THE UNIVERSITY
of ADELAIDE

Novel Data Analysis Techniques for BSM Physics Applications

Author:
Adam Leinweber

Supervisors:
Prof. Martin White
Prof. Paul Jackson

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy in Physics

Department of Physics
School of Physical Sciences

October, 2022

For Nikita, who will likely never read this.

Declaration of Authorship

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint award of this degree.

The author acknowledges that copyright of published works contained within the thesis resides with the copyright holder(s) of those works.

I give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

I acknowledge the support I have received for my research through the provision of an Australian Government Research Training Program Scholarship.

Signed: Adam Leinweber

Date: 4 October, 2022

"*Science is made up of so many things that appear obvious after they are explained.*"

Frank Herbert, *Dune*

Acknowledgements

First and foremost I would like to acknowledge my supervisor Martin White. His knowledge, wit, and willingness to sit down and tackle difficult problems with me made him a fantastic supervisor. I would like to thank Riley Patrick who guided me through Honours and the first year of my PhD. He really helped me gain a deeper understanding of both machine learning and particle physics. I would like to thank Melissa van Beekveld for being an excellent international collaborator to work with, and the Dark Machines collective for being a fantastic collaboration to contribute to. I would also like to thank all of the wonderful people in the physics department for making the office a great place to do research. An extra thank you to the fantastic people who have run S4S over the years. S4S is such a valuable resource for students to be able to present their research and learn new skills and I have very much enjoyed it.

Outside of the University, I would like to thank my partner Nikita Murfitt, who has been nothing but supportive throughout my years as a PhD candidate. She's always been there to listen to my frustrations and to make me laugh. I would also like to thank her family for being so supportive of me over the years. I would like to thank my parents Nancy and Derek for always being there for me, and my brothers Connor and Sean for always being a laugh. Finally I would like to thank my cat George for always stealing the spotlight in video meetings. All of these lovely souls have been instrumental in my successful completion of this massive undertaking and I can never thank them enough.

Abstract

Adam Leinweber

Novel Data Analysis Techniques for BSM Physics Applications

Since the discovery of the Higgs boson in 2012, the exact nature of new physics beyond the Standard Model (BSM) remains unknown. Modern experiments work to optimise analyses on specific regions of the parameter space where new physics is considered likely to exist. This thesis aims to identify issues with modern experimental techniques, and proposes solutions using a variety of novel data analysis techniques. Throughout this thesis, a particular emphasis is placed on the BSM theory known as supersymmetry which introduces superpartners for every particle in the standard model.

This thesis is broadly split into four parts. The first part is an overview of modern particle physics, including the standard model, supersymmetry, and high energy collider experiments. Additionally, an in depth introduction to machine learning is presented.

The following part concerns unsupervised anomaly detection in the context of high energy collider experiments. In a typical supervised experiment, one must specify a number of assumptions about the nature of the BSM signal being searched for. I show that by using unsupervised machine learning techniques, one is able to construct a quantity that is able to improve the performance of a typical analysis in a signal agnostic fashion. These techniques are tested on a wide array of BSM signals from various theories including supersymmetry.

The next part explores dimensional reduction of a supersymmetric theory. Typically supervised analyses are done on a small subset of the original parameter space, fixing the other parameters at arbitrary values. This shields the rich phenomenology of the BSM theory from the analysis, allowing many spectra to go undetected. By performing a dimensional reduction using machine learning, I am able to construct a 2-D space which captures the full phenomenology of the original high dimensional parameter space. Using this dimensionally reduced representation, I identify interesting regions of the parameter space, and exclude a number of previously unexcluded models.

The final part examines a optimisation algorithms in high dimensional spaces. Once a particular BSM model has been chosen, it is important to consider which parameter

values yield the closest match with current experimental data. This can be considered as an optimisation problem in a high dimensional space. By comparing the performance of a number of optimisation algorithms on analytic test functions, as well as a likelihood function based on a global fit of a supersymmetric theory performed by the GAMBIT collaboration, the strengths and weaknesses of each algorithm are identified. Ultimately I am able to draw conclusions on which algorithms are suitable for high dimensional particle astrophysics problems in general.

Contents

1 Introduction

Searches for new physics beyond the Standard Model of particle physics have so far been unsuccessful. After the discovery of the Higgs boson in 2012, physicists are in the interesting position of attempting to discover new physics from a theory of which the details are not known. While many Standard Model extensions exist, there is no unambiguous indication as to what phenomenology one can expect to observe. Modern collider experiments attempt to probe the limits of the Standard Model by optimising searches on specific regions of the parameter space where new physics is considered to be likely to exist. While these analyses have been very successful in excluding select regions of the parameter space, no strong evidence for beyond the Standard Model physics has been found. One of the major disadvantages of this technique is that the particular region of the parameter space must be precisely specified, excluding other regions where new physics may still exist.

An incredibly popular theory in modern particle physics is supersymmetry (SUSY), which introduces supersymmetric partners for every particle in the Standard Model. This theory quite naturally solves many of the problems that exist in the current Standard Model. As with any beyond the Standard Model theory, SUSY introduces a set of free parameters which govern the behaviour of the new particles it introduces. In order to make predictions using a SUSY model, one must specify the values of a number of these parameters. By comparing predicted results to experimental observations, one can constrain these parameters to identify which set of parameters best matches current or future observations.

Modern analyses are typically optimised on a small number of fundamental parameters of a particular BSM theory. This raises multiple problems which I will address throughout this thesis. The first of which is that optimising an analysis requires a number of presuppositions about the signal being searched for. These presuppositions include which BSM theory one is drawing from, and the values of the parameters which govern said theory. In Chapters 5 and 6 I explore using unsupervised anomaly detection in order to perform analyses in a signal agnostic fashion. This technique allows one to construct a quantity which is able to separate anomalous signals from the Standard Model background with minimal signal assumptions. The second issue is that even after a BSM theory is chosen, supervised analyses are not usually done using the entire parameter space. Typically a

simplified model featuring 2-3 parameters is chosen, fixing the remaining values arbitrarily. This obscures the rich phenomenology possible with some of these BSM theories, and allows many spectra to remain undetected. In Chapter 7 I explore using dimensional reduction in order to compress the high dimensional parameter space of a BSM physics model into 2-D where the full phenomenology of the theory is captured. This allows one to easily identify interesting regions on which to optimise analyses.

Once one has chosen a BSM theory to explore, a natural question to ask is what set of parameters best match with current experimental results? This problem can be thought of as an optimisation problem in a high dimensional parameter space. In Chapter 8 I explore a number of high dimensional optimisation algorithms and test them on analytic test functions, as well as on a likelihood function based on a global fit of a SUSY theory performed by the GAMBIT collaboration. Using this information I determine the strengths and weaknesses of each optimisation algorithm, and draw conclusions on which are suitable for high dimensional particle astrophysics problems.

Throughout this thesis I will be employing state of the art machine learning, anomaly detection, optimisation, and dimensional reduction techniques designed to aid in the discovery of new physics beyond the Standard Model. Chapter 2 covers the Standard Model of particle physics, its flaws, and how supersymmetry can address them. Chapter 3 discusses high energy collider experiments and their uses in modern particle physics analyses. Chapter 4 covers the basics of machine learning, and then explains the workings and applications of neural networks, including an architecture for dimensional reduction and unsupervised anomaly detection. In Chapter 5, a sophisticated technique for unsupervised anomaly detection in a particle physics context is presented. Using this method I am able to reject a number of novel physics signals with very minimal signal assumptions. Chapter 6 expands on the previous chapter, refining the algorithm, and testing on many more signals covering a wide range of supersymmetric and non-supersymmetric theories. Chapter 7 explores compressing the 4-dimensional minimally supersymmetric Standard Model into 2 dimensions and optimising analyses within this 2-D plane. This allows one to take all parameters into account when optimising analyses. In Chapter 8 I explore a number of optimisation algorithms for identifying supersymmetric model parameters that are most consistent with current experimental results. Finally, Chapter 9 provides a summary of the main results and concludes the thesis.

2 The Standard Model and Supersymmetric Extensions

2.1 The Standard Model

The Standard Model (SM) [1, 2, 3], first named in the 1970's, is the currently accepted model of particle physics, describing three of the four fundamental forces of nature (the electromagnetic, weak, and strong forces). The Standard Model is a gauge field theory, which means that the Lagrangian density describing the theory is invariant under certain local transformations. Particles in the Standard Model are treated as point-like fundamental objects which can be grouped into two classes using a quantum number known as "spin": fermions with half integer spin, and bosons with integer spin. Fermions obey Fermi-Dirac statistics, while bosons obey Bose-Einstein statistics. Figure 2.1 displays a table of the fundamental SM particles.

2.1.1 The Particles of the Standard Model

The spin-1 bosons are known as gauge bosons and are responsible for the fundamental forces. The photon (γ) is the mediator of the electromagnetic force, the $W^{\pm}$ and Z bosons are the mediators of the weak force, and gluons (g) mediate the strong force. The Higgs boson, instead of having spin-1 like all the other bosons, has spin-0. This makes it what is known as a "scalar" particle. The Higgs boson is responsible for giving mass to the massive gauge bosons, and contributes to the masses of all other known massive elementary particles.

The spin-$\frac{1}{2}$ fermions are the building blocks of matter, and are split into two main groups: leptons and quarks. Leptons with electric charge do not interact through the strong force, but rather via the electromagnetic and weak forces. Neutral leptons, known as "neutrinos", interact only via the weak force. Quarks, also known as partons, interact through the strong, weak, and electromagnetic forces. Each of these particles has a corresponding anti-matter particle with opposite quantum numbers and the same mass. For example, an electron's anti-particle is known as the positron, and while they both have the same mass and spin, the electron has electric charge $-1e$, and the positron has electric charge $+1e$.

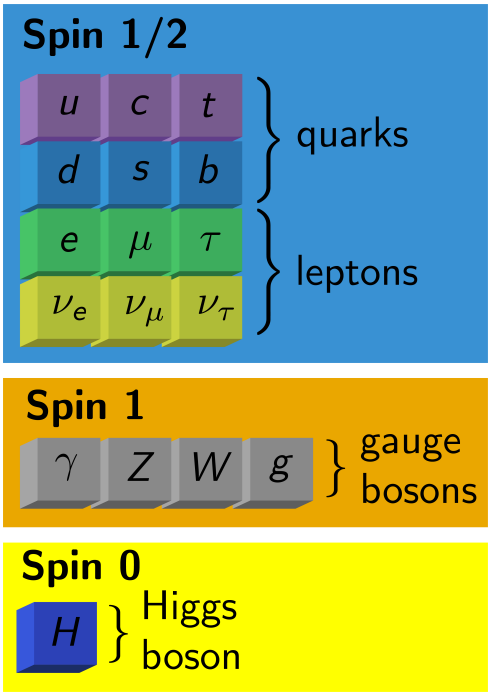


FIGURE 2.1: The particle content of the SM.

Quarks are never observed on their own due to the colour confinement hypothesis [4], instead combining to form composite particles known as hadrons, which consist of two or more quarks held together by the strong force. Note that the top quark decays before it can form bound states and so it does not form hadrons. The quarks in the SM are split into two types. "Up"-type quarks, referred to as up, charm, and top quarks (denoted u, c and t), and "down"-type quarks, referred to as down, strange and bottom quarks (dentoed d, s, and b). Up-type quarks have electric charge $\frac{+2}{3}$, while down-type quarks have electric charge $\frac{-1}{3}$.

Quarks and gluons carry what is known as "colour charge". Colour charge can take the form of red, green, and blue, as well as their "anti"-colours. Colour charge works analogously to the primary colours where red green and blue can sum to form white. Additionally a colour and it's anti-colour can sum to form white. All observable particles are "colourless" or white. It is this fact that means quarks are never observed on their own. "Baryons" are composite particles formed from three quarks, with each individual quark having red, green, or blue colour charge. "Mesons" are composite particles formed from a quark and an anti-quark, having colour and anti-colour charge.

2.1.2 Symmetries of the Standard Model

Symmetries play an important role in our fundamental understanding of nature. We understand that the total energy in an isolated system must remain constant under a

translation, or rotation of the coordinates. In the microscopic world of quantum mechanics the way we describe physical phenomena is through a mathematical framework known as Quantum Field Theory (QFT), where point-like particles are generalised to quantised fields.

The internal symmetry of the SM is represented by the gauge group $SU(3)_c \otimes SU(2)_L \otimes U(1)_Y$. These gauge groups are the origin of the gauge bosons. $SU(3)_c$ is the symmetry group responsible for colour charge, and $SU(2)_L \otimes U(1)_Y$ is the gauge symmetry group corresponding to electroweak interactions. The way in which particles transform under this gauge group specifies the charges that they have in the SM, and dictates the way in which particles interact with one another.

This gauge group is not preserved at the electroweak symmetry breaking scale. At these low energies the gauge group is broken via the Higgs mechanism. Hence our gauge group goes from $SU(3)_c \otimes SU(2)_L \otimes U(1)_Y$ to $SU(3)_c \otimes U(1)_Q$. In the unbroken symmetry $SU(2)_L \otimes U(1)_Y$, we have 3 W bosons of weak isospin, denoted W^1, W^2, W^3, and one B boson of weak hypercharge. After the symmetry has been broken to $U(1)_Q$ the weak bosons form "mixings", generating the massive weak gauge bosons we have already met, while $SU(3)$ symmetry is not broken.

2.1.3 Quantum Electrodynamics

Quantum Electrodynamics (QED) is a theory that attempts to combine Maxwell's theory of electromagnetism [5] with quantum mechanics. It describes phenomena involving electrically charged particles and photons, and is invariant under complex phase rotations applied to particle fields. It is based on the Abelian $U(1)_Q$ symmetry group, where the electric charge (Q) is the generator. QED has seen excellent predictive power, agreeing very strongly with experimental results.

Let us start by considering the Lagrangian density of a free Dirac fermion field, which is written as

$$\mathcal{L} = \bar{\psi}(i\gamma^\mu \partial_\mu - m)\psi \tag{2.1}$$

where ψ is a 4-component Dirac field, and γ^μ is the 4×4 gamma matrix.

We can see that Eq. 2.1 is invariant under the following global $U(1)_Q$ transformation

$$\psi \to \psi' = e^{-ig_e\theta}\psi, \tag{2.2}$$

as the other terms in Eq. 2.1 transform as

$$\bar{\psi} \to \bar{\psi}' = (\psi')^\dagger \gamma_0 = \left(e^{-ig_e\theta}\psi\right)^\dagger \gamma_0 = \bar{\psi}e^{ig_e\theta}, \qquad \partial_\mu\psi \to \partial_\mu\psi' = e^{-ig_e\theta}\partial_\mu\psi, \tag{2.3}$$

where g_e is the QED coupling.

When this global transformation is promoted to a local transformation, we introduce a space-time dependence on $\theta \to \theta(x)$. In this case we must introduce an additional term in order to preserve gauge invariance:

$$\mathcal{L} \to \mathcal{L} - g_e \bar{\psi} \gamma^\mu A_\mu \psi \tag{2.4}$$

where the gauge field A_μ transforms as

$$A_\mu \to A_\mu + \partial_\mu \lambda, \qquad \text{where } \lambda(x) = -\frac{\theta(x)}{g_e}. \tag{2.5}$$

This gauge field leads to the photon gauge boson. Under this local gauge transformation, the Dirac fermion field ψ and its derivative transform as

$$\psi \to \psi' = e^{-ig_e\theta(x)}\psi, \tag{2.6}$$
$$\bar{\psi} \to \bar{\psi}' = \bar{\psi} e^{ig_e\theta(x)}, \tag{2.7}$$
$$\partial_\mu \psi \to \partial_\mu \psi' = e^{-ig_e\theta(x)}(\partial_\mu \psi - ig_e \partial_\mu \theta(x)). \tag{2.8}$$

We now replace the normal derivative with a covariant derivative, a method known as "minimal substitution". The covariant derivative is defined as

$$\mathcal{D}_\mu \equiv \partial_\mu + ig_e A_\mu. \tag{2.9}$$

Upon replacing ∂_μ with $\mathcal{D}_\mu$ in Eq. 2.1, we find

$$\mathcal{D}_\mu \psi \to \mathcal{D}'_\mu \psi' = (\partial_\mu + ig_e A'_\mu)\psi' \tag{2.10}$$
$$= e^{-ig_e\theta(x)} \left[\partial_\mu \psi - ig_e \psi \partial_\mu \theta(x)\right] + ig_e \left[A_\mu + \partial_\mu \theta(x)\right] e^{-ig_e\theta(x)}\psi \tag{2.11}$$
$$= e^{-ig_e\theta(x)}(\partial_\mu + ig_e A_\mu)\psi \tag{2.12}$$
$$= e^{-ig_e\theta(x)} \mathcal{D}_\mu \psi. \tag{2.13}$$

So we see that $\mathcal{D}_\mu \psi$ transforms in the same way as $\partial_\mu \psi$ in Eq. 2.3. Using the covariant derivative, the rest of the terms defined in Eq. 2.1 are invariant under local transformations.

In order to describe the dynamics of the photon field A_μ, we must add its kinetic term to the Lagrangian. Using the electromagnetic field strength tensor, defined as $F_{\mu\nu} \equiv \partial_\mu A_\nu - \partial_\nu A_\mu$, we can write the non-interacting Lagrangian for the photon field A_μ as

$$\mathcal{L}_\gamma = -\frac{1}{4} F^{\mu\nu} F_{\mu_\nu} + \frac{1}{2} m_A^2 A^\mu A_\mu. \tag{2.14}$$

The mass term is not invariant under local gauge transformations, thus m_A is set to zero in order to preserve local gauge invariance. Finally we can write the QED Lagrangian as

$$\mathcal{L}_{\text{QED}} = \bar{\psi}(i\gamma^\mu \mathcal{D}_\mu - m)\psi - \frac{1}{4}F_{\mu\nu}F^{\mu\nu}. \tag{2.15}$$

This Lagrangian is invariant under both Lorentz and local $U(1)_Q$ transformations. One important implication of this local $U(1)_Q$ invariance is that a mass term for the photon field is not permitted.

2.1.4 Quantum Chromodynamics

The theory of Quantum Chromodynamics (QCD) [6, 7], just like QED, provides strong predictive power. Where QED is based on a $U(1)_Q$ symmetry, QCD is based on the $SU(3)_C$ symmetry group, where C is colour charge. When imposing local gauge invariance under this symmetry group on the QCD Lagrangian density, one introduces 8 coloured gluon fields, which correspond to the 8 generators of the group.

