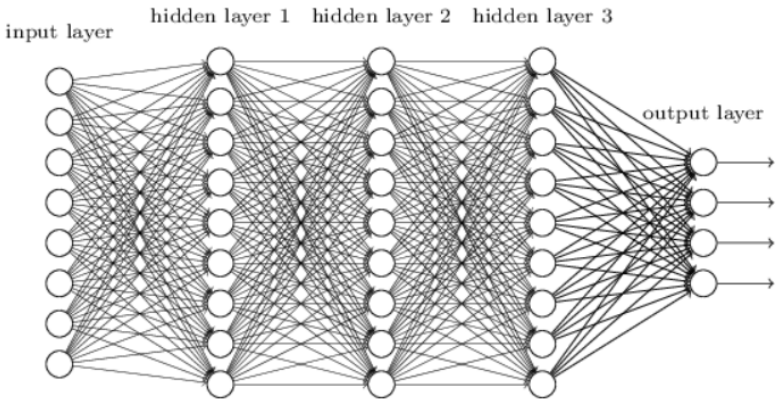


Figure 4.3: A typical neural network structure with 8 input features, 3 hidden layers and 4 output parameters. In a classification problem, there is only one output parameter. Figure taken from [66].

$$\begin{aligned} \mathbf{h}_0 &= \mathbf{x} \\ \mathbf{h}_1 &= g_1(\mathbf{w}_1^T \mathbf{h}_0 + \mathbf{b}_1) \\ &\vdots \\ \mathbf{h}_L &= g_L(\mathbf{w}_L^T \mathbf{h}_{L-1} + \mathbf{b}_L) \end{aligned} \tag{4.6}$$

Once the model is trained, the weights are kept fixed and the data my be sent through the network in order for it to produce an output $\mathbf{h}_L$.

4.3 Maximum likelihood estimation

In order for the ML model to learn, it needs a method to follow in order to achieve a better accuracy. Different algorithms will be discussed in section 4.6. The underlying principle used in most machine learning algorithms, is the *maximum likelihood principle*. The method attempts to find the weight parameters, discussed in section 4.2, that lead to the highest probability of producing the same output as the target variable. This is done by maximising the *likelihood function*.

We assume that $p_{model}(\mathbf{x}; \boldsymbol{\theta})$ maps a configuration $\mathbf{x}$, representing the $N \times M$ data array, into an estimation of the true probability or target $p_{data}(\mathbf{x})$. It is also assumed that the data in $\mathbf{x}$ are independent and identically distributed. $\boldsymbol{\theta} = [\theta_1, \theta_2, ..., \theta_M]^T$ represents the parameters that determine the distribution. In the case of neural networks, these are the weights. The maximum likelihood estimator [65] for θ is

$$\hat{\boldsymbol{\theta}} = \arg\max_{\theta} p_{model}(\mathbf{x}; \boldsymbol{\theta}) \tag{4.7}$$

A convenient way to express the maximum likelihood estimator is by using the logarithm, as it will not change the *arg max*. This is called the *log-likelihood*. The expression may then be transformed to a sum

$$\hat{\boldsymbol{\theta}} = \arg\max_{\theta} \sum_{i=1}^{M} p_{model}(\mathbf{x}^{(i)}; \boldsymbol{\theta}), \tag{4.8}$$

where the sum runs over the features. In a machine learning context, this may be generalized to

$$\hat{\boldsymbol{\theta}} = \arg\max_{\theta} \sum_{i=1}^{M} p_{model}(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}; \boldsymbol{\theta}) \tag{4.9}$$

where $\mathbf{y}$ are the targets. The maximum likelihood principle may be interpreted as minimizing the dissimilarity between the observed distribution p_{data} and the distribution produced by the neural network. This is best measured by the *Kullback–Leibler divergence* (KL-divergence)

$$D_{KL}(p_{data}||p_{model}) = \sum_{i=1}^{N} p_{data}(\mathbf{x})[\log p_{data}(\mathbf{x}) - \log p_{data}(\mathbf{x}; \boldsymbol{\theta})] \tag{4.10}$$

Minimizing the KL-divergence is mathematically equivalent to minimizing the cross-entropy loss. Therefore, a possible method for optimizing an ML model is using an algorithm that minimizes this quantity. This will be discussed further in section 4.6. However, we will first introduce the main types of activation functions.

4.4 Activation functions

The *activation function* g_i, introduced in section 4.2, produces the output of a node. It decides whether a neuron will fire or not, given some input, meaning it should generally return a value close to zero if the information in the input is not considered important to the output of the network. However, there are also activation functions with a range beyond 0 and 1. It should be noted that it, in principle, is possible to use the linear unit $\hat{y}$=$\mathbf{w}^T\mathbf{h}$+$\mathbf{b}$ without any further treatment. However, activation functions possess some desirable qualities which will be discussed in this section.

The type of activation function does not have to be the same in every node, although it is common to use the same activation function in all the nodes in the hidden layers and possibly a different one in the output layer, depending on the desired output. We will now introduce the activation functions that will be used in the ML analysis.

4.4.1 Sigmoid

The *sigmoid* function is a logistic activation function given by

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \tag{4.11}$$

The sigmoid function ranges from 0 to 1 in an s-curved shape and saturates at low and high values, as shown in figure 4.4 a). This means that it becomes flat and insensitive to changes when the argument is far from zero in the positive or negative direction. The sigmoid function is a natural function to use in the output layer for classification, as its output between 0 and 1 may be interpreted as probabilities. If the output contains several components, the element-wise sigmoid function is used, defined by

$$\mathbf{a} = \sigma(\mathbf{b}) \iff \mathbf{a} = (\sigma(\mathbf{b}_1), \sigma(\mathbf{b}_2), ..., \sigma(\mathbf{b}_N))^T. \tag{4.12}$$

The sigmoid function may be used in the hidden layers as well. However, the calculation time of the exponential may make it an inefficient choice. Also, it is subject to what is known as the *vanishing gradient problem* which may arise in the calculation of the gradient when the number of layers is large [65]. Therefore, other alternatives are often used in the hidden layers.

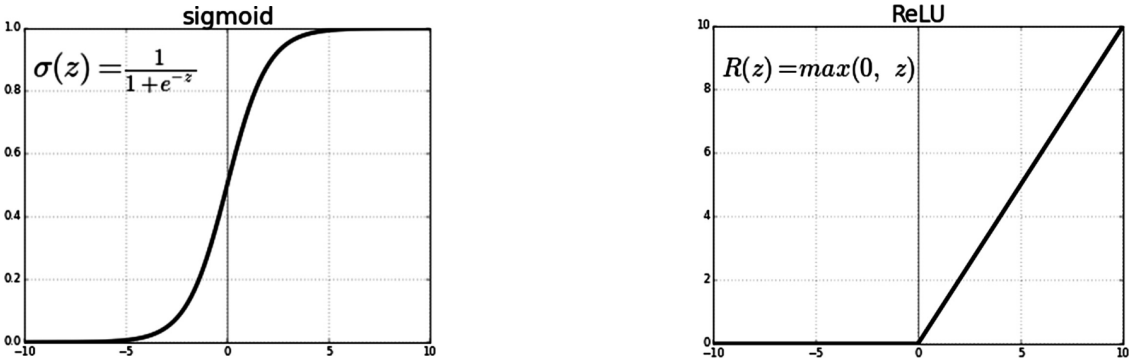
4.4.2 Rectified linear unit (ReLU)

The *rectified linear unit* (ReLU) function is defined by

$$ReLU(x) = \max(0, x). \tag{4.13}$$

The output is therefore zero for all negative input values and it returns the input for positive input values, as shown in figure 4.4 b). Outputs from nodes using ReLU are easy to optimize, as it is similar to the linear units, and solves some issues that arise from negative values. One of the main reasons it is used is that it solves the vanishing gradient problem. A drawback is that the network cannot learn

from gradient-based network on data points where the activation function is zero, as there will be no information flowing through the network. This does not mean that features can not be negative, as they are first multiplied by the weights. For multi-component output, an element-wise version of ReLU is used, similar to the element-wise sigmoid, defined in equation 4.12.



(a) (b)

Figure 4.4: Plots showing the general shapes of the a) sigmoid and b) ReLU activation functions. Figure taken from [67].

4.5 Backpropagation

In most optimization methods, it is necessary to calculate the gradient of the cost function with respect to the weights and the biases that were introduced in section 4.2. This may be done through a process known as *back-propagation*. The name comes from the fact that information in this case flows from the end of the network to the beginning.

We start by defining some quantities. The weights at each node are referred to as w^l_{jk}, which is the weight for the connection between the k^{th} neuron in layer $(l-1)$ and the j^{th} neuron in layer l. For the biases and activation functions, we use a similar approach, where b^j_l and a^j_l are the bias and activation function of the j^{th} neuron in layer l. The activation function a^j_l may then be written as

$$a^j_l = \sigma\Big(\sum_k w^l_{jk} a^k_{l-1} + b^j_l\Big) \tag{4.14}$$

which may be used in order to find the quantity

$$z^l = w^l a^{l-1} + b^l \tag{4.15}$$

which is called the weighted input to the neurons of layer l. As backpropagation measures how changing the weights and biases affect the cost function J (defined in equation 4.4), we define the error of neuron j in layer l as $\delta_j^l \equiv \frac{\partial J}{\partial z_j^l}$. The components of the error of the output layer L are then

$$\delta_j^L = \frac{\partial J}{\partial a_j^L}\sigma'(z_j^L) \tag{4.16}$$

where $\frac{\partial J}{\partial a_j^L}$ measures how fast the cost is changing as a function of the j^{th} activation output. $\sigma'(z_j^L)$ measures how fast the activation function changes at z_j^L. The expression may be rewritten in matrix form as

$$\delta^L = \nabla_a J \odot \sigma'(z^L) \tag{4.17}$$

where $\nabla_a J$ is the gradient with components $\frac{\partial J}{\partial a_j^L}$ and $\odot$ is the *Hadamard product*, which is the element-wise product of two matrices. If s and t are vectors, the components of their Hadamard product are then $(s \odot t)_j = s_j t_j$. It may be shown [66] that once δ^{l+1} is known, one may find δ^l by

$$\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l). \tag{4.18}$$

We will omit the proof of this expression, but it may be found in reference [66]. One may think of this expression as moving the error δ^{l+1} one layer backward through the network. As we have an expression for δ^L, the error for any layer in the network may be calculated. The rate of change of the cost in relation to the biases is given by

$$\frac{\partial J}{\partial b_j^l} = \delta_j^l. \tag{4.19}$$

The rate of change of the cost with respect to the weights follows from the chain rule, and is given by

$$\frac{\partial J}{\partial w_{jk}^l} = a_k^{l-1}\delta_j^l. \tag{4.20}$$

These equations are then combined in the following way in order to calculate the gradient: We have an input $\mathbf{x}$ that flows forward through the network using $z^l = w^l a^{l-1} + b^l$ and $a^l = \sigma(z^l)$ for $l = 2, 3, ...L$. Then, the output error δ^L may be computed by using equation 4.17. The backpropagation step consists of calculating δ^l for $l = L-1, L-2, ..., 2$ using equation 4.18. The gradient of the cost function is

then given by equation 4.20 and equation 4.19. In the next section, we will show how backpropagation is used in order to optimize the weights and biases of the neural network.

4.6 Optimization algorithms

The optimization algorithm is the method used in order to update the weights, thus giving the neural network as high predictive power as possible. It should not be confused with optimization of hyperparameters, network architecture etc. There are many different possibilities, although they often are variations of each other. We will now introduce some of the most common optimization algorithms.

4.6.1 Stochastic gradient descent

Stochastic gradient descent (SGD) [65] (and its variations) is likely the most used optimization algorithm for neural networks. The goal of the method is to optimize the weights in order to minimize the cost function. SGD uses a *hyperparameter* (a static parameter chosen before training the network) known as the *learning rate* ϵ. This affects how fast the network will learn, which means it determines how much the weights will be changed in each step. Each step t is called an *epoch*. There are drawbacks of having a too large or too small learning rate. A too large learning rate may prevent the network from converging, while a too small learning rate may prevent the network from learning efficiently. Therefore one usually needs to test different values manually to find the one that gives the best result.

The first step of the algorithm is to initialize the parameter $\boldsymbol{\theta_t}$ (at $t = 0$), representing the weights at step t. A batch of m data samples $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(m)}\}$ with targets $\mathbf{y}^{(i)}$ are then sent through the network, producing outputs $\hat{\mathbf{y}}^{(i)}$. The gradient

$$\hat{\mathbf{g}} = \frac{1}{m}\nabla_{\boldsymbol{\theta}}\Big(\sum_i L(\hat{\mathbf{y}}^{(i)}, \mathbf{y}^{(i)})\Big) \tag{4.21}$$

is calculated using backpropagation. The the next step is to update the weights to

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \epsilon\hat{\mathbf{g}}, \tag{4.22}$$

where $\hat{\mathbf{g}}$ is the gradient calculated in equation 4.21. These steps are repeated

until the network has finished training, either because it converges or because a specific number of training epochs is reached.

4.6.2 Adam

The *Adam optimizer* is an extension of SGD and was introduced in 2014 [68]. Its name is not an acronym, but derives from the phrase "adaptive moments". One of the possible advantages of this algorithm is its adaptive learning rate, which makes manual tuning of the global learning rate less important. Adam combines methods from other optimization algorithms called Momentum and RMSprop and is supposed to make use of the best features from these without inheriting their problems [68]. We will not go into the reasons for all the steps in the algorithm, but rather state how it is implemented.

Adam uses what is known as first and second moment variables $\mathbf{s}$ and $\mathbf{r}$ which are used when updating the weights. These have initial values of zero. As in SGD, we have a parameter $\boldsymbol{\theta}_t$ which represents the weights, as well as a learning rate ϵ. In addition, we have hyperparameters ρ_1 and ρ_2 which are decay rates for the first and second moments ($\mathbf{s}$ and $\mathbf{r}$), which need to be in the range of $[0, 1\rangle$. A small constant δ is also needed. This is used for numerical stabilization and is usually in the order of $\sim 10^{-8}$. The algorithm begins in the same way as SGD where we sample a batch of m data samples with targets $\mathbf{y}^{(i)}$ that produce outputs $\hat{\mathbf{y}}^{(i)}$. The gradient $\hat{\mathbf{g}}$ is then calculated using backpropagation as in equation 4.21. Then the first and second moment estimates $\mathbf{s}$ and $\mathbf{r}$ are updated by

$$\begin{aligned} \mathbf{s}_{t+1} &= \rho_1 \mathbf{s}_t + (1-\rho_1)\mathbf{g} \\ \mathbf{r}_{t+1} &= \rho_2 \mathbf{r}_t + (1-\rho_2)\mathbf{g} \odot \mathbf{g} \end{aligned} \tag{4.23}$$

these are then corrected with biases, so that

$$\begin{aligned} \hat{\mathbf{s}}_{t+1} &= \frac{\mathbf{s}}{1-\rho_1^t} \\ \hat{\mathbf{r}}_{t+1} &= \frac{\mathbf{r}}{1-\rho_2^t} \end{aligned} \tag{4.24}$$

where $\rho_1^t$and ρ_2^t denote ρ_1 and ρ_2 to the power of t. The update is then computed by

$$\Delta\boldsymbol{\theta} = -\epsilon \frac{\hat{\mathbf{s}}}{\hat{\mathbf{r}} + \delta} \tag{4.25}$$

and the weights are updated to

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \Delta\boldsymbol{\theta}_t \tag{4.26}$$

4.7 Regularization

As described in section 4.1.3, a common problem in ML is overfitting. This usually leads to a model that performs poorly when tested with new data, even though the main characteristics of the data are the same as in the training set. There are different strategies in ML that aim to reduce the amount of overfitting. These are called *regularization* methods, and may be defined as any modification made to the network in order to reduce the generalization error, defined in section 4.1.3. Ideally, these will do this without increasing the training error. However, in many cases it is a good idea to use regularization to reduce the generalization error even if it leads to a higher training error, as its real-world utility lies in its predictive power on new data. In the broader definition of regularization, it could mean implementing early stopping to avoid overfitting. However, specific methods for regularization that are built into the network have also been developed. These are usually based on regularizing estimators and often have the characteristic of trading increased bias for reduced *variance*, which is defined as

$$\sigma^2 = \sum_{i=1}^{N} \frac{(x_i - \bar{x})^2}{N-1}, \tag{4.27}$$

where x_i are the data samples, $\bar{x}$ is the mean of the samples and N is the number of samples. We will now introduce one of the most common methods known as L^2 *parameter regularization*. This method aims to make the weights stay close to the origin by adding a regularization term $\Omega(\boldsymbol{\theta}) = \|\mathbf{w}\|_2^2$ to the loss function, where $\|\mathbf{w}\|_2^2 = w_1^2 + w_2^2 + ... + w_n^2$ is the square of the euclidean norm $\|\mathbf{w}\|_2$. This means that weights with a large magnitude will lead to a significantly larger regularization term than those with a small magnitude. The term functions as a penalty, causing a larger perceived loss when the weights are far from the origin. In order to control how much the regularization influences the network, it is multiplied by a constant λ so that the regularized loss L_{L^2} becomes

$$L_{L^2} = L + \lambda\Omega(\boldsymbol{\theta}) \tag{4.28}$$

Now that the main theoretical aspects of neural networks have been described,

we will in the following section introduce a concept that is important to the interpretation of the output, called *feature importance*.

4.8 Feature importance

When features are chosen for the neural network, one often have an idea of which ones are likely to be most important for the predictions. However, when training the neural network, it does not automatically provide information about which connections it finds between each feature and the output, or between different variables in combination. Although it in principle would be possible to analyze the weights in the network, it is usually too many to discover useful information. However, it is still desirable to obtain some information about which features the network uses the most in order to predict the output, both because it may find new connections we were not aware of before, and also in order to check whether it has found spurious connections in some features that are not likely to have a significant connection to the target variable. Therefore, a way to find the feature importance of the different features is necessary. However, for neural networks it is not straightforward how this should be done, and the methods usually only provide partial information. However, they may tell you something about which ones have a large impact and if some features have no impact at all. The interpretation of feature importance that will be used in the ML analysis will be described in section 7.5.

The method that will be used in the analysis is called *permutation feature importance*. We start with a trained ML model $\hat{f}$ and a test set $\mathbf{x}$, as well as a cost function $J(y, \hat{y})$ where $\hat{y}$ is the output predicted by the model and y is the target. The test set is tested on the model which produces an original cost J_{orig}. In order to find the importance of a feature, we generate a new test set $\mathbf{x}_{perm}$ where the entries in one of the columns of $\mathbf{x}$ are permuted, resulting in the information from that feature (column) becoming noise. The model is then tested using $\mathbf{x}_{perm}$, which produces a new cost J_{perm}. If the feature is important for predictions in the model, it is likely that $J_{perm} > J_{orig}$ and if it is not important one expects $J_{perm} \approx J_{orig}$. The procedure is repeated for each feature. The permutation feature importance for feature j may therefore be defined as

$$FI_j = J_{perm} - J_{orig} \tag{4.29}$$

In some cases the feature importance may be negative, meaning that the cost

is lower when the feature is permuted. This may be a sign of overfitting for that feature as it implies that the model has found connections between the feature and the target that do not exist in the test set.

In this chapter, the theory of neural networks has been described, as well as how it may be used in order to separate signal and background events, using binary classification. We will now move on to discuss the data preparation for the analyses.

5 Data preparation[†]

In this chapter, we will explain the preparatory steps for the main analysis, which includes generating Monte-Carlo samples, event selection, plotting the signal distributions and comparison of Monte-Carlo samples and real data. In section 5.9, we will introduce the statistical tools that will later be used for performing the analysis in chapter 6 and 7. We will also make the choice of which features to use in the ML analysis.

5.1 Monte-Carlo simulations

5.1.1 Event generators

Monte-Carlo (MC) simulations are used in order to avoid performing complex calculations for every process. These are necessary for comparing the theory with real data. Different algorithms are combined into MC event generators. These usually take the model parameters as input and generate output of four-vectors of the momenta of final state particles. There are specialized theoretical groups that have written MC event generators, which are usually the ones used in analysis. MC simulations for both the signal and background processes that are needed for this analysis have already been generated by the ATLAS collaboration, and these will be used. We will now give a brief description of some of the most common MC event generators.

SHERPA [69] is a general-purpose event generator for particle collisions. It may be used to simulate all the SM processes that happen in hadron colliders, as well as many beyond SM processes. It has two built-in matrix element generators. It also has a parton shower model and a cluster hadronisation model. PYTHIA [70] is another general-purpose event generator. It is often used for generating events with several collisions, also known as pile-up events. MadGraph [71] is a matrix element generator, generating the Feynman diagram and calculating the matrix element for a process specified by the user. It must be combined with a different generator such as PYTHIA for simulation of the parton shower and hadronization. MC@NLO [72] computes partonic hard subprocesses by including the full next-to-leading-order (NLO) QCD corrections. MadGraph5_aMC@NLO [73] is a combina-

[†]Adapted from [1]

tion of MadGraph and MC@NLO. Another generator is the POWHEG-BOX [74], which uses a method called POWHEG [75] in order to perform NLO calculations.

5.1.2 Signal

The Monte-Carlo simulations for the signals use the parameters and framework outlined in section 3.2. The couplings to quarks and leptons are assumed to be the same, independent of generation. The LHC Dark Matter Working Group have provided recommendations for the coupling constants in vector models with small couplings to leptons, which apply to the models that are searched for here. These recommendations are followed in the simulations, which are couplings of $g_D = 1$, $g_q = 0.1$ and $g_l = 0.01$ [76]. As the coupling to leptons is small, the cross sections multiplied by the branching ratio of the signal processes is also low. As the masses of the dark Higgs and the dark sector fermions have been fixed to the values in table 3.1, the last free parameter is the mass of the Z' boson. The processes are simulated separately for electron and muon final states and range from 130 GeV to 1500 GeV. The two lowest simulated masses are 130 GeV and 200 GeV and then 100 GeV steps are made between each simulation up to 1500 GeV. The signal events are simulated with MadGraph using the NNPDF3.0LO [77] PDF set combined with PYTHIA 8 [70] for hadronization and the parton shower.

5.1.3 Background

As the signal processes produce two leptons and missing transverse energy (E_T^{miss}), all the background contributions included in section 3.3 must be simulated. The neutral current Drell-Yan/Z+jets process background are simulated with SHERPA 2.2.1 using the NNPDF3.0NNLO PDF set [77]. Backgrounds from top-antitop production ($t\bar{t}$) and single top quarks are generated with POWHEG-BOX, combined with PYTHIA 8 for the parton shower and hadronization, using the NNPDF3.0NLO PDF set [77]. Diboson production is simulated with SHERPA 2.2.1 for semi-leptonic final states and with SHERPA 2.2.2 for fully leptonic final states, using the NNPDF3.0NNLO PDF set. The W+jets background is simulated with SHERPA 2.2.1, using the NNPDF3.0NNLO PDF set. The generators used for each type of background are listed in table 5.1.

Process	Generator	PS + hadronization	Cross section
Drell-Yan	SHERPA 2.2.1	SHERPA 2.2.1	NNLO
Z+jets	SHERPA 2.2.1	SHERPA 2.2.1	NNLO
$t\bar{t}$	POWHEG-BOX v2	PYTHIA 8	NLO
Single top	POWHEG-BOX v2	PYTHIA 8	NLO
Diboson (semi-leptonic)	SHERPA 2.2.1	SHERPA 2.2.1	NNLO
Diboson (leptonic)	SHERPA 2.2.2	SHERPA 2.2.2	NNLO
W+jets	SHERPA 2.2.1	SHERPA 2.2.1	NNLO

Table 5.1: Event generators used for the different background types.

5.2 ROOT and data files

The data from the detector, as well as the MC simulated samples are reconstructed and made available as nTuples, put together by the ATLAS analysis groups in .root files. ROOT [78] is a c++ based object-oriented data analysis framework developed at CERN. The nTuples include the final state particles of the accepted events, as well as relevant variables from each accepted event. The nTuples in ROOT contains TTrees, and behave like an array of a data structure on storage. The trees may be divided into *branches*. A branch consists of values of any type that is known to ROOT's type system, such as vectors, floats and booleans. One may choose which branches to read when reading a tree. In a tree, each entry represents an event, while the branches represent variables. ROOT provides ways to read the data in the trees, as well as other functions specifically suited for physics analysis, such as built-in functions to calculate invariant mass once the (E, p_T, η, ϕ) vectors of an event are known. In addition, it provides the possibility of filling histograms with events and various ways of plotting them.

The data used in our case consists of both data from the detector, the MC background simulations, as well as MC simulations of the new physics signals. These come in separate files and are read individually. The data from the detector is separated into files for each year from 2015 to 2018. Each of these are divided into files of "minitrees" corresponding to different periods throughout each year, which must be read separately, but are combined afterwards. The MC samples are also, both for the signal and background, divided into sub-campaigns, although the 2015 and 2016 samples have been combined. These are divided into files corresponding to each type of process, which are then divided into a group of minitrees, which are the files that are read.

5.3 Event selection

The data files include many different types of events corresponding to all possible final states, most of which are not relevant for the analysis that is performed, and acts as extra background. In order to select the relevant events and organize them for further use, the *event selection algorithm* is used.

First, one needs to decide which trees are to be read. To link them together, a TChain (which is a ROOT object), can be created, which the trees are added to. This creates a chain of trees, which may be read as a single tree. Separate chains are usually created for the data from the detector and the MC samples. Also, it is possible to create separate chains for each year and each process in the MC samples, which is done in our case. Then, the variables one wants to read must be defined and linked to the corresponding branches of the tree. This may be done by using the TTreeReader function in ROOT, which is done in our analysis, or by using SetBranchAddress. Then histograms may be defined for every variable of interest, with the number of events on the y-axis, and separate histograms are defined for the electron and muon channels. Instead of doing this using ROOT, we will first fill new trees with the selected events which is a necessary preparation for the ML analysis. The histograms are later filled from these trees using Matplotlib [79]. As the final state of interest is that of two leptons the same flavor and opposite charge, we create a TLorentzVector, which is the ROOT version of a 4-vector, for each of them, as well as a dilepton vector for their sum.

The first step in the event selection is creating a loop over every event, with some data quality cuts to ensure the events are of high quality. The first main selection in our study is done in order to ensure that there are exactly two leptons of the same flavor (ee or $\mu\mu$). As these are features in the data, one skips the loop if the criteria are not met, and if it is met, the lepton's characteristics are added to each of the 4-vectors and kept in the new tree. The components added to the 4-vectors are $(E, p_T.\eta, \phi)$, as defined in section 2.3.2. When these are added together, variables such as the invariant mass may be calculated. Then it is possible to make cuts on other variables, such as m_{ll}, p_T and E_T^{miss}, which decide which events are to be kept in the analysis. The cuts used in our analysis will be described in section 6.1.

The real data may be added to the histograms after using the new trees to fill

them. However, the MC events need to be scaled before this is done. The MC sample files include a larger amount of events than the number of real events. This is done in order to make the results more realistic and reduce the statistical uncertainty by simulating more events in regions where few events are expected. Therefore, all events in the MC samples are associated with different kinds of weights. There are three different sets of weights. First, theoretical weights from the LO and NLO QCD/EW correction, generator weights etc. Then there are weights for scaling each event to its theoretical cross-section. In addition, we need weights (usually distributed around 1) to account for minor discrepancies observed in the acceptance and efficiencies of the various objects between data and simulations. We also have pile-up weights to correct the pile-up distribution in the simulations to be equal to the one in data, since the simulations were made before the data was taken and one had to guess on the pile-up distribution. Finally the events must be multiplied by the integrated luminosity for each period and divided by the sum of the weights corresponding to the type of process the events belongs to.

5.3.1 Preselection

Before analyzing the data sets, they must be cleaned as much as possible in order to get rid of unnecessary background, as discussed above. This is done by using the event selection algorithm before the main analysis begins. In the cut and count method one could also include these in the main cuts as the result would be the same. However, we do not have the ability to implement cuts directly in the neural networks. We also make cuts on small values of $m_{ll} > 70$ GeV in order to filter out regions where the agreement between the real data and MC simulations is not as good. On the final state leptons, we make cuts of $p_T > 30$ GeV, as well as $|\eta| < 2.5$. In addition, we make a cut on the momentum of jets of $p_{T,jets} > 20$ GeV. The precuts are shown in table 5.2.

5.4 Preparing data for ML environment

Once the event selection algorithm has selected the events of interest, they may be analyzed by ML methods. Although there exist some ML tools in ROOT, we are using PyTorch [80] in this thesis.

PyTorch is a machine learning library that is based on the Torch library which was originally developed by Meta AI [80]. It contains features for tensor computing

Variables	Precuts
Number of leptons	2
Flavour	Same flavour
Charge	Opposite charge
m_{ll}	> 70 GeV
p_T (leptons)	> 30 GeV
p_T (jets)	> 20 GeV
$\|\eta\|$ (leptons)	< 2.5

Table 5.2: Precuts used in the first event selection.

as well as for deep neural networks. It specifically contains features for automatic back-propagation, defined in section 4.5, as well as different standard optimization algorithms. This makes creating neural networks more efficient, as many of these features would be time-consuming to program and are similar in every neural network. PyTorch stores arrays of numbers in tensors, which are similar to NumPy [81] arrays in their structure.

It is not straightforward to convert a ROOT tree to a PyTorch tensor, as trees have different structure. Also, the trees contain a larger number of variables than those that are needed for the ML analysis. However, libraries have been developed that are specifically designed to convert trees to NumPy arrays with the possibility of selecting the necessary features. The NumPy arrays may then be converted to PyTorch tensors easily as their structures are similar.

5.5 Choice of features

Although one would prefer to put all the data into a neural network and let it attempt to find connections, this not the most efficient strategy and may also produce spurious results. For example, as the processes that are searched for produce exactly two leptons, it is not necessary to train the network on events with different final states, as these will act as noise. Also, it is not necessary to train on every feature, as many of the available features likely have no predictive power in relation to whether an event is a signal or background event. Although the neural network ideally ignores these features after training, the risk of it finding spurious results in these features is there. Every feature added also increases the time and computing power needed to train the network. In addition, for some variables the simulations do not resemble the real data as well as for other variables. This may lead the

neural network to perform well when tested on the simulated samples as well as the background data, but lack the ability to find the signal events in data if they exist. Because of this, carefulness is necessary when choosing the features. Only the features where the simulations resemble the real data should be chosen. Additionally, in order to find features with high predictive power, one may compare the signal distribution to background in order to find features with large discrepancies between the two.

Because of the reasons stated above, we choose to consider a limited number of features compared to the number of possibilities for the ML analysis. Some of these are known as *low-level features*, which are features that are measured directly or are not a combination of other features. *High-level features* are features that are found by combining other features, such as m_{ll}. The features include some that are expected to be important from fundamental reasons, which are m_{ll}, E_T^{miss}, $E_T^{miss,sig}$ and p_T. The invariant mass are important because they contain information about the mass of the Z'. The same applies to p_T, as a higher Z' mass in general results in a higher energies for the final state particles. E_T^{miss} and $E_T^{miss,sig}$ are important because the signal processes produce dark matter particles which are expected to result in an elevated amount of missing transverse energy. We will also use flavor and the charge of the first and second particle, as there may be differences in the electron and muon channel. The number of b-tagged jets is useful for filtering out top background, while m_T may be used in order to filter out W background. In addition, we will consider some other features where the neural network may find useful information. These are η, $\Delta\phi(l_1, l_2)$, $\Delta\phi(l, E_T^{miss})$ and H_T. We will keep these features even if their distributions do not show signs of usefulness, as ML methods analyze each event separately and may find connections that are not visible in the distributions as a whole. Because we have chosen a small number of such features, they are not as likely to cause overfitting. However, it is still necessary that the MC samples and data are in agreement for all features.

5.6 Signal model distributions

In order to get an idea of the main characteristics of the different signal models, we plot their distributions after making the preparatory cuts discussed in section 5.3.1. The results influence which features which will be useful for the ML analysis, as features with a similar shape to the background distribution usually are less useful.

They also influence the choice of cuts in the cut and count analysis. Although we will be using signal samples for many different values of Z' mass in the ML analysis, we will here only show some of them in order to have a general idea of their characteristics at different mass levels. The signal MC samples in each model are plotted separately for the electron and muon channel for m_{ll} and $E_T^{miss,sig}$ after making precuts, as shown in figure 5.1-5.4. Plots for other variables are shown in appendix A.

For m_{ll} in both models, a peak is observed at the Z' mass of the signal, followed by a sharp decrease at higher values of m_{ll}. However, in both models we observe a longer tail in values lower than the Z' mass. In the light vector model, the decrease is sharper before making a tail, while in the dark Higgs model for the light dark sector (LDS), there is a second peak before it decreases steadily. Because of this, the mass resonance is less distinct in this case than in the heavy sector and in the light vector models. For E_T^{miss}, the signals peak in different areas, providing the possibility of distinguishing between them if observed in the data, as well as the possibility of optimizing the cuts for a specific scenario. The peaks are more blunt in the muon channel than in the electron channel for m_{ll}, due to the electrons having higher resolution. This strengthens the reason to use flavor as a feature in the ML analysis. It should also be noted that the behaviour seen in the m_{ll} distribution is a characteristic of the models that are studied, as the Z' in figure 5.1-5.4 can also be virtual.

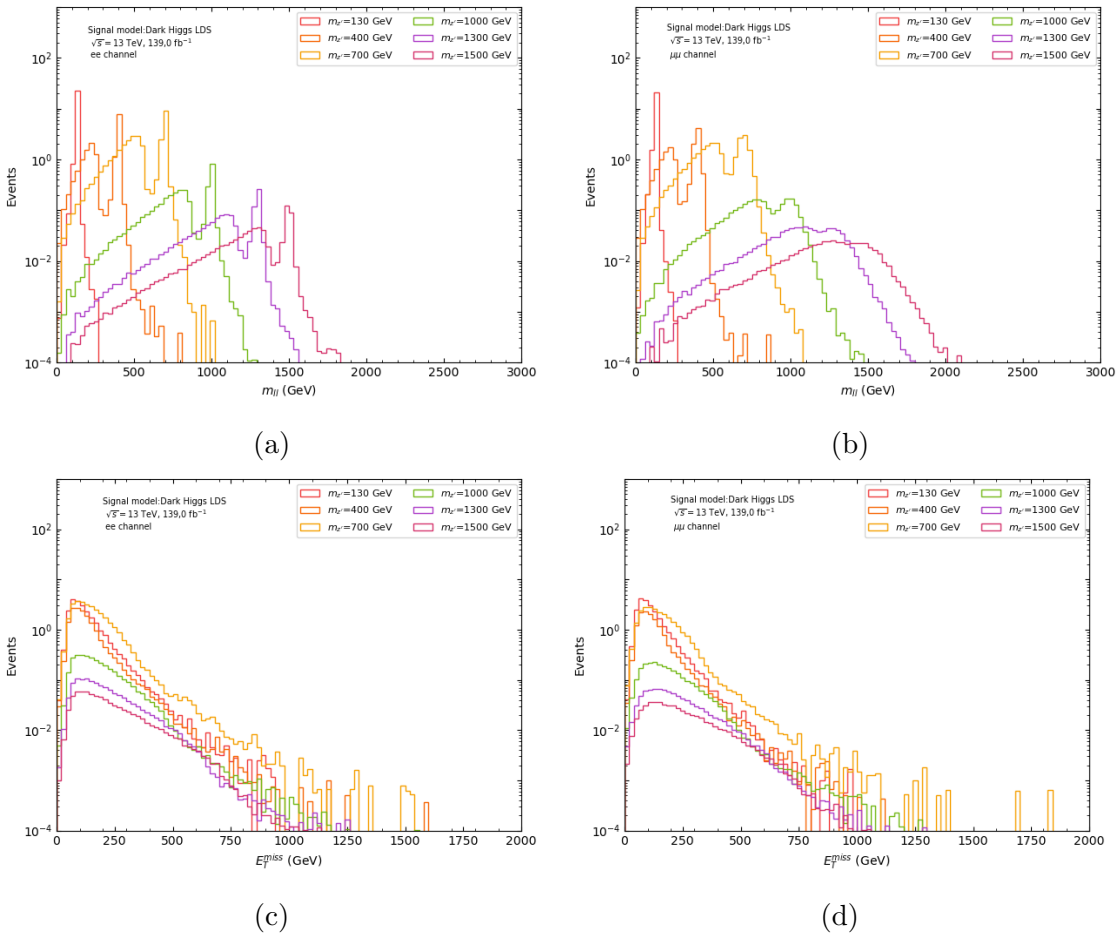


Figure 5.1: Electron (left) and muon (right) channel MC signal distributions of m_{ll} and E_T^{miss} in the dark Higgs LDS with precuts.

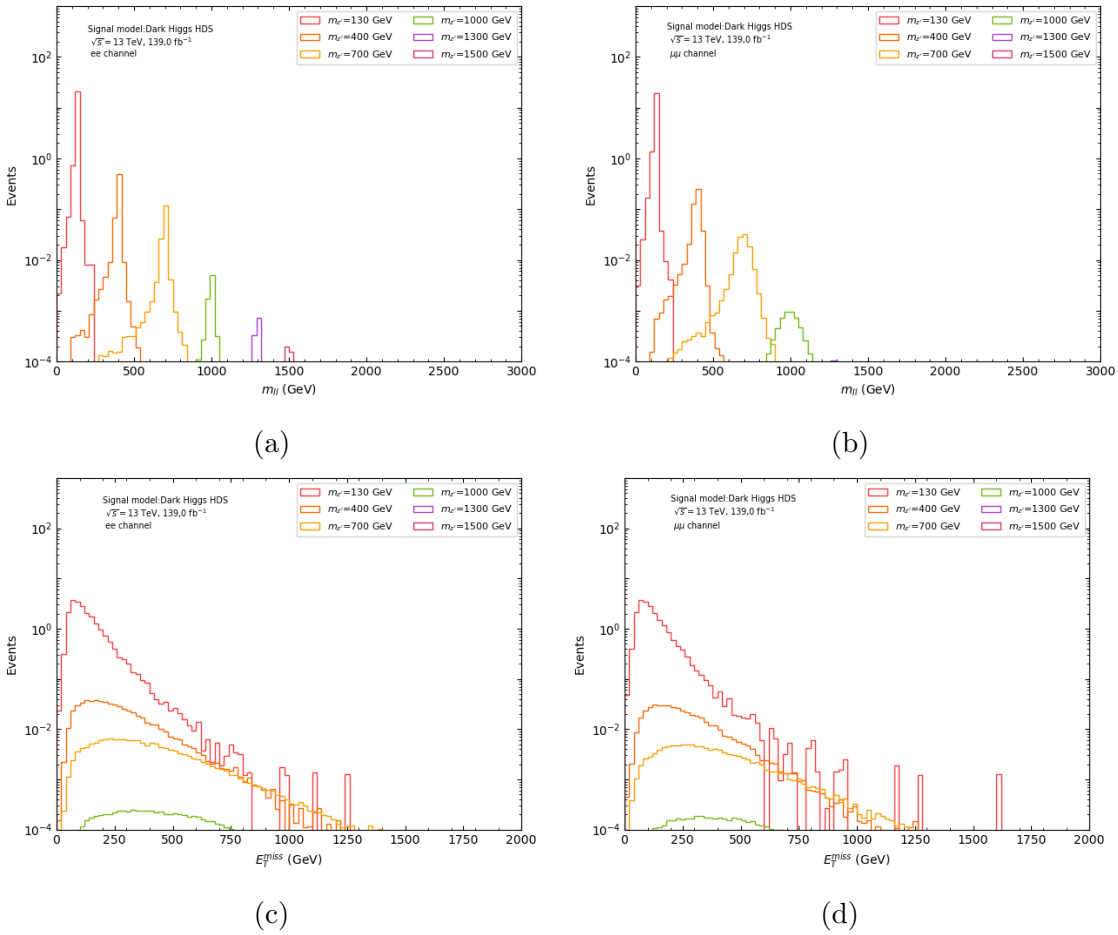


Figure 5.2: Electron (left) and muon (right) channel MC signal distributions of m_{ll} and E_T^{miss} in the dark Higgs HDS with precuts.

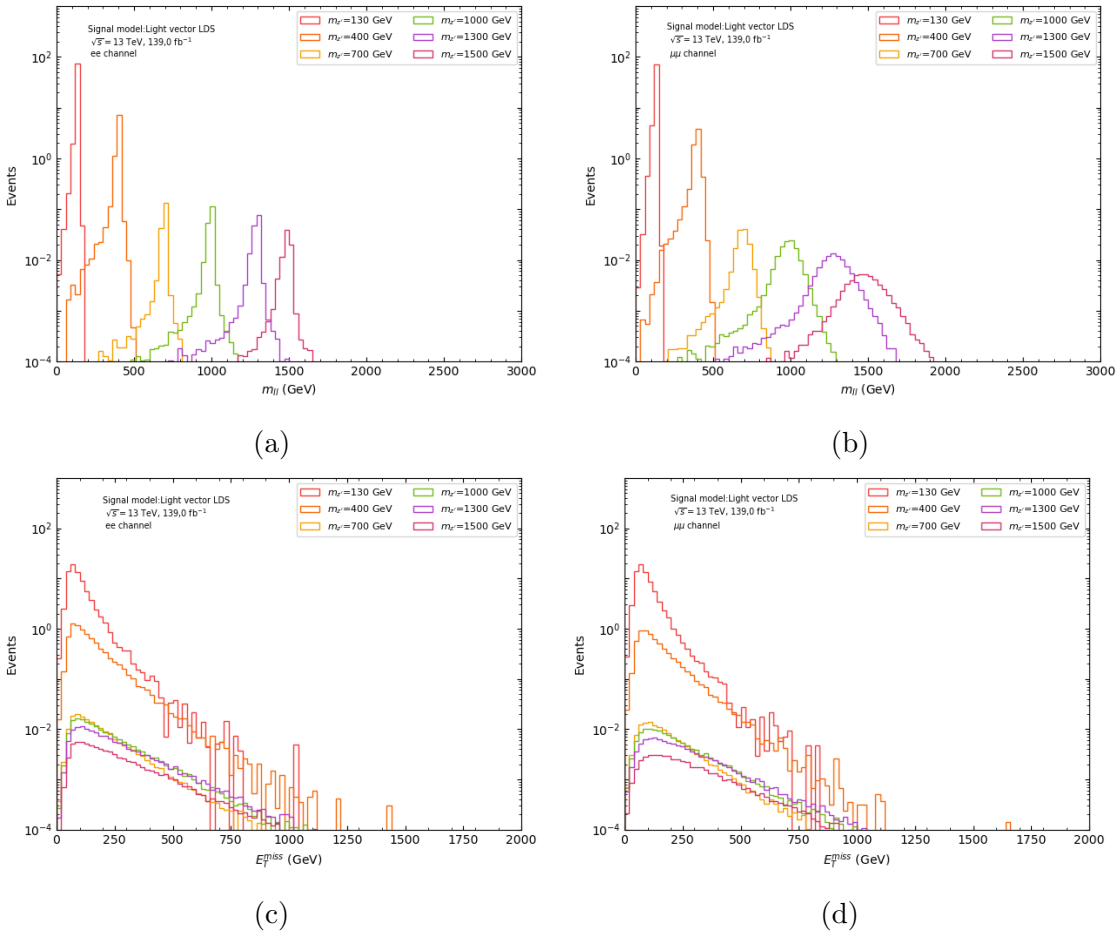


Figure 5.3: Electron (left) and muon (right) channel MC signal distributions of m_{ll} and E_T^{miss} in the light vector LDS with precuts.

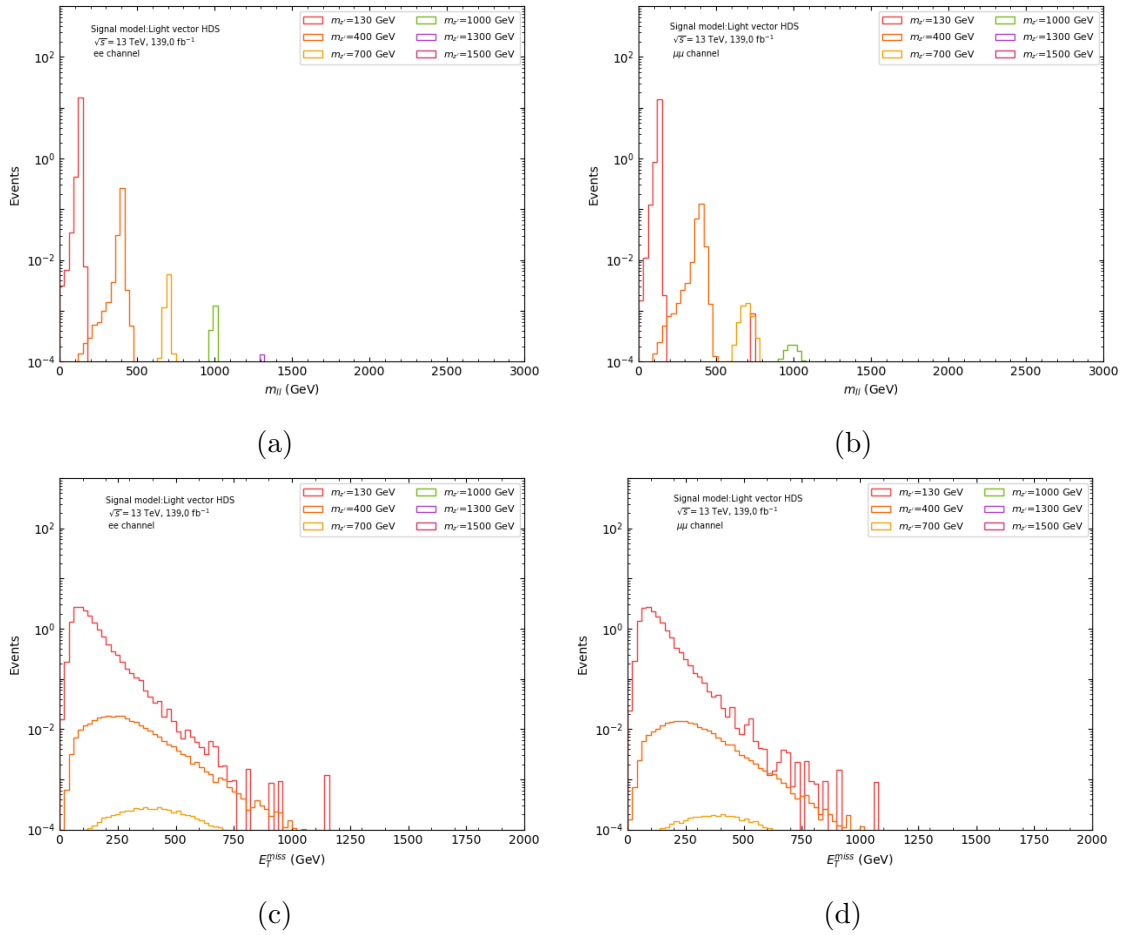


Figure 5.4: Electron (left) and muon (right) channel MC signal distributions of m_{ll} and E_T^{miss} in the light vector HDS with precuts.

5.7 Comparison of data and MC

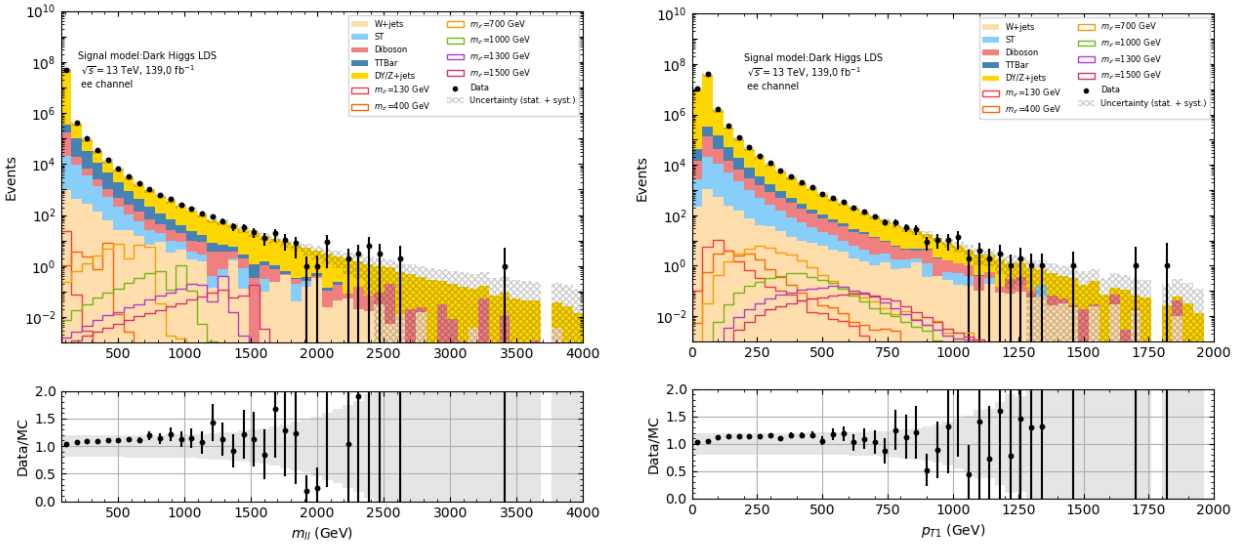
It is common to plot the background divided into the various processes and stacked on top of each other to represent the complete expected background from the SM. The real data and the signals are plotted individually. For the MC background, all histograms corresponding to a type of background are added together. These are divided into Drell-Yan (including Z+jets), single top, top quark pairs ($t\bar{t}$), diboson and W+jets. The sum of the SM backgrounds can be compared to data.

In order to determine whether there is any discrepancy between the data and the simulated background, and how large it is, we also plot the data divided by the MC background. This is ideally as close to 1.0 as possible, but some discrepancy is expected, especially in regions with few events or in areas where the agreement is known not to be perfect. Usually, adding systematic uncertainties cover for these effects. There may also be larger discrepancies in variables that are not well understood, which should be kept in mind when training machine learning models on the SM.

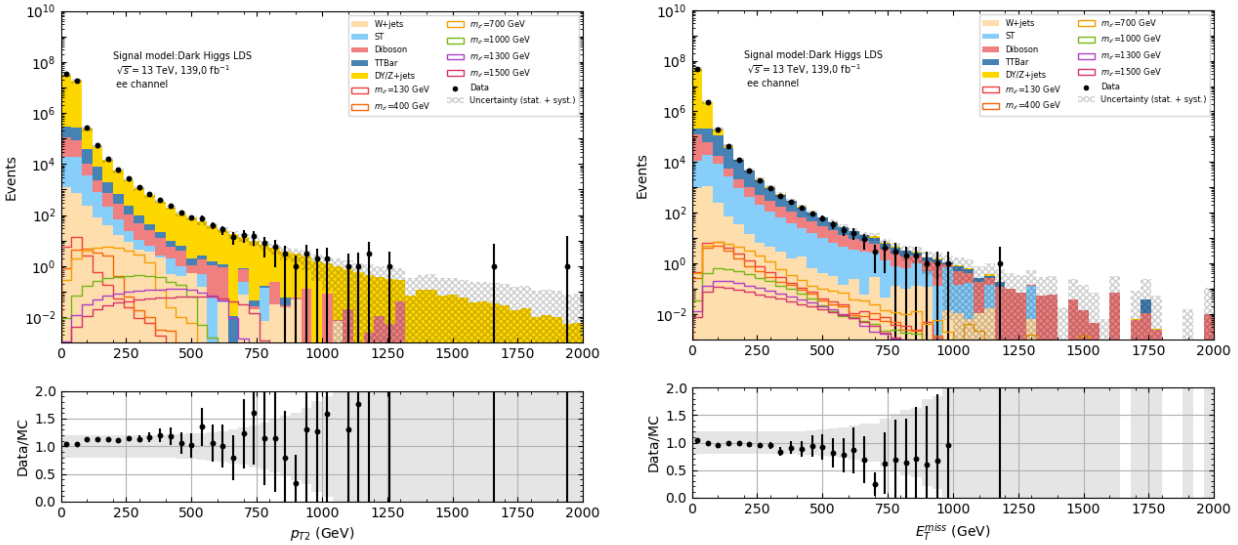
Plots of MC background alongside real data after precuts in the electron channel for the variables that are considered for the ML analysis are shown in figure 5.5, 5.6 and 5.7. Corresponding plots for the muon channel are shown in appendix B. These are plotted alongside the simulated contributions from the dark Higgs LDS simulations for reference. The ratio of the data to the total MC background is plotted below the histograms, where the grey bands show the sum of the statistical and assumed systematic uncertainties of 20%.

The MC samples are in reasonable agreement with the real data, suggesting that the data have been handled correctly and that the MC simulations has captured the main characteristics of the variables that are considered. Some large discrepancies are observed, but these occur in regions with few events, in some cases one event, which increases the uncertainty and statistical errors of the data points. There is also a relatively large discrepancy at small values (< 0.5) of $\Delta\phi_{l,l}$, which should be kept in mind if using it specifically. Also, as almost all regions are included, there is a large amount of background compared to the simulated signal. The background peaks between 10^7 and 10^8 events per bin for m_{ll}, with large regions containing more than 10^3 events per bin. The highest peak of the signal models shown is between

10 and 10^2 events per bin. Because of this, it should not be possible to be sensitive to the signals predicted by the models considered without a significant reduction of the SM background. Cuts on some of the variables defined earlier or a reduction of background from ML classification are therefore necessary for improving the sensitivity.

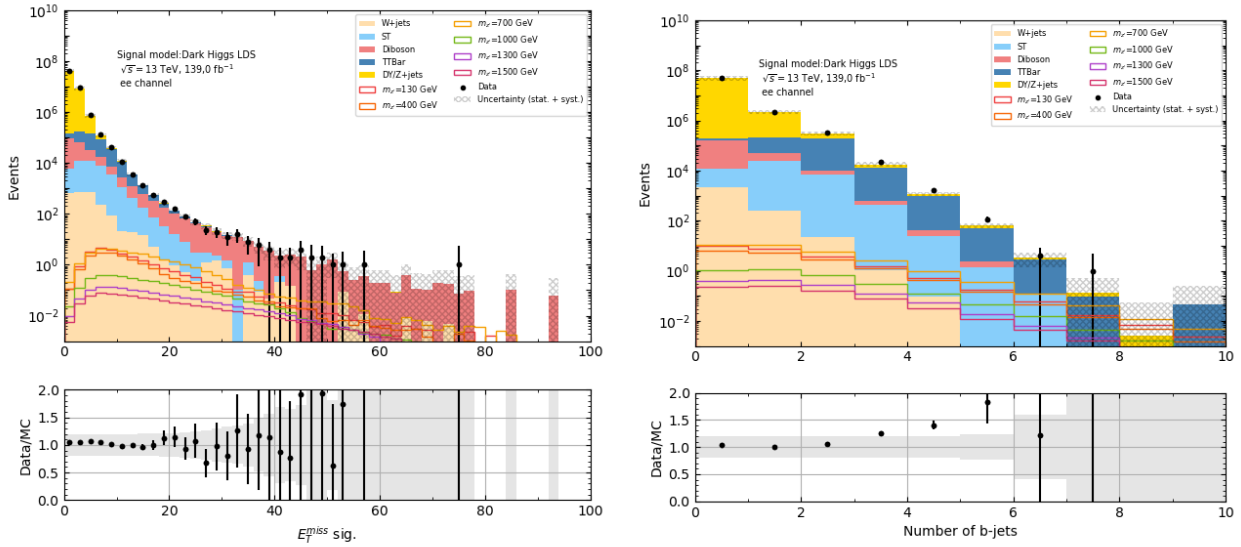


(a) (b)

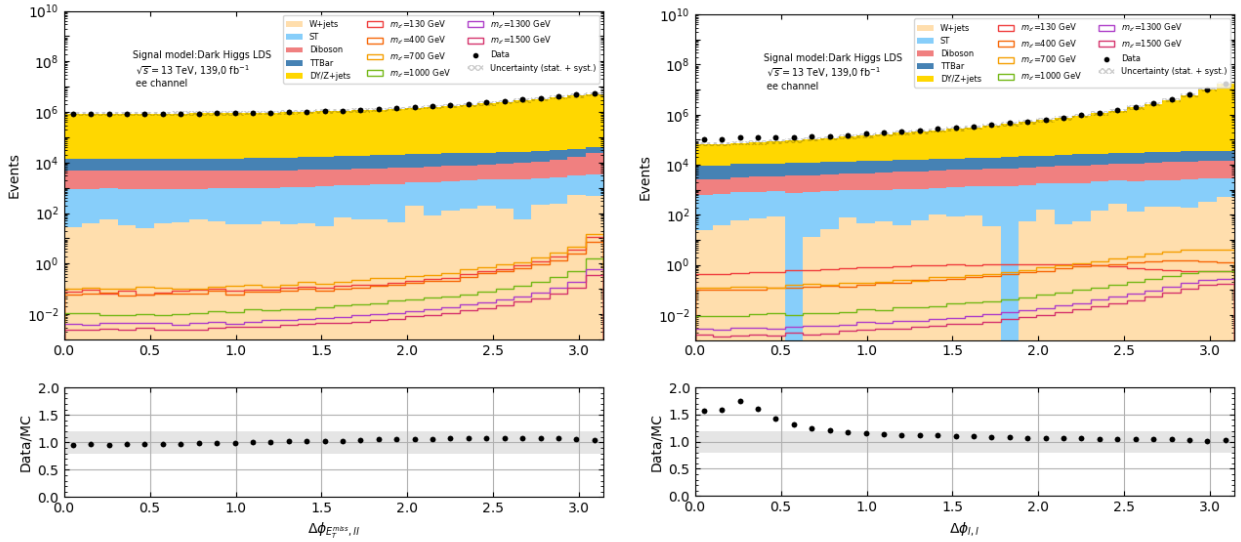


(c) (d)

Figure 5.5: Electron channel distributions for a) m_{ll}, b) p_{T1}, c) p_{T2} and d) E_T^{miss} with precuts. Data is shown along with MC background and MC signals in the dark Higgs model (LDS).

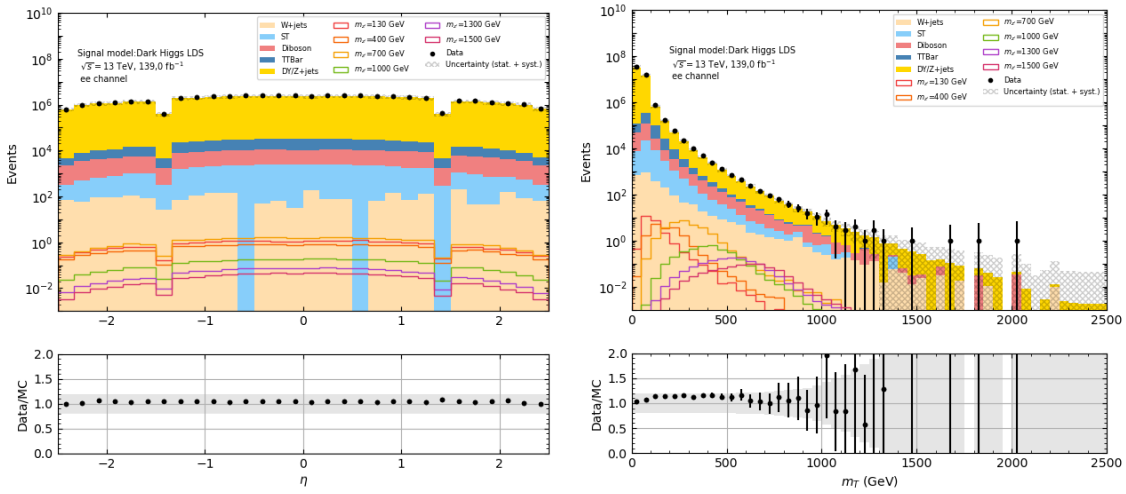


(a) (b)

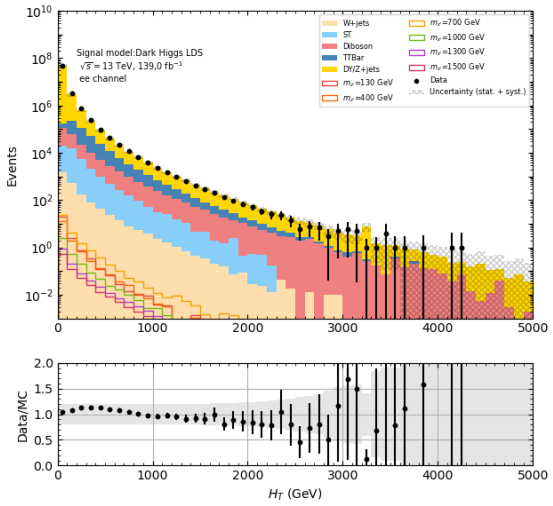


(c) (d)

Figure 5.6: Electron channel distributions for a) $E_T^{miss,sig}$, b) number of b-tagged jets, c) $\Delta\phi_{E_T^{miss},ll}$ and d) $\Delta\phi_{l,l}$ with precuts. Data is shown along with MC background and MC signals in the dark Higgs model (LDS).



(a) (b)



(c)

Figure 5.7: Electron channel distributions for a) η, b) m_T c) H_T with precuts. Data is shown along with MC background and MC signals in the dark Higgs model (LDS).

5.8 Systematic uncertainties

There are statistical uncertainties in both the MC data and the real data due to the limited number of collisions occurring and the fact that the resulting final states in each case are subject to a high degree of randomness. In regions with a large number of events, the statistical uncertainties will be small, and large in areas with few events occurring. However, there are also sources of systematic uncertainties

both in the theory and MC simulations. Estimating these to a high precision is beyond the scope of this thesis, and a constant value of systematic uncertainty of 20% will be assumed instead. Anyhow, we will here present some of the systematic uncertainties which are believed to dominate in this analysis.

5.8.1 Theoretical uncertainties

There are uncertainties in the MC simulations of the background due to the choice of the parton distribution function (PDF) of the colliding protons, in addition to the choice of parameters with theoretical uncertainties. These include parameters such as the coupling constants and the cross sections for different processes. The cross-sections are affected by which order is used in perturbation theory as well as the renormalization and factorisation scales used. In addition, there are uncertainties due to the choice of event generator, as there will be some discrepancy in the results depending on which one is used.

5.8.2 Experimental uncertainties

The experimental uncertainties arise due to possible imperfections in the resolution of different parts of the apparatus and in the handling of data. These come from a variety of different sources. Some of these are the integrated luminosity, pile-up re-weighting and trigger-related uncertainties. There are also uncertainties in the reconstruction, isolation and identification efficiency of different particles.

5.9 Statistical analysis method

The goal of the analysis is to find out whether we may expect to be able to detect the signal in the signal region if it exists or to exclude it if it is not there. The goal is also to see whether the cut and count or ML method leads to the highest sensitivity. For the first part, we need a method in order to know whether a model may be expected to be confirmed or excluded when analysing real data, in this case for different values of $m_{Z'}$. The sensitivity in particle physics is usually measured using *significance* or *p-values* for the signal regions.

When searching for a signal, two different hypotheses are tested. These are known as the *background only* (b) *hypothesis* and the *background+signal* ($s+b$) *hypothesis*. The b hypothesis acts as a null hypothesis and corresponds to the SM as it is currently understood, with no additional signal. The $s+b$ hypothesis corresponds to the SM with the additional signal that is searched for. In this case, a larger number of events are expected in the signal region compared to the number predicted by the SM. However, there may be an excess of events in the signal region even if the b hypothesis is correct due to randomness and systematic uncertainties. The extra number of events must therefore be large enough in order to be reasonably sure that it did not happen by chance. It is therefore common to use p-values to determine how likely it is that a number of events of a certain magnitude or larger is observed, given that the b hypothesis is true. In particle physics, the probability of observing n events if the predicted number of events is ν is given by the *Poisson distribution* [82]

$$P(n|\nu) = \frac{\nu}{n!}e^{-\nu} \tag{5.1}$$

As the p-value is the probability of measuring the observed number of events n_{obs} or higher the p-value is given by

$$p = P(n > n_{obs}|\nu) = \sum_{n=n_{obs}}^{\infty} \frac{\nu}{n!}e^{-\nu} \tag{5.2}$$

In particle physics, the p-value is usually converted into a *significance* Z. Z may be found from the p-value by [83]

$$Z = \Phi^{-1}(1-p) \tag{5.3}$$

where Φ^{-1} is the inverse of the cumulative distribution of the standard Gaussian. Z corresponds to the number of standard deviations, σ, that a Gaussian variable is observed from its mean, which has an upper-tail probability p. The norm for claiming a new discovery is an observed significance of 5σ, corresponding to a p-value of $\sim 3 \cdot 10^{-7}$, meaning that the chance of the observation occurring given the b hypothesis is approximately 1 in 3.5 million. In our case we will not use the real data to measure the significance, but instead use the MC simulations. In this case, the median number of extra events produced by the simulations lead to an expected significance under the $s + b$ hypothesis.