We begin the construction of a local $SU(3)_C$ invariant QCD Lagrangian in a similar way to the QED case. We again replace the normal derivative with the covariant derivative, this time defining it as

$$\mathcal{D}_\mu \equiv \partial_\mu - ig_s \mathbf{T} \cdot \mathbf{A}_\mu, \tag{2.16}$$

where the $T_a \equiv \frac{\lambda_a}{2}$, for $a = 1, \ldots, 8$, are the 8 generators of the $SU(3)_C$ group, and A^a_μ are the 8 gluon fields. λ_a is the set of 8 linearly independent Gell-Mann matrices, satisfying

$$[\lambda_a, \lambda_b] = 2if^{abc}\lambda_c \tag{2.17}$$

where f^{abc} for $a, b, c = 1, \ldots, 8$ are the structure constants of the $SU(3)_C$ group.

We define the six flavour quarks q, as fermionic fields populating a triplet such that

$$\psi \equiv \begin{bmatrix} q_r \\ q_g \\ q_b \end{bmatrix}, \qquad \bar{\psi} \equiv \begin{bmatrix} \bar{q}_r & \bar{q}_g & \bar{q}_b \end{bmatrix}. \tag{2.18}$$

Additionally we define the field strength tensor

$$G^a_{\mu\nu} \equiv \partial_\mu A^a_\nu - \partial_\nu A^a_\mu - ig_s[A^a_\mu, A^a_\nu]. \tag{2.19}$$

The commutation relation

$$[A_\mu, A_\nu]^a = if^{abc}A^a_\mu A^c_\nu \tag{2.20}$$

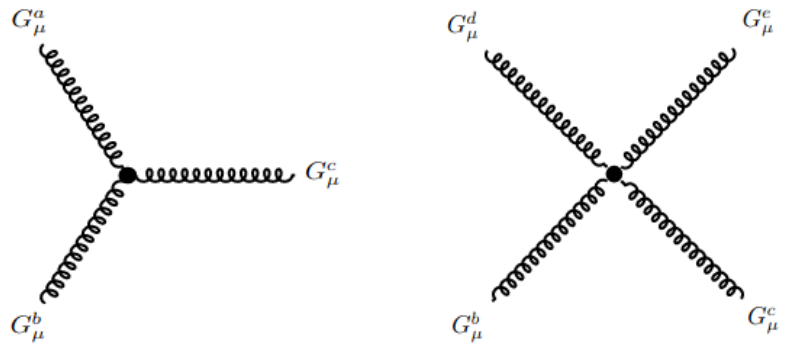


FIGURE 2.2: Gluon self interaction Feynman diagrams resulting from the commutation relation in Eq. 2.20

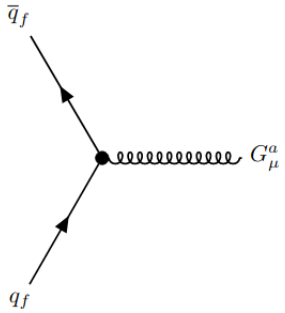


FIGURE 2.3: A Feynman diagram depicting an interaction between two quark fields and a gluon field resulting from Eq. 2.22

gives rise to triple and quartic gluon coupling terms. This means that gluons are able to self interact, as shown in Figure 2.2, because they carry colour charge. These self interaction terms make QCD a much richer and more complex theory than QED.

Using the above definitions, we can write the QCD Lagrangian in the form

$$\mathcal{L}_{\text{QCD}} = \bar{\psi}(i\gamma^\mu \mathcal{D}_\mu - m)\psi - \frac{1}{4} G^a_{\mu\nu} G^{\mu\nu}_a. \tag{2.21}$$

By replacing ∂_μ with $\mathcal{D}_\mu$, we create an interaction term between a gluon and two quark fields, as shown in Figure 2.3. We can write this term as

$$\mathcal{L}_{\text{int}} = 2g\bar{\psi}\gamma^\mu \lambda_a G^a_\mu \psi. \tag{2.22}$$

This means that quark-antiquark pairs can be generated by gluons and can similarly annihilate to gluons.

Now that we understand the intricacies of QED and QCD, we can explore the generation of the $W^\pm$ and Z bosons via the process of electroweak unification and move towards an understanding of the full Standard Model symmetry group.

2.1.5 Electroweak unification

Electroweak unification [8] refers to the unification of the electromagnetic and weak forces, represented by the gauge group $SU(2)_L \otimes U(1)_Y$. Here $SU(2)_L$ acts on the weak isospin I, and $U(1)_Y$ acts on the weak hypercharge Y. The $SU(2)_L$ doublet contains the chiral-left components of the elecrically charged leptons and their neutrinos, while the chiral-right components are in their own singlet. We can write these as

$$\psi_L^{doublet} \equiv \begin{bmatrix} \nu_l \\ l \end{bmatrix}_L, \begin{bmatrix} U \\ D \end{bmatrix}_L \tag{2.23}$$

$$\psi_R^{singlet} \equiv l_R,\, U_R,\, D_R, \tag{2.24}$$

where we define

$$l = (e, \mu, \tau) \tag{2.25}$$

$$\nu_l = (\nu_e, \nu_\mu, \nu_\tau) \tag{2.26}$$

$$U = (u, c, t) \tag{2.27}$$

$$D = (d, s, b). \tag{2.28}$$

Using this notation we can write the weak $SU(2)_L$ current as

$$J_i^\mu = \frac{1}{2} \begin{bmatrix} \bar{\nu}_l & \bar{l} \end{bmatrix}_L \gamma^\mu \tau_i \begin{bmatrix} \nu_l \\ l \end{bmatrix}_L, \tag{2.29}$$

where τ_i is the ith Pauli spin matrix. The third current, where $i = 3$ does not change the charge of the particles involved, and is known as the neutral current. The electromagnetic current for a lepton l is given by

$$J_{EM}^\mu = Q\bar{l}\gamma^\mu l = Q(\bar{l}_L \gamma^\mu l_L + \bar{l}_R \gamma^\mu l_R), \tag{2.30}$$

where Q is the elecromagnetic charge operator. This quantity is not invariant under local $SU(2)_L$ transformations and so we construct an $SU(2)_L$ invariant $U(1)_Y$ current which is written as

$$J_Y^\mu = Y_L \begin{bmatrix} \bar{\nu}_l & \bar{l} \end{bmatrix}_L \gamma^\mu \begin{bmatrix} \nu_l \\ l \end{bmatrix}_L + Y_R \bar{l}_R \gamma^\mu l_R, \tag{2.31}$$

where hypercharges Y_L and Y_R denote the conserved charge operators of the $U(1)_Y$ symmetry. Notice that J_Y^μ is a linear combination of the weak and electromagnetic currents J_3^μ and J_{EM}^μ. This implies that hypercharge Y can be constructed using electromagnetic

charge Q and the third component of weak isospin I_3. This relation can be written as

$$Y = 2(Q - I_3). \tag{2.32}$$

Using this we can calculate the weak quantum numbers of the quarks and leptons.

In order to write the electroweak Lagrangian, we need to first define the covariant derivative that acts on a matter field with weak isospin $\frac{1}{2}$, and weak hypercharge Y. This can be written as

$$\mathcal{D} \equiv \partial_\mu + ig_w \frac{1}{2}\tau_a W_\mu^a + ig_Y \frac{1}{2} Y B_\mu, \tag{2.33}$$

where τ_a are the 2×2 Pauli spin matrices, W_μ^a is the gauge field of the $SU(2)_L$ group, and B_μ is the gauge field of the $U(1)_Y$ group. g_w and g_Y are the two independent coupling constants of the theory.

We can hence write the electroweak Lagrangian as

$$\mathcal{L}_{\text{EW}} = \bar{\psi}(i\gamma^\mu \mathcal{D}_\mu - m)\psi - \frac{1}{4} W_a^{\mu\nu} W_{\mu\nu}^a - \frac{1}{4} B^{\mu\nu} B_{\mu\nu}, \tag{2.34}$$

where ψ can be either chiral-left or right fields. Similarly to QED, a gauge field mass term is forbidden due to local $SU(2)_L \otimes U(1)_Y$ invariance. This is an issue as we know the weak force carriers, the $W^\pm$ and Z bosons, are massive. This means we require some mechanism in order to generate gauge boson masses in a gauge invariant fashion. This is where the Higgs mechanism comes into play.

2.1.6 The Higgs Mechanism

The Higgs mechanism introduces a scalar field which exists at all space-time points. By introducing the Higgs scalar field which acquires a non-zero vacuum expectation value (vev), we break the local $SU(2)_L \otimes U(1)_Y$ gauge symmetry of the electroweak Lagrangian, and the gauge bosons acquire mass.

In order to do this we must first introduce a single Higgs doublet in the form [9]:

$$\phi = \begin{bmatrix} \phi^0 \\ \phi^+ \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} \phi_1 + i\phi_2 \\ \phi_3 + i\phi_4 \end{bmatrix} \tag{2.35}$$

for real fields ϕ_i with hypercharge $Y = \frac{1}{2}$. We also introduce a potential for the scalar field which spontaneously breaks the symmetry

$$V(\phi) = \frac{1}{2}\mu^2(\phi^\dagger \phi) + \frac{1}{4}\lambda(\phi^\dagger \phi)^2. \tag{2.36}$$

This potential has the "Mexican hat" form shown in Figure 2.4. At high energies, the vev of the Higgs approaches 0 which corresponds to the full gauge group discussed

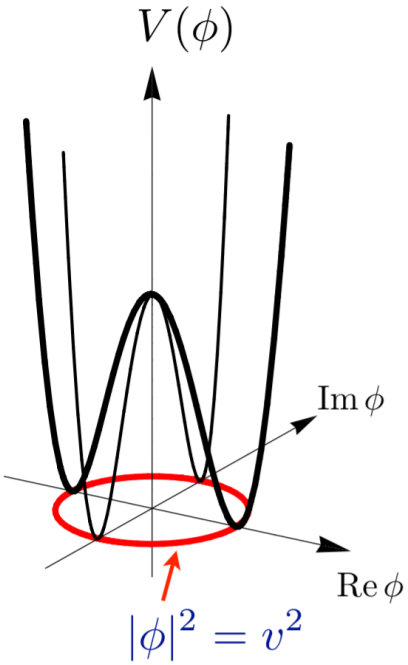


FIGURE 2.4: The vacuum expectation value of the Higgs as a function of ϕ. It has two extrema, one at $\phi = 0$, and one at $|\phi|^2 = v^2$ which corresponds to symmetry breaking.

earlier. As the energy density in the universe decreases, the Higgs field cannot maintain this unstable position, and acquires a non-zero vev. This non-zero vev results in the gauge bosons acquiring mass. We wish to find the minimum of this potential. Without loss of generality, we choose an axis such that $\langle 0|\,\phi_i\,|0\rangle = 0$, where $i = 1, 2, 4$, and $\langle 0|\,\phi_3\,|0\rangle = v \geq 0$. Thus we can write

$$\phi \to \langle 0|\,\phi\,|0\rangle \equiv v = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 \\ v \end{bmatrix} \tag{2.37}$$

$$V(\phi) \to V(v) = \frac{1}{2}\mu^2 v^2 + \frac{1}{4}\lambda v^4 \tag{2.38}$$

By choosing this axis, we have made a choice of the vacuum potential in equation 2.38. Thus to find the extrema, we take a derivative with respect to ϕ and evaluate it at v.

$$\frac{dV(v)}{dv} = v(\mu^2 + \lambda v^2) = 0 \tag{2.39}$$

We have two solutions.

- $\mu^2 > 0$ in which case $v = 0$ and the symmetry is not broken.
- $\mu^2 < 0$ in which case $v = \sqrt{\frac{-\mu^2}{\lambda}}$ and the symmetry is broken.

The mass terms can then be shown to exist after expanding the SM Lagrangian about this new minimum where $v = \sqrt{\frac{-\mu^2}{\lambda}}$.

We can finally write the part of the Lagrangian involving the scalar field ϕ as

$$\mathcal{L}_\phi = (\mathcal{D}_\mu \phi)^\dagger (\mathcal{D}^\mu \phi) - V(\phi), \tag{2.40}$$

where

$$\mathcal{D}_\mu \equiv \partial_\mu + \frac{i}{2} g_w \tau_a W_\mu^a + \frac{i}{2} g_Y Y B_\mu \tag{2.41}$$

is the covariant derivative for the $SU(2)_L \otimes U(1)_Y$ gauge group.

2.1.7 Standard Model Particle Masses

The Higgs, W and Z boson Masses

After expanding the SM Lagrangian in Eq. 2.40 about the minimum where $\mu^2 = -\lambda v^2$, we arrive at the expression

$$V(h) = -\frac{1}{2}\lambda v^2 (v+h)^2 + \frac{1}{4}\lambda (v+h)^4 \tag{2.42}$$

$$= \lambda v^2 h^2 - \frac{1}{4}\lambda v^4 + \lambda v h^3 + \frac{1}{4}\lambda h^4, \tag{2.43}$$

where h represents the Higgs field. The first term gives a mass term for the Higgs boson

$$m_h = \sqrt{2\lambda v^2} = \sqrt{-2\mu^2} > 0. \tag{2.44}$$

The last two terms represent cubic and quartic self interactions between Higgs bosons.

We can write the physical W, Z, and photon fields as linear combinations of the weak and hypercharge fields

$$W_\mu^\pm = \frac{1}{\sqrt{2}}(W_\mu^1 \mp i W_\mu^2), \tag{2.45}$$

$$Z_\mu = \cos(\theta_W) W_\mu^3 - \sin(\theta_W) B_\mu, \tag{2.46}$$

$$A_\mu = \cos(\theta_W) B_\mu + \sin(\theta_W) W_\mu^3, \tag{2.47}$$

where θ_W is the Weinberg angle which is given by the following relation

$$\cos(\theta_W) = \frac{g_w}{\sqrt{g_w^2 + g_Y^2}}. \tag{2.48}$$

The mass of the $W^{\pm}$ boson can be written as

$$m_{W^{\pm}} = \frac{1}{\sqrt{2}} v g_w, \tag{2.49}$$

and the mass of the Z boson can be written as

$$m_Z = \frac{1}{2} v \sqrt{g_w^2 + g_Y^2}. \tag{2.50}$$

Since g_w and g_Y are free parameters, the Standard Model makes no prediction for the masses of the gauge bosons.

Fermion Masses

Fermion fields have a mass term given by

$$m_\psi = m_\psi(\bar{\psi}_R \psi_L + \bar{\psi}_L \psi_R). \tag{2.51}$$

Such a term is not invariant under local $SU(2)_L \otimes U(1)_Y$ gauge transformations as the left hand terms form part of an isospin doublet whereas right hand terms are isospin singlets. In order to fix this issue, we utilise the Higgs doublet given by Eq. 2.35. The gauge invariant mass term for a fermion field is hence given by

$$\mathcal{L}_f = -\lambda_\psi(\bar{\psi}_L \phi \psi_R + \bar{\psi}_R \phi^\dagger \psi_L), \tag{2.52}$$

where λ_ψ is the fermion Yukawa coupling. Expanding this expression generates not only mass terms for the fermion field, but also interaction terms between the Higgs boson and said fermion fields.

Lepton Masses

As leptons are fermions, we begin from the gauge invariant mass term for the fermion field as in Eq. 2.52. Lepton fields form a part of an isospin doublet given by

$$L_l = \begin{bmatrix} \nu_l \\ l \end{bmatrix} \tag{2.53}$$

where $l = (e, \mu, \tau)$ are the three varieties of leptons. For the electron, one can write the interaction

$$\mathcal{L}_{e\phi} = -\lambda_e \left(\begin{bmatrix} \bar{\nu}_e & \bar{e} \end{bmatrix}_L \begin{bmatrix} \phi^+ \\ \phi^0 \end{bmatrix} e_R + \bar{e}_R \begin{bmatrix} \phi^- & \bar{\phi}^0 \end{bmatrix} \begin{bmatrix} \nu_e \\ e \end{bmatrix}_L \right). \tag{2.54}$$

One can substitute in Eq. 2.37 and get the expression

$$\mathcal{L}_{e\phi} = -\frac{\lambda_e}{\sqrt{2}}\left(v(\bar{e}_L e_R + \bar{e}_R e_L) + (\bar{e}_L e_R + \bar{e}_R e_L)H\right) \tag{2.55}$$

Hence for any given lepton,

$$\mathcal{L}_{l\phi} = -\frac{\lambda_l}{\sqrt{2}}\left(v\bar{l}l + \bar{l}lH\right). \tag{2.56}$$

The first term represents the mass term for a given lepton $m_l = \frac{\lambda_l v}{\sqrt{2}}$, while the second term represents the interaction between the Higgs boson and two lepton fields. As the Yukawa coupling λ_l is a free parameter, the Higgs mechanism does not predict masses for the leptons. Fermion masses are generated in a similar way, though one must introduce a different Higgs doublet term for the up-type quarks.

Quark Masses

The mass terms of the down-type quarks are generated as in Eq. 2.52. Up-type quarks on the other hand require a Higgs doublet with opposite hypercharge. Hence we can write the mass term as

$$\mathcal{L}_{\text{up}} = -\frac{\lambda_f}{\sqrt{2}}\left(\bar{\psi}_L \begin{bmatrix} v+h \\ 0 \end{bmatrix} \phi_R + \bar{\psi}_R \begin{bmatrix} v+h & 0 \end{bmatrix} \phi_L\right). \tag{2.57}$$

Hence the mass terms for the up and down quarks can be written as

$$\text{down-type:} \quad \frac{\lambda_d v}{\sqrt{2}}(\bar{d}_L d_R + \bar{d}_R d_L) \tag{2.58}$$

$$\text{up-type:} \quad \frac{\lambda_u v}{\sqrt{2}}(\bar{u}_L u_R + \bar{u}_R u_L). \tag{2.59}$$

The parameter $v \approx 246\text{GeV}$ is the scale that is responsible for all of the masses of the SM. We can see that the mass of any fermion in the SM can be written as

$$m_f = \frac{\lambda_f v}{\sqrt{2}}, \tag{2.60}$$

where λ_f is the Yukawa coupling that is ultimately determined by the experimentally measured particle mass.

2.2 Shortcomings of the Standard Model

While the Standard Model does explain a majority of experimental observations, there are phenomena for which it does not provide an explanation. In this section we will

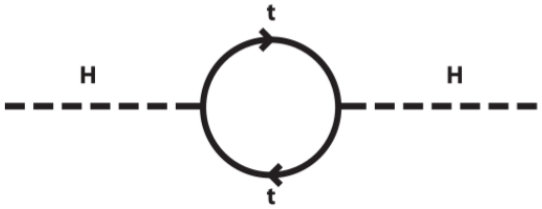


FIGURE 2.5: Feynman diagram of a one-loop correction to the Higgs in the SM.

outline a few issues that are relevant to the particular Standard Model extensions utilised throughout this thesis. One of the biggest issues is that it does not provide a suitable dark matter candidate. "Dark matter" is a term for a hypothetical form of matter which is usually thought to be composed of some as-of-yet undiscovered particle. Its presence has been observed through astronomical gravitational effects, which can be explained by introducing the presence of more matter than can be observed in the visible wavelengths. The Standard Model also does not incorporate gravity, which is a very weak force relative to the other forces on microscopic scales.

The Standard Model also fails to explain the hierarchy problem [10] which is a problem with the mass of the Higgs. This is due to quadratically divergent terms in the mass loop corrections. As we will see, this problem is solved by the introduction of the Minimal Supersymmetric Standard Model.

2.2.1 The Hierarchy Problem

As we have stated, the Standard Model is a low energy effective theory that works up to some cut-off scale which we will denote Λ. Figure 2.5 presents a Feynman diagram of the dominant loop corrections to the Higgs boson propagator from the top quark, which can be shown to shift the Higgs mass by

$$\Delta m_H^2 = -\frac{|\lambda_t|^2}{8\pi^2}\Lambda^2 + \dots \tag{2.61}$$

where λ_t is the Yukawa coupling of the Higgs to the top quark.

The mass of the Higgs boson is measured to be 125 GeV [11], while these quadratic mass corrections are $\mathcal{O}(\Lambda^2)$. Assuming that the SM is valid up to the Planck scale where quantum gravity begins to have an effect, these mass corrections are around 10^{15} GeV. Unless there is an incredibly large fine-tuning cancellation between these quadratic corrections and the bare mass, one would expect the Higgs mass to be very high. These enormous cancellations in the various contributions to the Higgs mass are what is referred to as the hierarchy problem.

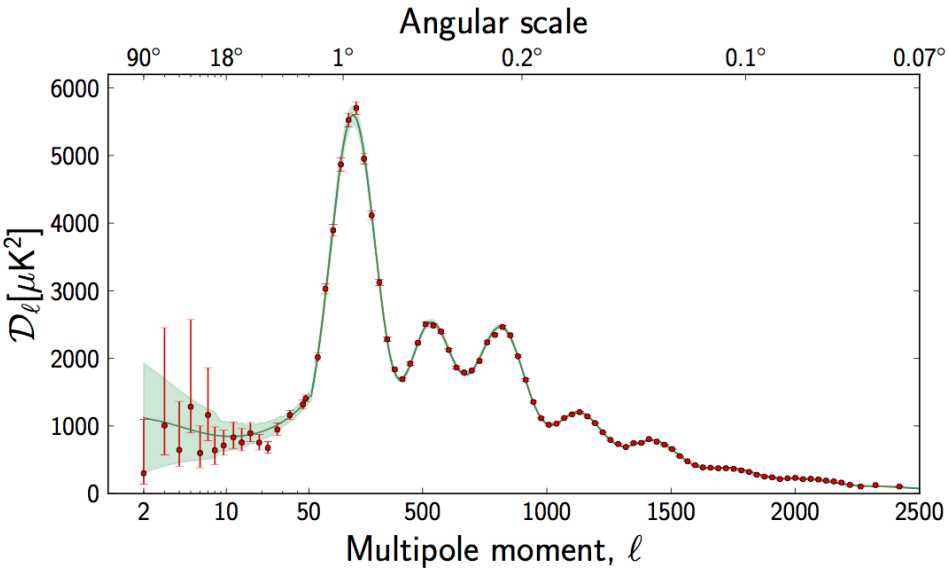


FIGURE 2.6: The power spectrum of temperature fluctuations in the CMB and its fit using the ΛCDM model [18].

2.2.2 Dark Matter

Current models predict approximately 85% of the mass in the universe is of a form that is yet unknown [12]. The earliest signs of dark matter (DM) [13] came from observations of the velocity distribution of stars in the Milky Way in the 1930s. The velocity as a function of the radius in a spiral galaxy should yield Keplerian behaviour, where the velocity should gradually fall off as the radius increases. What we observe instead is a roughly constant rotation velocity, implying a large unseen mass density distributed at the edges of the galaxy [14]. Other evidence for DM comes in the form of gravitational lensing [15], where light is bent due to a large presence of mass. An Einstein ring is created when light from behind a large mass is distorted by its gravitational pull. The radius of this ring, referred to as the Einstein radius, can be used to determine the mass of the distorting body. When looking at distant galaxy clusters, the mass required to reproduce the Einstein radius is much larger than what is inferred by the luminosity of the cluster. Using this information it is possible to map the dark matter distributions of distant galaxies [16]. The cosmic microwave background (CMB) [17] also contains evidence of dark matter. When observing the power spectrum of temperature fluctuations in the CMB, the angular scale and height of each peak are used to probe various cosmological parameters including the fraction of dark matter in the universe. This is shown in Figure 2.6. Matching all of the peaks implies that 26% of the total mass in the universe comes from dark matter, and 5% comes from atoms in the form of stars and galaxies.

Theoretical explanations for DM are numerous and span a wide range of possible masses, from very light axions to extremely heavy black holes. It is also possible that

DM is not made up of one single particle, but of numerous particles. In this thesis, I focus on a class of DM theories known as Weakly Interacting Massive Particles (WIMPs). Theoretical DM particles must follow a few basic rules: they must be stable, otherwise they would decay into SM particles. They must have no electric charge, and be colourless, otherwise they would form strongly bound states, and be easily detectable via direct detection experiments like the XENON1T experiment [19]. DM must also reproduce the observed relic density [20], which is the observed density of non-baryonic DM divided by the total critical density, written as

$$\Omega_{DM}h^2 = h^2\frac{\rho_{DM}}{\rho_C} \approxeq 0.1186 \pm 0.0020, \tag{2.62}$$

where $h^2 \approxeq 0.5$ is the Hubble constant in units of $100\text{kms}^{-1}\text{Mpc}^{-1}$. $\rho_C = \frac{3H_0}{8\pi G_N} \approxeq 10h^2\text{GeVm}^{-3}$ yields a spatially flat homogeneous universe and $\rho_{DM} \approxeq 1.2 \times 10^{-6}\text{GeVcm}^{-3}$ is roughly equal to 6 protons every 5m^3.

WIMPs can come from many well motivated BSM theories, and a GeV scale WIMP with a weak interaction cross section is able to reproduce the observed DM abundance. In the next section we explore supersymmetry, which is one of the most powerful BSM theories, and is able to fill many of the gaps which are observed in the SM.

2.3 Overview of Supersymmetry and the MSSM

Supersymmetry (SUSY) [10] is a theoretical extension to the Standard Model which introduces supersymmetric partners for every Standard Model particle. Each pair of superpartners has identical quantum numbers, except for a half-unit difference in spin, meaning fermions have bosonic superpartners and bosons have fermionic superpartners.

The Minimal Supersymmetric Standard Model (MSSM) is a particular supersymmetric extension of the Standard Model which considers the minimum number of particle states and interactions necessary to obtain a theory that can yield results that agree with modern experiments. A supersymmetry transformation turns a bosonic state into a fermionic state and vice versa. This can be denoted with an operator Q which generates such transformations.

$$Q\left|Boson\right\rangle = \left|Fermion\right\rangle, Q\left|Fermion\right\rangle = \left|Boson\right\rangle \tag{2.63}$$

We know that if SUSY exists, it must be broken at low energies. If SUSY were a perfect symmetry, we would already have observed superpartners with the same mass as the SM counterparts. It is thought that, below some energy scale, the masses of the superpartners

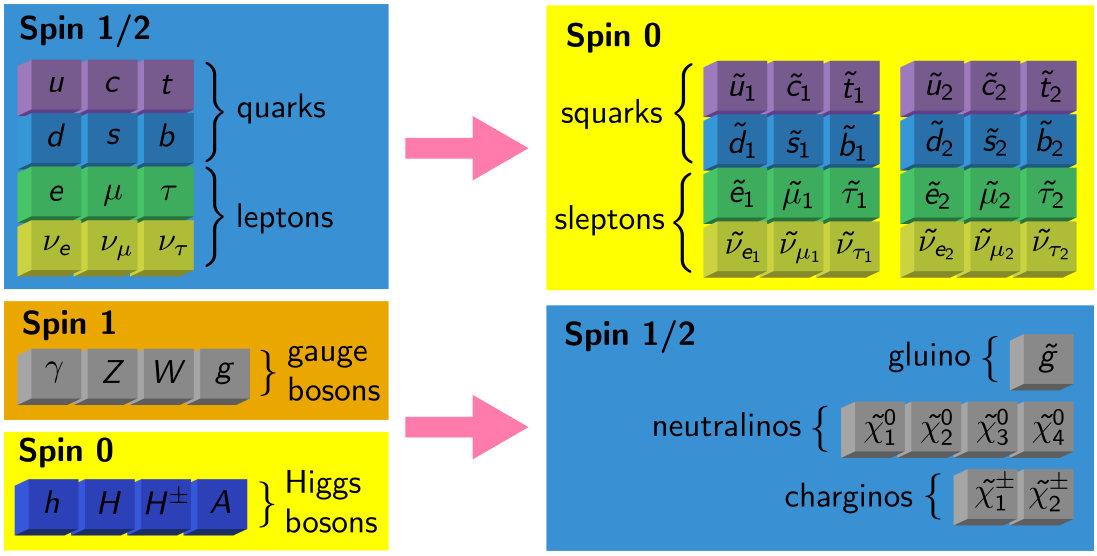


FIGURE 2.7: The particle content of the MSSM.

must be much higher than that of their SM partners. Above this energy scale, however, the masses of the superpartners approach that of their SM partners.

2.3.1 Particle Content of the MSSM

The superpartners of the left and right-handed chiral components of fermionic SM particles will mix to form two mass eigenstates, yielding two bosonic superpartners. Bosonic SM particles will only have one fermionic superpartner. The naming convention for these superpartners is to add an "s-" prefix for bosonic SUSY particles and an "-ino" suffix for fermionic SUSY particles. Additionally it is convention to mark SUSY particles with a tilde. Figure 2.7 presents a table of the Standard Model, as well as its minimally supersymmetric extension.

The mixing of the left and right-handed chiral components of the supersymmetric top quark (the stop) can be represented by the following matrix equation:

$$\begin{bmatrix} \tilde{t}_1 \\ \tilde{t}_2 \end{bmatrix} = \begin{bmatrix} \cos(\theta_t) & \sin(\theta_t) \\ -\sin(\theta_t) & \cos(\theta_t) \end{bmatrix} \begin{bmatrix} \widetilde{t_L} \\ \widetilde{t_R} \end{bmatrix} \tag{2.64}$$

for some mixing angle θ_t. This mixing of left and right-handed chiral components is similar for all sfermionic particles. Whilst the top quark is by far the heaviest quark, the light stop quark is typically the lightest of the squarks in order to cancel the fermionic component of the Higgs field, and solve the hierarchy problem. This means that it should be the easiest to observe in nature.

Before SUSY is broken we have superpartners for the gauge bosons called "gluinos" ($\tilde{g}$), "winos" ($\widetilde{W}^{\pm}, \widetilde{W}^0$) and "binos" ($\widetilde{B}^0$). We also have more Higgs bosons in the MSSM.

Unlike the SM, the MSSM has two Higgs doublets, which can be written in the following way:

$$H_u = \begin{bmatrix} H_u^+ \\ H_u^0 \end{bmatrix}, \qquad H_d = \begin{bmatrix} H_d^0 \\ H_d^- \end{bmatrix} \tag{2.65}$$

where the labels u, and d correspond to the Yukawa couplings to up and down type fermions. Upon breaking, the winos, binos, and Higgsinos mix together to form the neutralinos and charginos that are present in Figure 2.7. This mixing can be represented by the following matrix equation:

$$\begin{bmatrix} \widetilde{\chi}_1^0 \\ \widetilde{\chi}_2^0 \\ \widetilde{\chi}_3^0 \\ \widetilde{\chi}_4^0 \end{bmatrix} = \begin{bmatrix} N_{11} & N_{12} & N_{13} & N_{14} \\ N_{21} & N_{22} & N_{23} & N_{24} \\ N_{31} & N_{32} & N_{33} & N_{34} \\ N_{41} & N_{42} & N_{43} & N_{44} \end{bmatrix} \begin{bmatrix} \widetilde{W}^0 \\ \widetilde{B}^0 \\ \widetilde{H}_u^0 \\ \widetilde{H}_d^0 \end{bmatrix} \tag{2.66}$$

We denote the left hand side of Equation 2.66 χ_i^0 with $i \in \{1, 2, 3, 4\}$. N_{ij} are the mixing parameters, and are constrained by $N_{i1}^2 + N_{i2}^2 + N_{i3}^2 + N_{i4}^2 = 1$ in order to ensure the unity of probability. The neutralinos are ordered in increasing mass, with χ_1^0 being the lightest and χ_4^0 being the heaviest.

The formation of the positive charginos can be represented by:

$$\begin{bmatrix} \widetilde{\chi}_1^+ \\ \widetilde{\chi}_2^+ \end{bmatrix} = \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix} \begin{bmatrix} \widetilde{W}^+ \\ \widetilde{H}^+ \end{bmatrix} \tag{2.67}$$

Similarly, the formation of the negative charginos can be represented by:

$$\begin{bmatrix} \widetilde{\chi}_1^- \\ \widetilde{\chi}_2^- \end{bmatrix} = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} \begin{bmatrix} \widetilde{W}^- \\ \widetilde{H}^- \end{bmatrix} \tag{2.68}$$

2.3.2 Soft SUSY breaking in the MSSM

As mentioned prior, SUSY must necessarily be broken at low energies. In order to do this we add a "soft" Lagrangian, which breaks the symmetry only at low energies. The full MSSM Lagrangian density is given by

$$\mathcal{L}_{MSSM} = \mathcal{L}_{SUSY-SM} + \mathcal{L}_{soft}, \tag{2.69}$$

where $\mathcal{L}_{SUSY-SM}$ is obtained by sypersymmetrising the SM Lagrangian density. Let us now look at the term $\mathcal{L}_{soft}$. This contains all renormalisable soft SUSY-breaking terms

which conserve baryon number B, and lepton number L. It can be written as

$$\mathcal{L}_{soft} = -\frac{1}{2}(M_1\bar{\widetilde{B}}\widetilde{B} + M_2\bar{\widetilde{W}}\widetilde{W} + M_3\bar{\widetilde{g}}\widetilde{g} + c.c.) \tag{2.70}$$

$$- (\widetilde{Q}^\dagger \boldsymbol{m_Q^2} \widetilde{Q} + \widetilde{L}^\dagger \boldsymbol{m_L^2} \widetilde{L} + \widetilde{u}^\dagger \boldsymbol{m_u^2} \widetilde{u} + \widetilde{d}^\dagger \boldsymbol{m_d^2} \widetilde{d} + \widetilde{e}^\dagger \boldsymbol{m_e^2} \widetilde{e}) \tag{2.71}$$

$$- (\widetilde{u}\boldsymbol{A_u}\widetilde{Q}H_u - \widetilde{d}\boldsymbol{A_d}\widetilde{Q}H_d - \widetilde{e}\boldsymbol{A_e}\widetilde{L}H_d + c.c.) \tag{2.72}$$

$$- (m_{H_u}^2|H_u|^2 + m_{H_d}^2|H_d|^2 + (bH_uH_d + c.c.)). \tag{2.73}$$

The parameters of this Lagrangian density dictate the behaviour of a given MSSM model once SUSY is broken. It is common to only examine a subset of these parameters in analyses, freezing a number of them in order to work with a smaller parameter set. The first set of terms, line 2.70, contains the parameters M_1, M_2, and M_3. These are the bino, wino, and gluino mass parameters. There is a second term with parameters M_1', M_2', and M_3' that is not shown here, as it violates CP and so must thus be very small. The second set of terms, line 2.71, contains squark and slepton mass terms. $\boldsymbol{m_Q^2}$, $\boldsymbol{m_L^2}$, $\boldsymbol{m_u^2}$, $\boldsymbol{m_d^2}$, and $\boldsymbol{m_e^2}$ are 3×3 Hermitian mass-squared matrices. The third set of terms, line 2.72, contains the three trilinear scalar interaction terms $\boldsymbol{A_u}$, $\boldsymbol{A_d}$, and $\boldsymbol{A_e}$. Each of these is a complex 3×3 matrix in one-to-one correspondence with the Yukawa couplings. These denote the triliear couplings between the Higgs and squarks or sleptons. Finally, line 2.73 contains the real Higgs sector mass terms $m_{H_u}^2$, and $m_{H_d}^2$, as well as a bilinear coupling with a complex parameter b.

2.3.3 R-Parity

Baryon number (B) and Lepton number (L) are explicitly required to be conserved in the SM. This is in order for the proton to be stable, as proton decay has not ever been observed. The MSSM, however, does not explicitly require that Baryon Number and Lepton Number are conserved. Instead it uses what is known as R-Parity to do so. R-Parity is a $\mathbb{Z}_2$ symmetry which is added to the MSSM, and is defined as:

$$P_R = (-1)^{3(B-L)+2s}, \tag{2.74}$$

where s is the spin of a particle. All SM particles have $P_R = 1$, and all MSSM particles have $P_R = -1$. Conservation of R-Parity has the consequence that the lightest supersymmetric particle (LSP) cannot decay into lighter SM particles, as this would violate R-Parity. This means that the LSP does not decay at all, which makes it a perfect dark matter candidate. R-Parity also requires that all sparticles are produced in pairs at the LHC, and that each sparticle decay chain ends in an odd number of LSPs. Since the LSP only interacts through the weak force, and does not decay, it is impossible to directly observe.

2.3.4 Supersymmetry's Solution to the Hierarchy Problem

We now return to the hierarchy problem, and state how it is solved with the introduction of superpartners. Recall the form of the Higgs mass loop corrections:

$$\Delta m_H^2 = -\frac{|\lambda_f|^2}{8\pi^2}\Lambda^2 + \dots \tag{2.75}$$

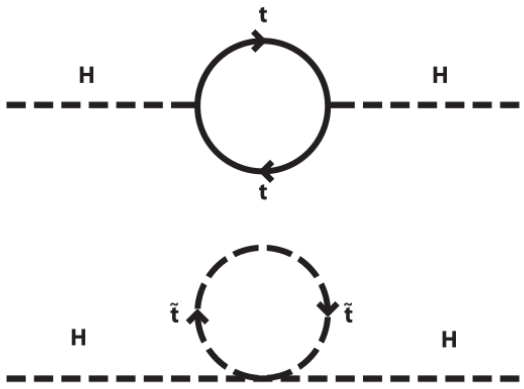


FIGURE 2.8: Feynman diagram of Higgs loop corrections in the MSSM.