The goal of a search is often not only to attempt to make a discovery, but to exclude the theory fully or in specific regions. In this case, the p-values are used as *confidence levels*. This measures the probability, given the $s + b$ hypothesis, that the number of observed events is equal to or less than the number observed. The confidence level for the $s + b$ hypothesis is

$$CL_{s+b} = P(n < n_{obs}|s + b) = \sum_{n=0}^{n_{obs}} \frac{(s + b)^n}{n!} e^{-(s+b)} \tag{5.4}$$

where s is the number of signal events and b is the number of background events. This may be converted to an expected significance Z_N by

$$Z_N = \Phi^{-1}(1 - CL_{s+b}) \tag{5.5}$$

It is common to exclude the signal model if $CL_{s+b} \leq 0.05$, and the probability to falsely exclude an existing signal(+background) is then 5%, corresponding to a significance $Z_N \leq 1.64\sigma$. However, if the signal model produces a very small amount of events, it may not be possible to exclude the model, and using the CL_{s+b} method may be inappropriate, as the b and $s + b$ hypotheses are almost identical. Then one may use the CL_s method [84] instead.

5.9.1 Calculating significance in a counting experiment

In a particle physics context, it is desirable to obtain a formula for the expected significance, given a number of expected signal events and background events, possibly including the uncertainty σ_b. A Poisson distribution with mean $s + b$ may be approximated by a Gaussian variable with mean $s + b$ and standard deviation

$\sqrt{s+b}$, where s and b here represent the number of signal and background events, respectively. Using the likelihood function for a Poisson counting experiment, it is possible to obtain the so-called *Asimov approximation formula* [85]

$$Z_A = \sqrt{2\Big((s+b)\ln\Big(1+\frac{s}{b}\Big)-s\Big)} \tag{5.6}$$

Expanding the logarithm in s/b, one finds

$$Z_A = \frac{s}{\sqrt{b}}(1+\mathcal{O}(s/b)) \tag{5.7}$$

The first formula is more precise and is recommended, but the $\frac{s}{\sqrt{b}}$ approximation gives good results when $s \ll b$. In the analyses in chapter 6 and 7, we will use equation 5.6 to calculate the expected significance. It is also possible to generalize equation 5.6 to include the background uncertainty σ_b. The formula then becomes

$$Z_A = \left[2\left((s+b)\ln\left[\frac{(s+b)(b+\sigma_b^2)}{b^2+(s+b)\sigma_b^2}\right]-\frac{b^2}{\sigma_b^2}\ln\left[1+\frac{\sigma_b^2 s}{b(b+\sigma_b^2))}\right]\right)\right]^{1/2}. \tag{5.8}$$

In this chapter, we explained the different data preparation stages. These included Monte-Carlo simulations of the signal and background events, event selection, as well as verifying that there is agreement between the simulated and real data. We also discussed the systematic uncertainties, and introduced the statistical methods that will be used in the analyses in chapter 6 and 7. We are now ready for the main analyses, and we will start with the cut and count analysis.

6 Cut and count analysis

As the MC samples and the real data are in good agreement (except for some discrepancies commented on in section 5.7), as shown in section 5.7, the MC samples may be used for modelling the background in the analysis. A cut and count analysis is performed in order to later compare its sensitivity with the ML analysis. It should be noted that it may be possible to further improve the sensitivity using more sophisticated methods, such as significance scans for the cuts on each variable, as well as making a larger amount of signal regions with cuts specifically chosen for each variable. However, such improvements are likely marginal and unlikely to affect the main results. Also, in the ML analysis the networks are not trained specifically for each Z' mass, which makes the analysis method similar.

In this chapter we will begin by explaining the considerations made when choosing the cuts. Signal regions are then constructed by applying the cuts that are chosen. The effectiveness of the cuts are discussed and the expected significances are measured in order to find out whether it may be possible to discover or exclude the signals with real data using the signal region that is constructed.

6.1 Cuts

The cuts are made with the intent of removing as many background events as possible while keeping as many signal events as possible. Considering the distributions shown in the previous sections, there are several kinematic variables with large differences between the signal and background distributions. These are m_{ll}, E_T^{miss}, $E_T^{miss,sig}$, p_{T1}, p_{T2} and to some extent $\Delta\phi_{E_T^{miss}}$, $\Delta\phi_{l,l}$ and the number of b-jets. For η, the signal and background distributions are similar. It is also necessary to keep in mind that some of the variables are correlated, meaning that if cuts are made on both of them, it is likely that a large amount of the same events are removed in both cuts.

A cut is made on m_{ll}, as the distributions in the signal regions are relatively narrow around the Z' mass, allowing the possibility of searching for resonances. We want a cut that includes most of the Z' masses in the signal models while removing a large amount of the background. We also want to filter out the Z mass, as it is a source of irreducible background. Therefore, a cut that also filters out the model with Z' mass of 130 GeV is necessary in order to be reasonably far away from the

Variable	Cut
m_{ll}	> 180 GeV
$E_T^{miss,sig}$	> 8
Number of b-jets	0

Table 6.1: Cuts used in the cut and count analysis. In combination with the precuts, these define the signal region.

Z peak. In order to keep the rest of the signals, a cut of $m_{ll} > 180$ GeV is made. This cut filters out the m_{ll} peak of the background.

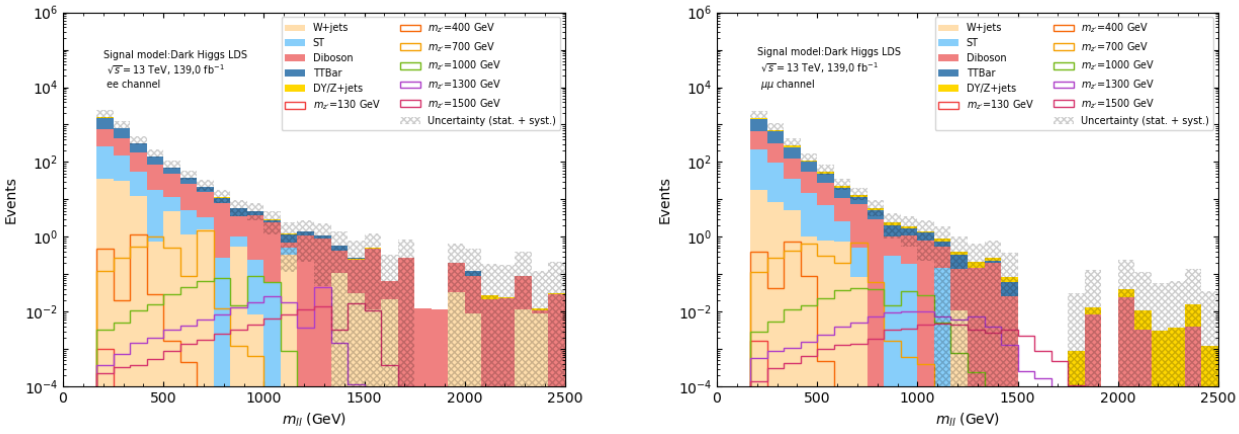
In addition, we want to make use of E_T^{miss}, as an elevated amount of missing transverse energy is expected from the production of dark matter. However, it is also possible to use $E_T^{miss,sig}$. This is expected to be a more effective variable to cut on, as a larger decrease in the signal models are observed in the distributions when approaching zero, while the background peaks when approaching zero, as shown in figure 5.6 and B.2. Most of the signal models peak between $E_T^{miss,sig} = 8$ and $E_T^{miss,sig} = 12$, and a large amount is found from $E_T^{miss,sig} = 5$ and above. However, a cut at 5 will not filter out the same amount of background. Therefore, a cut of $E_T^{miss,sig} > 8$ is made. Another possibility would be to split this variable into several signal regions depending on where each signal is largest. However, as explained earlier, we will only make one signal region in this case, which is expected to be effective for most of the signals. In addition, a cut is made on the number of b-tagged jets by setting it equal to zero. Although the distributions for the signals and backgrounds are relatively similar for this variable, it is an effective way to filter out the top background. The cuts are summarized in table 6.1.

6.2 Dark Higgs model

The results for the signal region in the LDS are shown for m_{ll} in figure 6.1 and for $E_T^{miss,sig}$ in figure 6.2. The corresponding plots for the HDS are shown in appendix D. There is a significant reduction in background events. The number of background events before making cuts peaked at $\sim 10^8$ events per bin for m_{ll} while peaking below 10^4 events per bin in the signal region. There is also a reduction in signal events, as some signal events will fall outside the signal region. However, the signal reduction is significantly smaller than the background reduction in relation to its original amount. The same is observed in the $E_T^{miss,sig}$ plots. Therefore, the

cuts have succeeded in improving the sensitivity of the expected signal. However, as seen in the signal plots before the cuts, the amount of expected signal is very small, where the highest peaks reach only a few events for the Run 2. This makes it unlikely that it is possible to discover these models in the data at the current stage.

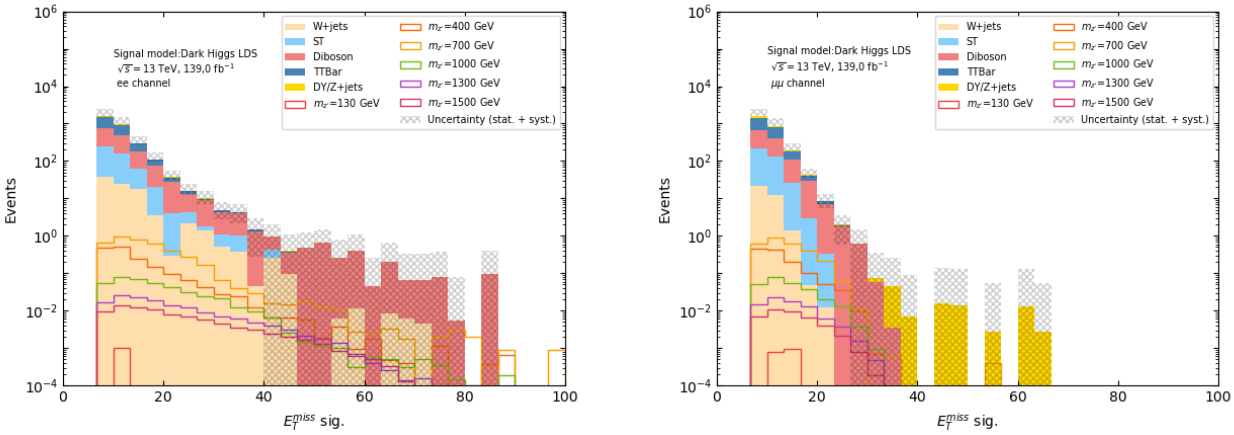
Almost all of the Drell-Yan/Z+jets backgrounds are removed, which is likely mostly due to the $E_T^{miss,sig}$ and m_{ll} cuts. The Z' resonances are still seen in the m_{ll} distribution, which opens for the possibility of searching for such resonances in the data at a later stage. However, the resonance would likely not be exactly the same as any of the simulated signals as the $m_{Z'}$ mass is a free parameter in the model. Also, we observe that the $m_{z'} = 130$ GeV signal is almost completely filtered out due to the m_{ll} cut. To search for this signal, a wider m_{ll} range in the signal region is necessary.



(a) (b)

Figure 6.1: Dark Higgs LDS signal regions for m_{ll} in the a) ee and b) $\mu\mu$ channel.

The expected significances are listed in table 6.2 for the ee and $\mu\mu$ channels. As expected, the significance for $m_{Z'} = 130$ GeV is the smallest, at the order of 10^{-5}. The highest is achieved between $m_{Z'} = 200$ GeV and $m_{Z'} = 700$ GeV in a range between $1.7 \cdot 10^{-2}$ and $7.6 \cdot 10^{-2}$, before gradually decreasing for increasing Z' masses. The expected significance is generally somewhat lower in the $\mu\mu$ channel than in the ee channel, possibly partly due to the signal model having somewhat sharper peaks in the ee channel for m_{ll} (due to higher resolution). As the expected significances are very small, we do not expect to be sensitive enough to exclude the



(a) (b)

Figure 6.2: Dark Higgs LDS signal regions for $E_T^{miss,sig}$ in the a) ee and b) $\mu\mu$ channel.

signal by analysing real data. This is mainly due to the low cross section of the signals, which is a result of the choice of the coupling to leptons, g_l. However, it may be possible to obtain a more sensitive signal region. We will later compare the expected significances with those found using the ML method to find out whether it leads to a higher sensitivity.

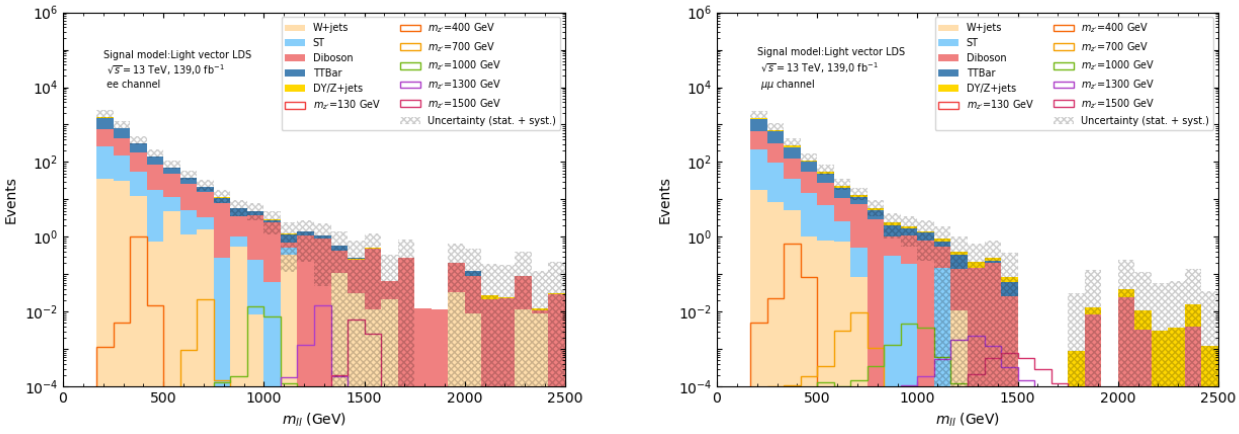
Dark Higgs LDS		
	ee channel	$\mu\mu$ channel
$m_{Z'}$ (GeV)	Expected significance (Z)	
130	$1.9 \cdot 10^{-5}$	$3.3 \cdot 10^{-5}$
200	$5.3 \cdot 10^{-2}$	$4.7 \cdot 10^{-2}$
300	$4.2 \cdot 10^{-2}$	$3.6 \cdot 10^{-2}$
400	$3.1 \cdot 10^{-2}$	$2.5 \cdot 10^{-2}$
500	$3.2 \cdot 10^{-2}$	$2.6 \cdot 10^{-2}$
600	$2.2 \cdot 10^{-2}$	$1.7 \cdot 10^{-2}$
700	$7.6 \cdot 10^{-2}$	$5.4 \cdot 10^{-2}$
800	$1.2 \cdot 10^{-2}$	$8.6 \cdot 10^{-3}$
900	$8.9 \cdot 10^{-3}$	$6.0 \cdot 10^{-3}$
1000	$7.8 \cdot 10^{-3}$	$5.1 \cdot 10^{-3}$
1100	$4.8 \cdot 10^{-3}$	$3.1 \cdot 10^{-3}$
1200	$3.8 \cdot 10^{-3}$	$2.2 \cdot 10^{-3}$
1300	$2.8 \cdot 10^{-3}$	$1.6 \cdot 10^{-3}$
1400	$2.1 \cdot 10^{-3}$	$1.1 \cdot 10^{-3}$
1500	$1.7 \cdot 10^{-4}$	$7.9 \cdot 10^{-4}$

Table 6.2: Expected significances for the dark Higgs LDS at different Z' masses in the ee and $\mu\mu$ channel using the cut and count method.

6.3 Light vector model

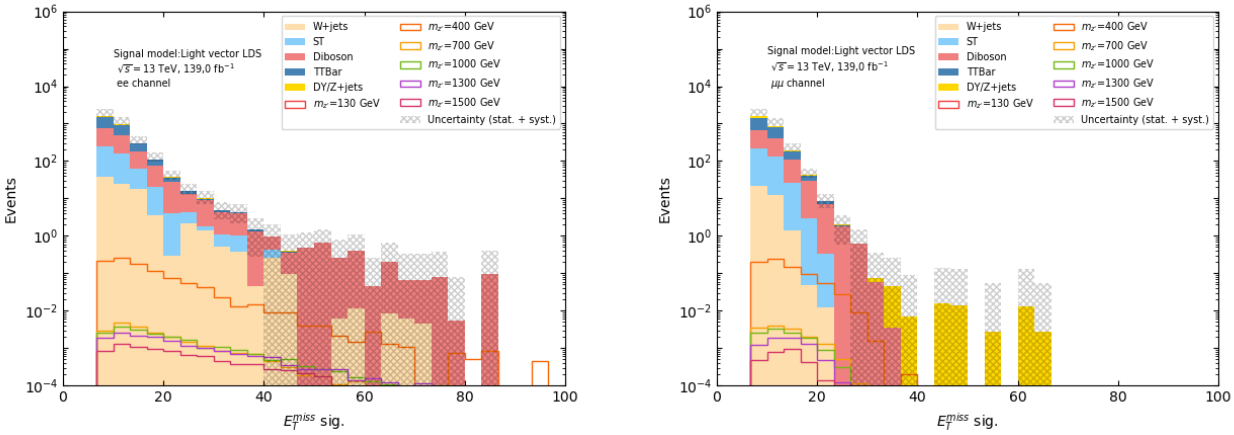
The results for the signal region in the light vector LDS are shown for m_{ll} in figure 6.3 and for $E_T^{miss,sig}$ in figure 6.4. The corresponding plots for the HDS are shown in appendix E. The reduction in background is the same as for the dark Higgs model as the cuts are the same. However, the reduction of signal is different, depending on how effective the cuts are for each signal. Most signal peaks are reduced by a factor of ~ 10, which is significantly less than the reduction in background, where peaks are reduced by a factor of $\sim 10^4$. This suggests that the cuts have been effective.

Similar patterns are observed for the expected significance as in the dark Higgs model. The expected significance at $m_{Z'} = 130$ GeV is 0, as it is completely filtered out by the m_{ll} cut. The highest expected significance occur at $m_{Z'} = 200$ GeV in the ee channel at $Z = 8.9 \cdot 10^{-2}$, as shown in table 6.3. It then generally decreases with increasing Z' mass until the lowest value of $Z = 5.6 \cdot 10^{-5}$ in the $\mu\mu$ channel $m_{Z'} = 1500$ GeV. The expected significance is somewhat lower in the $\mu\mu$ channel than in the ee channel for most of the signals. For most signals, the expected significance is lower than in the dark Higgs model. This may be partly due to the effectiveness of the cuts, but most is due to a lower number of total expected events because pf the low cross section. Similarly to the result for the dark Higgs model, the expected significances are not large enough to be sensitive to the signal by analysing real data in this signal region.



(a) (b)

Figure 6.3: Light vector LDS signal regions for m_{ll} in the a) ee and b) $\mu\mu$ channel.



(a) (b)

Figure 6.4: Light vector LDS signal regions for $E_T^{miss,sig}$ in the a) ee and b) $\mu\mu$ channel.

In this chapter we performed a standard cut and count analysis by making cuts on specific variables in order to construct signal regions. The sensitivity to the signal was measured by calculating the expected significances. We will now move on to the machine learning based analysis, in order to compare its sensitivity to the cut and count approach.

Light vector LDS		
$m_{Z'}$ (GeV)	Expected significance (Z)	
	ee channel	$\mu\mu$ channel
130	0	0
200	$8.9 \cdot 10^{-2}$	$8.1 \cdot 10^{-2}$
300	$2.4 \cdot 10^{-2}$	$7.9 \cdot 10^{-2}$
400	$1.9 \cdot 10^{-2}$	$2.0 \cdot 10^{-2}$
500	$5.2 \cdot 10^{-3}$	$3.9 \cdot 10^{-3}$
600	$5.9 \cdot 10^{-3}$	$3.9 \cdot 10^{-3}$
700	$4.2 \cdot 10^{-4}$	$2.8 \cdot 10^{-4}$
800	$2.1 \cdot 10^{-3}$	$1.4 \cdot 10^{-3}$
900	$1.4 \cdot 10^{-3}$	$8.4 \cdot 10^{-4}$
1000	$4.1 \cdot 10^{-4}$	$2.2 \cdot 10^{-4}$
1100	$6.2 \cdot 10^{-4}$	$3.3 \cdot 10^{-4}$
1200	$4.3 \cdot 10^{-4}$	$2.1 \cdot 10^{-4}$
1300	$3.2 \cdot 10^{-4}$	$1.4 \cdot 10^{-4}$
1400	$2.2 \cdot 10^{-4}$	$8.6 \cdot 10^{-5}$
1500	$1.6 \cdot 10^{-4}$	$5.6 \cdot 10^{-5}$

Table 6.3: Expected significances for the light vector LDS at different Z' masses in the ee and $\mu\mu$ channel using the cut and count method.

7 Machine learning analysis

As was found in chapter 6, the cut and count approach was successful in removing a significant amount of background and increasing the sensitivity to the signals. However, it was far from sensitive enough to reach a level where the signal may be discovered or excluded. In this chapter, we will use an ML-based approach to find out whether it can be used to increase the sensitivity further. We will begin by explaining some of the methodical aspects of the analysis in section 7.1-7.3. Then, we explain the optimization process in section 7.4. The training and performance of the neural networks are then discussed for the dark Higgs and light vector models, before measuring the expected significances they achieve. The ML-based approach is compared with the cut and count method in section 7.8. For theoretical parts of this chapter, we use references [65, 66].

7.1 Method

In order to be able to compare the ML results with the cut and count results, separate analyses are performed for each Z' mass, and separately for the ee channel and $\mu\mu$ channels. This is done separately for each model and for the light dark sector (LDS) and heavy dark sector (HDS). However, some of these will be combined in the training process. Each network is trained on one of the models (dark Higgs or light vector) and either the LDS or HDS, while training on both the ee and $\mu\mu$ channels as well as all Z' mass signals combined. As flavor is a feature used in the training, the network may still distinguish electrons and muons. There are several advantages to this approach. It increases the number of signal events the model can be trained on, which may result in better performance, although the network will have to learn a larger variety of models. Furthermore, for possible further use of the neural networks on data, it is better if it recognizes a larger variety of signals, as it is expected that any signal found in the data may deviate from the exact models studied in this analysis.

We will define the signal region as the region where the classification score is above 0.90. Although it is likely that the expected significance from the MC samples will be higher if we choose a smaller region, for example above 0.99, one should keep in mind that if the signal is observed in the physical data, there will be deviations in the free parameters which may cause it to receive a lower classification score, that would place it outside the signal region.

7.2 Event weights

A common problem in ML classification problems is that the amount of available data often is not equally distributed among the different classes, meaning that some classes may contain a significantly larger number of events to train on than others. If the network trains on such a data set without any modification, it will learn the classes with more statistics better than the ones with less statistics. This is because each data point has a similar effect on the loss, which means that the impact of a class on the loss will depend on the number of samples for that class. If the statistics is much smaller in one of the classes, the effect on the loss will become negligible and the network will not learn from that class.

In our case, a data set containing a significantly larger amount of background events than signal events is used, with a ratio of $\sim 1000/1$. In this case, the network will learn to recognize background events better than signal events, or may not learn to recognize signal events at all. Therefore the effect of background events on the loss must be scaled down in order for both types to have the same impact during the training. Each event is therefore associated with a weight, which is a constant that the loss contribution from the event is multiplied with.

The contribution to the event weights that compensates for the data imbalance is called the *class weight*. The class weight is given by

$$class\,weight = 1 - \frac{n_{class}}{n_{total}}, \tag{7.1}$$

where n_{class} is the number of samples in the given class, and n_{total} is the total number of samples. Even though this solves the imbalance problem, it is still necessary that the data set for the signal events is sufficiently large in order to represent the variety of event characteristics within that class.

In addition to the class weights, the background events are scaled with the event weights calculated in the event selector algorithm in section 5.3 in order for the distribution of events the network is trained on to resemble the distribution of events in the real data.

7.3 Normalization of data

In gradient-based optimization algorithms, problems may arise when the magnitudes of the features are in different ranges. This is because the value of the feature influences the step size. This means that features within a range of high values will influence the network more than features with smaller values, although the features are equally important. If the features with the largest values turn out not to be important for predicting the target value, it may result in the network not learning at all or not using important, available information from other features. However, the exact values of the features of each data point are not important in order for the network to learn. Instead, the relation of the values to each other are important, as the network learns the patterns. Therefore, it is common to use normalization methods. These are methods used to make sure that the different features are in the same range, while conserving the characteristics of the data. It is common to normalize the data set in order for the features to stay within the ranges of $[0, 1]$ or $[-1, 1]$. This must be done to both the training and test sets, as the model will only learn to recognize data within these ranges.

One of the most common normalization methods is called *standardization*. In this method, the distribution for each feature is centered at 0 while setting the variance to 1. This is done by setting

$$X' = \frac{X - \bar{X}}{\sigma} \tag{7.2}$$

where X is an array containing all the training data points for a feature, $\bar{X}$ is the mean of the distribution, σ^2 is the variance and X' is the new, normalized array. A different method is *linear scaling*. In this case, the features are scaled to the range of $[0, 1]$ for every data point x of a feature, setting

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{7.3}$$

where x_{max} is the maximum value of the feature in the data set, x_{min} is the minimum value and x' is the new value of the data point. As these methods are variations of each other and the main goal of both is to transform the features to a similar scale, the difference in performance between the two are expected to be small. In our case, the linear scaling is used.

7.4 Network architecture and hyperparameters

The neural network consists of 13 input nodes (one for each feature), as well as one output node. The activation function used in the input layer and the hidden layers is the rectified linear unit (ReLU), defined in section 4.4, in order to avoid the vanishing gradient problem. In the output layer, the sigmoid function (introduced in section 4.4) is used. This is used in order for the network to produce an output between 0 and 1. The Adam optimization algorithm (introduced in section 4.6) is used, due to its advantage of having an adaptive learning rate.

As it is not possible to decide the optimal hyperparameters by using a formula, they must be determined by manual testing. There are often limitations for some of the hyperparameters because of the time it takes to train the network. The parameters that have a significant effect on training time are the number of hidden layers, the number of neurons per layer and the number of epochs. The batch size may also have an impact. As the number of MC events are in the order of 10^7, we meet computing time constraints when these parameters are large, even though supercomputers are used. It is therefore necessary to make a trade-off to decide the limit on each hyperparameter and then optimize within that region. If only one hidden layer is used, it is possible to use a few hundred neurons in the layer without significant computing time. However, when the number of hidden layers increases, the computing time increases significantly. A general rule is that it is often better to prioritize a deeper network (more hidden layers) rather than a wider (more neurons per layer), although this is not true in every case. We decide to test a maximum of fur hidden layers with a maximum of 100 neurons per layer, as time constraints become significant at this point. Also, we decide to use a maximum of 50 epochs with the possibility of early stopping for the same reason, since the network usually has converged at that point. However, a higher number of hidden layers and neurons per layer are not necessarily better, and different combinations should therefore be tested. We also use a batch size of 10% of the training set in order to make sure there is a large number of signal events in each batch, as the training set is imbalanced.

Other hyperparameters include the learning rate (ϵ) and the $L2$ weight decay (λ) for regularization, which were introduced in section 4.6 and 4.7, in order to prevent overfitting. These do not have a large impact on computing time and may be tested across a wide range of values. Common default values that may be used

as starting points are $\epsilon = 10^{-2}$ and $\lambda = 10^{-5}$. However, there is no specific reason for choosing these values, as they may not be optimal for the network constructed for our task. Values around these will therefore be tested, in addition to testing how increasing or decreasing them impact the performance. A too small value of ϵ may result in the network not changing the weights of the model enough to reach a global minimum for the loss. A too large ϵ may result in the weights not converging. As we are using the Adam optimizer, defined in section 4.6.2, the choice of learning rate is not expected to be as important as when using stochastic gradient descent (because of its adaptive learning rate), but it still has an impact. A low value of λ may result in overfitting, as this is what the regularization technique attempts to avoid. However, a too large value may result in underfitting which prevents the model from learning efficiently. We decide to test $\epsilon = \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ as we then test the default value as well as deviations from it in both directions. For the same reason, we test $\lambda = \{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}\}$. For the number of layers, we test from 1 to 4 hidden layers, as well as number of neurons $= \{5, 10, 50, 100\}$. The effect of these parameters are expected to be smaller than the effect of the choice of ϵ and λ, as networks often may reach a high accuracy with only one hidden layer and a limited number of neurons, although this depends on the complexity of the problem. For the hyperparameters that are specific to the Adam optimizer, we will use the default values [68], which are $\rho_1 = 0.90$, $\rho_2 = 0.99$ and $\delta = 10^{-8}$, defined in section 4.6.2.

The parameters are tested by *grid searches*, which means that we train the network with different combinations of hyperparameters and test the performance. The combination that leads to the best performance are then usually chosen. However, one may also use other types of reasoning (such as checking for overfitting). We test ϵ and λ in combination, as well as the number of hidden layers and neurons per layer in combination. Although the number of neurons per hidden layer does not have to be the same in all layers, we keep it constant if the results are good, as variations only lead to small deviations in the results. Grid searches are performed for each model in case there are differences in performance. The hyperparameters are therefore chosen separately for each model. The grid search is first performed for ϵ and λ and then for the number of hidden layers and neurons. The combinations of the optimized hyperparameters may therefore lead to a difference between the performance seen in the grid searches and in the final networks. In section 7.6.1 and 7.7.1, the grid searches are shown using accuracy, defined in section 4.1.2, as

measure. The grid searches using AUC are shown in appendix C. These are both considered when choosing parameters, but the deviations in AUC are much smaller, and the accuracy therefore gives a clearer answer to which parameters result in the best performance.

7.5 Interpretation of the feature importance

Permutation feature importance, defined in section 4.8, will be used in order to find information about how the network has learned and which features have a large effect on its predictions. However, one should be aware that permutation feature importance for neural networks provides an imperfect result when used to evaluate how well each feature may be used to predict the class of an event. This is because it measures the increase or decrease in loss when the events are permuted for a specific feature. Therefore, if a feature is highly correlated with another or several other features, a large amount of the same information may still be contained in those features, which leads to a smaller increase in the loss. This may be the case for variations of E_T^{miss}, such as $E_T^{miss,sig}$. Because of this, the exact value of the permutation feature importance is not necessarily a precise measure of the importance of a feature. However, it may still be an indication of whether a feature has some utility for predicting the class or not. If the feature importance is close to zero, the network has most likely not found information in the feature that may be used for prediction. In the following sections, we will perform the optimization of the neural networks for each model, measure their performance for each model, as well as their expected significances.