The Higgs boson's coupling to a given fermion is proportional to its mass. As such, the top quark, being the heaviest quark, is the dominant particle in this process. With the addition of bosonic superpartners for the top quark, these quadratically divergent contributions cancel precisely. This is due to fermionic loop contributions incurring a minus sign relative to bosonic loop contributions. Hence, the total Higgs mass loop correction in the MSSM is given by:

$$\Delta m_H^2 = m_{soft}^2 \left(\frac{\lambda^2}{16\pi^2} \ln\left(\frac{\Lambda}{m_{soft}}\right) + \dots \right) \tag{2.76}$$

where m_{soft}^2 is the mass scale at which supersymmetry is broken. Figure 2.8 displays these loop corrections with SUSY present.

2.3.5 Decay Phenomenology

Now that we have a basic understanding of SUSY and the MSSM, we examine the decay phenomenology of the main MSSM particles. Here we denote leptons, neutrinos and quarks using the generic labels l, ν, and q.

Neutralino Decays

Neuralino decay modes can be denoted as

$$\widetilde{\chi}_j^0 \to Z\widetilde{\chi}_k^0,\ W^{\pm}\widetilde{\chi}_i^{\mp},\ h^0\widetilde{\chi}_k^0,\ \widetilde{ll},\ \widetilde{\nu\nu},\ A^0\widetilde{\chi}_k^0,\ H^0\widetilde{\chi}_k^0,\ H^{\pm}\widetilde{\chi}_i^{\mp},\ \widetilde{qq} \tag{2.77}$$

where $j \in [1,2,3,4]$, $i \in [1,2]$, and $k < j$. The first five decay modes, involving the Z, $W^{\pm}$, and h^0 bosons, as well as lepton and neutrino decays, are the dominant decay modes when permitted. If two body decays are prohibited then the dominant decay modes will involve off-shell Z, $W^{\pm}$, or h^0 boson decays. Searches that target these particles typically involve multiple leptons as the dominant decay modes include Z and $W^{\pm}$ bosons.

Chargino Decays

Chargino decay modes can be denoted as

$$\widetilde{\chi}_j^{\pm} \to Z\widetilde{\chi}_k^{\pm},\ W^{\pm}\widetilde{\chi}_i^0,\ h^0\widetilde{\chi}_k^{\pm},\ l\widetilde{\nu},\ \nu\widetilde{l},\ A^0\widetilde{\chi}_k^{\pm},\ H^0\widetilde{\chi}_k^{\pm},\ H^{\pm}\widetilde{\chi}_i^0,\ q\widetilde{q}', \tag{2.78}$$

where $j \in [1,2]$, $i \in [1,2,3,4]$, and $k < j$. Here, q and $\widetilde{q}'$ are different flavours of quark. Three-body decays via off-shell $W^{\pm}$, Z, or h^0 bosons are dominant when two body decays are prohibited. The most favoured decay mode is $\widetilde{\chi}_1^{\pm} \to W^{\pm}\widetilde{\chi}_1^0$ via an on or off shell $W^{\pm}$ boson.

Squark Decays

Squark decay modes can be denoted as

$$\widetilde{q} \to q\widetilde{g},\ q\widetilde{\chi}_i^0,\ q'\widetilde{\chi}_j^{\pm}, \tag{2.79}$$

where $i \in [1,2,3,4]$, and $j \in [1,2]$. Due to the QCD strength of the vertex, the first mode is dominant. Right-handed squarks are more likely to decay to the LSP, while left-handed quarks are more likely to decay to intermediate states involving the chargino and neutralino. The third generation of squarks are more likely to decay to Higgsino-dominated electroweakinos than the other two generations due to a larger Yukawa coupling. The lightest squark, the $\widetilde{t}$, is worth noting as when the mass difference between the $\widetilde{t}$ and a $\widetilde{\chi}_i^0$ is small, the stop decays are dominated by the following processes

$$\widetilde{t}_1 \to t\widetilde{\chi}_i^0,\ b\widetilde{\chi}_j^{\pm},\ bW\widetilde{\chi}_i^0,\ c\widetilde{\chi}_i^0, bW^*\widetilde{\chi}_i^0. \tag{2.80}$$

Slepton Decays

Slepton decays can be denoted as

$$\widetilde{l}^{\pm} \rightarrow l^{\pm}\widetilde{\chi}_i^0,\ \nu\widetilde{\chi}_j^{\pm}, \tag{2.81}$$

$$\widetilde{\nu} \rightarrow l\widetilde{\chi}_i^0,\ l^{\mp}\widetilde{\chi}_j^{\pm}, \tag{2.82}$$

where $i \in [1, 2, 3, 4]$, and $j \in [1, 2]$. Right-handed sleptons will generally decay to the LSP $\widetilde{\chi}_1^0$, while left-handed sleptons will prefer decay channels containing heavier electroweakinos.

Gluino Decays

Gluino decays are the simplest and can be denoted as

$$\widetilde{g} \rightarrow q\widetilde{q}^{(*)}. \tag{2.83}$$

These decay modes can only take place through a real or virtual squark and are typically dominated by modes containing $\widetilde{t}_1$ and $\widetilde{b}_1$ as they are assumed to be the lightest squarks in many SUSY scenarios.

2.3.6 The Current State of SUSY

Recent experiments have found no evidence for the existence of SUSY [21]. Current exclusion limits are dominated by results from high energy particle colliders which are used to search for these high mass BSM particles. However, these exclusion limits are often plotted in a 2-D plane of two particular SUSY particle masses, leaving the masses of the other sparticles frozen at arbitrary values. These vanishingly thin hyperplanes of the total parameter space do give some insight into more general exclusion limits, but they are not the whole picture. This is explored in more detail in Chapter 7.

Figures 2.9 and 2.10 show the most up to date exclusion curves in two different mass planes from the ATLAS experiment. Figure 2.9 explores gluino decays, while Figure 2.10 explores the process $\widetilde{\chi}_2^0\widetilde{\chi}_1^{\pm} \rightarrow Wh\widetilde{\chi}_1^0\widetilde{\chi}_1^0$. In order to better understand these exclusions, and eventually develop our own analyses using novel data analysis techniques, we must first explore how modern particle collider experiments work.

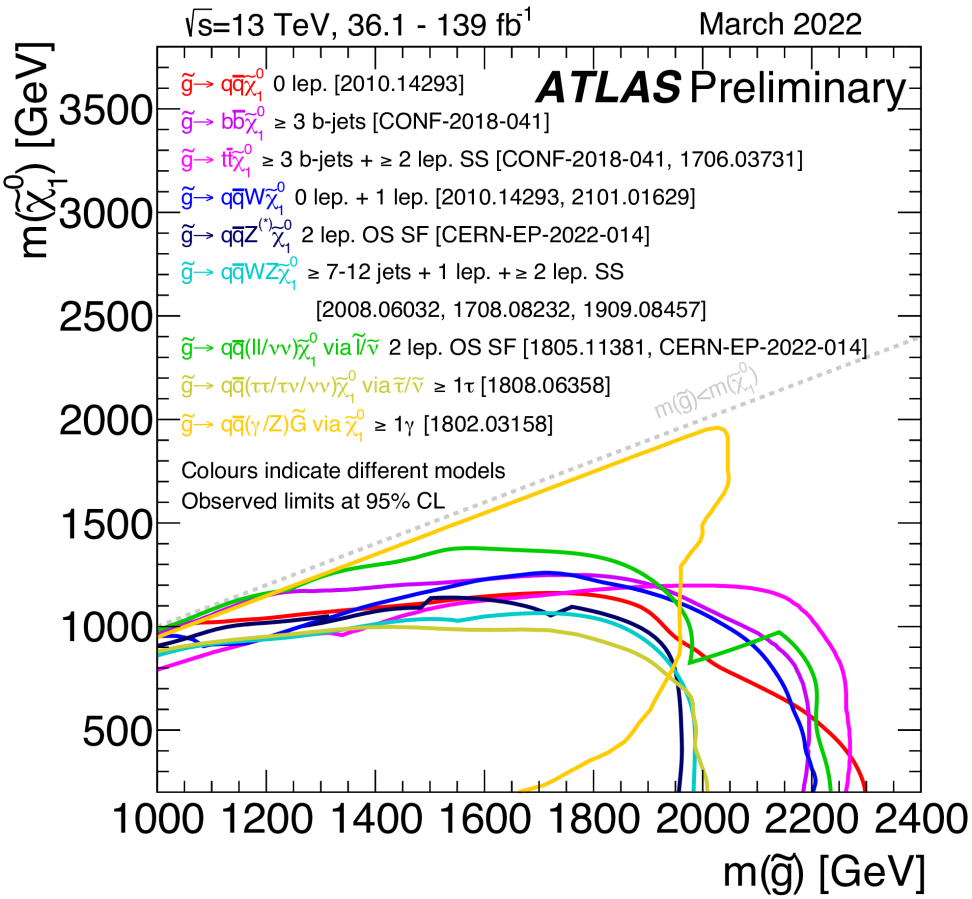


FIGURE 2.9: Exclusion limits at 95% CL based on 13 TeV data in the $\widetilde{\chi}_0^1$-$\widetilde{g}$ mass plane for a number of BSM searches [22].

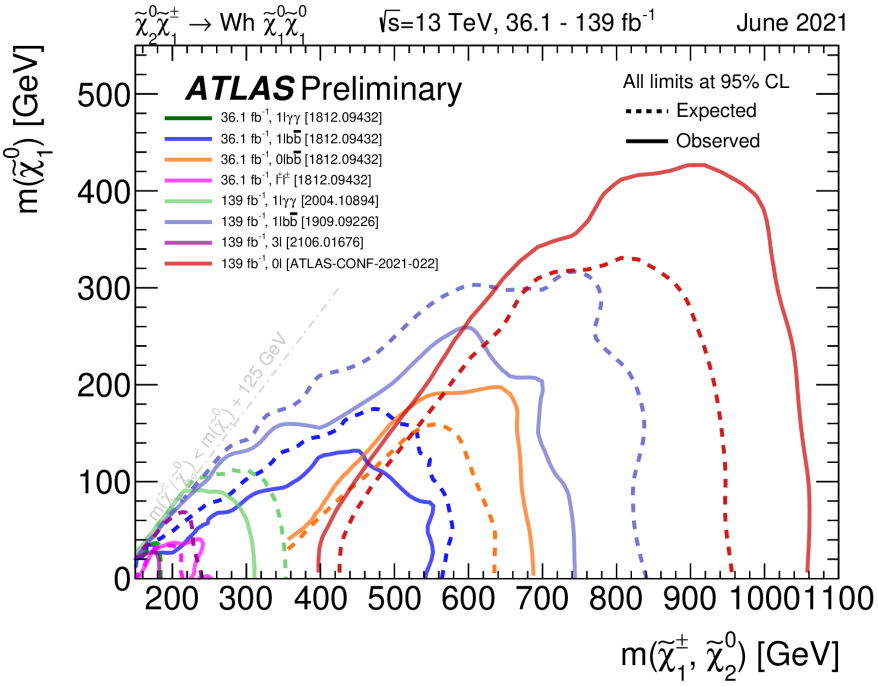


FIGURE 2.10: Exclusion limits at 95% CL based on 13 TeV data in the $\tilde{\chi}_0^1$-$(\tilde{\chi}_1^\pm, \tilde{\chi}_2^0)$ mass plane for a number of BSM searches [22].

3 BSM Searches at High Energy Collider Experiments

There are many experiments around the world geared towards searching for BSM physics. For example, different experiments use different methods of dark matter detection, represented by Figure 3.1. Direct detection experiments, such as LUX [23], and Xenon1T [24] search for DM by examining the recoil of SM nuclei due to collisions with DM particles. Indirect detection experiments, such as Pamela [25], AMS [26], IceCube [27], and Fermi-LAT [28] examine DM annihilation to SM particles such as γ rays. Collider experiments search for the direct production of DM in high energy particle collisions. These experiments presuppose the possibility of producing DM particles from SM particles and attempt to observe some inconsistency with the known SM background. Colliders must run at very high energies, as according to the Einstein formula $E \propto m$, a particle of mass m requires enough energy E to be produced by colliding particles. Some BSM theories require that particles be produced in pairs, further restricting their mass limits. Higher energies mean more massive particles can be produced.

The number of events for any given process in a particular final state that one might observe in a collider experiment can be expressed as

$$N = \sigma \times BR \times \epsilon \int \mathcal{L} dt, \tag{3.1}$$

where σ is the production cross section of the process, BR is the decay branching ratio to the channel yielding the final state, ϵ represents the efficiencies and acceptances on the reconstruction of each object in the final state (this is covered in more detail in Section 3.2), and $\int \mathcal{L} dt$ is the luminosity of the detector integrated over time.

The goal in any BSM physics search is to discriminate signal-like events from background-like events and compare the number of observed and expected events. At collider experiments this is done by performing various selection cuts on parameters such as the number of observed leptons, or the missing transverse momentum of an event.

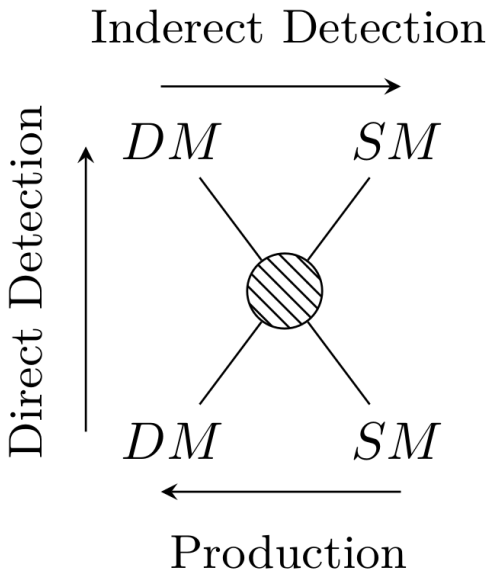


FIGURE 3.1: Dark Matter search techniques.

3.1 The Large Hadron Collider

The Large Hadron Collider (LHC) [29] is the largest particle accelerator in the world, with a circumference of 27 kilometers and ran at a centre-of-mass energy of 13 TeV from 2015-2018. Protons are accelerated to speeds approaching the speed of light and smashed together in the middle of the various detectors located about the ring where the resulting "stuff" is collected and examined.

The protons are supplied in the form of a hydrogen gas. The electrons are stripped from the atoms by an electromagnetic field and the protons are then accelerated to an energy of 450 GeV by a series of linear and synchrotron accelerators. The proton beam is then split into two, each orbiting in opposite directions through the main tunnel. The proton beams are then accelerated to their peak energy of 6.5 TeV and then collided at various interaction points. There are a total of four interaction points where protons are collided corresponding with the four main detectors, ATLAS [30], CMS [31], ALICE [32], and LHCb [33].

ATLAS (A Toroidal LHC ApparatuS), and CMS (Compact Muon Solenoid) are the detectors that most of the work in this thesis is focused on. ATLAS is designed to be a general purpose detector, aimed at measuring a broad range of signals rather than focusing on a particular physical process. The ATLAS detector consists of various layers of sub-detectors, each designed to target a specific phenomena. The innermost layer is designed to take high precision measurements of the position and momentum of charged particles. The next layers, the calorimeters, measure the energies of easily stopped particles. The outer layer is the muon spectrometer, designed to measure high energy muons.

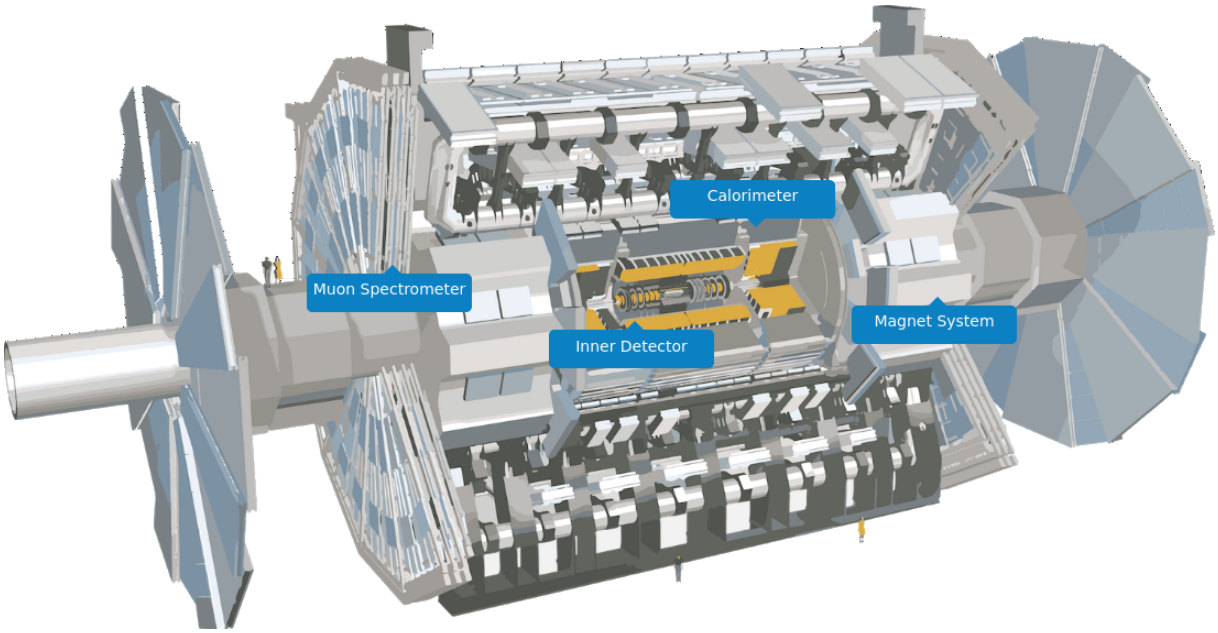


FIGURE 3.2: The ATLAS detector, and the locations of the inner detector, calorimeter, magnet system, and muon spectrometer. People included for scale.

The momenta of charged particles are able to be measured with the aid of the magnet systems, which bend charged particles in the inner and muon detectors proportional to their momenta. Each detector also has a series of "triggers" which are used to limit the amount of data that is recorded by the detector. A trigger imposes a set of requirements on each event to rapidly decide which events should be kept. This is necessary when it is not possible to record the results of every single collision at the LHC due to the rate of incoming data and storage capacity. Some triggers are designed specifically to target certain final states, such as requiring at least one electron or muon. By using these triggers, it is possible to not only reduce the amount of data to be stored, but also efficiently target only the events of interest for a given analyses.

3.1.1 What's in an LHC Event?

Recall the last term of Equation 3.1. This is the integral over time of the instantaneous machine luminosity, which is written as

$$\mathcal{L} = \frac{N_b^2 n_b f_{rev} \gamma_r}{4\pi \epsilon_n \beta^*} F \tag{3.2}$$

where N_b is the number of particles per bunch, n_b is the number of bunches per beam, f_{rev} is the number of revolutions per second, γ_r is the relativistic gamma factor, ϵ_n is the normalised transverse beam emittance, β^* is the beta function at the collision point, and F is a geometric reduction factor due to the non-zero angle of incidence between the

beams at the interaction point [29]. The ATLAS and CMS experiments both aim for a high peak luminosity of $\mathcal{L} = 10^{34}\text{cm}^2\text{s}$. This luminosity factor determines how many events are produced, and hence, how sensitive the experiment is to various BSM signals, since collecting more data allows rarer processes to be observed.

Each particle seen by ATLAS has a measured energy-momentum 4-vector which can be written as $\vec{p} = (\frac{E}{c}, p_x, p_y, p_z)$. Note that the proton beams travel along the z axis with the interaction point at the origin. Since the proton is a composite object, composed of quarks and gluons, the centre of mass frame of the collision is different to the centre of mass frame of the quarks which interact with each other. For this reason, it is necessary to work with variables which are longitudinally boost invariant. The transverse momentum, defined as $\vec{p}_T = (p_x, p_y)$, is one such quantity. The two angular components used to define the 4-vectors typically used in an analysis are η and ϕ. Looking at the coordinates, shown in Figure 3.3, we see that ϕ is defined as the angle about the beam axis in the x y plane. The other angular variable, η is defined using the angle to the beam axis θ and can be written as

$$\eta = -\log\left[\tan\left(\frac{\theta}{2}\right)\right]. \tag{3.3}$$

This quantity is referred to as the "pseudo-rapidity", and is useful because the difference in η between two particles, $\Delta\eta$, is invariant under longitudinal boosts. The difference in ϕ between two particles is also a Lorentz invariant quantity, and is used to construct various physical variables which are detailed in Section 3.3.1. Another important variable is the missing transverse momentum defined as

$$p_T^{miss} = -\sum_i \vec{p}_{Ti}, \tag{3.4}$$

where $\vec{p}_{Ti}$ refers to the transverse momentum of the ith particle. This quantity gives the transverse momentum of particles not picked up by the detector and is very important in DM searches due to escaping DM particles.

Recall that particles with colour charge cannot exist on their own due to the requirement that particles exist in colourless states. In order to conserve colour charge, quarks combine with other particles in a process known as "hadronisation", forming jets of mesons and baryons. These jets are commonly used to analyse the physical properties of the original interaction.

3.2 Monte Carlo Simulation

In order to compare theory to experiment one must use simulated Monte Carlo (MC) data. This method also allows one to simulate BSM signals to tune analyses on. In this

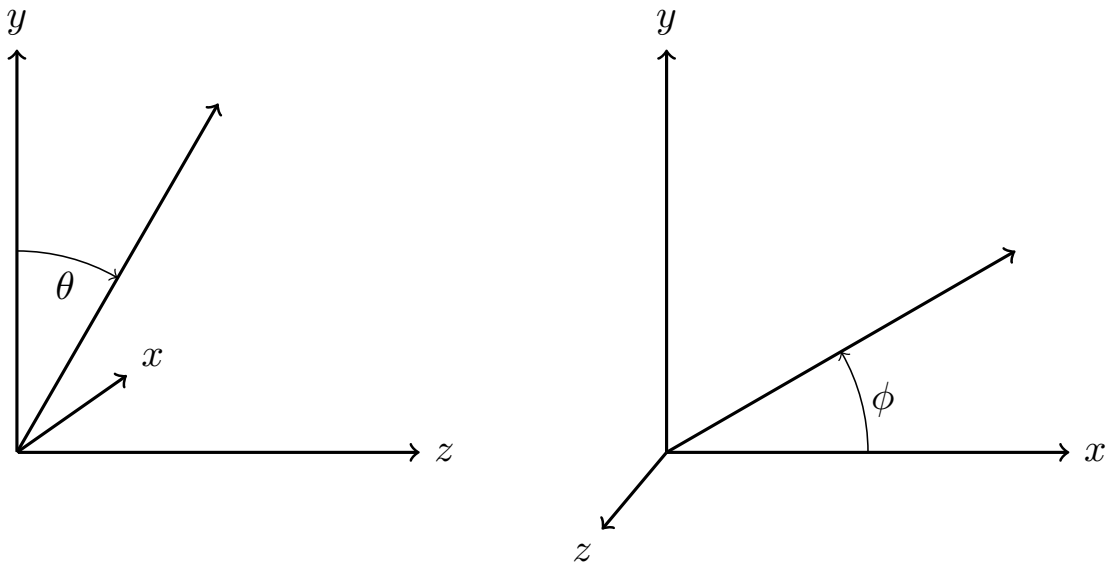


FIGURE 3.3: The coordinate system typically used at the LHC, where the z axis indicates the beam direction.

thesis, I primarily use a dataset described in detail in Section 5.1. This dataset is made up of simulated proton-proton collisions at $\sqrt{s} = 13$ TeV generated at leading order with up to two extra partons (quarks) in `Madgraph` [34]. The parton showering and hadronisation is done in `Pythia 8.2` [35], and the detector simulation is done in `Delphes 3` [36].

In order to avoid phase space overlap between `Madgraph` and `Pythia` when extra partons are generated, one must employ a parton-jet matching procedure. The one used in the datasets described in this thesis is the k_T-jet MLM scheme. This algorithm clusters final-state partons at the matrix-element level. The cutoff scale `xqcut` determines the smallest value of k_T a parton is allowed to have. After these final-state partons are showered by Pythia, the final-state partons are clustered to form jets once again using the k_T algorithm before they are hadronised. A different cutoff scale `QCUT` is defined for this second round of clustering. The jets are then compared to the partons, matching them when the measured k_T is less than `QCUT`. When each jet is matched with a parton, the event is not discarded unless extra jets are allowed. A non-matched event occurs if the initial final-state partons are too close to create a unique jet or if a single parton has too little transverse momentum to be reconstructed as a jet.

3.2.1 Definition of Physical Objects

The job of simulating the detector and identifying the physical final state objects is done by Delphes. Delphes simulates the detector response by recreating the tracking of a charged particle moving through the detector. The electromagnetic and hadronic energy deposited in the detector by the particle are independently smeared using a log-normal

distribution. This section discusses the reconstruction requirements, efficiencies, and some key physical features for use in analyses for each particle type.

Isolation of a particle is a fundamental part of identifying it. In order to remove contributions from jet backgrounds, a particle $P = e, \mu, \gamma$ is defined to be isolated if the following ratio is satisfied for the i nearby particles.

$$\frac{1}{p_T(P)} \sum_{i \neq P}^{\Delta R < R, p_T(i) > p_T^{min}} p_T(i) < I^{min}. \tag{3.5}$$

Here the following parameters are defined as $I^{min} = 0.1$, $R = 0.3$, and $p_T^{min} = 0.1$ GeV. Particles which fail to be isolated, are referred to as misidentified or non-prompt leptons. These leptons can be hadrons mimicking lepton signatures, or leptons produced in in-flight decays of hadrons. Simulation of the rate of misidentified or non-prompt electrons, muons, and photons requires very large statistics, and a much more detailed detector simulation. This is not implemented in Delphes 3.

Photons

A photon object is defined when a true photon or an electron without a reconstructed track reaches the electromagnetic calorimeter and is isolated with $p_T > 10$ GeV. The efficiencies are

- $\epsilon = 0.9635$ for $|\eta| \leq 1.5$
- $\epsilon = 0.9624$ for $1.5 < |\eta| \leq 2.5$.

Photons are occasionally used in chargino-neutralino searches where the second neutralino decays to two photons via a higgs boson, plus the lighest neutralino. Photons have also been used to search for gauge-mediated supersymmetry breaking (GSMB) scenarios which allow the lightest neutralino to decay to gravitinos (the superpartner of the theoretical graviton) plus photons. In this scenario the gravitino is the LSP and the photons are generated by a Higgs boson produced alongside the LSP.

Electrons

An electron object is defined when isolated with transverse momentum $p_T > 10$ GeV. The efficiencies are given by

- $\epsilon = 0.98$ for $|\eta| \leq 1.5$
- $\epsilon = 0.90$ for $1.5 < |\eta| \leq 2.5$.

In this thesis, electrons will be important in SUSY searches, particularly those using a 3-lepton search strategy which targets processes that create $W^{\pm}$ and Z bosons together.

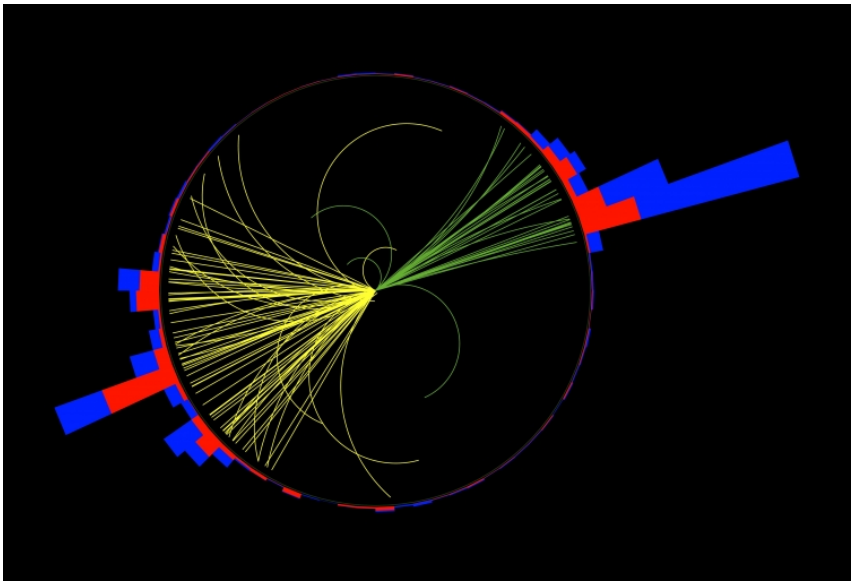


FIGURE 3.4: A scattering event from CMS showing jets of particles resulting from two top quarks.

Muons

A muon object is defined when isolated with transverse momentum $p_T > 10$ GeV. The efficiencies are identical to those of the electrons. Muons are the only SM particles to interact with the outer layers of the CMS and ATLAS detectors, and the final momentum is calculated by performing a Gaussian smearing of the initial four-momentum vector. Muons are used in a similar fashion to electrons in this thesis, being a core component of a 3-lepton search strategy.

Jets

Jets are some of the most complicated objects produced at the LHC. A jet is formed from a cascade of hadrons which tend to arrange themselves in collimated "jets" of particles. Figure 3.4 shows an image from CMS of an event which contains two jets from top quarks. These jets deposit their energy in the hadronic calorimeter and are identified using a jet-finding algorithm.

b-jets are jets where a b quark forms bound states in which a single B or D meson carries a majority of the energy. They are common objects to use in BSM physics searches. b-jet events can be differentiated from typical jets by the displacement of their vertices from the primary vertex as b hadrons have a significant flight length. For sufficiently high p_T b-jets (> 50 GeV), the typical tagging efficiency is over 70%.

Missing Transverse Momentum

The missing transverse momentum must be inferred due to conservation of momentum in the transverse plane. It is defined as the negative vector sum of the measured transverse momentum, as detailed in equation 3.4. Delphes measures the energies of particles, and so calculate the missing transverse energy rather than the missing transverse momentum. In Delphes, this is done using only the simulated calorimetric cells, meaning that muons and neutrinos are not taken into account. This quantity is defined as

$$\vec{E}_T^{miss} = -\sum_{i}^{cells} \vec{E}_{T_i} \tag{3.6}$$

Note that Delphes does not perform any overlap removal when calculating $\vec{E}_T^{miss}$. In ATLAS and CMS more sophisticated algorithms are employed to remove double counting of hits and tracks between objects.

3.3 Typical Search Strategy

Direct production experiments target processes that produce new particles. These new particles might be produced on their own, or alongside other Standard Model particles. These particles will then decay according to their phenomenology and they or their decay products will be picked up by the detector. A typical search for BSM physics at the LHC first involves understanding the signal model being searched for and by extension, the relevant background processes. One aims to isolate the presupposed signal from the background by constructing what are referred to as "signal regions". These are regions of the observed data where one expect to see a statistically significant signal to background ratio given by $r = \frac{N_S}{N_B}$. The parameters used in this process are the four-vectors of electron, muon, tau, photon and (b-)jet objects, the missing transverse momentum, and the identified particle labels of each measured object. Using this information one can construct sophisticated physical variables which aim to discriminate the signal from the background in some way.

Figure 3.5 shows the inclusive cross sections of a variety of SM processes for different values of centre-of-mass collision energy. The multi-jet processes are collectively referred to as the QCD background and are orders of magnitude larger than many of the other backgrounds. The QCD background is typically suppressed using missing energy, lepton, or transverse momentum cuts as the QCD background has low missing energy, few leptons and low transverse momentum.

The first step of an analysis is to apply a number of pre-selection cuts. These are cuts which select the particular final state of interest by imposing object multiplicity

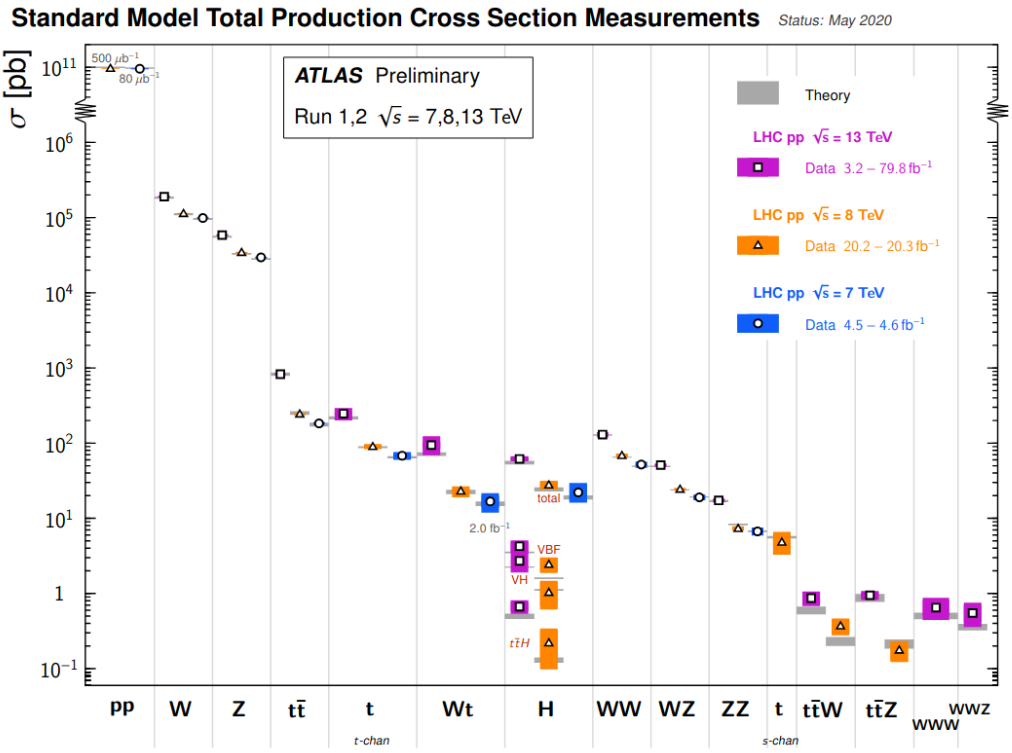


FIGURE 3.5: A summary of the Standard Model total production cross section measurements performed by the ATLAS experiment as of June 2020 [37].

requirements. We also apply lower bounds on the p_T of objects to ensure they pass trigger requirements. As a general rule, these pre-selection cuts should be as minimal as possible so as to not cut down on the amount of available data to optimise on and potentially lose sensitivity.

3.3.1 Discriminating Variables

Once one has decided on a signal model, generated a complete set of Monte Carlo signal/background samples, and imposed a set of pre-selection criteria, it is time to create the signal regions that will give the best discriminating power. In practice this involves defining a number of "physical variables" from the information contained in each event and performing selection cuts to remove as much of the background as possible, while leaving as much of the signal as possible. Some of these physical variables are variables we have seen before such as missing transverse energy, and some are complicated functions of four-vector components. Here a number of common physical variables are detailed and their applications are discussed.

- E_T^{miss}: Missing transverse energy highlights models with invisible particles not seen by the detector as we have mentioned. Models with extra neutrinos, or those with

high mass SUSY particles that decay to the lightest neutralino will be expected to have high missing transverse energy.

- H_T: Defined as the scalar sum of the p_T of jet objects. This value is correlated with the energy scale of the hard process, meaning it has a broader distribution for events containing heavy BSM particles compared to SM backgrounds.

- m_{eff}: Defined as the scalar sum of the p_T of jet objects plus E_T^{miss}. This value tends to have a higher value for events containing high mass particles. Hence this value is extremely useful for detection of models containing high mass particles, especially SUSY analyses.

- m_{12}: Defined as $m_{12} = \sqrt{(E_{T_1} + E_{T_2}) - (p_{T_1} + p_{T_2})}$ for two particles 1 and 2, the invariant mass between two objects of the same kind is regularly used in discovering or rejecting resonances.

- m_T: Typically defined as $\sqrt{2p_T^l E_T^{miss}(1 - \cos\left(\phi_l - \phi_{E_T^{miss}}\right)}$ for a lepton l. The transverse mass is commonly used in understanding W boson backgrounds and constructing signal regions containing W boson production from BSM particles. This is especially useful in searches done within the electroweakino sector of the MSSM. Note that in principal, this variable can be defined between any two particles, not just a lepton and the missing transverse momentum.

- m_{T2}: Known as the stransverse mass, this variable was originally designed for SUSY analyses [38] but is applicable to any scenario where a pair-produced object decays semi-invisibly. Consider two particles which decay to visible and invisible components, where the invisible components have masses M_{χ_1}, and M_{χ_2}. It is not possible to know the transverse momentum of a single invisible object so a minimisation over the under-constrained kinematic degrees of freedom associated with the weakly interacting particles is performed. Hence $m_{T2}^2 = \min_{p_T^{\chi 1} + p_T^{\chi 2} = E_T^{miss}} (\max[m_{T_1}^2, m_{T_2}^2])$, where $m_{T_1}^2$ is the transverse mass between the visible and invisible particles associated with decay 1.