7.6 Dark Higgs model

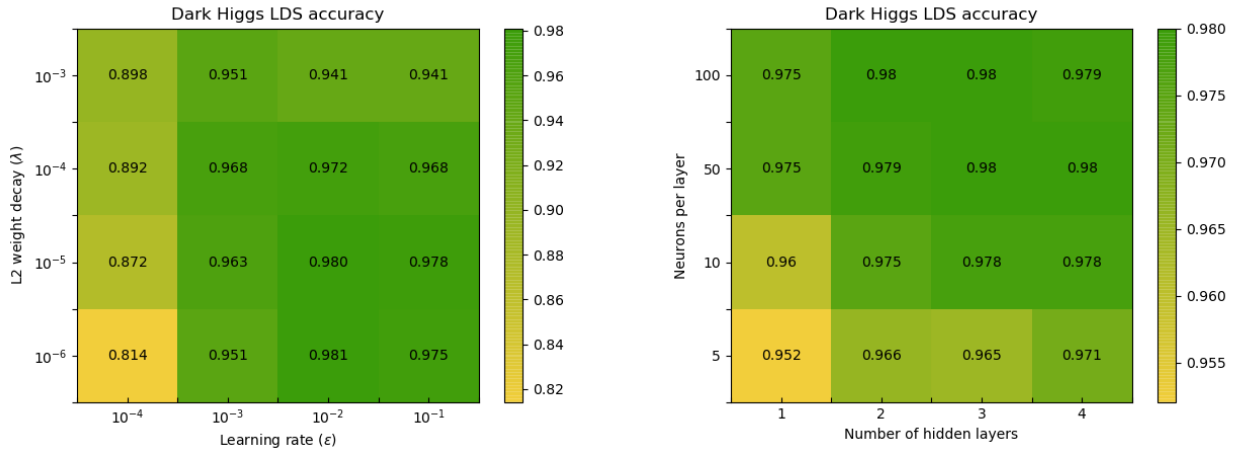
7.6.1 Hyperparameters

A grid search measured using accuracy for the Dark Higgs LDS model is shown in figure 7.1. The same grid searches using the area under the ROC curve (AUC), defined in section 4.1.2, are shown in appendix C. The optimal parameters are found to be $\epsilon = 10^{-2}$ and $\lambda = 10^{-6}$. However, as the difference in accuracy between $\lambda = 10^{-5}$ and $\lambda = 10^{-6}$ is negigible, we decide to use $\lambda = 10^{-5}$ in order to prevent overfitting. The optimal number of layers and neurons per layer are found to be at least 3 hidden layers and at least 50 neurons. However, no improvement is achieved by increasing the parameters further, as seen in figure 7.1. Instead, the accuracy is

Dark Higgs LDS	
Hyperparameter	Value
Number of hidden layers	3
Neurons per layer	50
Learning rate (ϵ)	10^{-2}
$L2$ weight decay (λ)	10^{-5}
Epochs	50
Batch size	10% of training set
Exponential decay rate (ρ_1)	0.90
Exponential decay rate (ρ_2)	0.99
Stabilization constant (δ)	10^{-8}

Table 7.1: Hyperparameters used for training the neural network on the dark Higgs LDS.

maintained at the same level of ~ 0.980. Similar results are observed when using AUC, as shown in appendix C. Therefore, 3 hidden layers and 50 neurons per layer are chosen for efficiency purposes. All of the hyperparameters used for training the network on the dark Higgs model for the light dark sector are listed in table 7.1.



(a) (b)

Figure 7.1: Grid searches for optimal hyperparameters a) $L2$ weight decay (λ) and learning rate (ϵ), and b) number of hidden layers and neurons per layer for ML training on the dark Higgs LDS using accuracy as measure.

7.6.2 Performance

The loss on the training and validation set during training for each epoch is shown in figure 7.2. The training and validation loss decreases quickly until epoch 10, before

decreasing slowly and converging. However, a large drop in the validation loss and a corresponding increase in training loss is observed around epoch $30 - 35$. Although the opposite is usually a sign of overfitting, which therefore likely is not happening in this case, the network may be moving away from the global loss minimum. As the training loss increases, the network recognizes it as a mistake and corrects it, before converging.

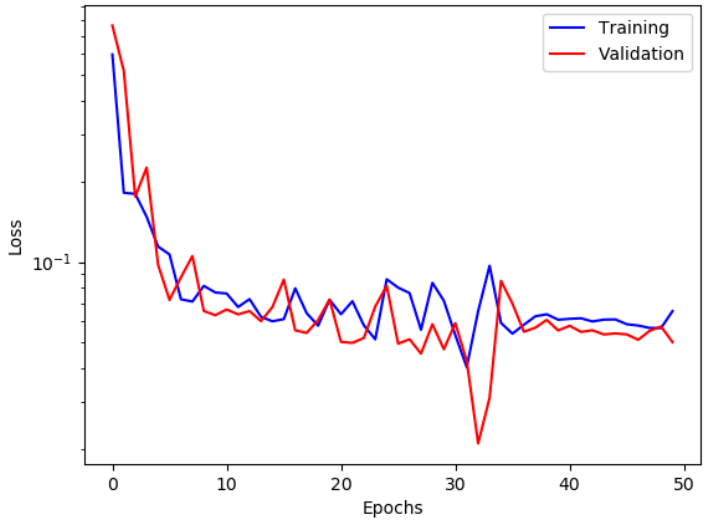
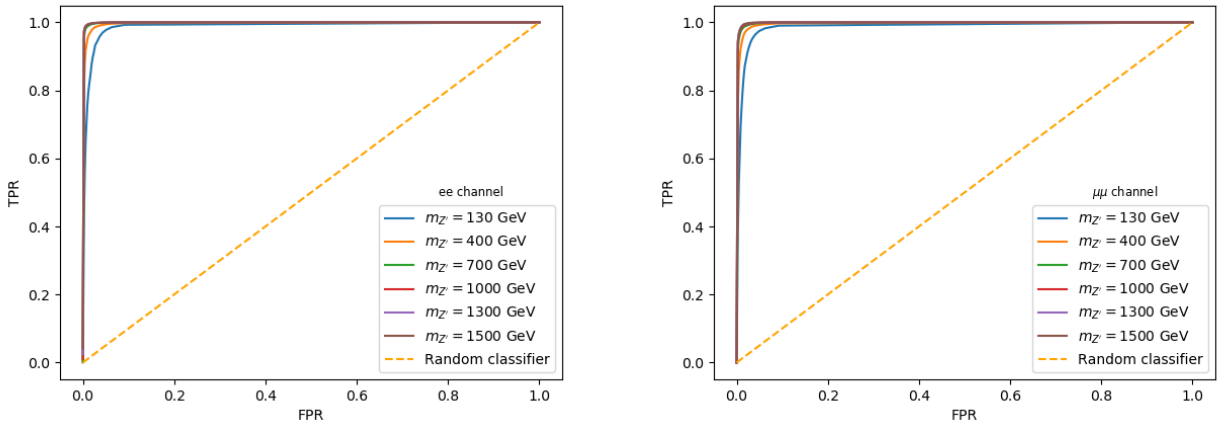


Figure 7.2: Training and validation loss as a function of epochs during training on the dark Higgs LDS.

The performance of the trained network is tested on the test set using equal amounts of signal and background events in order to measure its performance on both types. The accuracy and AUC obtained for each signal are shown in table 7.2 and the ROC curves for some of the Z' masses are shown in figure 7.3. The network achieves an accuracy above 0.93 for all Z' masses, and significantly higher for most signals, suggesting the network has learned to recognize the main characteristics of signal and background events. The difference in performance between the models is best captured using accuracy rather than AUC as the AUC reaches a level close to 1.0 quickly. The accuracy and AUC generally increase with increasing $m_{Z'}$ until $m_{Z'} \sim 700$ GeV where it converges and maintains a similar performance for further increasing mass. The highest accuracy is 0.989 for $m_{Z'} = 1100$ GeV and $m_{Z'} = 1400$ GeV in the ee channel. The performance is somewhat better for the ee channel than the $\mu\mu$ channel, but the difference is relatively small.



(a) (b)

Figure 7.3: ROC curve plots for a selection of Z' mass signals in the a) ee and b) $\mu\mu$ channel in the dark Higgs LDS.

Dark Higgs LDS				
	ee channel		$\mu\mu$ channel	
$m_{Z'}$ (GeV)	Accuracy	AUC	Accuracy	AUC
130	0.933	0.991	0.931	0.989
200	0.970	0.995	0.965	0.993
300	0.955	0.994	0.947	0.992
400	0.979	0.997	0.975	0.997
500	0.984	0.998	0.979	0.998
600	0.986	0.999	0.983	0.999
700	0.987	0.999	0.984	0.999
800	0.988	1.0	0.985	0.999
900	0.988	1.0	0.985	0.999
1000	0.988	1.0	0.985	0.999
1100	0.989	1.0	0.987	0.999
1200	0.988	1.0	0.986	0.999
1300	0.988	1.0	0.987	0.999
1400	0.989	1.0	0.987	0.999
1500	0.989	1.0	0.987	0.999

Table 7.2: Accuracy and AUC achieved by the neural network for different Z' mass signals in the dark Higgs LDS.

7.6.3 Results

The feature importance for the features used in the ML model are shown in figure 7.4. The $E_T^{miss,sig}$ has a significantly larger feature importance than the other features, meaning that the model loses a significant amount of predictive power when events are permuted for this feature. It may also suggest that it contains some information that is not contained in E_T^{miss}, which has a relatively large, but much smaller feature importance. This may be due to the two features being correlated, and it is possible that E_T^{miss} had a larger feature importance if $E_T^{miss,sig}$ was not used during training. The invariant mass and the number of b-jets also have a relatively large feature importance, while p_{T1}, p_{T2}, the transverse mass and H_T have small positive values. The flavor, charge, $\Delta\phi_{E_T^{miss},ll}$, $\Delta\phi_{l,l}$ and η have negligible values. This suggests that the network did not find important information in these features which could help distinguishing the SM background and the signal. Also, that the feature importance is not negative for these features suggests that they have not contributed to overfitting.

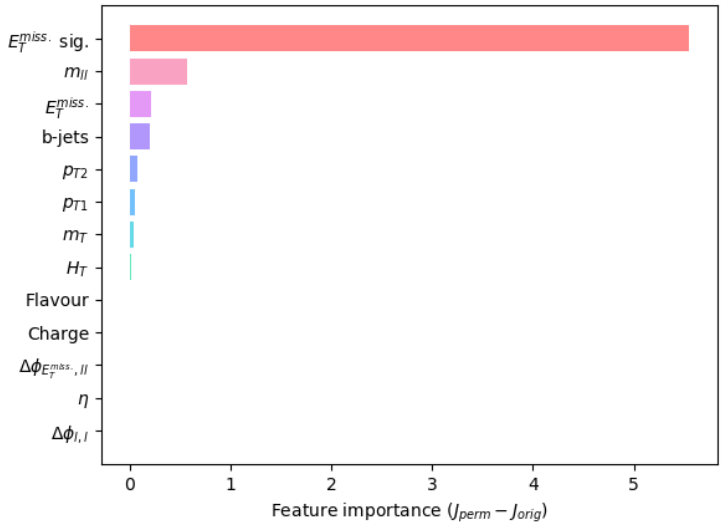
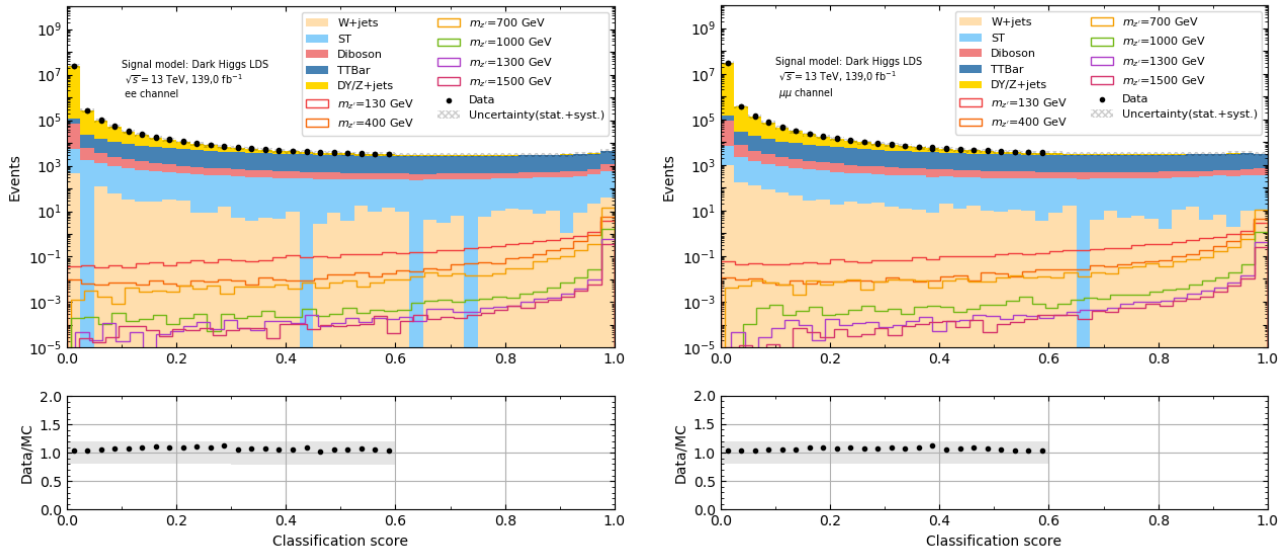


Figure 7.4: Permutation feature importance of the features used to train the neural network for the dark Higgs LDS.

As shown in figure 7.5, the model produces similar output for the physical data as it does for the MC background, suggesting that the model works properly. The real data is cut off at classification scores above 0.60 in order not to show the data in the signal region. The signal region is defined as events receiving a score above

0.90. The amount of background increases exponentially when the classification score approaches 0, while the same happens to the signal when approaching 1. The DY and $Z+$ jets are mostly filtered out, while a larger amount of the other background types are within the signal region. As the model performs well for all of the Z' mass signals, it is likely that it would be able to identify a signal where the Z' mass deviates somewhat from the masses the network is trained on.

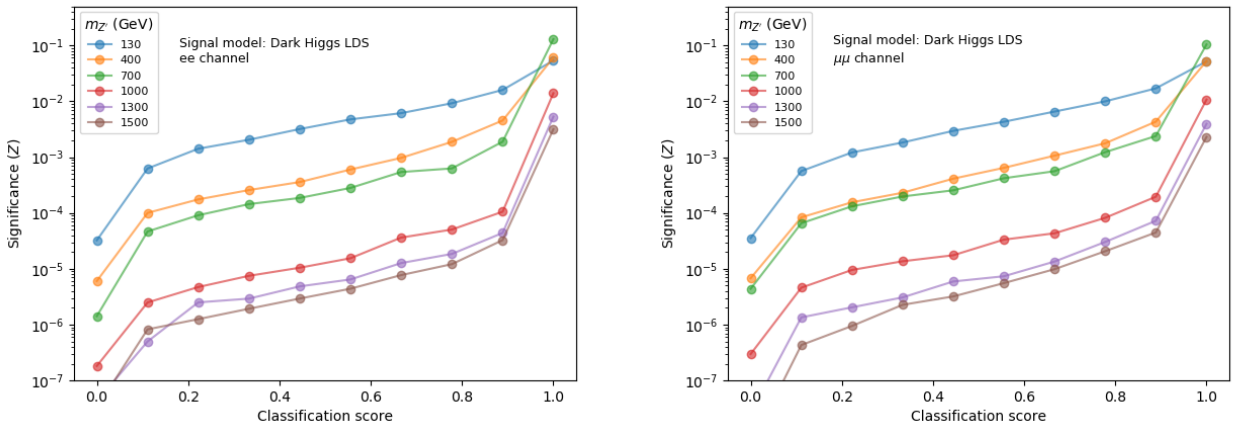


(a) (b)

Figure 7.5: Classification score distributions for the background and signal for the neural network in the dark Higgs LDS. The results are shown in the a) ee and b) $\mu\mu$ channel and compared with real data outside of the signal region.

The expected significance is plotted for each Z' mass in figure 7.6. These are calculated for each bin (from 0.0 to 0.1, etc.), and show that the highest expected significance is obtained in the signal region (from 0.9 to 1.0). As the expected significance is undefined in some of the regions with low classification score when using the full equation 5.6 (due to negative values in the square root), we instead plot the expected significance $Z = \frac{s}{\sqrt{b}}$ as this is a good approximation when $s \ll b$. As expected, the expected significance increases when the classification score increases for all of the signals and is significantly higher in the signal region (above 0.90) than outside it. In table 7.3 we list the expected significance for each Z' mass and both the ee and $\mu\mu$ channels using the full formula, defined in section 5.9. The highest expected significance is 0.13 for $m_{Z'} = 700$ GeV in the ee channel. As the mass

variations are tested using the same neural network, they have the same amount of background events in the signal region and varying amounts of signal, which depends on the performance of the network on each signal and the total amount of expected signal. The expected significance is generally decreasing with larger Z' mass although the performance of the network increases.



(a) (b)

Figure 7.6: Expected significance for the different signals in the dark Higgs LDS at different classification scores in the a) ee and b) $\mu\mu$ channels. The approximation $Z = \frac{s}{\sqrt{b}}$ is used in order to prevent undefined values.

Dark Higgs LDS		
	ee channel	$\mu\mu$ channel
$m_{Z'}$ (GeV)	Expected significance (Z)	
130	$5.3 \cdot 10^{-2}$	$5.2 \cdot 10^{-2}$
200	$7.1 \cdot 10^{-2}$	$6.6 \cdot 10^{-2}$
300	$7.0 \cdot 10^{-2}$	$6.0 \cdot 10^{-2}$
400	$6.0 \cdot 10^{-2}$	$5.2 \cdot 10^{-2}$
500	$5.6 \cdot 10^{-2}$	$4.7 \cdot 10^{-2}$
600	$3.8 \cdot 10^{-2}$	$3.1 \cdot 10^{-2}$
700	$1.3 \cdot 10^{-1}$	$1.0 \cdot 10^{-2}$
800	$2.1 \cdot 10^{-2}$	$1.7 \cdot 10^{-2}$
900	$1.6 \cdot 10^{-2}$	$1.2 \cdot 10^{-2}$
1000	$1.4 \cdot 10^{-2}$	$1.1 \cdot 10^{-2}$
1100	$8.9 \cdot 10^{-3}$	$6.9 \cdot 10^{-3}$
1200	$6.9 \cdot 10^{-3}$	$5.2 \cdot 10^{-3}$
1300	$5.2 \cdot 10^{-3}$	$3.9 \cdot 10^{-3}$
1400	$4.1 \cdot 10^{-3}$	$3.0 \cdot 10^{-3}$
1500	$3.1 \cdot 10^{-3}$	$2.3 \cdot 10^{-3}$

Table 7.3: Expected significances for different signals in the dark Higgs LDS for the ee and $\mu\mu$ channels.

Light vector LDS	
Hyperparameter	Value
Number of hidden layers	3
Neurons per layer	50
Learning rate (ϵ)	10^{-2}
$L2$ weight decay (λ)	10^{-6}
Epochs	50
Batch size	10% of training set
Exponential decay rate (ρ_1)	0.90
Exponential decay rate (ρ_2)	0.99
Stabilization constant (δ)	10^{-8}

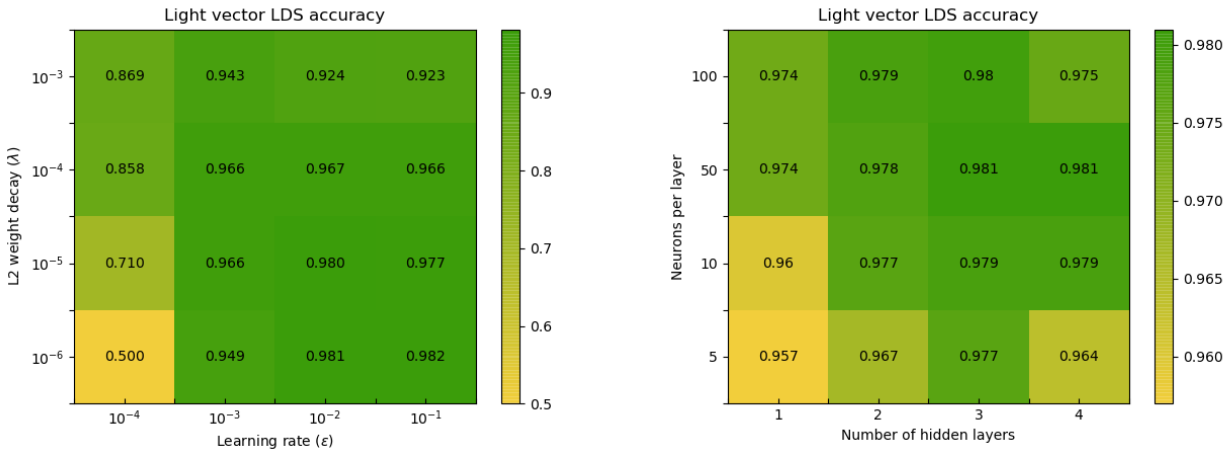
Table 7.4: Hyperparameters used for training the neural network on the light vector LDS.

7.7 Light vector model

We will now present the results for the neural networks trained on the light vector model. We will here show the results for the light dark sector, while results for the heavy dark sector are shown in appendix G.

7.7.1 Hyperparameters

Grid searches measured using accuracy for the Dark Higgs LDS model are shown in figure 7.7. The same grid searches using the area under the ROC curve (AUC), defined in section 4.1.2, are shown in appendix C.3. The hyperparameters resulting in the highest accuracy and AUC are at least 3 hidden layers with 50 neurons per layer, as well as a learning rate of $\epsilon = 10^{-1}$ and $L2$ weight decay $\lambda = 10^{-6}$. However, as shown in figure 7.8, the network then has problems converging, likely because of a too large step size (learning rate). Therefore, this parameter is changed to $\epsilon = 10^{-2}$ in order for the network to converge more easily. The difference in accuracy obtained using these values in the grid search is marginal. The final choices of hyperparameters are listed in table 7.4.



(a) (b)

Figure 7.7: Grid searches for optimal hyperparameters a) $L2$ weight decay (λ) and learning rate (ϵ), and b) number of hidden layers and neurons per layer for ML training on the light vector LDS using accuracy as measure.

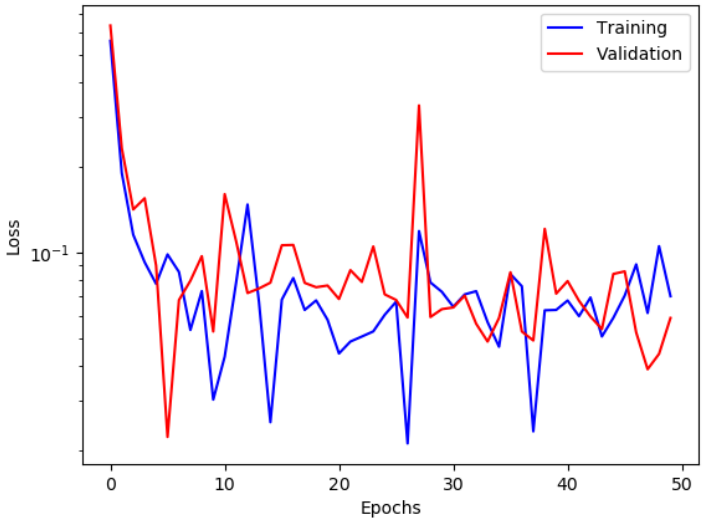


Figure 7.8: Training and validation loss as a function of epochs during training on the light vector LDS, when testing $\epsilon = 10^{-1}$.

7.7.2 Performance

The loss on the training and validation set during training for each epoch are shown in figure 7.9. The training and validation loss decrease quickly until approximately epoch 10. After that, there is a large spike in both training and validation loss, before they converge. There is a smaller increase in training loss around epoch 40-45, which is quickly corrected. The gap between training and validation loss is small, which suggests that the network generalizes well.

The accuracy and AUC obtained for each signal are listed in table 7.5, and the ROC curves are shown in figure 7.10. The lowest accuracy achieved is 0.881 for $m_{Z'} = 130$ GeV in the ee channel, which means that the network has learned the main features of the signal. However, this is not a satisfactory result, as it means that $\sim 12\%$ of events will be misclassified. However, as the Z' mass increases, the performance improves quickly, with an accuracy of > 0.980 for all signals with Z' mass of 400 GeV or more. The highest accuracy reached is 0.989 for $m_{Z'} = 1400$ GeV in the ee channel. The AUC quickly exceeds 0.999, which means that the performance appears perfect when using this metric, and the difference in performance between the different signals is not as clear when using this metric in this case. The accuracy achieved in the ee channel is generally somewhat higher than in the $\mu\mu$ channel, except at the lowest Z' masses.

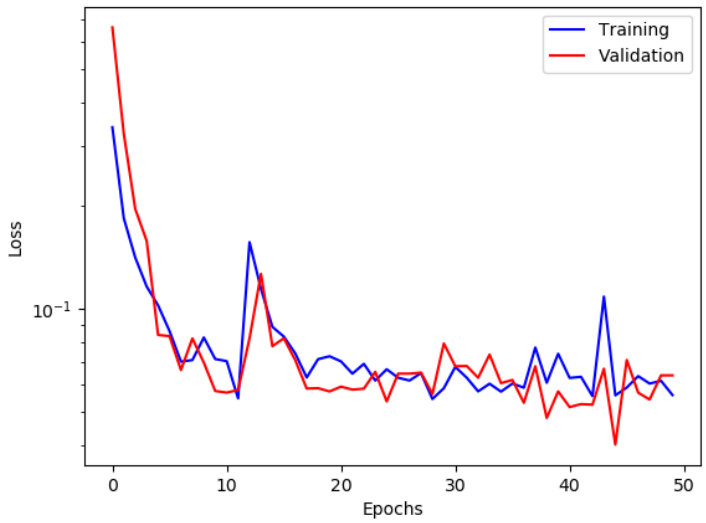
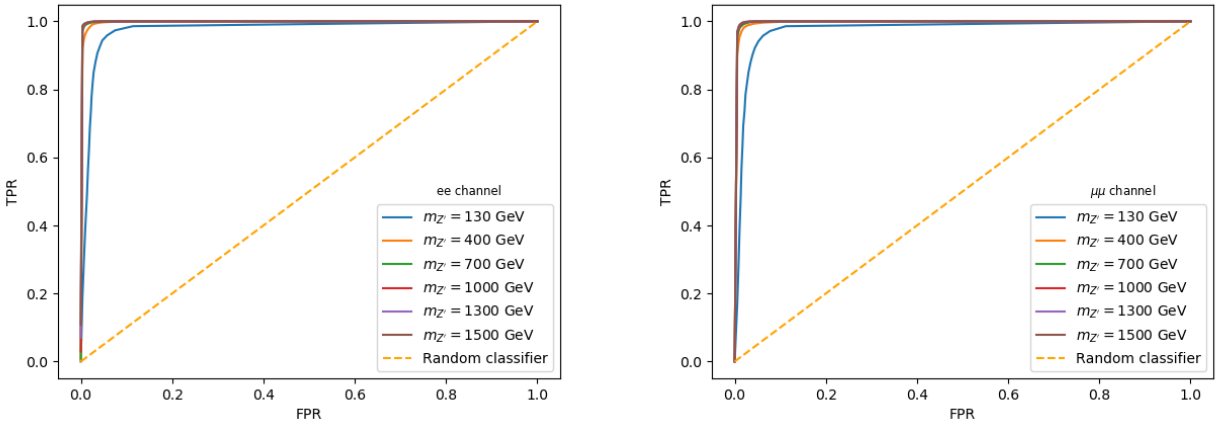


Figure 7.9: Training and validation loss as a function of epochs during training on the light vector LDS.



(a) (b)

Figure 7.10: ROC curve plots for a selection of Z' mass signals in the a) ee and b) $\mu\mu$ channel in the light vector LDS.

Light vector LDS				
	ee		$\mu\mu$	
$m_{Z'}$ (GeV)	Accuracy	AUC	Accuracy	AUC
130	0.881	0.983	0.888	0.980
200	0.965	0.994	0.965	0.991
300	0.979	0.997	0.977	0.996
400	0.982	0.998	0.982	0.998
500	0.986	0.999	0.982	0.998
600	0.988	1.0	0.983	0.999
700	0.987	1.0	0.984	0.999
800	0.988	1.0	0.985	0.999
900	0.989	1.0	0.985	0.999
1000	0.987	1.0	0.987	1.0
1100	0.989	1.0	0.987	0.999
1200	0.989	1.0	0.988	1.0
1300	0.988	1.0	0.988	0.999
1400	0.989	1.0	0.987	0.999
1500	0.988	1.0	0.985	0.999

Table 7.5: Accuracy and AUC achieved by the neural network for different Z' mass signals in the light vector LDS.

7.7.3 Results

The feature importance for the features the ML model is trained on, are shown in figure 7.11. Similarly to what was the case in the dark Higgs LDS, the $E_T^{miss,sig}$ has a significantly higher feature importance than the other features. The invariant mass and E_T^{miss} also have relatively high feature importance. As these are the ones expected to be most important in classifying events as signal or background, the result is a sign of the network working properly. It also indicates that the method of measuring feature importance provides real information about the neural network. The features that have a small positive feature importance, are the number of b-tagged jets, p_{T1}, p_{T2} and the transverse mass. The feature importance of p_{T2} is higher than that of p_{T1}. However, these are expected to be highly correlated, which may result in either of them having a very low feature importance, as some of the information is still contained in the other feature. The flavor, charge, H_T, η, $\Delta\phi_{l,l}$ and $\Delta\phi_{E_T^{miss},ll}$ all have a negligible feature importance, meaning that the network has not found information in these features that may be used for classifying events as signal or background for this signal model.

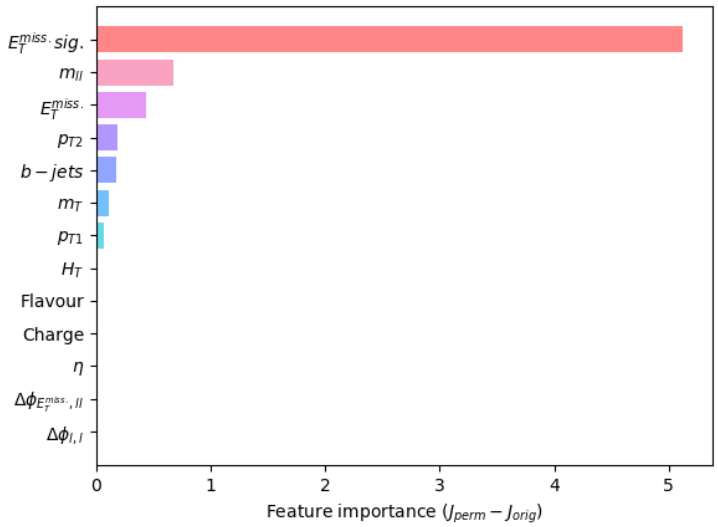
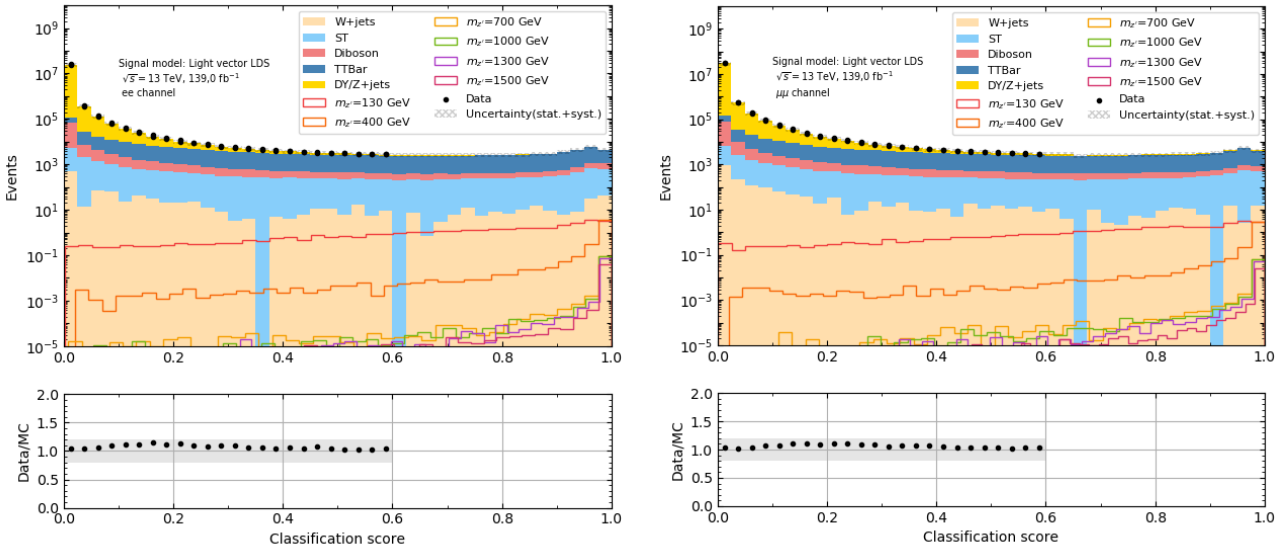


Figure 7.11: Permutation feature importance of the features used to train the neural network for the light vector LDS.