- $m_T^{b,min}$: This variable is defined as the transverse mass between E_T^{miss} and the b-tagged jet closest in ϕ to the missing transverse momentum. It is regularly used to reject events containing a W boson decaying via a lepton and a neutrino. This is useful in rejecting $t\bar{t}$ background events in searches for BSM particles that decay to top quarks such as stop pair production.

- $\Delta\phi(\text{obj}, E_T^{miss})$: Defined as the polar angle between the direction of an object and the missing transverse momentum. In a typical SM multijet event, the main source

of missing transverse momentum comes from jet mismeasurements. In these cases the jets with the highest transverse momentum are typically closely aligned with the direction of the missing transverse momentum. In BSM searches containing multiple jets it is common to place a lower bound on this value for some number of the highest p_T jets.

There are many physical variables used in BSM analyses that are not detailed here, however these are some of the most commonly used ones, and indeed, the ones used in this thesis. These variables have been used to great success in order to measure heavy SM particles and to probe for BSM physics. Most of these variables have some sensitivity to SUSY if the mass scale is larger than the SM scale. Many experiments have been performed using these physical variables to try to probe interesting regions of the parameter space for BSM physics. However there is a crucial drawback to this method in that one must assume a signal model to begin an analysis.

In this thesis I explore applications of novel data analysis techniques to physics data in the search for BSM physics. A quite new and novel approach to this problem of identifying a BSM signal is consider the problem as an anomaly detection problem. In order to most effectively tackle this problem machine learning methods are utilised in order to "learn what the SM background looks like".

4 Machine Learning

Machine learning [39, 40, 41, 42, 43] combines computer science and mathematics to create algorithms which can "learn" the features of a data-set without being told explicitly what to look for. These techniques have been used in a wide array of scenarios, from computer vision to targeted advertising and they all revolve around the same basic principals. Machine learning algorithms rely on what is known as a "training set" of data in order to extract some desired output. Let us first look at the example of image recognition, specifically with the MNIST handwritten digit dataset [44]. The most conventional way to solve this problem is by using supervised machine learning techniques. In this context, "supervised" means that the algorithm is being supplied answers as to how well it is performing during the training process. It must be given labelled data. For example, the algorithm is shown a handwritten digit and then takes a guess at the written value. It is then told how accurate it was with that prediction and uses this information to update its own parameters.

A primary focus of this thesis is *unsupervised* anomaly detection. An unsupervised algorithm is not supplied labelled data at any point in its training, and must use other methods to tune its output. For example, an unsupervised way of tackling the MNIST handwritten digit classification problem is to cluster each image with images that appear similar. This technique will not use any labels associated with the data, and so is considered unsupervised. While a supervised algorithm will usually outperform an unsupervised example, there are cases where a supervised algorithm is unsuitable. When dealing with large amounts of unlabelled data, only an unsupervised analysis is possible.

4.1 Anomaly Detection

Anomaly detection [45] is a subset of machine learning which aims to identify "anomalous" points in a given dataset. While this can be done in a supervised manner, having labelled "typical" and "atypical" data points, I instead choose to focus on unsupervised anomaly detection. In an unsupervised problem, one must assume that the training set is comprised mostly of typical data points, with very few atypical points. Indeed, if this is not the case, then the distinction between typical and atypical becomes rather blurred. Unsupervised anomaly detection algorithms aim to learn what is "normal" for a given dataset. Once this information is learned, a measure of anomalousness can be assigned to each point.

Let us consider the simple thought experiment of attempting to identify images of cats and dogs. A supervised solution to this problem would involve labelling a number of images, and training a machine learning algorithm on this labelled training data. However, this is relatively inflexible. For example what happens when the algorithm is shown something outside of the range of the training set such as an image of a bird? Likely it will attempt to classify it as either a cat or a dog. This issue can be addressed by approaching this problem as an unsupervised anomaly detection problem. Instead of creating a set of labelled data, let us simply create a training set containing only pictures of dogs and build an algorithm which learns what a typical image of a dog looks like. Using this algorithm one can essentially make a dog detector, assigning a measure of anomalousness or "dog-likeness" to each image. In this way, the problem can be solved without having to create a new set of labelled data, and the algorithm is more flexible than the aforementioned supervised method. For example, when shown an image of a bird it will return a high measure of anomalousness, whereas in the supervised example it would have classified it incorrectly.

Adapting this thought experiment to a physics scenario, the words "dog" and "cat" can be replaced with "SM background" and "BSM signal". With this approach in mind, it becomes clear how an anomaly detection algorithm could be useful in solving some issues with searching for new physics at high energy colliders. When treating the scenario as an anomaly detection problem, the particular BSM model being optimised on does not need to be specified, which allows the algorithm to be sensitive to any potential BSM signatures.

4.2 Neural Networks

One of the most flexible and powerful machine learning algorithms is the neural network [46]. Neural networks are used in almost every subsequent chapter in some form. As such, this section will cover the fundamentals of how a simple neural network works. Loosely modelled after the human brain, a neural network consists of a series of nodes containing activation functions, interconnected with weights and biases. Let us first consider a single node as displayed in Figure 4.1.

The output y for a single node in a neural network can be written as

$$y^L = f(y_1^{L-1}w_1 + y_2^{L-1}w_2 + ... + y_n^{L-1}w_n + b), \tag{4.1}$$

where f is some activation function, typically a nonlinear monotonically increasing function. y_i^{L-1} is the ith incoming value from the previous layer, w_i is the weight associated with that incoming value, and b is the bias associated with the node. These nodes, and

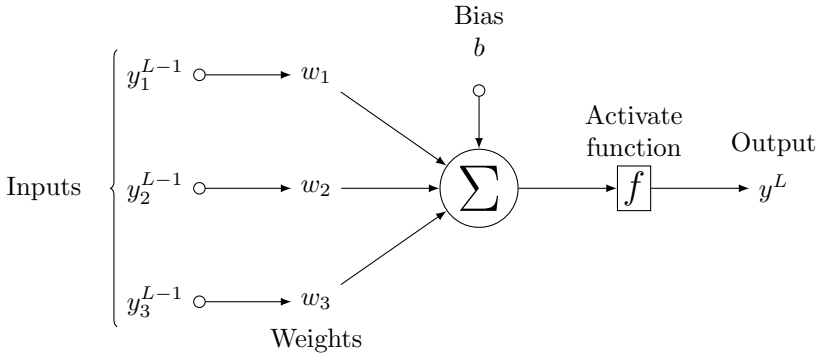


FIGURE 4.1: A single node in a neural network. Incoming values y_i^{L-1} are multiplied with weights w_i and summed together along with a bias b. This value is used as input for an activation function f which in turn produces the output, y^L.

their connections, weights, and biases, are what make up a neural network. When it comes time to train a network, it becomes a matter of tuning the weights and biases of the network via gradient descent. First a metric known as a "loss function" is established in order to decide how to change the weights and biases. It is the job of the loss function to indicate how far a given output is from some desired value, and by extension, how to modify the weights and biases. In the case of a supervised classification problem, this output would be a class label.

The parts that change the behaviour of a neural network, that is, all the weights and biases, can be thought of as a big vector of values $\vec{W}$. At each update step in the training process, the gradient of the loss function with respect to these weights and biases is calculated, and their values are updated. Hence, the new value of the ith element of this weight vector can be represented with the following equation:

$$\vec{W}_i^{new} = \vec{W}_i - \alpha \nabla L_i, \tag{4.2}$$

where α is the learning rate which can be adjusted to make each training step smaller. Let us demonstrate the calculation of the gradient of the loss function first using the simplest possible neural network as shown in Figure 4.2. This consists of two nodes, an input and an output node, connected by a single weight.

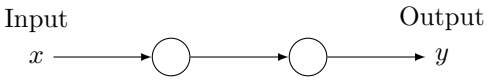


FIGURE 4.2: A very simple neural network consisting of two nodes. The input x is multiplied by a weight, added to a bias, and passed through an activation function to yield y

Let us now compare the output of this neural net, y to the true value y^T, using a loss function. For simplicity, the squared difference function will be used in this example, though in practice this loss function can take any form.

$$\mathcal{L} = (y - y^T)^2 \tag{4.3}$$

The output of this function can be written in the form

$$y = f(xw + b). \tag{4.4}$$

For ease of use, let us define

$$z = xw + b \tag{4.5}$$
$$\therefore y = f(z). \tag{4.6}$$

In order to calculate the gradient of the loss function with respect to the weight, the chain rule must be used.

$$\frac{\partial \mathcal{L}}{\partial w} = \frac{\partial z}{\partial w}\frac{\partial y}{\partial z}\frac{\partial L}{\partial y} \tag{4.7}$$
$$\frac{\partial \mathcal{L}}{\partial y} = 2(y - y^T) \tag{4.8}$$
$$\frac{\partial y}{\partial z} = f'(z) \tag{4.9}$$
$$\frac{\partial z}{\partial w} = x \tag{4.10}$$
$$\therefore \frac{\partial \mathcal{L}}{\partial w} = 2xf'(z)(y - y^T). \tag{4.11}$$

The same process is then followed for the bias.

$$\frac{\partial \mathcal{L}}{\partial b} = \frac{\partial z}{\partial b}\frac{\partial y}{\partial z}\frac{\partial L}{\partial y} \tag{4.12}$$
$$\therefore \frac{\partial \mathcal{L}}{\partial b} = 2f'(z)(y - y^T). \tag{4.13}$$

Now that all the components making up ∇L have been calculated, the weights and biases can be updated using Equation 4.2.

Now let us consider a fully connected network of nodes as in Figure 4.3. The process is near identical to the simple example from before, except with some more indices. Here, the layer l layers away from the output layer is referred to as layer $L - l$. The output of the jth node in layer $L - l$ is written as y_j^{L-l}. The bias for this node is written similarly, b_j^{L-l}. The weight connecting the kth node in layer $L - l - 1$ to the jth node in layer $L - l$

is written as w_{jk}^{L-l}. Using this information the desired partial derivatives can be written

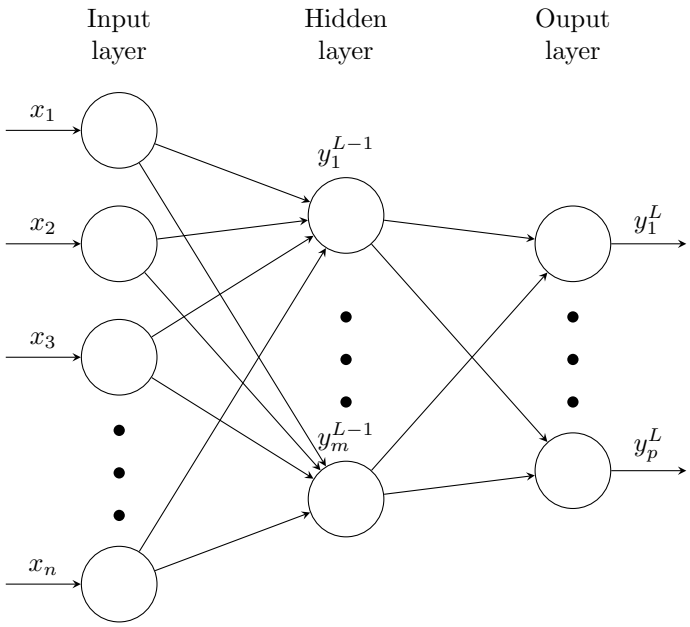


FIGURE 4.3: A simple neural network where the output of a given node is indicated by either its input value x_i, or its activation function value y_i^{L-l}, where l indicates the number of layers it is from the output layer L. Each arrow indicates a connection between two nodes by a weight.

in a similar fashion to Equations 4.11 and 4.13.

$$\mathcal{L} = \sum_{j=0}^{n^L-1} (y_j^L - y_j^T)^2 \tag{4.14}$$

$$z_j^L = \sum_{k=0}^{n^{L-1}-1} w_{jk}^L y_k^{L-1} + b_j^L \tag{4.15}$$

$$y_j^L = f(z_j^L) \tag{4.16}$$

$$\frac{\partial L}{\partial w_{jk}^{L-l}} = \sum_{j=0}^{n^L-1} \frac{\partial z_j^{L-l}}{\partial w_{jk}^{L-l}} \frac{\partial y_j^{L-l}}{\partial z_j^{L-l}} \frac{\partial L}{\partial y_j^{L-l}} \tag{4.17}$$

$$\frac{\partial L}{\partial b_j^{L-l}} = \sum_{j=0}^{n^L-1} \frac{\partial z_j^{L-l}}{\partial b_j^{L-l}} \frac{\partial y_j^{L-l}}{\partial z_j^{L-l}} \frac{\partial L}{\partial y_j^{L-l}}, \tag{4.18}$$

where n^L is the number of nodes in layer L.

In practice, instead of updating the weights and biases for every single training sample by gradient descent, the value is averaged over a number of training samples known as a "batch". This dramatically improves the training speed, as the weights and biases only need to be updated a fraction of the number of times they otherwise would.

This neural network architecture is a very simple feed forward neural network, "feed forward" meaning that each node is only connected to nodes in the next layer, and not previous layers. In practice, this simple kind of neural network has limited use. Modern day neural networks utilise sophisticated techniques such as convolutional and max pooling layers, radial basis functions, recurrence, adversarial training, and more to further improve the performance of a neural network on a given problem. In the following chapters I utilise a particular architecture known as a "Variational Autoencoder" (VAE) for anomaly detection and dimensional reduction purposes.

4.2.1 Variational Autoencoders

An autoencoder [47] is a neural network that maps a given input to itself through a dimensionally reduced "latent space". Autoencoders have a wide variety of uses in anomaly detection, dimensional reduction, image processing, and information retrieval. In the following chapters variational autoencoders (VAEs) will be used for the purposes of anomaly detection and dimensional reduction.

Since an autoencoder maps an input to itself, it can be used as an anomaly detector by training it on "normal" data points. This means when it attempts to reconstruct an anomalous point, the reconstruction will be poorer, and so the loss will be greater. One can then use the value of the loss function to obtain an effective anomaly score on a point by point basis. In Chapter 5 this is done in order to develop an effective anomaly detector.

One can use the latent space in order to obtain a dimensionally reduced representation of a given data point. The thought is that the latent space contains only the bare information needed to reconstruct the sample, so the latent space can be thought of as the "essence" of a given point. In Chapter 5 I experiment with training machine learning models on dimensionally reduced latent space representations of LHC events and observe a remarkable improvement in performance. In Chapter 7 a VAE is used to compress high dimensional physics model parameters to a 2-D plane where trends can be observed and exclusion limits can be drawn.

Up to this point autoencoders have been described, but not variational autoencoders. A variational autoencoder [48] modifies each latent space node to map to two parameters describing the mean and standard deviation of a Gaussian. These Gaussians are then randomly sampled to feed forward into the following layer. This technique yeilds better reconstruction, generalisation, and training speed over the traditional method. Figure 4.4 shows the architecture of a typical VAE. The loss function of a VAE can vary, just as with any neural network, however it will always attempt to map its output to the given input. Additionally, VAE's ensure regularisation within the latent space through the use of a Kullback-Leibler (KL) divergence term [49]. This means that the latent space is both continuous, meaning that two close points in the latent space should be close in the original

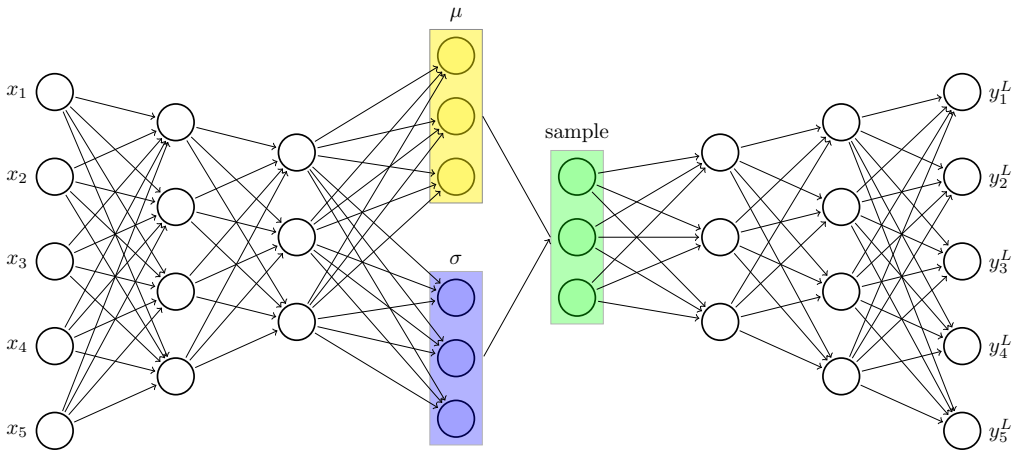


FIGURE 4.4: A diagram of the architecture of a Variational Autoencoder (VAE). In this case x_i is the input, and y_i^L is the reconstructed value of x_i. This particular VAE has a 3 dimensional latent space.

space, and complete, meaning that for any given point sampled from the latent space, the reconstructed output should be "meaningful". The KL divergence term is calculated between the latent distribution $q(z|x) = \mathcal{N}(z|\mu(x), \Sigma(x))$, where $\Sigma = \text{diag}(\sigma_1^2, ..., \sigma_n^2)$, and the latent prior: $p(z) = \mathcal{N}(0, I)$. This term can intuitively be thought of as the information gain if one were to use the latent distribution $q(z|x)$ instead of $p(z)$. The KL divergence is defined as:

$$\text{KL}[q(z|x)||p(z)] = \frac{1}{2}\left[-\sum_i(\log\sigma_i^2 + 1) + \sum_i \sigma_i^2 + \sum_i \mu_i^2\right]. \tag{4.19}$$

A very basic VAE loss function could look something like

$$\mathcal{L} = \beta(x_n - y_n)^2 + (1-\beta)\sum_i^d \text{KL}(\mathcal{N}(\mu_i, \sigma_i), \mathcal{N}(0,1)), \tag{4.20}$$

where d is the dimensionality of the latent space, and β is a term set by the user, determining the weight of the reconstruction term vs the KL term. This type of VAE is typically referred to as a β-VAE.

Now that the fundamentals of machine learning have been explored, especially β-VAE's, let us look at an application to a physics problem. BSM physics searches at the LHC have utilised machine learning in a variety of ways in the past, but have mostly been done in a supervised fashion. In the following chapter I explore using unsupervised anomaly detection to search for BSM signatures, allowing one to perform an analysis in a signal-agnostic fashion.

5 Combining Anomaly Detection Algorithms For BSM Physics Searches

Recent searches at the ATLAS and CMS experiments of the LHC have not yet uncovered evidence of BSM physics. Recall from Chapter 3 that in a standard BSM search at the LHC, one must begin with a signal definition which is then optimised on, removing as much of the Standard Model background as possible to uncover the signal. This method has a high efficiency for signals that resemble the model chosen for optimisation, but requires one to already know what to look for. If nature has not chosen the same Standard Model extension as the signal definition, the search is not guaranteed to find anything. A basic outline of the steps involved in this method are as follows.

1. Select a signal and final state.
2. Model all relevant background processes (for example, by Monte Carlo or data-driven methods).
3. Optimise kinematic selections on functions of the four momenta in the given final state, to define regions of the data that have a high signal-to-background ratio for the simulated signal.
4. Compare a detailed background estimate in the signal regions with the observed yield in the LHC data, and determine the statistical significance of any noted excess.

In this chapter I recap and expand upon my paper on unsupervised machine learning [50], in which I propose using a "signal agnostic" search method in which one merely looks for non-standard-model-like processes instead of searching for an assumed signal model. As briefly discussed in Chapter 4, and explored in Ref. [51], this is possible to do using unsupervised machine learning, namely anomaly detection. Machine learning anomaly detection has been used in a high energy physics context in Ref. [52] however while this approach is weakly supervised, the method detailed in this thesis is entirely unsupervised. The process that I propose is similar to the method detailed prior, but with a few key changes. Note that it is necessary to apply a minimal preselection to ensure

that selected events are compatible with detector trigger requirements. It is also arguably easier to search each final state separately in order to uncover anomalies, to make the background estimation easier to obtain in each search. This does introduce some model dependence, but this preselection is intentionally kept minimal so as to introduce as little model dependence as possible. The procedure is as follows:

1. Model background processes, creating training and testing datasets.
2. Apply a minimal preselection.
3. Train an unsupervised anomaly detection algorithm on the simulated background and obtain an "anomaly score" calculator.
4. Calculate the anomaly scores for both the simulated background and observed LHC data.
5. Compare the event yield in the high anomaly score regions of the simulated background with the observed yield in the LHC data, and determine the statistical significance of any noted excess.

This technique aims to have an unsupervised machine learning algorithm learn the Standard Model background and then, on an event by event basis, assign each event a measure of anomalousness. This allows one to distinguish between non-SM events and background events without making any signal assumptions. Using this method, one can identify interesting regions of the parameter space for further model dependant searches. It is also important to note that while this method could identify new physics on its own, it gives no explanation for the anomalies that one might find so further analysis will be required.

5.1 Dataset

While this technique is designed to provide a model-independent approach to LHC searches, in order to assess its performance, I will test it on particular models of interest. A variety of supersymmetric benchmark models are explored, using the supersymmetric signal and SM background processes from the dataset published and described in Ref. [53].

The dataset consists of simulated proton-proton collision events akin to what is generated at the LHC with a centre of mass energy of 13 TeV. The signal and background events are generated at leading order with up to two extra partons using `Madgraph` [34] with the NNPDF PDF set [54] in the 5 flavour proton scheme. In order to add parton showering to the parton-level samples `Madgraph` was interfaced with `Pythia 8.2` [35] using MLM matching. Detector effects are simulated by `Delphes 3` [36] using a modified

version of the ATLAS detector card. `FastJet` [55] was used to perform jet clustering with the anti-k_t algorithm and a jet radius of $R = 0.4$. b-jets are tagged by `Delphes` in a similar fashion to [56]. A summary of the supersymmetric benchmark models and SM backgrounds used can be seen in Table 5.1 and 5.2 respectively.

Process	Process ID	σ (pb)	N_{tot} ($N_{10\text{fb}^{-1}}$)
$pp \to \tilde{g}\tilde{g}$ (1 TeV)	Gluino 01	0.20	50000 (2013)
$pp \to \tilde{g}\tilde{g}$ (1.2 TeV)	Gluino 02	0.05	50000 (508)
$pp \to \tilde{g}\tilde{g}$ (1.4 TeV)	Gluino 03	0.014	50000 (144)
$pp \to \tilde{g}\tilde{g}$ (1.6 TeV)	Gluino 04	0.004	50000 (44)
$pp \to \tilde{g}\tilde{g}$ (1.8 TeV)	Gluino 05	0.001	50000 (14)
$pp \to \tilde{g}\tilde{g}$ (2 TeV)	Gluino 06	4.8×10^{-4}	50000 (5)
$pp \to \tilde{t}_1\tilde{t}_1$ (220 GeV), $m_{\tilde{\chi}_1^0} = 20$ GeV	Stop 01	26.7	500000 (267494)
$pp \to \tilde{t}_1\tilde{t}_1$ (300 GeV), $m_{\tilde{\chi}_1^0} = 100$ GeV	Stop 02	5.7	500000 (56977)
$pp \to \tilde{t}_1\tilde{t}_1$ (400 GeV), $m_{\tilde{\chi}_1^0} = 100$ GeV	Stop 03	1.25	250000 (12483)
$pp \to \tilde{t}_1\tilde{t}_1$ (800 GeV), $m_{\tilde{\chi}_1^0} = 100$ GeV	Stop 04	0.02	250000 (201)

TABLE 5.1: Summary of the supersymmetric benchmark models that are used to test each method. The details include the production cross-section at $\sqrt{s} = 13$ TeV, the number of events that were generated, and the number of events expected in 10fb^{-1} of LHC data [53].

The data is stored in a one-line-per-event csv file, with each event having a weight of 1. Each line consists of a number of final-state physics objects, defined in Table 5.3. Each final state object is subject to a handful of requirements:

- jet or b-jet: $p_T > 20$ GeV and $|\eta| < 2.8$,
- electron/muon: $p_T > 15$ GeV and $|\eta| < 2.7$,
- photon: $p_T > 20$ GeV and $|\eta| < 2.37$.

Once these checks on each final state object have been satisfied, a given event is stored only when it meets at least one of the following critera:

- At least one jet or a b-jet with transverse momentum $p_T > 60$ GeV and pseudorapidity $|\eta| < 2.8$, or
- at least one electron with $p_T > 25$ GeV and $|\eta| < 2.47$, except for $1.37 < |\eta| < 1.52$, or
- at least one muon with $p_T > 25$ GeV and $|\eta| < 2.7$, or
- at least one photon with $p_T > 25$ GeV and $|\eta| < 2.37$.

Process	Process ID	σ (pb)	N_{tot} ($N_{10\text{fb}^{-1}}$)
$pp \to jj$	njets	$19718_{H_T>600\text{GeV}}$	415331302 (197179140)
$pp \to W^{\pm}(+2j)$	w_jets	$10537_{H_T>600\text{GeV}}$	135692164 (105366237)
$pp \to \gamma(+2j)$	gam_jets	$7927_{H_T>600\text{GeV}}$	123709226 (79268824)
$pp \to Z(+2j)$	z_jets	$3753_{H_T>600\text{GeV}}$	60076409 (37529592)
$pp \to t\bar{t}(+2j)$	ttbar	541	13590811 (5412187)
$pp \to W^{\pm}t(+2j)$	wtop	318	5252172 (3176886)
$pp \to W^{\pm}\bar{t}(+2j)$	wtopbar	318	4723206 (3173834)
$pp \to W^{+}W^{-}(+2j)$	ww	244	17740278 (2441354)
$pp \to t + \text{jets}(+2j)$	single_top	130	7223883 (1297142)
$pp \to \bar{t} + \text{jets}(+2j)$	single_topbar	112	7179922 (1116396)
$pp \to \gamma\gamma(+2j)$	2gam	47.1	17464818 (470656)
$pp \to W^{\pm}\gamma(+2j)$	Wgam	45.1	18633683 (450672)
$pp \to ZW^{\pm}(+2j)$	zw	31.6	13847321 (315781)
$pp \to Z\gamma(+2j)$	Zgam	29.9	15909980 (299439)
$pp \to ZZ(+2j)$	zz	9.91	7118820 (99092)
$pp \to h(+2j)$	single_higgs	1.94	2596158 (19383)
$pp \to t\bar{t}\gamma(+2j)$	ttbarGam	1.55	95217 (15471)
$pp \to t\bar{t}Z$	ttbarZ	0.59	300000 (5874)
$pp \to t\bar{t}h(+1j)$	ttbarHiggs	0.46	200476 (4568)
$pp \to \gamma t(+2j)$	atop	0.39	2776166 (3947)
$pp \to t\bar{t}W^{\pm}$	ttbarW	0.35	279365 (3495)
$pp \to \gamma\bar{t}(+2j)$	atopbar	0.27	4770857 (2707)
$pp \to Zt(+2j)$	ztop	0.26	3213475 (2554)
$pp \to Z\bar{t}(+2j)$	ztopbar	0.15	2741276 (1524)
$pp \to t\bar{t}t\bar{t}$	4top	0.0097	399999 (96)
$pp \to t\bar{t}W^{+}W^{-}$	ttbarWW	0.0085	150000 (85)

TABLE 5.2: Summary of the background processes included in the analysis. The details include the production cross-section at $\sqrt{s} = 13$ TeV, the number of events that were generated, and the number of events expected in $10\,\text{fb}^{-1}$ of LHC data [53]. Note that for the `njets`, `w_jets`, `gam_jets`, and `z_jets` backgrounds, the cross section is only calculated with an H_T cut of 600 GeV applied.

Where the electron η restrictions emulate a veto in the crack regions of the detector, which are often applied in ATLAS analyses. These requirements are unrealistic in terms of what a real experiment could afford to record after the online trigger system, however the aim of this dataset is to be a flexible resource that allows for many types of studies and selection criteria.

Table 5.2 displays a summary of the SM background processes generated for this dataset. For each process, the total number of generated events (N_{tot}) is greater than the number of events required for 10 fb^{-1} of data ($N_{10\text{fb}^{-1}}$). In order to ensure that the background data generation is sensible, I present Figures 5.1-5.4, which show stacked histograms for the E, p_T, η, and ϕ of jets and leptons for each background process.

Symbol ID	Object
j	jet
b	b-jet
e-	electron (e^-)
e+	positron (e^+)
m-	muon (μ^-)
m+	antimuon (μ^+)
g	photon (γ)

TABLE 5.3: Definition of symbols used to refer to the final-state objects.

Figure 5.5 displays the stacked histograms for the number of jets and leptons. Figure 5.6 shows the E_T^{miss} and $\phi_{E_T^{\text{miss}}}$ histograms, and Figure 5.7 shows the H_T distribution. Note that for Figure 5.7 events with $H_T < 600$ GeV have been removed. For the other figures no restriction on H_T is placed, except for `njets`, which has $H_T > 600$ GeV. The `w_jets`, `gam_jets` and `z_jets` backgrounds all have $H_T > 100$ GeV.

Each event is formatted as follows:

```
event ID; process ID; event weight; MET; METphi; obj1, E1, pt1, eta1,
phi1; obj2, E2, pt2, eta2, phi2;  ...
```

`event ID` is an event specifier, used to identify the generation of that particular event, and is used for debugging and reproducability. `process ID` is a string that identifies the physical process that generated the event. `MET` and `METphi` are the magnitude of the missing transverse energy E_T^{miss}, and its azimuthal angle $\phi_{E_T^{\text{miss}}}$. Recall that E_T^{miss} refers to the transverse energy of objects that escaped detection by the detector, such as neutrinos and weakly interacting stable particles. `obj1, obj2,...` refer to the particle identifiers listed in Table 5.3, and `E1, pt1, eta1, phi1` refer to the energy E, transverse momentum p_T, pseudorapidity η, and azimuthal angle ϕ of the first physics object.

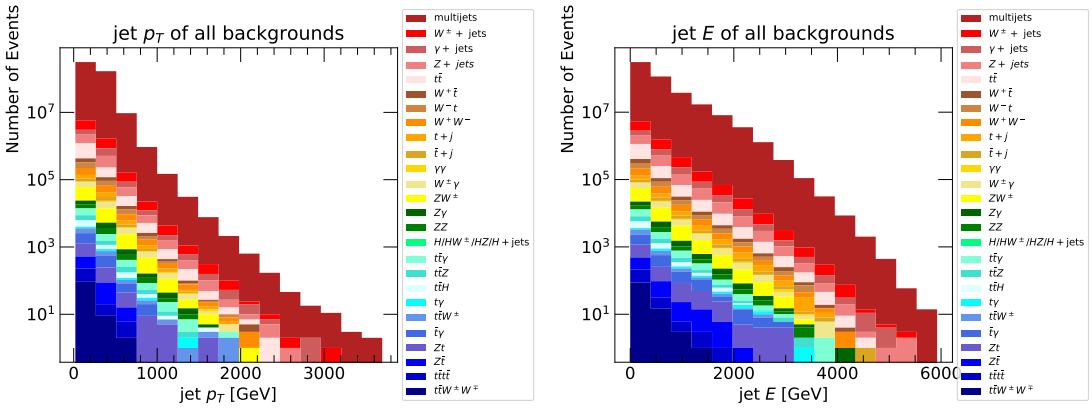


FIGURE 5.1: Transverse momentum p_T (left) and energy E (right) in GeV of the jets for all backgrounds.

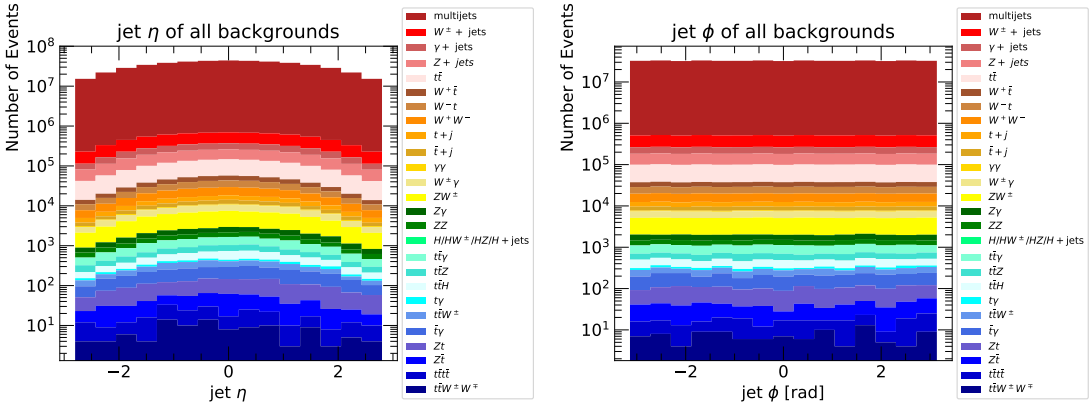


FIGURE 5.2: Pseudorapidity η (left) and azimuthal angle ϕ (right) of the jets for all backgrounds.

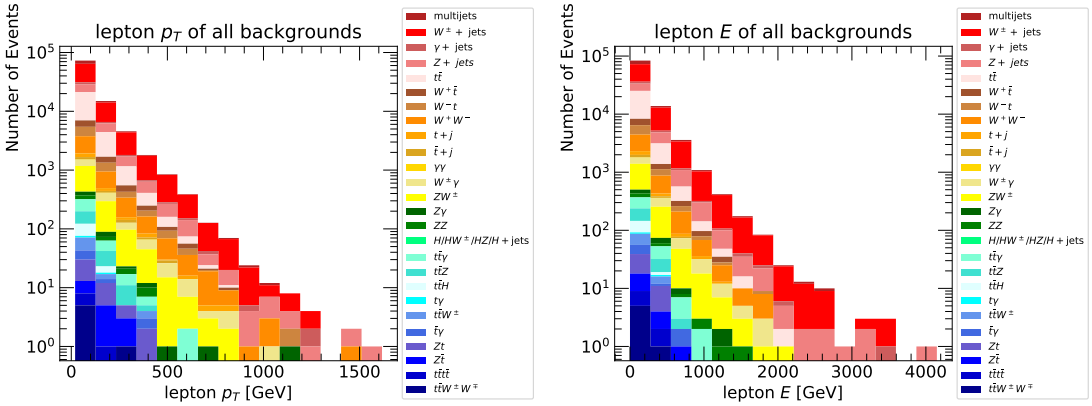


FIGURE 5.3: Transverse momentum p_T (left) and energy E (right) in GeV of the leptons (e^+, e^-, μ^+, μ^-) for all backgrounds.

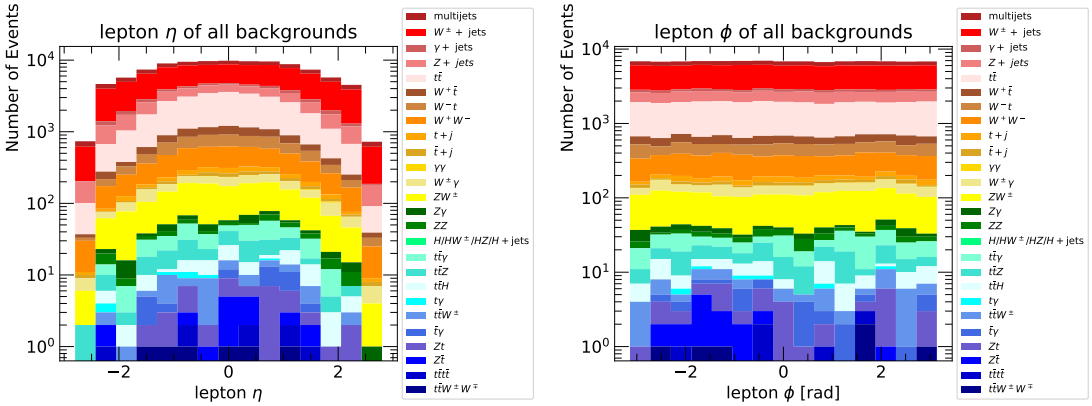


FIGURE 5.4: Pseudorapidity η (left) and azimuthal angle ϕ (right) of the leptons (e^+, e^-, μ^+, μ^-) for all backgrounds.