Figure 7.12 shows the output of the neural network for the different background types and some of the signals. The outputs show that the network classifies the signals with a high Z' mass better than those with a low mass, as their curves

are steeper when approaching a classification score of 1.0. However, the amount of events in the signal region is still higher for $m_{Z'} = 130$ GeV than for most other signal due to a higher total number of expected signal events. The amount of signal is significantly higher in the bin with the highest classification score than in the ones below, which means that the expected significance achieved would be higher if the signal region was more narrow. The Drell-Yan and Z+jets backgrounds are almost completely filtered out in the signal region, wile a larger fraction of the other background types are misclassified as signal.

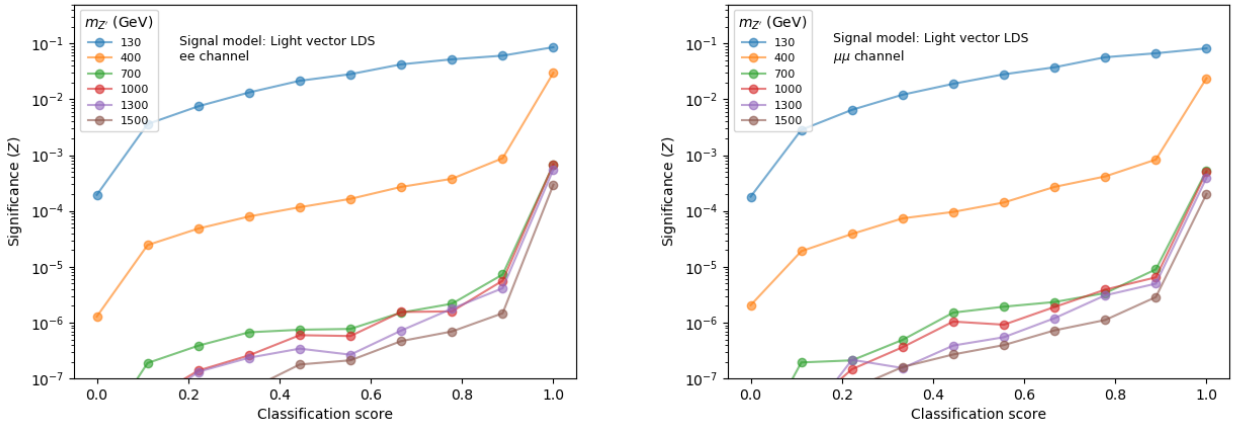


(a) (b)

Figure 7.12: Classification score distributions for the background and signal for the neural network in the light vector LDS. The results are shown in the a) ee and b) $\mu\mu$ channel and compared with real data outside of the signal region.

The expected significance for some of the signals at different classification scores is shown in figure 7.13, using the approximation $Z = \frac{s}{\sqrt{b}}$ in order to avoid undefined values at the low classification scores. The expected significance for all of the signals increase with increasing classification scores, but the increase is steeper for the highest masses, as expected from the results observed in figure 7.12. The expected significances obtained in the signal region for all of the signals are listed in table 7.6, using the full formula for Z, introduced in section 5.9. Although the accuracy obtained generally is higher for the signals with a larger Z' mass, the expected significance is highest for $m_{Z'} = 130$ GeV with $Z = 8.5 \cdot 10^{-2}$ in the ee channel, and then

generally decreasing with increasing mass due to a lower number of total expected events. The expected significance is also somewhat higher in the *ee* channel than in the $\mu\mu$ channel. Although the network achieved a relatively high accuracy, the expected significances are still very small, and the signal regions are not sensitive enough to discover or exclude the signals by analyzing real data. This is mainly due to a low number of total expected signal events. However, the expected significance could be improved further by removing a larger amount of background.



(a) (b)

Figure 7.13: Expected significance for the different signals in the light vector LDS at different classification scores in the a) *ee* and b) $\mu\mu$ channels. The approximation $Z = \frac{s}{\sqrt{b}}$ is used in order to prevent undefined values.

Light vector LDS		
$m_{Z'}$ (GeV)	Expected significance (Z)	
	ee channel	$\mu\mu$ channel
130	$8.5 \cdot 10^{-2}$	$8.1 \cdot 10^{-2}$
200	$1.2 \cdot 10^{-2}$	$1.1 \cdot 10^{-2}$
300	$3.6 \cdot 10^{-2}$	$3.0 \cdot 10^{-2}$
400	$3.0 \cdot 10^{-2}$	$2.4 \cdot 10^{-2}$
500	$8.2 \cdot 10^{-3}$	$6.5 \cdot 10^{-3}$
600	$9.0 \cdot 10^{-3}$	$7.1 \cdot 10^{-3}$
700	$6.7 \cdot 10^{-4}$	$5.2 \cdot 10^{-4}$
800	$3.5 \cdot 10^{-3}$	$2.6 \cdot 10^{-3}$
900	$2.3 \cdot 10^{-3}$	$1.7 \cdot 10^{-3}$
1000	$6.8 \cdot 10^{-4}$	$5.0 \cdot 10^{-4}$
1100	$1.1 \cdot 10^{-4}$	$7.8 \cdot 10^{-4}$
1200	$1.1 \cdot 10^{-3}$	$5.6 \cdot 10^{-4}$
1300	$5.5 \cdot 10^{-4}$	$3.9 \cdot 10^{-4}$
1400	$3.9 \cdot 10^{-4}$	$2.8 \cdot 10^{-4}$
1500	$3.0 \cdot 10^{-4}$	$2.0 \cdot 10^{-4}$

Table 7.6: Expected significances for different signals in the light vector LDS for the ee and $\mu\mu$ channels.

7.8 Comparison of methods

We will now compare the cut and count method and the ML method by evaluating the expected significances achieved. There are however a few things that should be mentioned before making a judgement on which method is performing better. In the cut and count method, we chose to make only one signal region which was expected to give a high sensitivity to most of the signals, except the one with $m_{Z'} = 130$ GeV. Dividing it into two or tree regions would likely increase the sensitivity to some of the signals. One could also have made a mass cut for each $m_{Z'}$ which would also likely give a high sensitivity to each of them. However, as the aim of the cuts is to search for the signals in the data at some point and because there are several unknown free parameters in the models, the signal region should not be made too tight, as this could cause the real signal to fall outside the signal region. Therefore, it is often a good idea that each signal region is sensitive to a range of signals. The same consideration is made in the ML analysis, where a tighter signal region could have been chosen, which would increase the sensitivity to the MC samples, as observed in figure 7.5. Therefore, one should not only judge the methods by the expected significance they obtain, but also how well they generalize when some parameters are modified. However, in both approaches, a conservative signal region is chosen, which generalizes well for all of the $m_{Z'}$ signals (except $m_{Z'} = 130$ GeV in the cut and count method). Therefore, it is reasonable to judge their performance by the expected significance when the first criterion is met by both methods.

The ratio of the expected significances $Z_{NN}/Z_{C\&C}$ obtained by the neural network and the cut and count method are listed in table 7.7. Similar tables for the other signal models are shown in appendix H. We observe that the neural network consistently reaches a higher expected significance for the signals. Excluding $m_{Z'} = 130$ GeV, the ratio is between 1.34 and 18.2. The higher performance of the neural network relative to the cut and count method generally increases with increasing $m_{Z'}$ and is mostly higher in the $\mu\mu$ channel. The difference between the ee and $\mu\mu$ channels may indicate that the cuts where more effective in the ee channel, as the neural network generally obtained a slightly higher accuracy in the ee channel than the $\mu\mu$ channel. For $m_{Z'} = 130$ GeV, the ratio is of the order of 10^3, but this is because the signal is almost completely filtered out by the mass cut in the cut and count method. However, it should be noted that the neural network is able to perform well on this signal while also performing better on the other signals.

Dark Higgs LDS		
$m_{Z'}$ (GeV)	Expected significance ($Z_{NN}/Z_{C\&C}$)	
	ee channel	$\mu\mu$ channel
130	$2.79 \cdot 10^3$	$1.58 \cdot 10^3$
200	1.34	1.40
300	1.67	1.67
400	1.94	2.08
500	1.75	1.81
600	3.17	1.82
700	1.71	1.85
800	1.75	1.98
900	1.80	2.00
1000	1.79	2.16
1100	1.85	2.23
1200	1.82	2.36
1300	1.86	2.44
1400	1.95	2.72
1500	$1.82 \cdot 10^1$	2.91

Table 7.7: Ratio of expected significance $Z_{NN}/Z_{C\&C}$ obtained by the neural network (NN) and cut and count (C&C) method for the dark Higgs LDS.

It is clear that the ML method has been most successful in this case. However, it is still not sensitive enough for the signal model to be expected to be discovered or excluded if searched for by analyzing real data.

We begun this chapter with explaining the implementation of the neural networks for the analysis, including the optimization of the network architecture and hyperparameters. The networks were trained, and were successful in separating signal and background with a relatively high accuracy. The expected significances of the signals were calculated and compared with the cut and count method, and the neural networks turned out to reach higher sensitivities.

8 Conclusion

In this thesis, we have searched for a new gauge boson Z', decaying to a dilepton pair, and dark matter, measured as missing transverse energy (E_T^{miss}). This was done by studying Monte-Carlo simulated signal events based on two different models, namely the dark Higgs model and the light vector model in combination with the Standard Model background. We begun by comparing the MC background with real data recorded by the ATLAS detector during the full Run 2 at the LHC in order to check that it is an accurate representation of the real event distributions. Signal regions were then constructed in order to maximize the sensitivity to the two models by using two different methods. First, the standard cut and count method was used, by making cuts on some of the variables in order to remove a large amount of background while keeping as much signal as possible, and then measuring the expected significance for each signal model. Signal regions were also created by training neural networks to classify events as either signal or background. This was done by first carefully selecting variables that were expected to be useful for the classification, and then optimizing the network in order for it to perform as well as possible. The neural networks performed well, consistently reaching an accuracy above 0.98 for most of the signals. The expected significance was then measured and compared with the ones obtained by the cut and count method. The neural networks performed better for almost all of the signals, and significantly better for some of them, even though the signal region could have been made narrower in order to increase the significance further.

However, by looking at the expected significances, it is clear that a search using real data and the signal regions used in this analysis will not be sensitive enough to either discover or exclude the signals. This is due to the signal models predicting a very small amount of events. This is partly a result of the choices of the free parameters that were used for simulating the signals. In future studies, it may therefore be useful to also consider signals with higher coupling constants. In the cut and count method, which here was used in order to be compared with the machine learning method, one may also create several signal regions designed to be more sensitive to each of the signals. It may also be possible to train neural networks on several different signal models that share some of the main characteristics in order to be able to detect a larger variety of possible signals in the data. This would be useful, as there are several uncertainties in the free parameters of the signals, if the signals

exist. However, a trade-off must be made between making the network general and making it precise as generalization may result in a lower accuracy. Although the sensitivity is not high enough to exclude or discover the signals used in this study, we have seen that it may be possible to discover dark matter particles at the LHC if they exist, as they may be involved in processes leading to specific characteristics in the final states that may be recognized, for example by a neural network, especially if they result in an increased amount of E_T^{miss}. It will therefore be possible to use the methods used in this thesis to search for other dark matter signal models as well as other new physics processes. Run 3 at the LHC will increase the amount of available data further, which increases the possibility of new discoveries. Another aspect that may be useful to study further is how neural networks and other machine learning methods may best be used in particle physics. Although the main aspects of it are understood, it may be possible to make breakthroughs in order for them to remove a significantly larger amount of background. However, there will likely be a limit to how precise they can become, as some of the background events for most models will be very similar to the signal.

References

[1] Andreas Middelthon. Preparation of proton-proton collision data for an ML-based search for a new dark matter aware gauge boson Z' at the LHC with the ATLAS detector. Norwegian University of Science and Technology, 2022.

[2] Edvige Corbelli and Paolo Salucci. The Extended Rotation Curve and the Dark Matter Halo of M33. *Mon. Not. Roy. Astron. Soc.*, 311:441–447, 2000.

[3] Virginia Trimble. Existence and nature of dark matter in the universe. *Annual review of astronomy and astrophysics*, 25(1):425–472, 1987.

[4] Steven W. Allen, August E. Evrard, and Adam B. Mantz. Cosmological parameters from observations of galaxy clusters. *Annual Review of Astronomy and Astrophysics*, 49(1):409–470, sep 2011.

[5] J. R. Espinosa, D. Racco, and A. Riotto. Cosmological Signature of the Standard Model Higgs Vacuum Instability: Primordial Black Holes as Dark Matter. *Phys. Rev. Lett.*, 120(12):121301, 2018.

[6] George F. Chapline and Paul H. Frampton. A new direction for dark matter research: intermediate-mass compact halo objects. *Journal of Cosmology and Astroparticle Physics*, 2016(11):042, nov 2016.

[7] Gianfranco Bertone, Dan Hooper, and Joseph Silk. Particle dark matter: evidence, candidates and constraints. *Physics Reports*, 405(5):279–390, 2005.

[8] Edward A. Baltz, Marco Battaglia, Michael E. Peskin, and Tommer Wizansky. Determination of dark matter properties at high-energy colliders. *Phys. Rev. D*, 74:103521, Nov 2006.

[9] Stefano Profumo. *An introduction to particle dark matter*. World Scientific Publishing Company, 2017.

[10] Leszek Roszkowski, Enrico Maria Sessolo, and Sebastian Trojanowski. Wimp dark matter candidates and searches—current status and future prospects. *Reports on Progress in Physics*, 81(6):066201, may 2018.

[11] Francis Halzen and Alan D Martin. *Quark & Leptons: An introductory course in modern particle physics*. John Wiley & Sons, 2008.

[12] Gregg Jaeger. Exchange forces in particle physics. *Foundations of Physics*, 51(1):1–31, 2021.

[13] Mark Thomson. *Modern particle physics*. Cambridge University Press, 2013.

[14] David Griffiths. *Introduction to elementary particles*. John Wiley & Sons, 2020.

[15] ATLAS Collaboration. Observation of a new particle in the search for the standard model higgs boson with the ATLAS detector at the LHC. *Physics Letters B*, 716(1):1–29, sep 2012.

[16] W Noel Cottingham and Derek A Greenwood. *An introduction to the standard model of particle physics*. Cambridge university press, 2007.

[17] Antonio Pich. The Standard Model of Electroweak Interactions. In *2010 European School of High Energy Physics*, pages 1–50, 1 2012.

[18] Walter Greiner, Stefan Schramm, and Eckart Stein. *Quantum chromodynamics*. Springer Science & Business Media, 2007.

[19] Matthew Robinson. *Symmetry and the standard model*. Springer, 2011.

[20] CMS Collaboration. Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC. *Physics Letters B*, 716(1):30–61, sep 2012.

[21] Wikipedia. Standard model of elementary particles. https://en.wikipedia.org/wiki/File:Standard_Model_of_Elementary_Particles.svg, sep 2019. Accessed: 7.12.2022.

[22] Herbert W Hamber. *Quantum gravitation: The Feynman path integral approach*. Springer Science & Business Media, 2008.

[23] Q. R. Ahmad et al. Measurement of the rate of $\nu_e + d \rightarrow p + p + e^-$ interactions produced by ^{8}B solar neutrinos at the Sudbury Neutrino Observatory. *Phys. Rev. Lett.*, 87:071301, 2001.

[24] Y. Fukuda et al. Evidence for oscillation of atmospheric neutrinos. *Phys. Rev. Lett.*, 81:1562–1567, 1998.

[25] C. D. Froggatt. The Hierarchy problem and an exotic bound state. In *10th International Symposium on Particles, Strings and Cosmology (PASCOS 04 and Pran Nath Fest)*, pages 325–334, 12 2004.

[26] Nima Arkani-Hamed, Savas Dimopoulos, and Gia Dvali. The hierarchy problem and new dimensions at a millimeter. *Physics Letters B*, 429(3-4):263–272, 1998.

[27] Tim et al. Schrabback. Evidence of the accelerated expansion of the universe from weak lensing tomography with cosmos. *Astronomy & Astrophysics*, 516:A63, 2010.

[28] Pierre Astier and Reynald Pain. Observational evidence of the accelerated expansion of the universe. *Comptes Rendus Physique*, 13(6-7):521–538, 2012.

[29] Scott Dodelson. *Modern cosmology*. Elsevier, 2003.

[30] Salma Alrasheed. *Principles of mechanics: Fundamental university physics*. Springer Nature, 2019.

[31] Eleni Skorda. *Search for Dark Matter in events with missing transverse momentum and a Higgs boson decaying into bottom quarks with the ATLAS detector*. Lund University, 2022.

[32] JA Sellwood and RH Sanders. A maximum disc model for the galaxy. *Monthly Notices of the Royal Astronomical Society*, 233(3):611–620, 1988.

[33] Katherine Freese. Status of dark matter in the universe. *International Journal of Modern Physics D*, 26(06):1730012, 2017.

[34] Peter AR Ade, Nabila Aghanim, MIR Alves, Charmaine Armitage-Caplan, M Arnaud, M Ashdown, F Atrio-Barandela, J Aumont, H Aussel, C Baccigalupi, et al. Planck 2013 results. I. Overview of products and scientific results. *Astronomy & Astrophysics*, 571:A1, 2014.

[35] Planck Collaboration. Planck 2015 results. XIII. Cosmological parameters. *Astronomy &amp Astrophysics*, 594:A13, sep 2016.

[36] C. Skordis, D. F. Mota, P. G. Ferreira, and C. Bœhm. Large scale structure in bekenstein's theory of relativistic modified newtonian dynamics. *Physical Review Letters*, 96(1), jan 2006.

[37] Planck collaboration. Planck 2018 results. VI. Cosmological parameters. *Astronomy &amp Astrophysics*, 641:A6, sep 2020.

[38] Frank Wilczek. Problem of strong P and T invariance in the presence of instantons. *Physical Review Letters*, 40(5):279, 1978.

[39] John Preskill, Mark B Wise, and Frank Wilczek. Cosmology of the invisible axion. *Physics Letters B*, 120(1-3):127–132, 1983.

[40] Roberto D Peccei and Helen R Quinn. CP conservation in the presence of pseudoparticles. *Physical Review Letters*, 38(25):1440, 1977.

[41] Roberto D Peccei and Helen R Quinn. Constraints imposed by CP conservation in the presence of pseudoparticles. *Physical Review D*, 16(6):1791, 1977.

[42] Scott Dodelson and Lawrence M. Widrow. Sterile neutrinos as dark matter. *Physical Review Letters*, 72(1):17–20, jan 1994.

[43] Glenn D Starkman, Andrew Gould, Rahim Esmailzadeh, and Savas Dimopoulos. Opening the window on strongly interacting dark matter. *Physical Review D*, 41(12):3594, 1990.

[44] David N Spergel and Paul J Steinhardt. Observational evidence for self-interacting cold dark matter. *Physical review letters*, 84(17):3760, 2000.

[45] ATLAS. LHC Run 3: Physics at record energy starts tomorrow. https://atlas.cern/Updates/Press-Statement/LHC-Run3-Starts. Accessed: 17.12.2022.

[46] ATLAS. ATLAS event display of top-pair production in 13.6 tev collisions during run 3. https://cds.cern.ch/record/2842591. Accessed: 17.12.2022.

[47] R et al. Bruce. LHC Run 2: Results and challenges. 57th ICFA Advanced Beam Dynamics Workshop on High-Intensity and High-Brightness Hadron Beams, jul 2016.

[48] Julie Haffner. The CERN accelerator complex. https://cds.cern.ch/images/OPEN-PHO-ACCEL-2013-056-1, October 2013.

[49] ATLAS. ATLAS Fact Sheets. https://atlas.cern/Resources/Fact-sheets. Accessed: 17.12.2022.

[50] G. et al. Aad. The ATLAS Experiment at the CERN Large Hadron Collider. *JINST*, 3:S08003, 2008.

[51] L Pontecorvo. The ATLAS muon spectrometer. *The European Physical Journal C-Particles and Fields*, 34(1):s117–s128, 2004.

[52] A Ruiz Martínez, ATLAS Collaboration, et al. The run-2 ATLAS trigger system. In *Journal of Physics: Conference Series*, volume 762, page 012003. IOP Publishing, 2016.

[53] Georges Aad et al. ATLAS data quality operations and performance for 2015–2018 data-taking. *JINST*, 15(04):P04003, 2020.

[54] P Puzo. ATLAS calorimetry. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 494(1-3):340–345, 2002.

[55] JT Boyd. LHC Run-2 and future prospects. *arXiv preprint arXiv:2001.04370*, 2020.

[56] P Grafström. Lifetime, cross-sections and activation. 2007.

[57] Werner Herr and Bruno Muratori. Concept of luminosity. 2006.

[58] Vladimir Shiltsev and V Lebedev. *Accelerator physics at the Tevatron collider*. Springer, 2014.

[59] Matthias Schott and Monica Dunford. Review of single vector boson production in pp collisions at $\sqrt{s} = 7$ TeV. *Eur. Phys. J. C*, 74:2916, 2014.

[60] The CMS collaboration. Missing transverse energy performance of the CMS detector. *Journal of Instrumentation*, 6(09):P09001, sep 2011.

[61] Nathan Mirman, Yimin Wang, and James Alexander. Missing transverse energy significance at CMS. In *2nd Large Hadron Collider Physics Conference*, 9 2014.

[62] J. Beringer et al. Review of Particle Physics (RPP). *Phys. Rev. D*, 86:010001, 2012.

[63] Marcelo Autran, Kevin Bauer, Tongyan Lin, and Daniel Whiteson. Searches for dark matter in events with a resonance and missing transverse energy. *Phys. Rev. D*, 92(3):035007, 2015.

[64] Aaboud et al. Luminosity determination in pp collisions at $\sqrt{s}$= 8 TeV using the ATLAS detector at the LHC. *The European Physical Journal C*, 76(12):1–45, 2016.

[65] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[66] Michael A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015. http://neuralnetworksanddeeplearning.com/.

[67] Akhilesh A Waoo and Brijesh K Soni. Performance analysis of sigmoid and Relu activation functions in deep neural network. In *Intelligent Systems: Proceedings of SCIS 2021*, pages 39–52. Springer, 2021.

[68] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[69] T Gleisberg, S Hoeche, F Krauss, A Schaelicke, S Schumann, and J Winter. SHERPA 1. , a proof-of-concept version. *Journal of High Energy Physics*, 2004(02):056–056, feb 2004.

[70] Torbjörn Sjöstrand, Stephen Mrenna, and Peter Skands. A brief introduction to PYTHIA 8.1. *Computer Physics Communications*, 178(11):852–867, jun 2008.

[71] Johan Alwall, Michel Herquet, Fabio Maltoni, Olivier Mattelaer, and Tim Stelzer. Madgraph 5: going beyond. *Journal of High Energy Physics*, 2011(6):1–40, 2011.

[72] Stefano Frixione, Fabian Stoeckli, Paolo Torrielli, Bryan R Webber, and Chris D White. The MC@ NLO 4.0 event generator. *arXiv preprint arXiv:1010.0819*, 2010.

[73] Johan Alwall, R Frederix, S Frixione, V Hirschi, Fabio Maltoni, Olivier Mattelaer, H-S Shao, T Stelzer, P Torrielli, and M Zaro. 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*, 2014(7):1–157, 2014.

[74] Simone Alioli, Paolo Nason, Carlo Oleari, and Emanuele Re. A general framework for implementing NLO calculations in shower monte carlo programs: the POWHEG BOX. *Journal of High Energy Physics*, 2010(6):1–58, 2010.

[75] Stefano Frixione, Paolo Nason, and Carlo Oleari. Matching NLO QCD computations with Parton Shower simulations: the POWHEG method. *JHEP*, 11:070, 2007.

[76] Andreas Albert et al. Recommendations of the LHC Dark Matter Working Group: Comparing LHC searches for dark matter mediators in visible and invisible decay channels and calculations of the thermal relic density. *Phys. Dark Univ.*, 26:100377, 2019.

[77] Richard D Ball, Valerio Bertone, Stefano Carrazza, Christopher S Deans, Luigi Del Debbio, Stefano Forte, Alberto Guffanti, Nathan P Hartland, José I Latorre, Juan Rojo, et al. Parton distributions for the LHC Run II. *Journal of High Energy Physics*, 2015(4):1–148, 2015.

[78] Rene Brun and Fons Rademakers. ROOT—an object oriented data analysis framework. *Nuclear instruments and methods in physics research section A: accelerators, spectrometers, detectors and associated equipment*, 389(1-2):81–86, 1997.

[79] J. D. Hunter. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[80] Meta AI. Pytorch. https://ai.facebook.com/tools/pytorch/. Accessed: 30.05.2022.

[81] R. Harris Charles and et al. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[82] Ronald E. Walpole, Sharon L. Myers, Raymond Myers, and Keying Ye. *Probability Statistics for Engineers Scientists*. Pearson Education Limited, 2021.

[83] Eilam Gross. Practical statistics for high energy physics. *CERN yellow reports: school proceedings*, 4(0):165, 2017.

[84] Alexander L Read. Presentation of search results: the CLs technique. *Journal of Physics G: Nuclear and Particle Physics*, 28(10):2693, 2002.

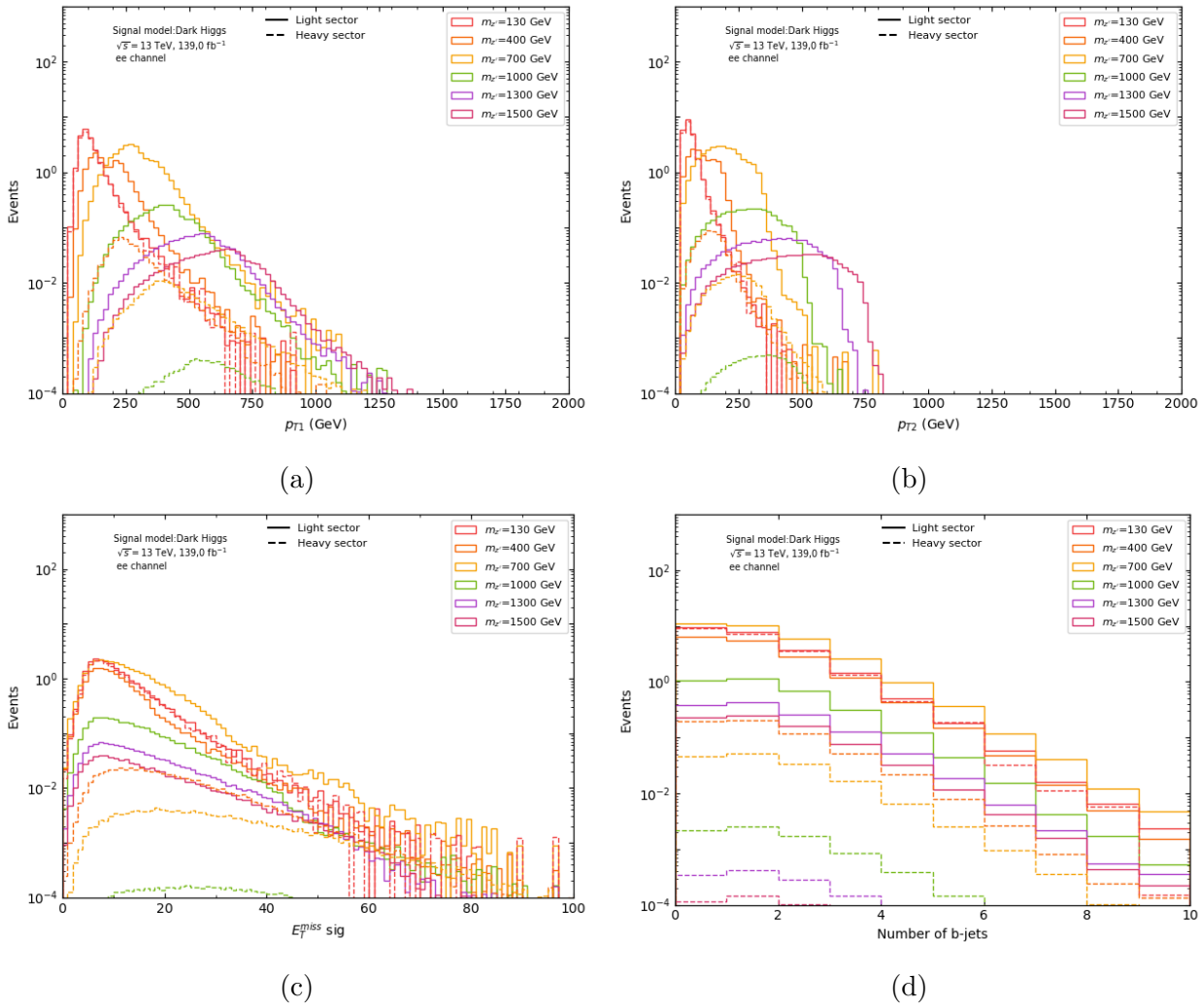
[85] Glen Cowan, Kyle Cranmer, Eilam Gross, and Ofer Vitells. Asymptotic formulae for likelihood-based tests of new physics. *The European Physical Journal C*, 71:1–19, 2011.

A Signal model distributions

Below are the dark Higgs and light vector signal distributions (with precuts) for p_{T1}, p_{T2}, $E_T^{miss.sig}$, the number of b-jets, $\Delta\phi_{E_T^{miss},ll}$, $\Delta\phi_{l,l}$, η and H_T in both the ee and $\mu\mu$ channels.

A.1 Dark Higgs model

A.1.1 ee channel



(a) (b)

(c) (d)

Figure A.1: Dark higgs model signal distributions for a) p_{T1}, b) p_{T2}, c) $E_T^{miss,sig}$ and d) number of b-jets in the ee channel with precuts.

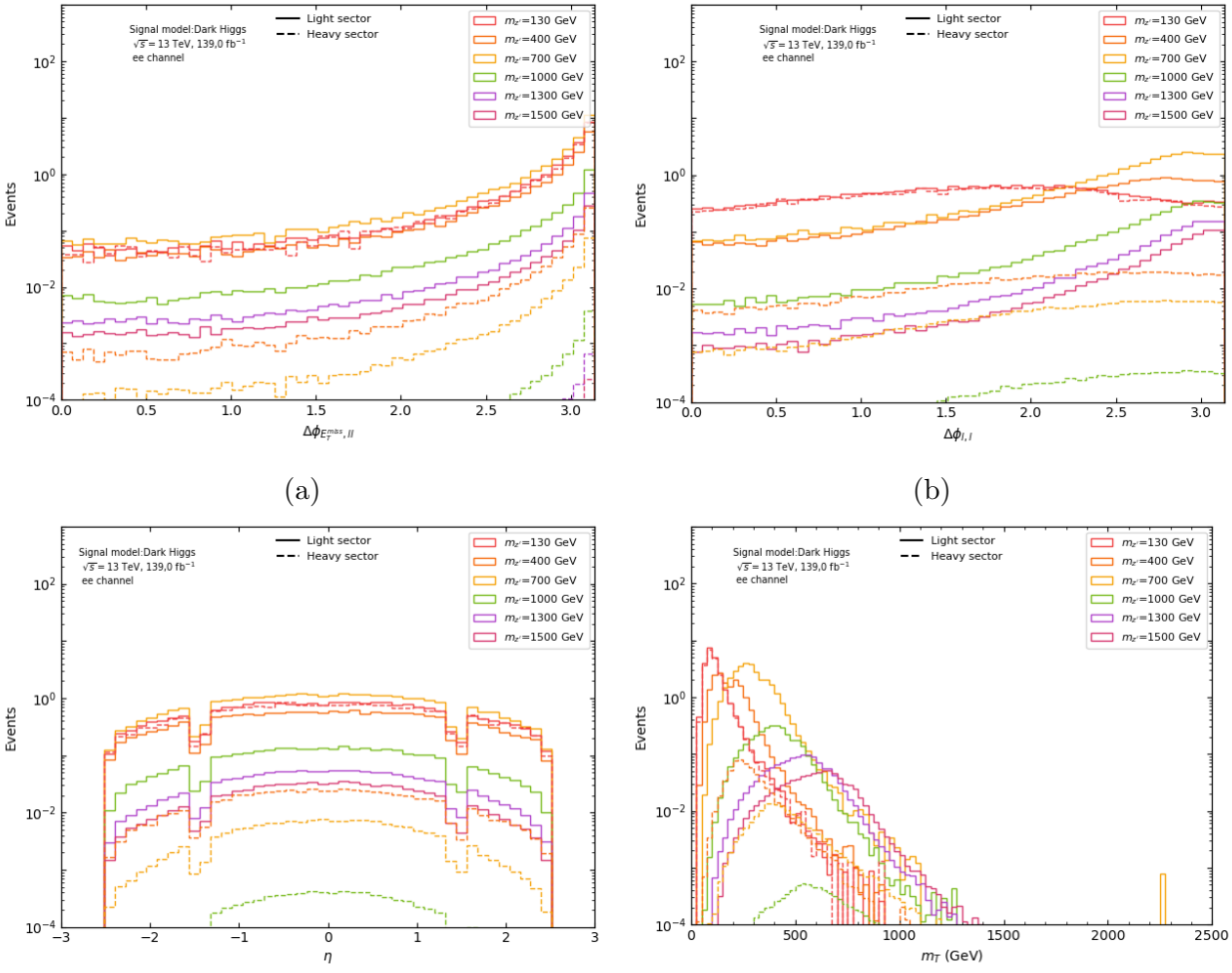


Figure A.2: Dark higgs model signal distributions for a) $\Delta\phi_{E_T^{miss},ll}$, b) $\Delta\phi_{l,l}$, c) η and d) m_T in the ee channel with precuts.

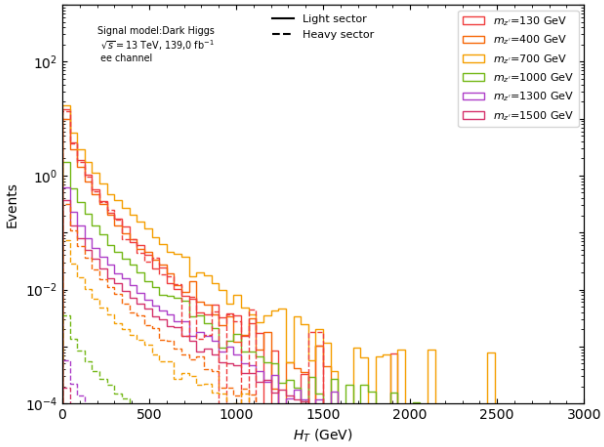


Figure A.3: Dark Higgs model signal distribution for H_T with in the *ee* channel with precuts.

A.1.2 $\mu\mu$ channel

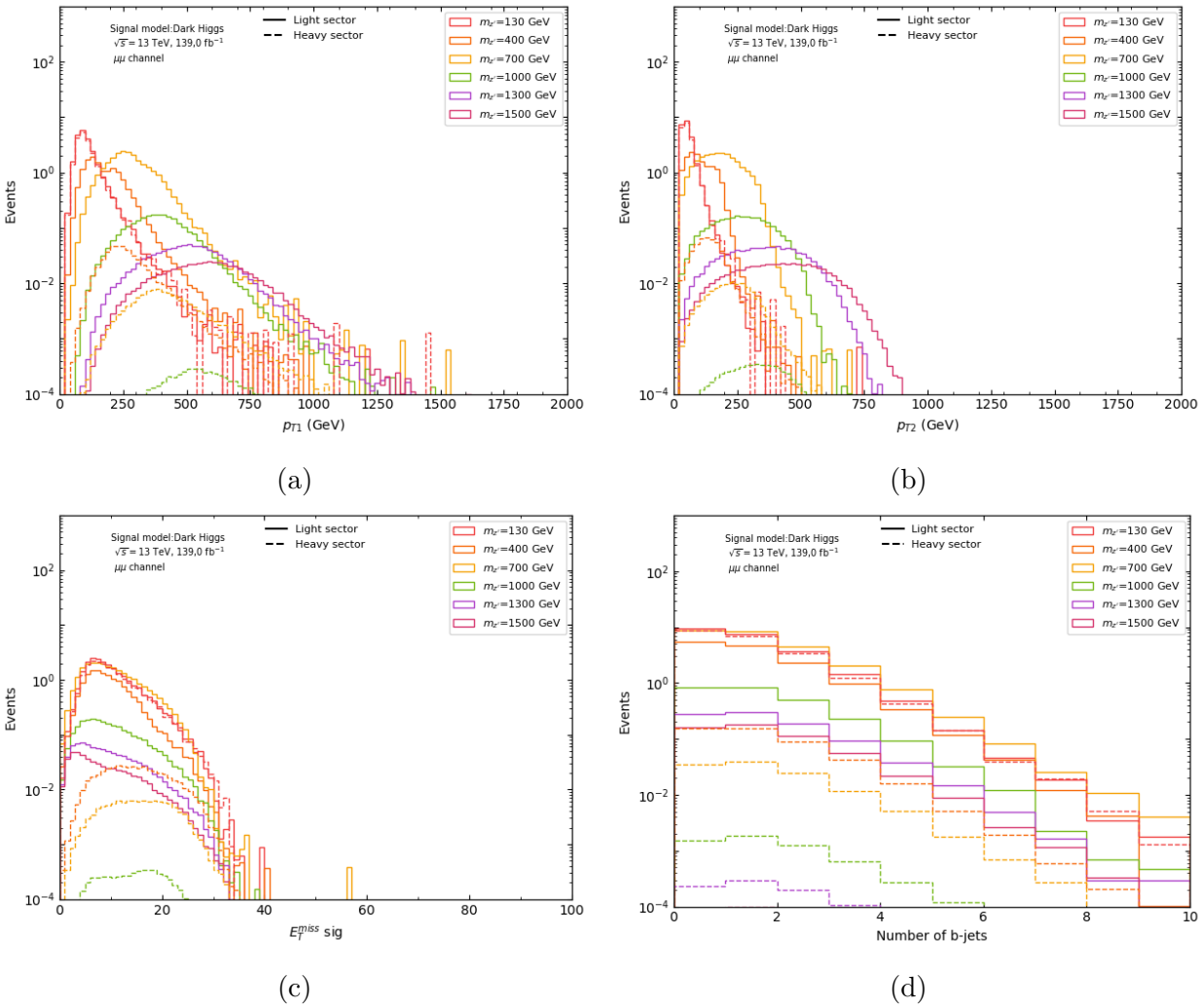


Figure A.4: Dark higgs model signal distributions for a) p_{T1}, b) p_{T2}, c) $E_T^{miss,sig}$ and d) number of b-jets in the $\mu\mu$ channel with precuts.

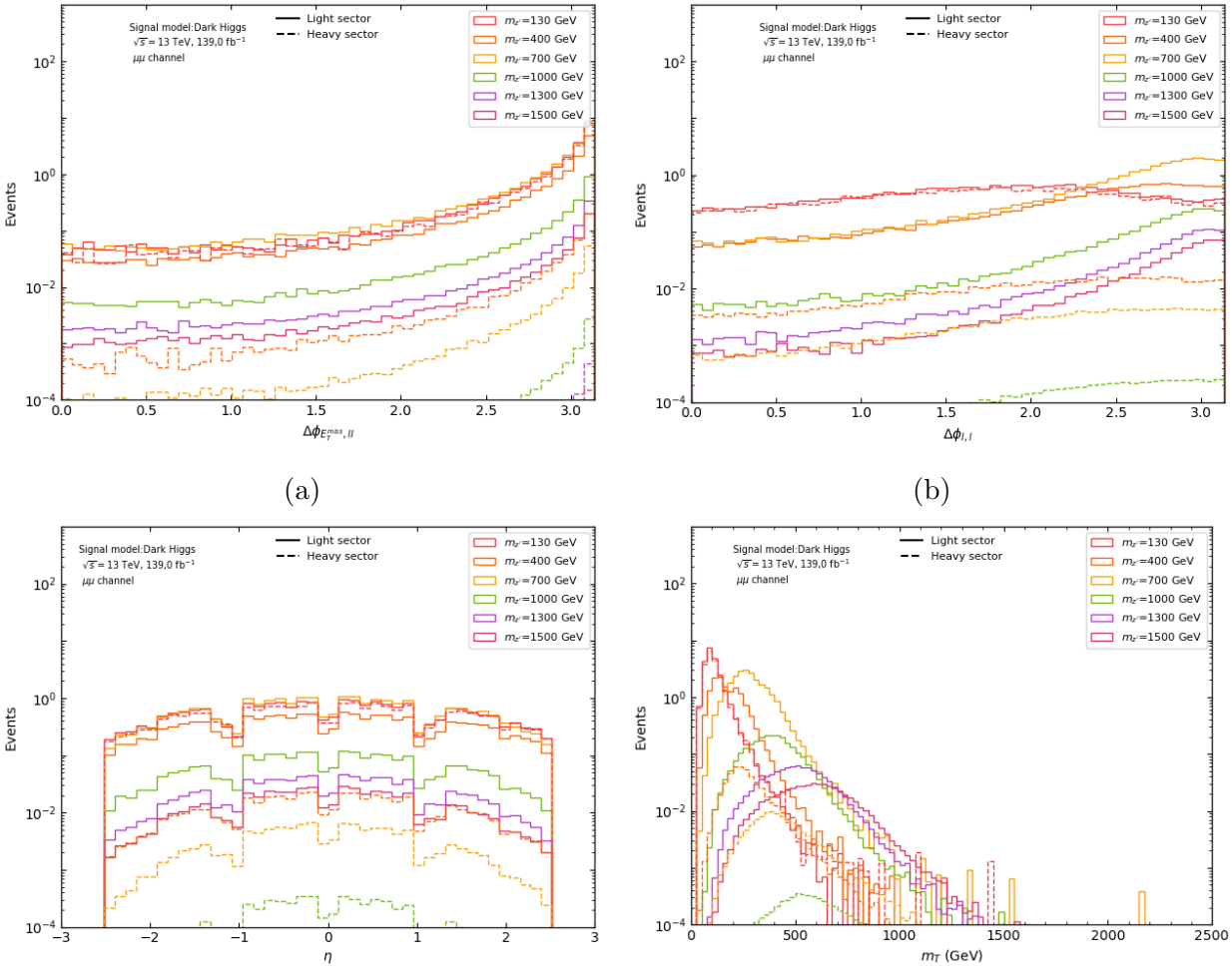


Figure A.5: Dark higgs model signal distributions for a) $\Delta\phi_{E_T^{miss},ll}$, b) $\Delta\phi_{l,l}$, c) η and d) m_T in the $\mu\mu$ channel with precuts.

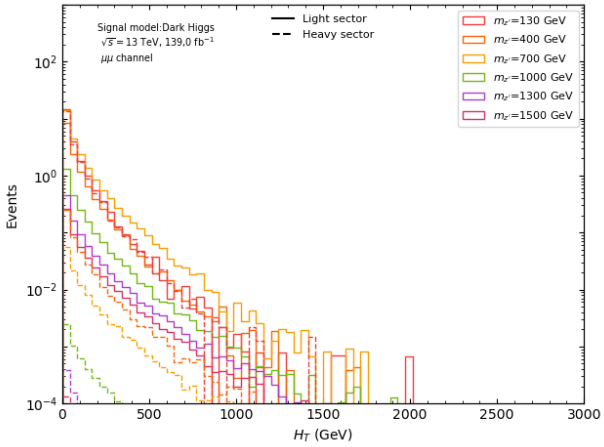


Figure A.6: Dark Higgs model signal distribution for H_T with in the $\mu\mu$ channel with precuts.

A.2 Light vector model

A.2.1 *ee* channel

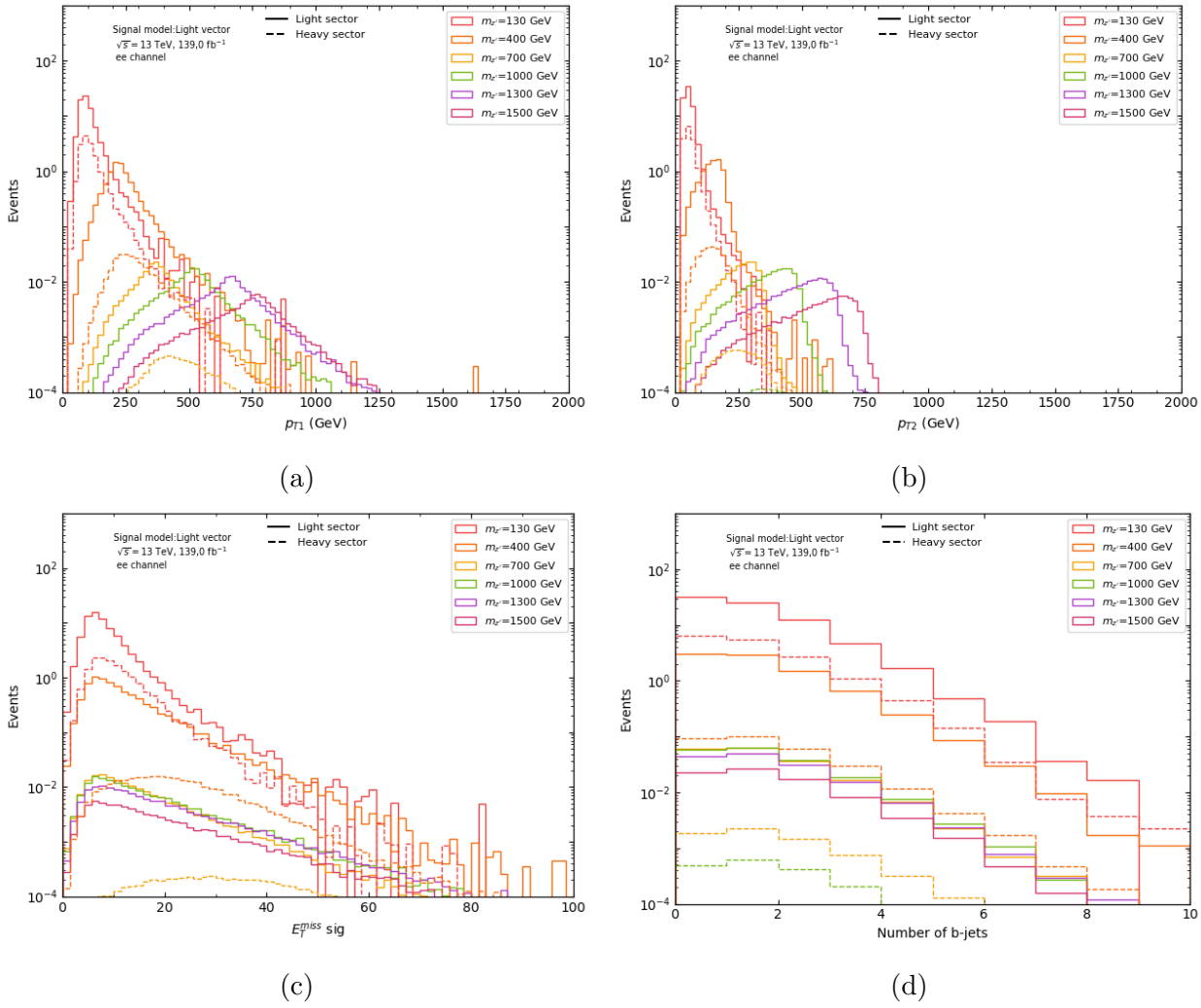


Figure A.7: Light vector model signal distributions for a) p_{T1}, b) p_{T2}, c) $E_T^{miss,sig}$ and d) number of b-jets in the ee channel with precuts.

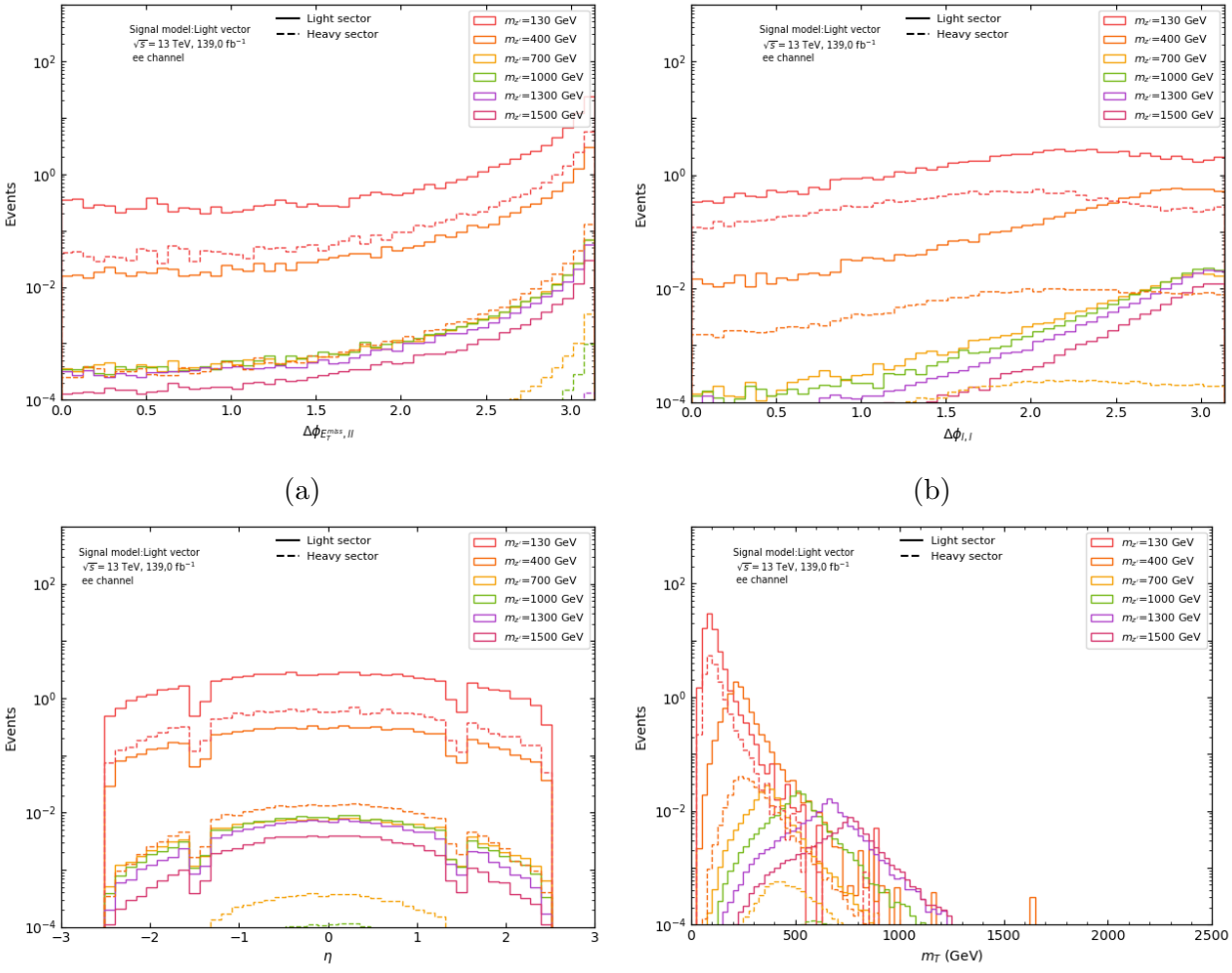


Figure A.8: Light vector model signal distributions for a) $\Delta\phi_{E_T^{miss},ll}$, b) $\Delta\phi_{l,l}$, c) η and d) m_T in the *ee* channel with precuts.

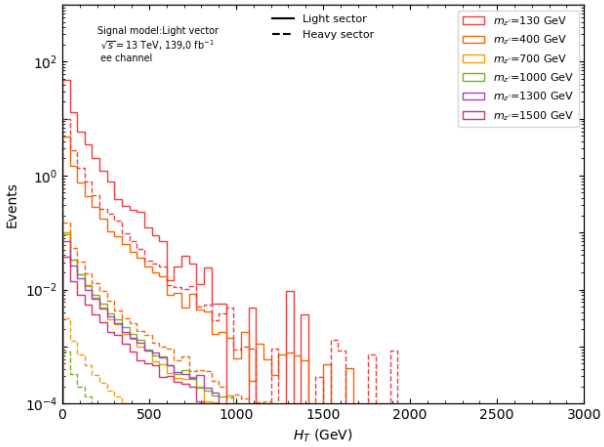


Figure A.9: Light vector model signal distribution for H_T with in the ee channel with precuts.

A.2.2 $\mu\mu$ channel

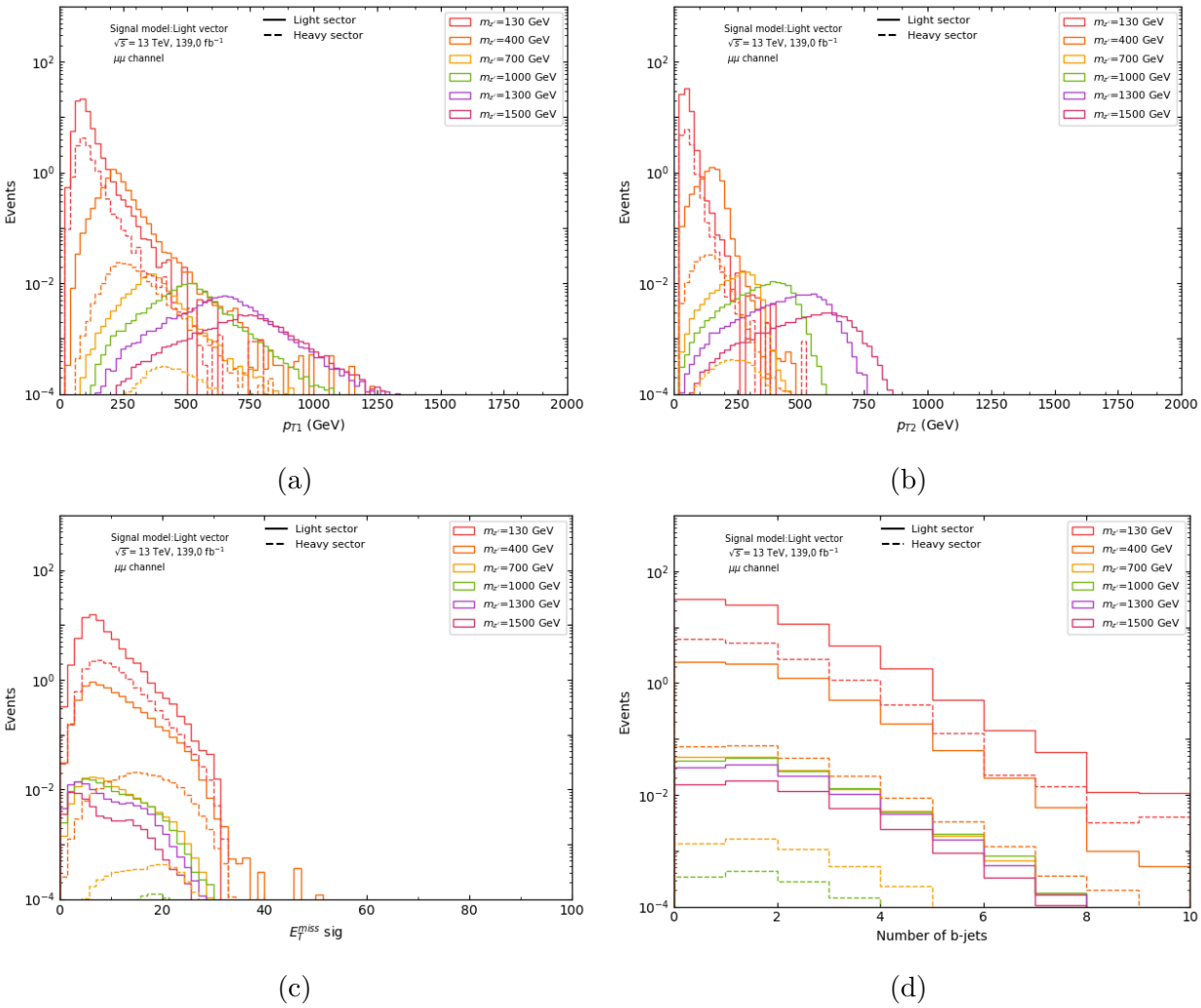


Figure A.10: Light vector model signal distributions for a) p_{T1}, b) p_{T2}, c) $E_T^{miss,sig}$ and d) number of b-jets in the $\mu\mu$ channel with precuts.

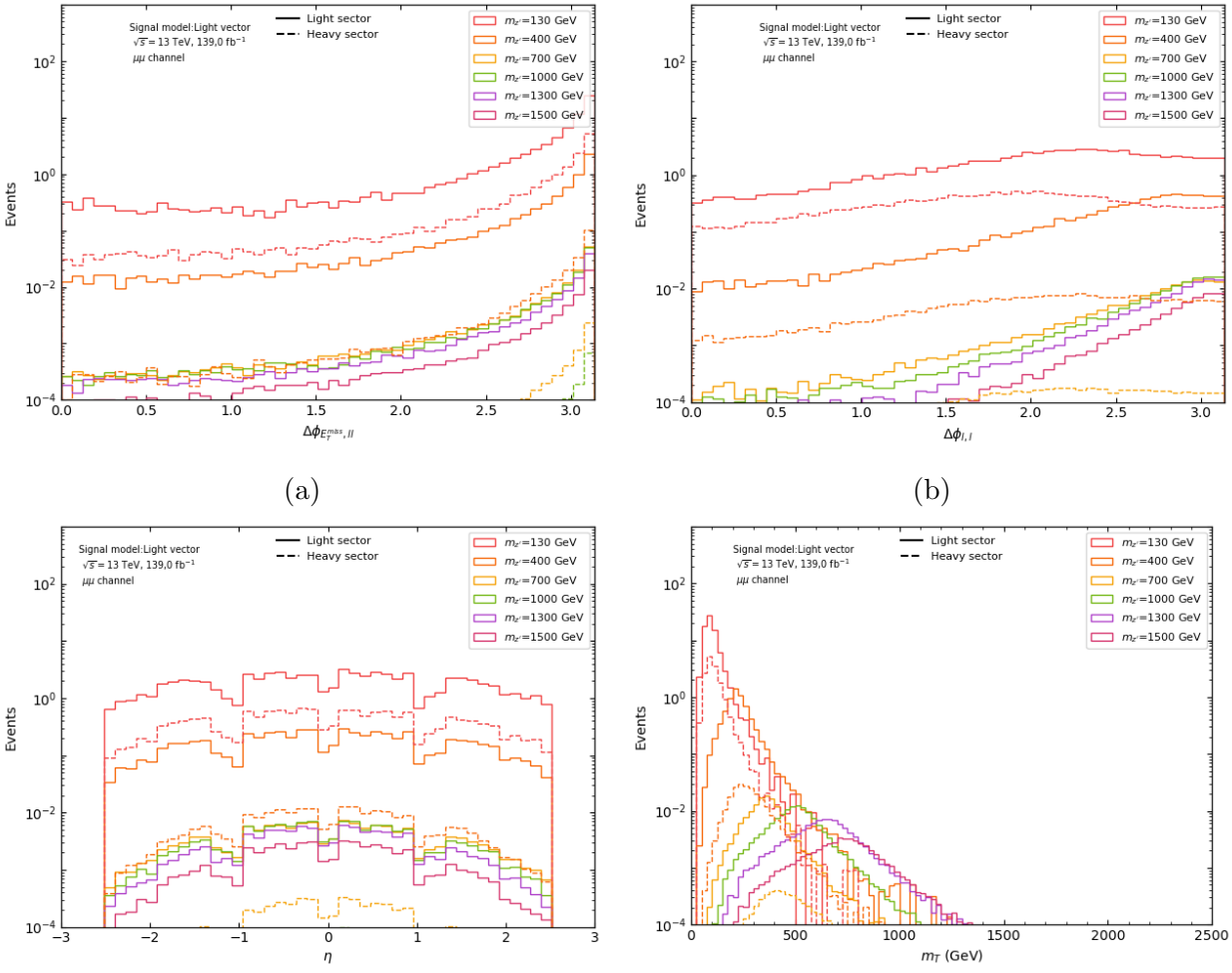


Figure A.11: Light vector model signal distributions for a) $\Delta\phi_{E_T^{miss},ll}$, b) $\Delta\phi_{l,l}$, c) η and d) m_T in the $\mu\mu$ channel with precuts.

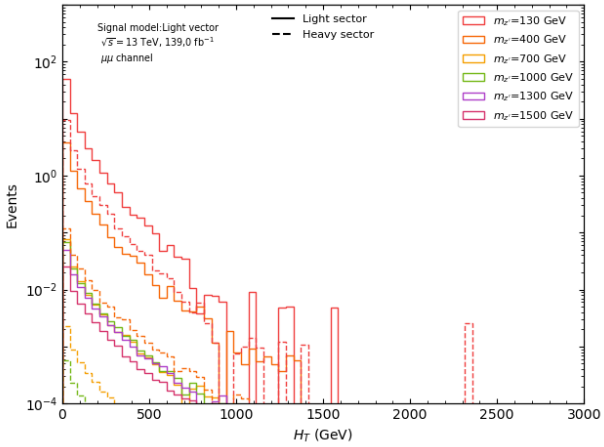
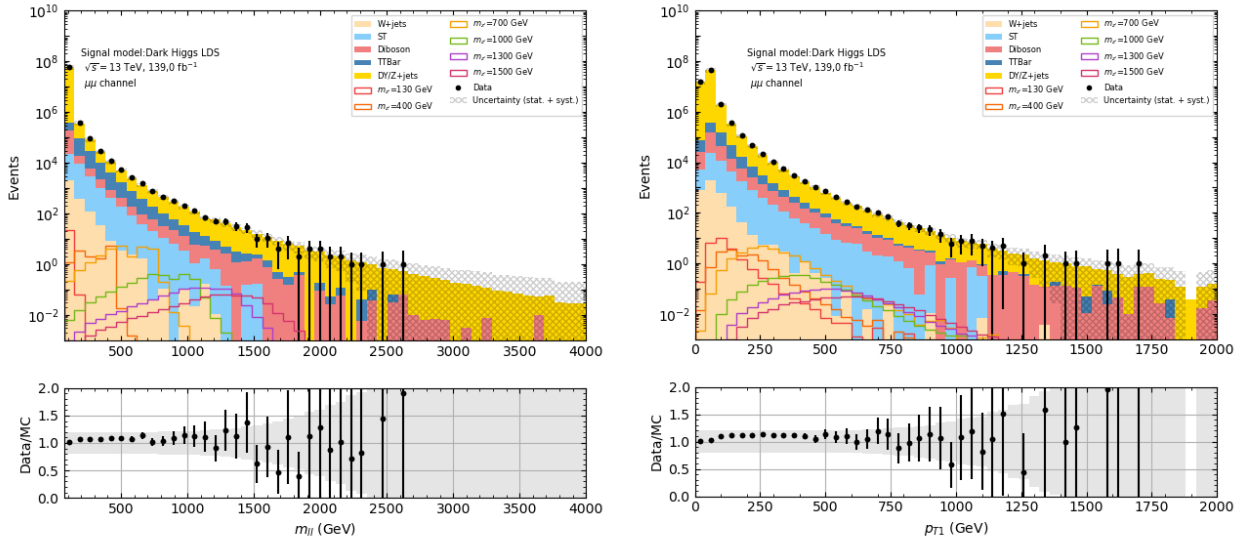


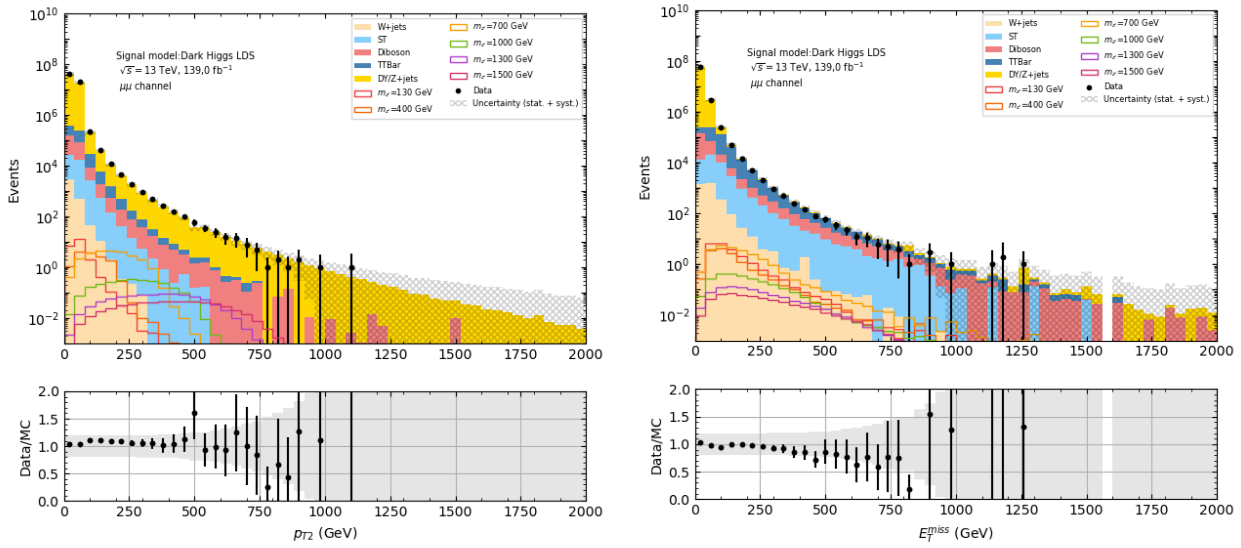
Figure A.12: Light vector model signal distribution for H_T with in the $\mu\mu$ channel with precuts.

B Comparison of MC and data for $\mu\mu$ channel

Below are plots of MC background alongside real data after precuts in the muon channel for the variables that are considered for the ML analysis. These are plotted alongside the simulated contributions from the dark Higgs LDS simulations for reference. The ratio of the data to the total MC background is plotted below the histograms, where the grey bands show the sum of the statistical and assumed systematic uncertainties of 20%.

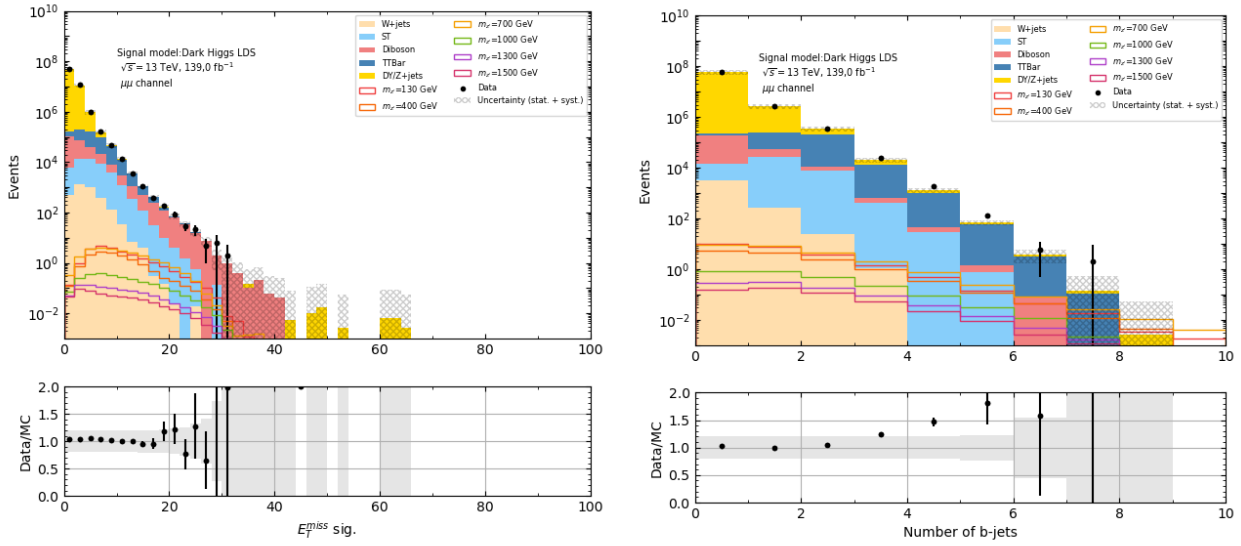


(a) (b)

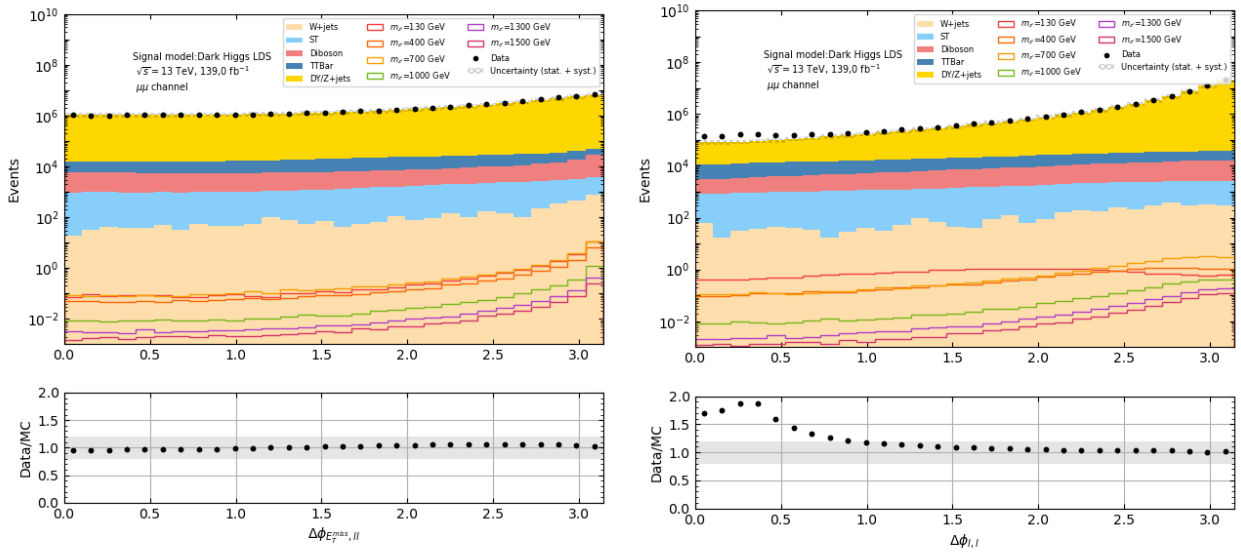


(c) (d)

Figure B.1: Muon channel distributions for a) m_{ll}, b) p_{T1}, c) p_{T2} and d) E_T^{miss} with precuts. Data is shown along with MC background and MC signals in the dark Higgs model (LDS).

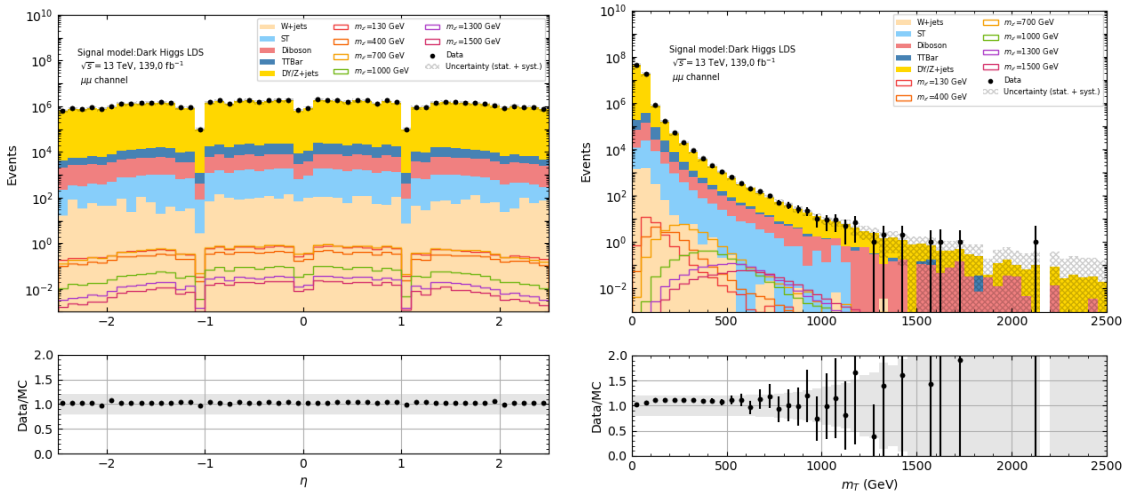


(a) (b)

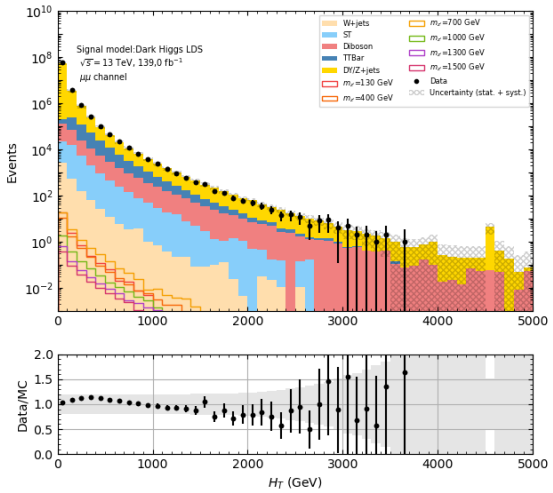


(c) (d)

Figure B.2: Muon channel distributions for a) $E_T^{miss,sig}$, b) number of b-tagged jets, c) $\Delta\phi_{E_T^{miss},ll}$ and d) $\Delta\phi_{l,l}$ with precuts. Data is shown along with MC background and MC signals in the dark Higgs model (LDS).



(a) (b)

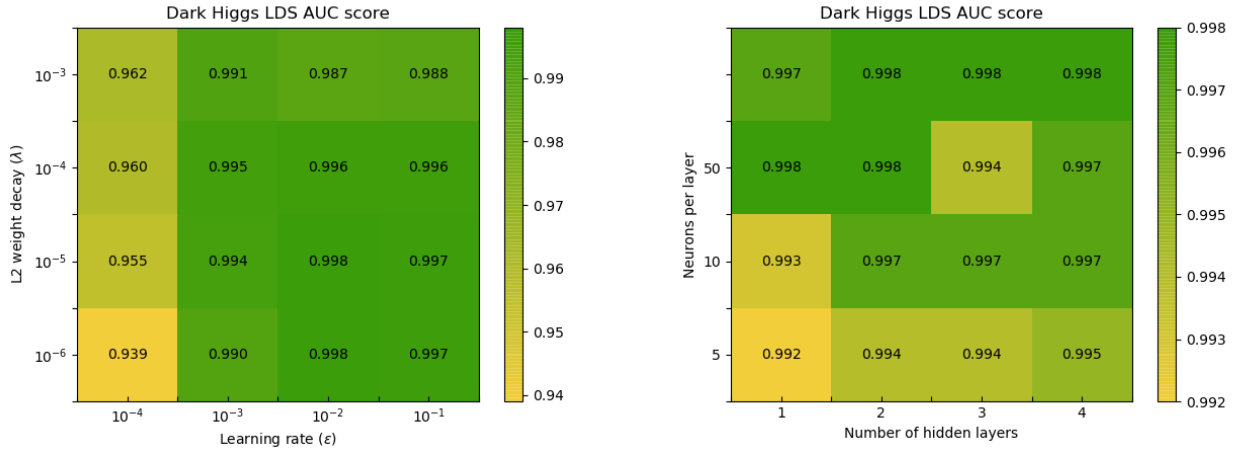


(c)

Figure B.3: Muon channel distributions for a) η, b) m_T c) H_T with precuts. Data is shown along with MC background and MC signals in the dark Higgs model (LDS).

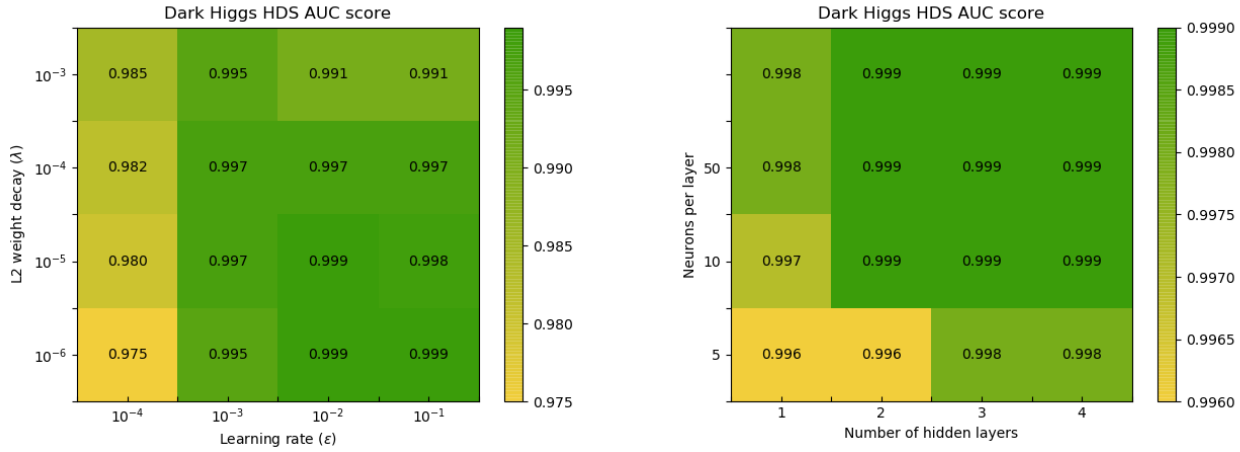
C Grid searches using AUC

Below are grid searches for the optimal values of the L_2 weight decay (λ), learning rate ϵ, the number of hidden layers and neurons per layer. These are used for optimizing the hyperparameters and network architecture.



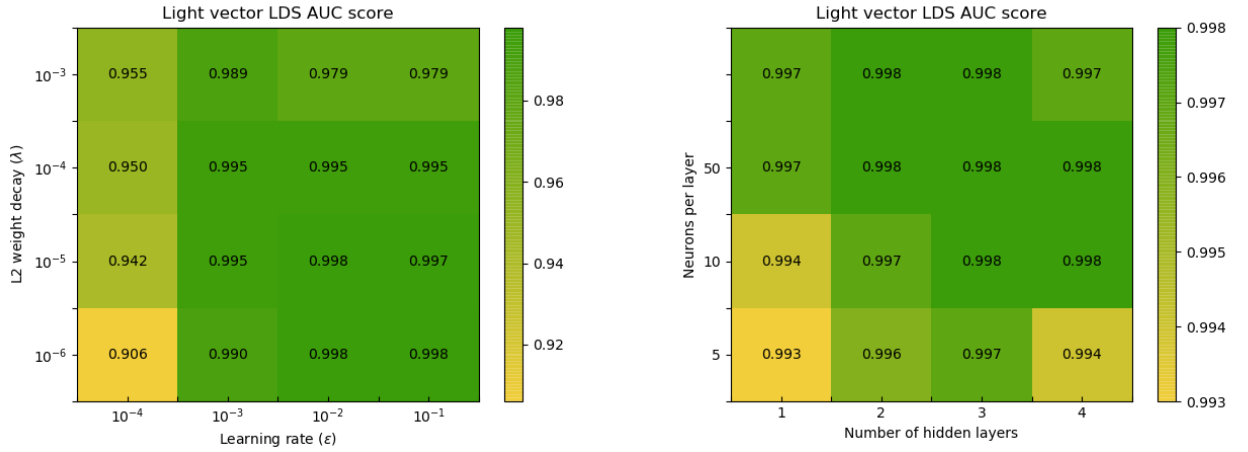
(a) (b)

Figure C.1: Grid searches for optimal hyperparameters a) $L2$ weight decay (λ) and learning rate (ϵ) and b) number of hidden layers and neurons per layer for ML training on the dark Higgs LDS using AUC as measure.



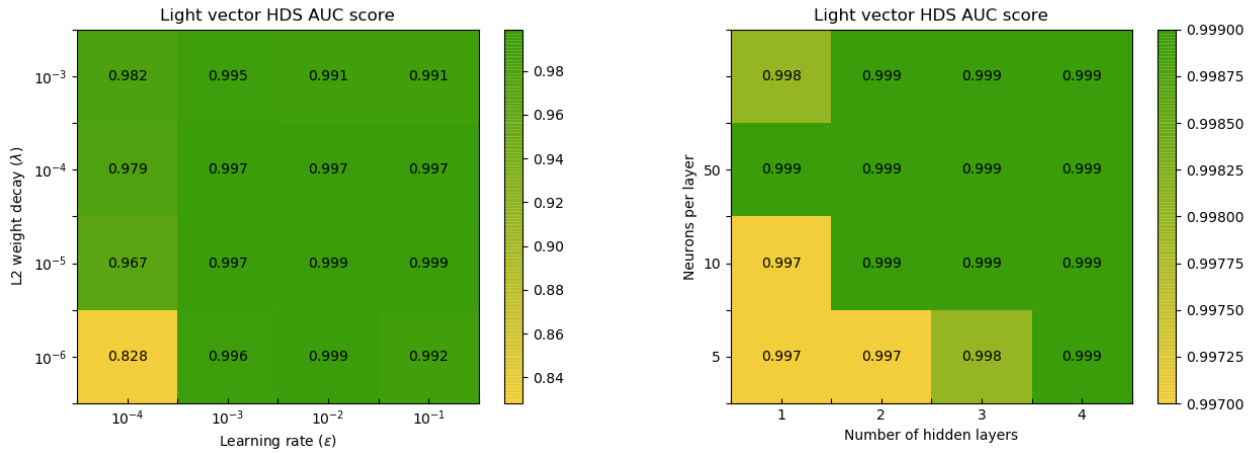
(a) (b)

Figure C.2: Grid searches for optimal hyperparameters a) $L2$ weight decay (λ) and learning rate (ϵ) and b) number of hidden layers and neurons per layer for ML training on the dark Higgs HDS using AUC as measure.



(a) (b)

Figure C.3: Grid searches for optimal hyperparameters a) $L2$ weight decay (λ) and learning rate (ϵ) and b) number of hidden layers and neurons per layer for ML training on the light vector LDS using AUC as measure.

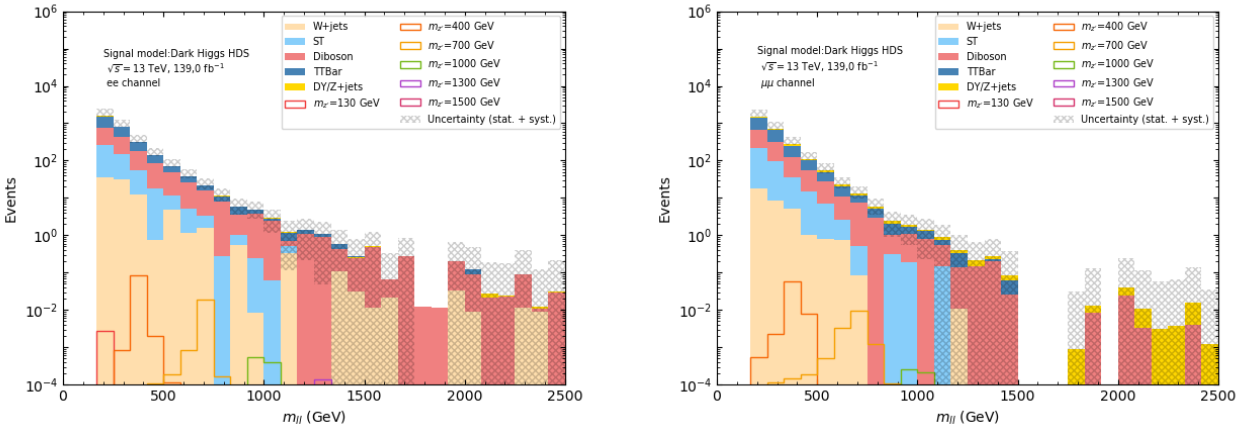


(a) (b)

Figure C.4: Grid searches for optimal hyperparameters a) $L2$ weight decay (λ) and learning rate (ϵ) and b) number of hidden layers and neurons per layer for ML training on the light vector HDS using AUC as measure.

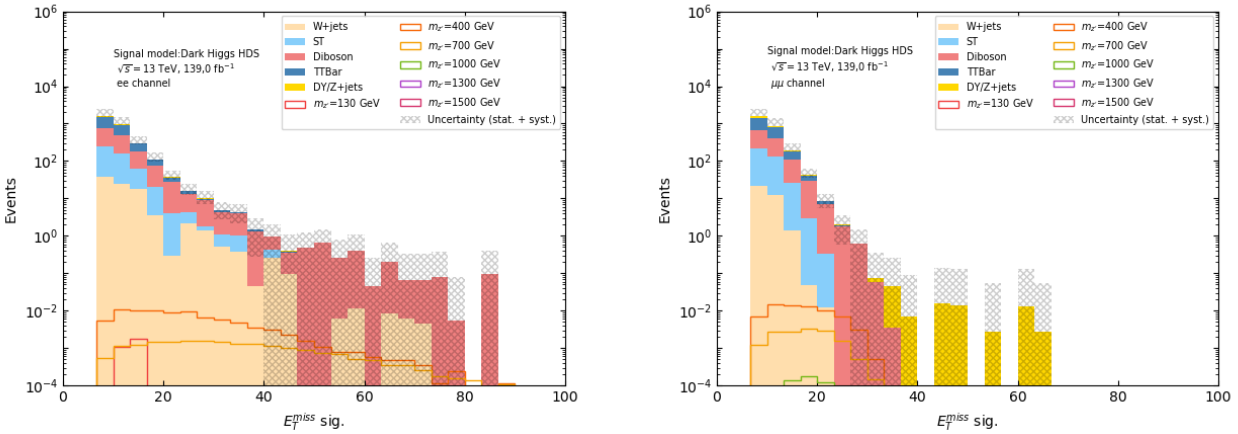
D Dark Higgs HDS cut and count

In this section, we show the m_{ll} and $E_T^{miss,sig}$ signal regions in the the ee and $\mu\mu$ channels for the dark Higgs HDS, as well listing the expected significances of the signals.



(a) (b)

Figure D.1: Dark Higgs HDS signal regions for m_{ll} in the a) ee and b) $\mu\mu$ channel.



(a) (b)

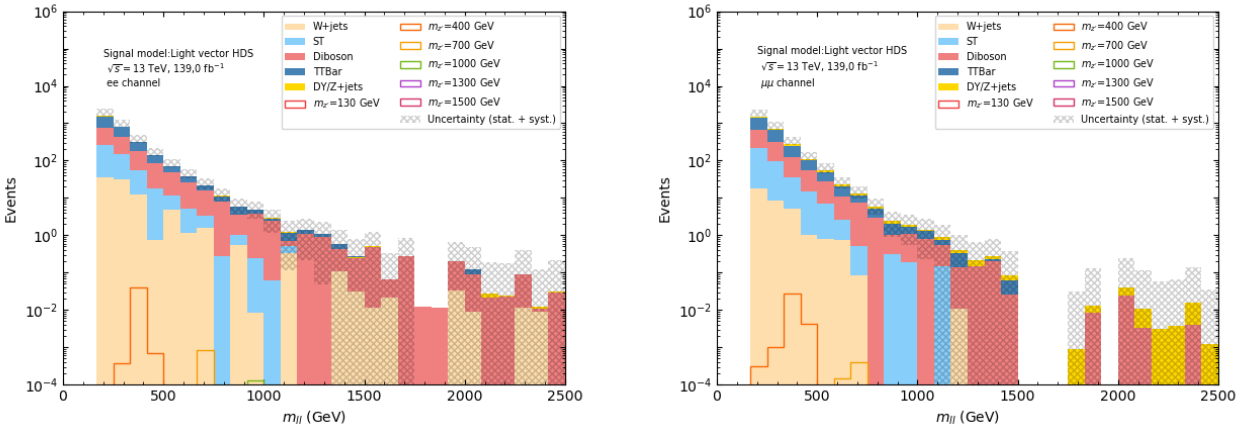
Figure D.2: Dark Higgs HDS signal regions for $E_T^{miss,sig}$ in the a) ee and b) $\mu\mu$ channel.

Dark Higgs HDS		
	ee channel	$\mu\mu$ channel
$m_{Z'}$ (GeV)	Expected significance (Z)	
130	$5.3 \cdot 10^{-5}$	0
200	$1.9 \cdot 10^{-2}$	$1.7 \cdot 10^{-2}$
300	$6.4 \cdot 10^{-3}$	$5.2 \cdot 10^{-3}$
400	$1.6 \cdot 10^{-3}$	$1.4 \cdot 10^{-3}$
500	$6.9 \cdot 10^{-3}$	$5.6 \cdot 10^{-3}$
600	$2.5 \cdot 10^{-4}$	$2.0 \cdot 10^{-4}$
700	$3.9 \cdot 10^{-4}$	$2.9 \cdot 10^{-4}$
800	$5.6 \cdot 10^{-5}$	$4.4 \cdot 10^{-5}$
900	$3.1 \cdot 10^{-5}$	$2.2 \cdot 10^{-5}$
1000	$1.9 \cdot 10^{-5}$	$1.2 \cdot 10^{-5}$
1100	$9.2 \cdot 10^{-6}$	$6.0 \cdot 10^{-6}$
1200	$5.0 \cdot 10^{-6}$	$3.2 \cdot 10^{-6}$
1300	$2.9 \cdot 10^{-6}$	$1.5 \cdot 10^{-6}$
1400	$1.6 \cdot 10^{-6}$	$9.8 \cdot 10^{-7}$
1500	$1.1 \cdot 10^{-6}$	0

Table D.1: Expected significances for the dark Higgs HDS at different Z' masses in the ee and $\mu\mu$ channel using the cut and count method.

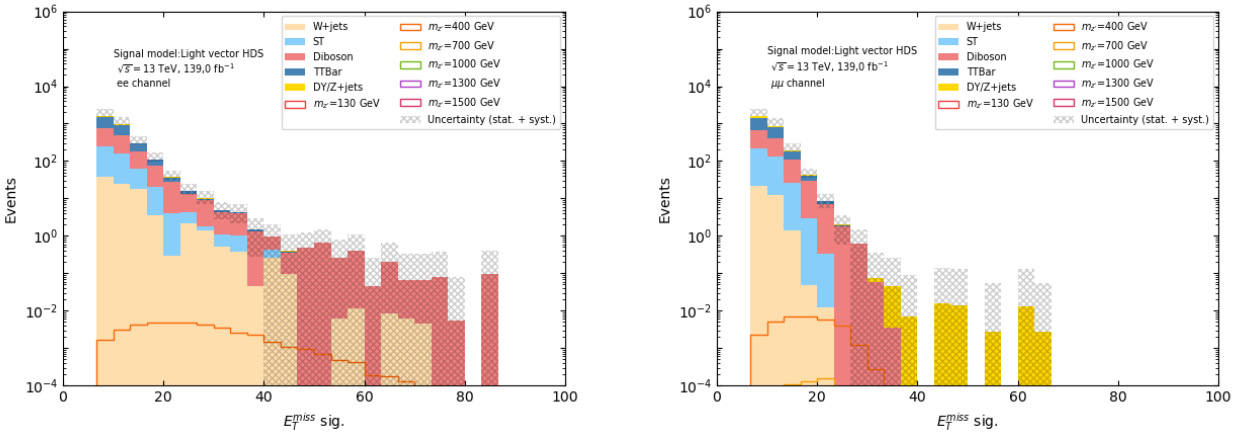
E Light vector HDS cut and count

In this section, we show the m_{ll} and $E_T^{miss,sig}$ signal regions in the the ee and $\mu\mu$ channels for the light vector HDS, as well listing the expected significances of the signals.



(a) (b)

Figure E.1: Light vector HDS signal regions for m_{ll} in the a) ee and b) $\mu\mu$ channel.



(a) (b)

Figure E.2: Light vector HDS signal regions for $E_T^{miss,sig}$ in the a) ee and b) $\mu\mu$ channel.

Light vector HDS		
$m_{Z'}$ (GeV)	Expected significance (Z)	
	ee channel	$\mu\mu$ channel
130	0	0
200	$1.3 \cdot 10^{-2}$	$1.1 \cdot 10^{-2}$
300	$3.4 \cdot 10^{-3}$	$2.9 \cdot 10^{-3}$
400	$7.8 \cdot 10^{-4}$	$6.4 \cdot 10^{-4}$
500	$3.1 \cdot 10^{-4}$	$2.4 \cdot 10^{-4}$
600	$1.0 \cdot 10^{-4}$	$7.7 \cdot 10^{-5}$
700	$1.7 \cdot 10^{-5}$	$1.2 \cdot 10^{-5}$
800	$1.9 \cdot 10^{-5}$	$1.4 \cdot 10^{-5}$
900	$8.6 \cdot 10^{-6}$	$6.3 \cdot 10^{-6}$
1000	$4.3 \cdot 10^{-6}$	$3.0 \cdot 10^{-6}$
1100	$2.2 \cdot 10^{-6}$	$1.3 \cdot 10^{-6}$
1200	$1.0 \cdot 10^{-6}$	$8.3 \cdot 10^{-7}$
1300	$6.1 \cdot 10^{-7}$	$4.9 \cdot 10^{-7}$
1400	$6.7 \cdot 10^{-7}$	$7.3 \cdot 10^{-7}$
1500	$8.2 \cdot 10^{-7}$	$5.9 \cdot 10^{-7}$

Table E.1: Expected significances for the light vector HDS at different Z' masses in the ee and $\mu\mu$ channel using the cut and count method.

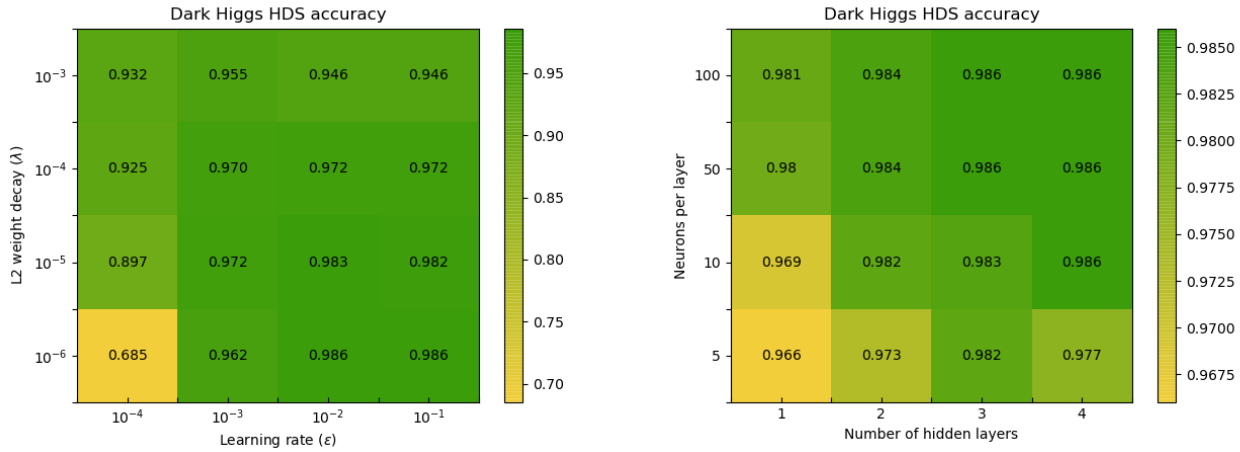
F Dark Higgs HDS ML analysis

In this section, we show the results of the ML analysis for the dark Higgs HDS, corresponding to the steps in the light dark sector analysis in section 7.6.

F.1 Hyperparameters

Dark Higgs HDS	
Hyperparameter	Value
Number of hidden layers	3
Neurons per layer	50
Learning rate (ϵ)	10^{-2}
$L2$ weight decay (λ)	10^{-6}
Epochs	50
Batch size	10% of training set
Exponential decay rate (ρ_1)	0.90
Exponential decay rate (ρ_2)	0.99
Stabilization constant (δ)	10^{-8}

Table F.1: Hyperparameters used for training the neural network on the dark Higgs HDS.



(a) (b)

Figure F.1: Grid searches for optimal hyperparameters a) $L2$ weight decay (λ) and learning rate (ϵ), and b) number of hidden layers and neurons per layer for ML training on the dark Higgs HDS using accuracy as measure.

F.2 Performance

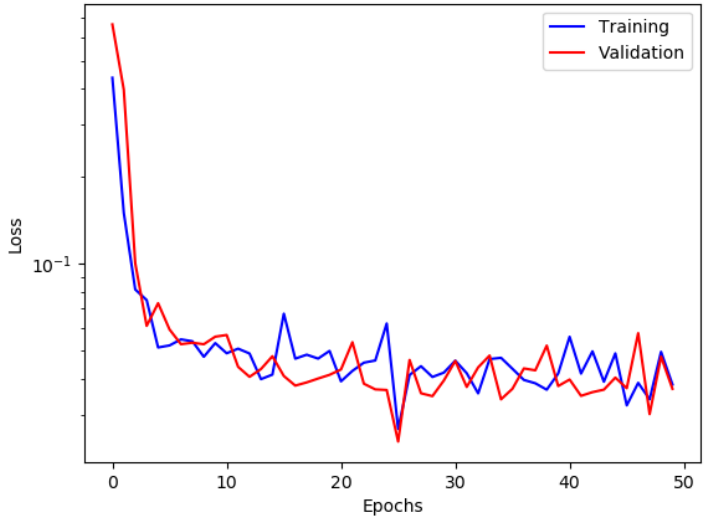
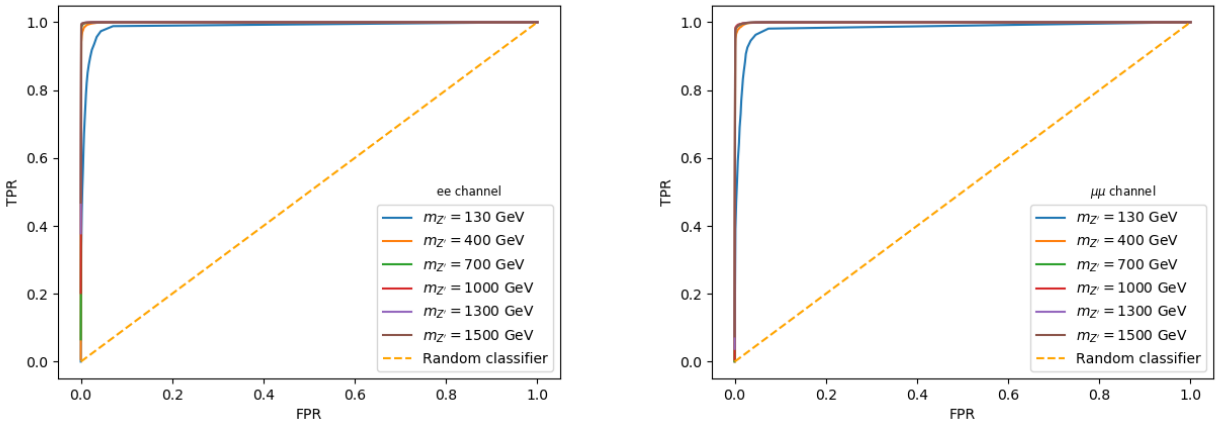


Figure F.2: Training and validation loss as a function of epochs during training on the dark Higgs HDS.



(a) (b)

Figure F.3: ROC curve plots for a selection of Z' mass signals in the a) ee and b) $\mu\mu$ channel in the dark Higgs HDS.

Dark Higgs HDS				
	ee channel		$\mu\mu$ channel	
$m_{Z'}$ (GeV)	Accuracy	AUC	Accuracy	AUC
130	0.896	0.991	0.887	0.989
200	0.968	0.997	0.966	0.997
300	0.984	0.999	0.981	0.998
400	0.990	0.999	0.988	0.999
500	0.990	1.0	0.989	1.0
600	0.992	1.0	0.991	1.0
700	0.992	1.0	0.990	1.0
800	0.992	1.0	0.991	1.0
900	0.994	1.0	0.992	1.0
1000	0.992	1.0	0.991	1.0
1100	0.993	1.0	0.991	1.0
1200	0.993	1.0	0.993	1.0
1300	0.993	1.0	0.990	1.0
1400	0.993	1.0	0.993	1.0
1500	0.993	1.0	0.991	1.0

Table F.2: Accuracy and AUC achieved by the neural network for different Z' mass signals in the dark Higgs HDS.

F.3 Results

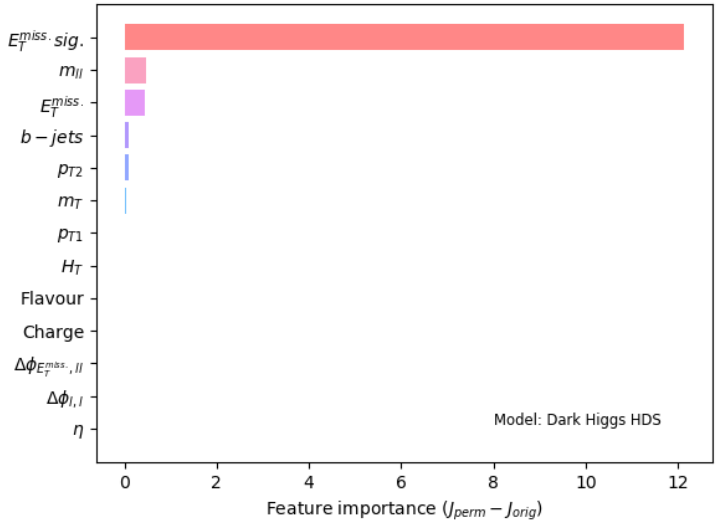
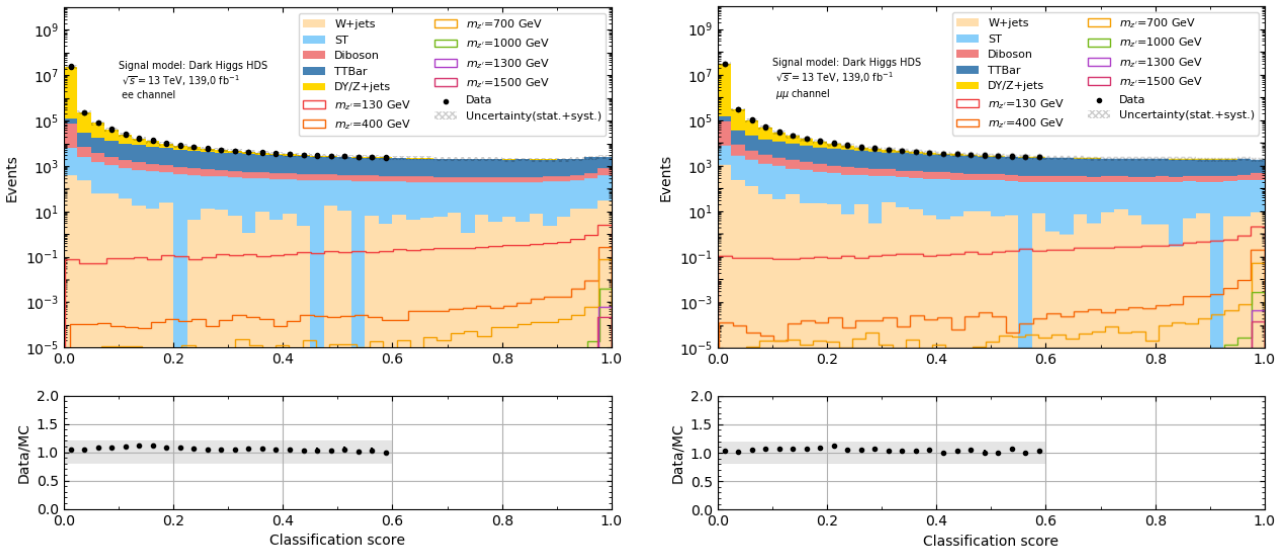
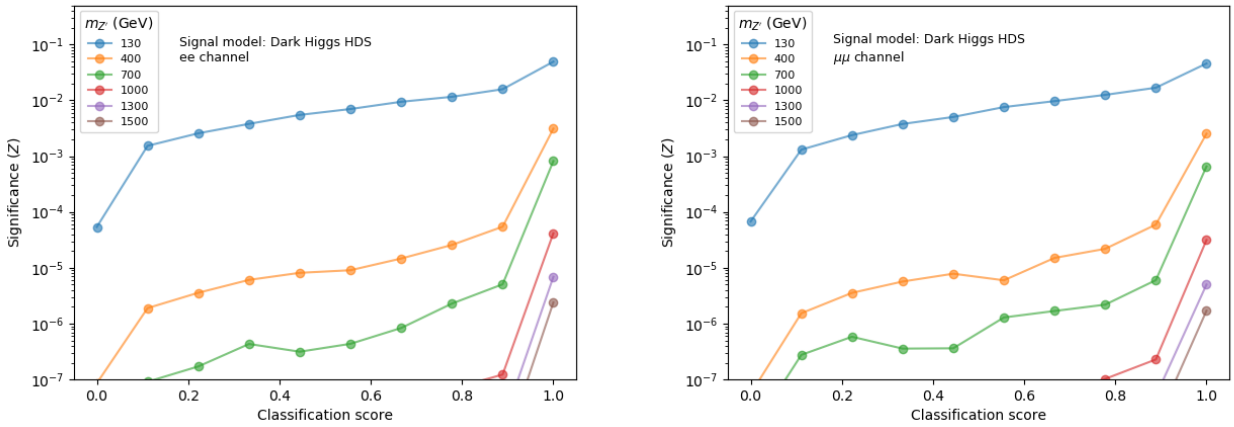


Figure F.4: Permutation feature importance of the features used to train the neural network for the dark Higgs HDS.



(a) (b)

Figure F.5: Classification score distributions for the background and signal for the neural network in the dark Higgs HDS. The results are shown in the a) ee and b) $\mu\mu$ channel and compared with real data outside of the signal region.



(a) (b)

Figure F.6: Expected significance for the different signals in the dark Higgs HDS at different classification scores in the a) ee and b) $\mu\mu$ channels. The approximation $Z = \frac{s}{\sqrt{b}}$ is used in order to prevent undefined values.

Dark Higgs HDS		
	ee channel	$\mu\mu$ channel
$m_{Z'}$ (GeV)	Expected significance (Z)	
130	$4.9 \cdot 10^{-2}$	$4.5 \cdot 10^{-2}$
200	$2.9 \cdot 10^{-2}$	$2.6 \cdot 10^{-2}$
300	$1.4 \cdot 10^{-2}$	$9.3 \cdot 10^{-3}$
400	$3.1 \cdot 10^{-3}$	$2.5 \cdot 10^{-3}$
500	$1.4 \cdot 10^{-3}$	$1.3 \cdot 10^{-3}$
600	$5.4 \cdot 10^{-4}$	$4.3 \cdot 10^{-4}$
700	$8.1 \cdot 10^{-4}$	$6.5 \cdot 10^{-4}$
800	$1.3 \cdot 10^{-4}$	$1.0 \cdot 10^{-4}$
900	$6.8 \cdot 10^{-5}$	$5.3 \cdot 10^{-5}$
1000	$4.1 \cdot 10^{-5}$	$3.1 \cdot 10^{-5}$
1100	$2.1 \cdot 10^{-5}$	$1.6 \cdot 10^{-5}$
1200	$1.2 \cdot 10^{-5}$	$8.6 \cdot 10^{-6}$
1300	$6.8 \cdot 10^{-6}$	$5.1 \cdot 10^{-6}$
1400	$4.2 \cdot 10^{-6}$	$2.6 \cdot 10^{-6}$
1500	$2.5 \cdot 10^{-6}$	$2.0 \cdot 10^{-6}$

Table F.3: Expected significances for different signals in the dark Higgs HDS for the ee and $\mu\mu$ channels.

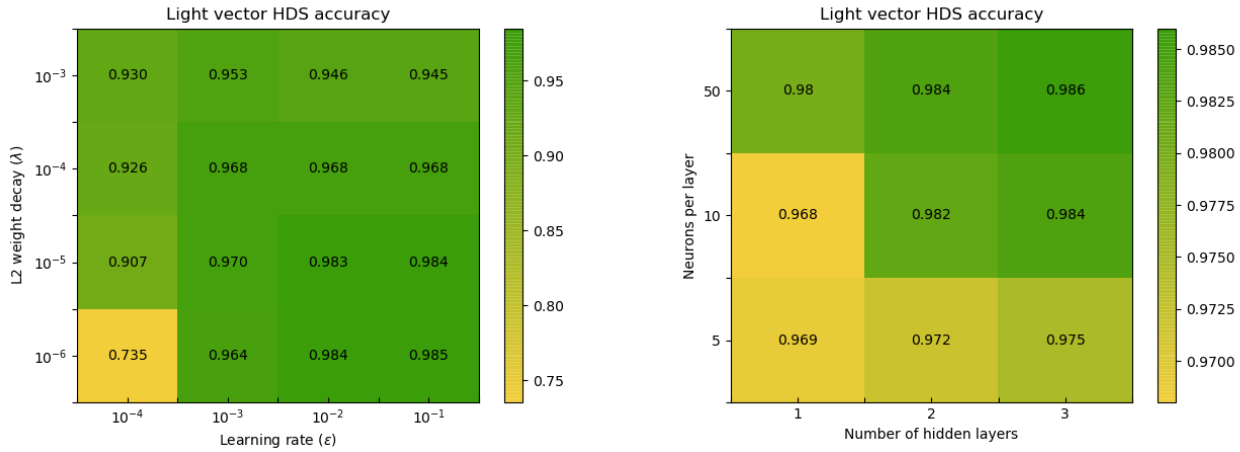
G Light vector HDS ML analysis

In this section, we show the results of the ML analysis for the light vector HDS, corresponding to the steps in the light dark sector analysis in section 7.7.

G.1 Hyperparameters

Light vector HDS	
Hyperparameter	Value
Number of hidden layers	3
Neurons per layer	50
Learning rate (ϵ)	10^{-2}
$L2$ weight decay (λ)	10^{-6}
Epochs	50
Batch size	10% of training set
Exponential decay rate (ρ_1)	0.90
Exponential decay rate (ρ_2)	0.99
Stabilization constant (δ)	10^{-8}

Table G.1: Hyperparameters used for training the neural network on the light vector HDS.



(a) (b)

Figure G.1: Grid searches for optimal hyperparameters a) $L2$ weight decay (λ) and learning rate (ϵ), and b) number of hidden layers and neurons per layer for ML training on the light vector HDS using accuracy as measure.

G.2 Performance

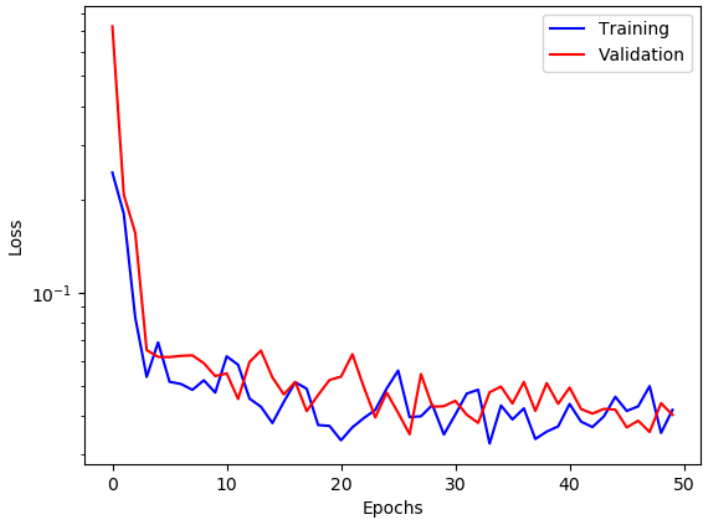
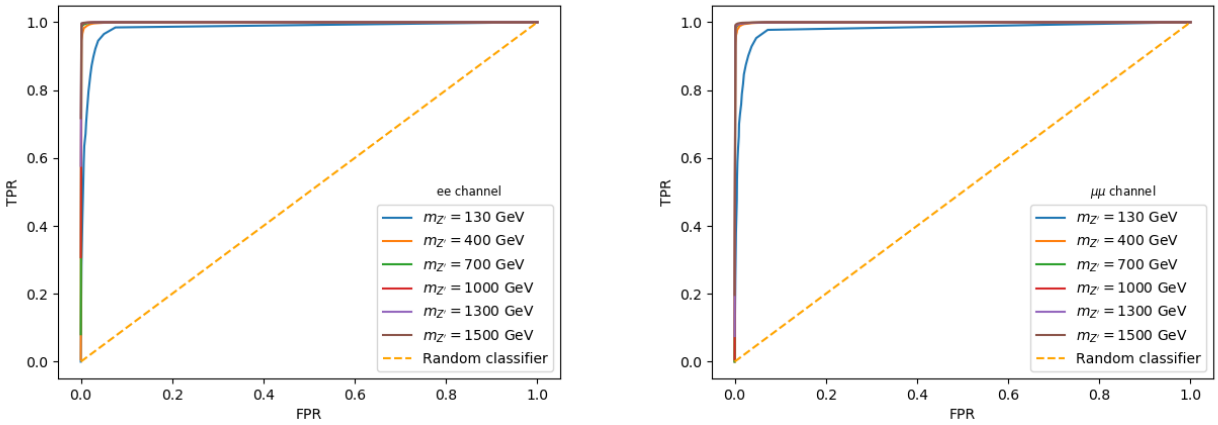


Figure G.2: Training and validation loss as a function of epochs during training on the light vector HDS.



(a) (b)

Figure G.3: ROC curve plots for a selection of Z' mass signals in the a) ee and b) $\mu\mu$ channel in the light vector HDS.

Light vector HDS				
	ee channel		$\mu\mu$ channel	
$m_{Z'}$ (GeV)	Accuracy	AUC	Accuracy	AUC
130	0.883	0.989	0.867	0.989
200	0.970	0.996	0.963	0.996
300	0.985	0.999	0.93	0.998
400	0.988	0.999	0.988	0.999
500	0.989	1.0	0.991	1.0
600	0.989	1.0	0.990	0.999
700	0.992	1.0	0.990	1.0
800	0.992	1.0	0.991	1.0
900	0.991	1.0	0.992	1.0
1000	0.992	1.0	0.991	1.0
1100	0.993	1.0	0.992	1.0
1200	0.992	1.0	0.991	1.0
1300	0.992	1.0	0.991	1.0
1400	0.993	1.0	0.991	1.0
1500	0.993	1.0	0.992	1.0

Table G.2: Accuracy and AUC achieved by the neural network for different Z' mass signals in the light vector HDS.

G.3 Results

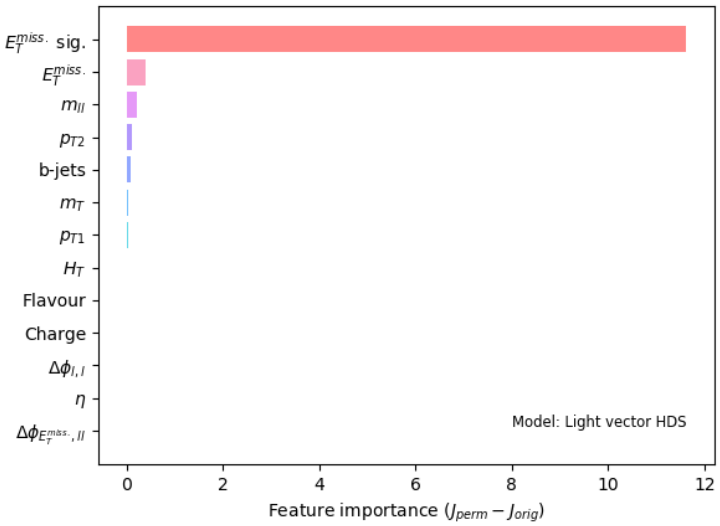
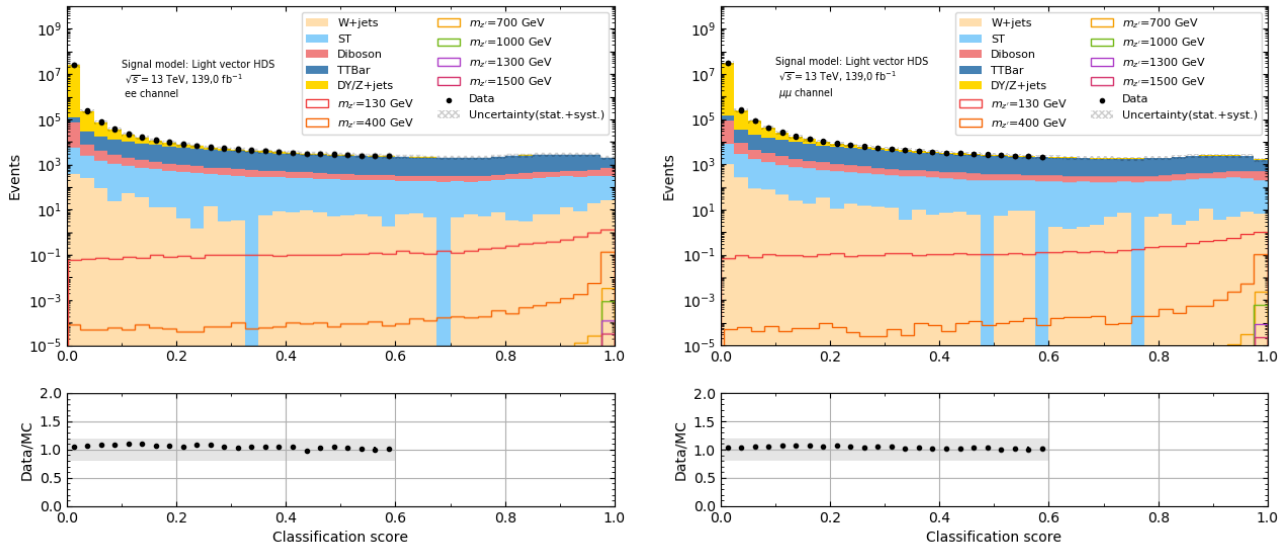
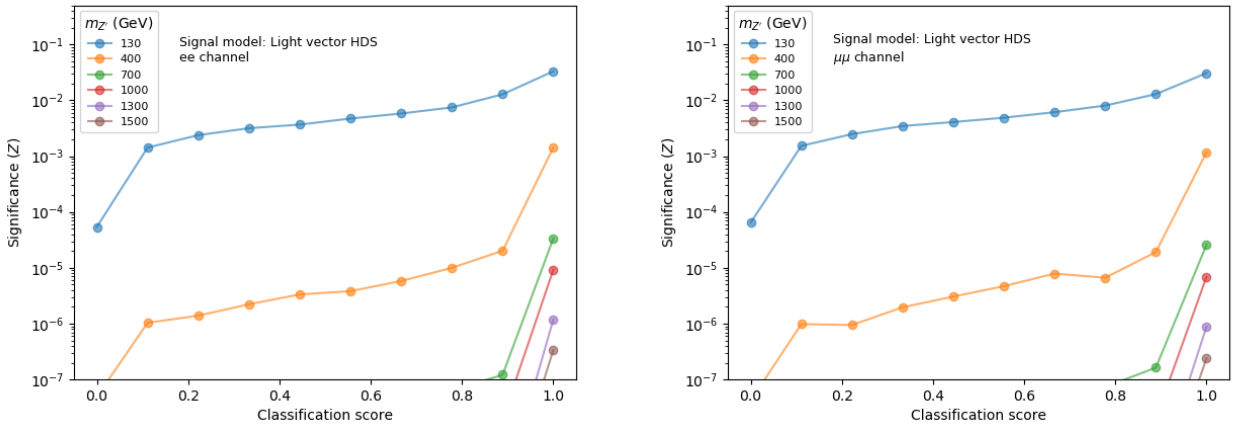


Figure G.4: Permutation feature importance of the features used to train the neural network for the light vector HDS.



(a) (b)

Figure G.5: Classification score distributions for the background and signal for the neural network in the light vector HDS. The results are shown in the a) ee and b) $\mu\mu$ channel and compared with real data outside of the signal region.



(a) (b)

Figure G.6: Expected significance for the different signals in the light vector HDS at different classification scores in the a) ee and b) $\mu\mu$ channels. The approximation $Z = \frac{s}{\sqrt{b}}$ is used in order to prevent undefined values.

Light vector HDS		
$m_{Z'}$ (GeV)	Expected significance (Z)	
	ee channel	$\mu\mu$ channel
130	$3.3 \cdot 10^{-2}$	$3.1 \cdot 10^{-2}$
200	$1.8 \cdot 10^{-2}$	$1.6 \cdot 10^{-2}$
300	$5.9 \cdot 10^{-3}$	$4.9 \cdot 10^{-3}$
400	$1.4 \cdot 10^{-3}$	$1.2 \cdot 10^{-3}$
500	$5.8 \cdot 10^{-4}$	$4.5 \cdot 10^{-4}$
600	$2.0 \cdot 10^{-4}$	$1.6 \cdot 10^{-4}$
700	$3.4 \cdot 10^{-5}$	$2.6 \cdot 10^{-5}$
800	$3.9 \cdot 10^{-5}$	$3.0 \cdot 10^{-5}$
900	$1.9 \cdot 10^{-5}$	$1.4 \cdot 10^{-5}$
1000	$9.4 \cdot 10^{-6}$	$6.9 \cdot 10^{-6}$
1100	$4.5 \cdot 10^{-6}$	$3.3 \cdot 10^{-6}$
1200	$2.4 \cdot 10^{-6}$	$2.0 \cdot 10^{-6}$
1300	$1.9 \cdot 10^{-6}$	$7.5 \cdot 10^{-7}$
1400	0	$1.3 \cdot 10^{-6}$
1500	$1.5 \cdot 10^{-6}$	$8.9 \cdot 10^{-7}$

Table G.3: Expected significances for different signals in the light vector HDS for the ee and $\mu\mu$ channels.

H Comparison of methods

Below, we show the ratio of the expected significances obtained by using neural networks to the expected signifcances obtained by the cut and count method in the dark Higgs HDS, light vector LDS and light vector HDS.

Dark Higgs HDS		
$m_{Z'}$ (GeV)	Expected significance ($Z_{NN}/Z_{C\&C}$)	
	ee channel	$\mu\mu$ channel
130	$6.23 \cdot 10^2$	-
200	1.52	1.53
300	2.18	1.79
400	1.93	1.79
500	2.01	2.32
600	2.16	2.15
700	1.45	2.24
800	2.32	2.27
900	2.19	2.41
1000	2.16	2.58
1100	2.28	2.67
1200	2.40	2.69
1300	2.34	3.40
1400	2.63	2.65
1500	2.27	-

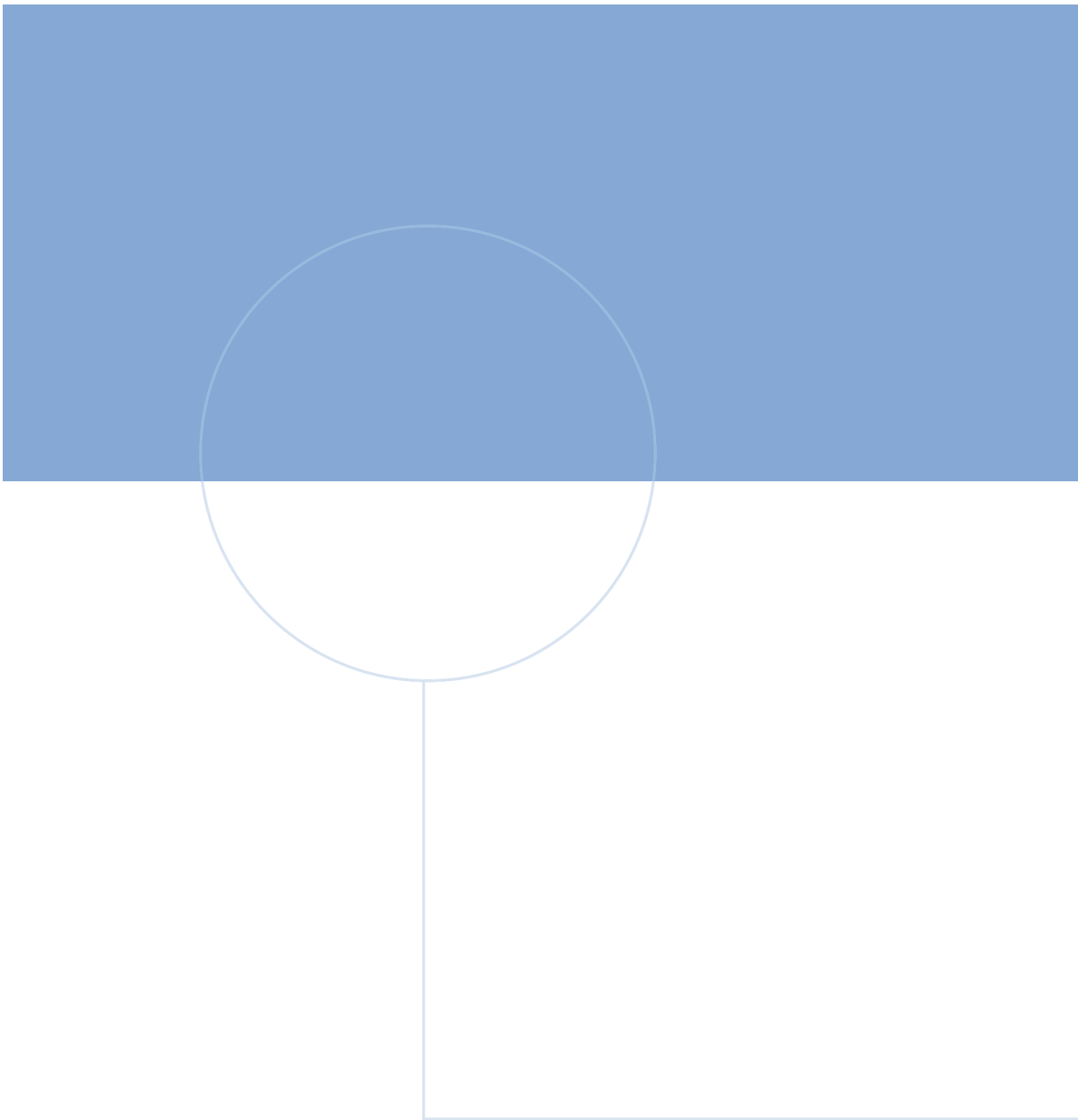
Table H.1: Ratio of expected significance $Z_{NN}/Z_{C\&C}$ obtained by the neural network (NN) and cut and count (C&C) method for the dark Higgs HDS.

Light vector LDS		
$m_{Z'}$ (GeV)	Expected significance ($Z_{NN}/Z_{C\&C}$)	
	ee channel	$\mu\mu$ channel
130	-	-
200	0.13	0.14
300	1.50	3.80
400	1.58	1.2
500	1.58	1.67
600	1.53	1.82
700	1.60	1.86
800	1.67	1.86
900	1.64	2.02
1000	1.66	2.27
1100	1.77	2.36
1200	2.56	2.67
1300	1.72	2.79
1400	1.77	3.26
1500	1.88	3.57

Table H.2: Ratio of expected significance $Z_{NN}/Z_{C\&C}$ obtained by the neural network (NN) and cut and count (C&C) method for the light vector LDS.

Light vector HDS		
$m_{Z'}$ (GeV)	Expected significance ($Z_{NN}/Z_{C\&C}$)	
	ee channel	$\mu\mu$ channel
130	-	-
200	1.38	1.45
300	1.73	1.69
400	1.79	1.88
500	1.87	1.88
600	2.00	2.08
700	2.00	2.17
800	2.05	2.14
900	2.21	2.22
1000	2.19	2.30
1100	2.05	2.54
1200	2.40	2.41
1300	3.11	1.53
1400	0.00	1.78
1500	1.83	1.51

Table H.3: Ratio of expected significance $Z_{NN}/Z_{C\&C}$ obtained by the neural network (NN) and cut and count (C&C) method for the light vector HDS.



NTNU

Norwegian University of
Science and Technology