The following supersymmetric benchmark model points are chosen for their differing behaviour and while some have been excluded by dedicated ATLAS and CMS searches, they still provide adequate benchmarks to test these unsupervised anomaly detection techniques. The first set of BSM models involves supersymmetric gluino pair production,

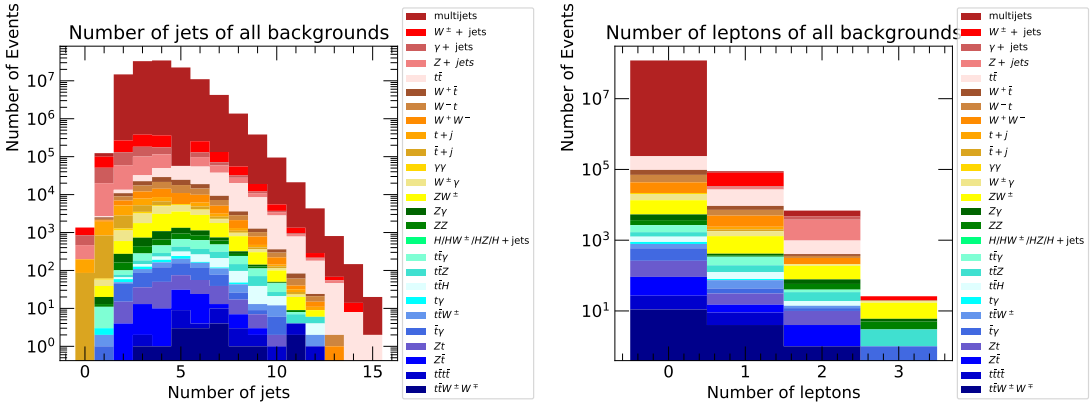


FIGURE 5.5: Number of jets (left) and leptons (right).

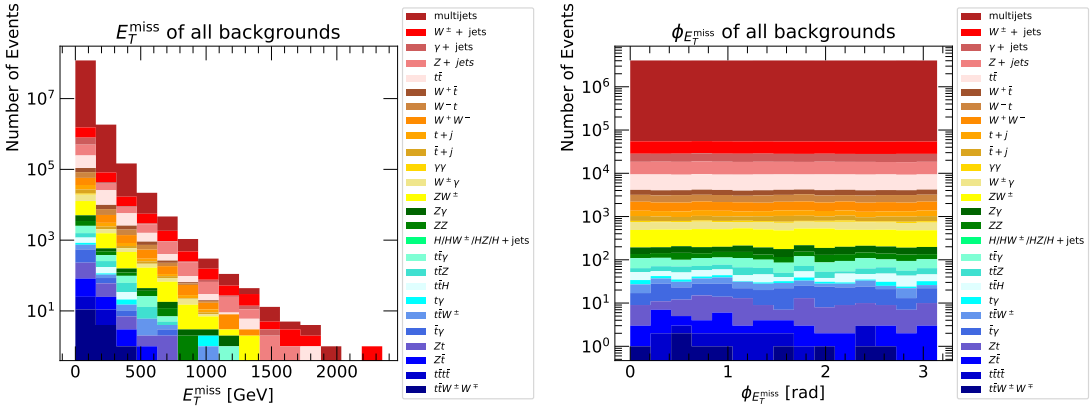


FIGURE 5.6: Missing transverse energy E_T^{miss} in GeV and azimuthal angle $\phi_{E_T^{\text{miss}}}$ for all backgrounds.

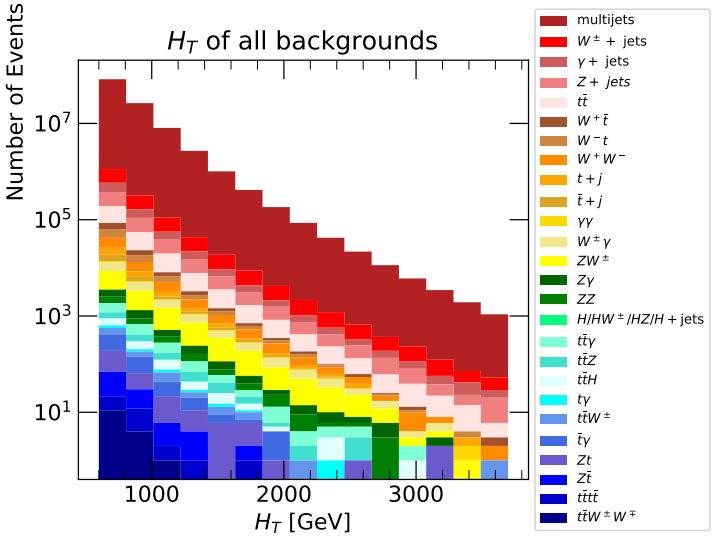


FIGURE 5.7: The scalar sum of the jet transverse momenta H_T in GeV for all backgrounds, with $H_T > 600$ GeV imposed.

with each gluino subsequently decaying to a boosted top-quark pair and the lightest neutralino, which is stable by assuming R-parity conservation. The gluinos are assumed

to have a mass of 1-2.2 TeV (in steps of 200 GeV), while the neutralinos have a mass of 1 GeV. The branching ratio of the decay process $\tilde{g} \to t\bar{t}\tilde{\chi}_1^0$ is taken to be 100%.

In the second scenario two stop quarks ($\tilde{t}_1$) are produced, with each stop decaying into an on-shell top quark and a lightest neutralino ($\tilde{t}_1 \to t\tilde{\chi}_1^0$). Four different benchmark scenarios have been chosen. In the first model, the lightest neutralino has a mass of 20 GeV and the lightest stop has a mass of 220 GeV. In the other models, the mass of the lightest neutralino is 100 GeV and the stops have masses of 300, 400 and 800 GeV.

Although the production cross-section for the lowest-mass stop quark model is the highest out of all assumed signal hypothesis, it is actually the most difficult model to discover using traditional search methods. The mass difference of the $\tilde{t}_1$ and $\tilde{\chi}_1^0$ is close to the mass of the top quark, which makes the kinematics very similar to those of the background. The techniques described in the next section are designed to find anomalies, but this model does not result in an obviously anomalous signal. Therefore, it is to be expected that the techniques will show least sensitivity to the 200 GeV stop scenario, although this might be compensated for by the fact that its cross section is the highest. On the other hand, the gluino signals are more anomalous as they result in four top quarks and a sizable missing transverse energy. This is a rare final state for SM production, and since the 1 TeV gluino carries the highest production cross-section, it is to be expected that this scenario will be the easiest.

The csv file is modified to better enable the training of each algorithm. All data is first zero-padded so every event has the same dimensionality. Next, the continuous data and the categorical data are split and the number of objects in the events is counted. From this, the following event structure is defined:

$$\mathbf{x} = \left(N, \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{19} \end{bmatrix}, \begin{bmatrix} (p_T, \eta, \phi)_0 \\ (p_T, \eta, \phi)_1 \\ \vdots \\ (p_T, \eta, \phi)_{19} \end{bmatrix} \right). \tag{5.1}$$

In this vector, N is the number of objects in the event, c_i is the object type as a one-hot encoded vector, p_T is the transverse momentum, η the pseudorapidity and ϕ the azimuthal angle of an object. This layout is used to train the unsupervised machine learning algorithms on the 4-vector representations of the data. When the non-VAE techniques are trained on the latent space variables of the VAE, it is still true to say that the starting point for the analysis is this 4-vector representation.

Each background is split into two subsets, 80% in a training/validation set and the remaining 20% in a testing set. The training/validation split ratio is also 80/20. The preselection for the preliminary results using the stop and gluino signals is as follows:

- $E_T^{\text{miss}} \geq 150\text{GeV}$,
- ≥ 4 jets,
- $H_T \geq 600\text{GeV}$.

5.2 Machine Learning Algorithms

The following algorithms are used for unsupervised anomaly detection. Note that a detailed hyperparameter scan has not been conducted for these algorithms, and the performance could potentially be improved by doing so. Nonetheless, these algorithms are able to successfully identify anomalies with their current architectures.

5.2.1 Isolation Forests

First outlined in Ref. [57], "Isolation Forest" is an unsupervised learning algorithm that assigns each point in a dataset a value based on the ease in which it is isolated from the other points in the dataset. It is attractive due to its simple concept, linear time complexity and low memory requirement.

Given a set of data $X = \{x_1, x_2, \ ... \ , x_n\}$ from a d-variate distribution, one first randomly chooses an attribute q, and a "split value" p which lies between the maximum value and minimum value of q. If, for each instance in the dataset, $q < p$, the point is placed in a set of points called X_l whilst, if $q \geq p$, it is placed in a set called X_r. This process is repeated recursively, until value x_i is isolated (or if a limit imposed on the number of splits is reached). The sequence of splits generated are called "trees", and the number of splits in them is called the "path length" of the tree. Each split is a "node" of the tree, nodes which do not begin or end trees are "internal", and those which do are "external".

Anomalies are by definition "few and far between" thus an anomaly should on average require a fewer number of splits to become isolated. This measure of anomalousness is therefore defined via the average path length of the trees. This average path length is normalized using:

$$c(n) = 2H(n-1) - \left(\frac{2(n-1)}{n}\right) \tag{5.2}$$

where n is the total number of external nodes in a tree and $H(i)$ is the harmonic number (approximately $ln(i) + 0.5772156649$). The anomaly score of a point x is then defined as:

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}}, \tag{5.3}$$

where $h(x)$ is the path length, and $E(h(x))$ is the mean path length of all trees constructed for x. Empirically it is found that $s \approx 1$ implies a high level of anomalousness, $s \approx 0$ indicates no anomaly at all and, if the whole sample generates $s \approx 0.5$, the entire sample is likely devoid of anomaly.

Note that the normalisation used by `scikit-learn` sets $s \approx -1$ to be indicative of no anomaly at all, and $s \approx 1$ to be indicative of a high level of anomalousness. This means that if the whole sample generates $s \approx 0$, the entire sample is likely devoid of anomaly.

5.2.2 Gaussian Mixture Models

The job of clustering a set of data into smaller subsets can be thought of as an expectation minimization problem, that is to pose the question "what is the distribution, or set of distributions, from which this set of data was most likely randomly sampled?". Mixture models are a methodology by which one can approximate the most accurate set of N-dimensional statistical distributions which represent a cluster of data and its substructure. Specifically, Gaussian Mixture Models (GMMs) [58] are an implementation of this methodology where the statistical distributions being fitted are N-dimensional Gaussian distributions.

To do this, a vector of latent variables, denoted γ, is defined. This vector represents, for each datapoint, the corresponding probability that it was generated by a given Gaussian. This is often referred to as the distribution having 'responsibility' for that data point. The job is then to maximize the overall probability of the full dataset being generated from the fitted distributions:

$$\log p(X|\Theta) = \log \left\{ \sum_i p(X, \gamma_i|\Theta) \right\} \tag{5.4}$$

where 'X' is the dataset and Θ is the set of parameters of the distributions being fit to. Maximising the left hand side of the equation can be very tricky and so is generally done using the Expectation-Maximisation (EM) algorithm.

Expectation-Maximisation Algorithm

There are two steps to the EM algorithm, the expectation step (E-step) and the maximisation step (M-step). If the probability of a given point being sampled from the GMM is given by $p(x)$, the posterior distribution of the responsibilities that each Gaussian has for each datapoint can be written as $\gamma(z_{nk})$. The probability $p(x)$ is given by:

$$p(x) = \sum_{k=1}^{K} \pi_k N(x|\mu_k, \Sigma_k) \tag{5.5}$$

where there are K Gaussian's with weights $\vec{\pi}$, means $\vec{\mu}$, and variances $\vec{\Sigma}$. Using this one can write:

$$\gamma(z_{nk}) = \frac{\pi_k N(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_n|\mu_j, \Sigma_j)} \tag{5.6}$$

for the posterior distribution of the responsibilities that each of the K Gaussians have for each of the N datapoints.

Once the posterior has been calculated, the parameters of each Gaussian can be estimated defined by:

$$\mu'_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n \tag{5.7}$$

$$\Sigma'_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu'_k)(x_n - \mu'_k)^T \tag{5.8}$$

$$\pi'_k = \frac{N_k}{N} \tag{5.9}$$

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk}) \tag{5.10}$$

where the primes on π, μ and Σ denote that they are updated versions of the previous parameters. Finally with the updated parameters the new log-likelihood can be calculated:

$$\log p(X|\mu, \Sigma, \pi) = \sum_{n=1}^{N} \log \left\{ \sum_{k=1}^{K} \pi_k N(x_n|\mu_k, \Sigma_k) \right\} \tag{5.11}$$

However this equation 5.11 is a stumbling block as described above, so instead, a lower bound is calculated using "Jensen's Inequality" [59] which takes the form:

$$L = \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbf{E}[\mathbf{z_{n,k}}](\log \pi_\mathbf{k} + \log \mathbf{N}(\mathbf{x_n}|\mu_\mathbf{k}, \sigma_\mathbf{k})) - \sum_{\mathbf{n=1}}^{\mathbf{N}} \sum_{\mathbf{k=1}}^{\mathbf{K}} \mathbf{E}[\mathbf{z_{n,k}}] \log \mathbf{E}[\mathbf{z_{n,k}}] \tag{5.12}$$

Whilst this equation may appear more complicated the equation contains only the sum of log terms, and not the log of summed terms.

Once this is done, the process is performed iteratively until some stopping point defined by the user is reached. The resulting parameters define the distributions which best classify the data and its substructure. The Gaussian mixture model used in this chapter uses 10 Gaussian components.

5.2.3 Neural networks

Recall Section 4.2 where the workings of a simple neural network were explored in great detail. Additionally, their use in anomaly detection problems was briefly considered. In this chapter two neural networks are used for anomaly detection and dimensional reduction purposes. A simple static autoencoder is used as an anomaly detector, and a more sophisticated variational autoencoder is used as both an anomaly detector and a dimensional reduction tool.

Autoencoders

Autoencoders are a special class of neural networks where the input and output of the network are equal. This means autoencoders can be trained without labels in an unsupervised fashion. The loss function typically is the reconstruction loss: the difference between the output and input, quantified by, for example, the mean squared error on every dimension of the data. Generally, the number of hidden neurons in the neural network first decreases and then increases again, so the data needs to be squeezed in a lower dimensional representation. The lowest dimensional representation, usually in the middle of the network, is called the latent space. If the latent space dimensionality is too high, the neural network can simply learn the identity function to make the output equal to the input. When it is too low, too much information needs to be removed in order to have a good reconstruction ability. The part of the network that transforms the input to latent space representation is called the encoder, while the part of the network that transforms the latent space representation to output is called the decoder.

If the latent space dimensionality is just right, the input data is transformed into a (highly correlated) lower dimensional representation with only relevant information that is required for reconstruction of the original input. If an autoencoder is trained on a dataset without any anomalies and applied to a dataset with both normal and anomalous data, the autoencoder will have a low reconstruction loss for the normal events and a high reconstruction loss for the anomalous events. This is because the anomalous events are different from the normal events, and thus are placed in unexpected locations in the latent space. These anomalous events are then reconstructed badly. An autoencoder can thus be used as an anomaly detector [60].

In this work, an autoencoder is used as an anomaly detector to distinguish signal events from the Standard Model background. The autoencoder is defined to have 5 hidden layers, with 40, 20, 8, 20 and 40 nodes, respectively as shown in Figure 5.8. This shape is modelled after Ref. [61]. The loss function used is a Sliced Wasserstein Distance Metric [62]. The Wasserstein Distance (sometimes referred to as “Earth Movers Distance” or “EMD”) between two points can be thought of as the minimum amount of

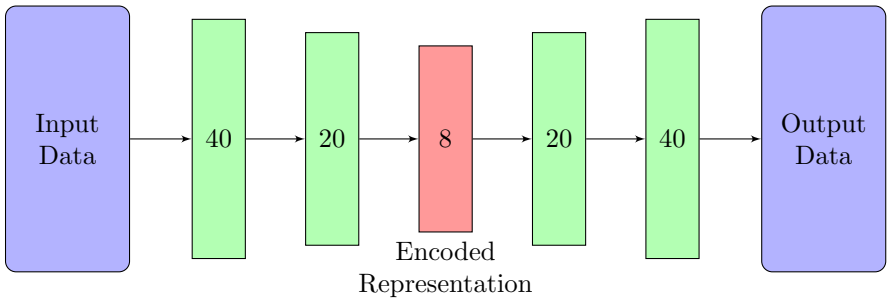


FIGURE 5.8: The structure of the autoencoder used in this analysis.

energy required to transform one into the other. It is a useful tool as it metrizes the energy flow between two events. The Sliced Wasserstein Distance is the Wasserstein Distance between a projection of the data onto a 1-D distribution. It has similar properties to the Wasserstein Distance metric, and is more computationally efficient.

Variational Autoencoder

A variational autoencoder, differs from a static autoencoder within its latent space. In a VAE, the encoder outputs two numbers per latent space dimension, which represent the mean and standard deviation of a Gaussian distribution (see Figure 5.9). The decoder works by taking a random sample of this distribution and decoding the sample back into the original input. Variational autoencoders have been used in particle physics applications to great effect in Ref. [63].

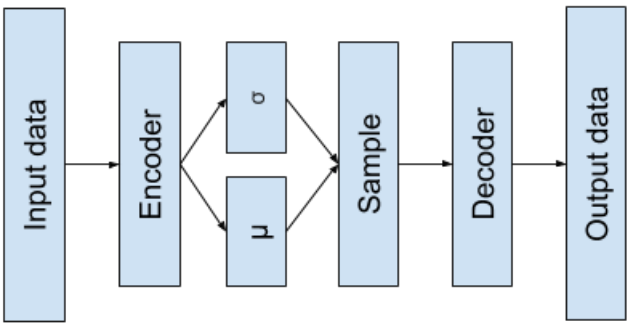


FIGURE 5.9: The structure of a VAE

The loss function in the VAE used in this chapter is constructed such that the KL-divergence [64] between these Gaussians and a standard normal distribution is as low as possible. The loss function of a VAE then is given by a function that encodes the ability to reconstruct the original data point, and a KL-divergence term. The former encourages optimal reconstruction, while the KL-divergence term forces ordering within the latent

space: all input should be encoded as close as possible to $\vec{0}$ within the latent space. The relative importance of these two terms can be tuned with a β term. In this work a range of values of beta are explored. These β values range from 10^{-5} to 1, incrementing by order of magnitude in order to determine the optimal weighting of these two terms.

The reconstruction loss consists of three different components: a mean squared error on the number of objects x_n, a-mean-squared error on 4-vector terms ($\vec{x}_{r,i} = p_T, \eta$ or ϕ), and a categorical cross-entropy (see e.g. from Ref. [65]) on the categorical variables $x_{c,i}$ that represent different objects in an event (jet, b-jet, electron, etc.). The total loss function of the VAE is then defined as

$$\begin{aligned} \mathcal{L} \;=\; & 100\beta\,(x_n - \hat{x}_n)^2 \\ & + \frac{\beta}{d_r}\sum_i^{d_r}(x_{r,i} - \hat{x}_{r,i})^2 \\ & - \frac{10\beta}{d_c}\sum_i^{d_c}(x_{c,i}\log(\hat{x}_{c,i}) + (1 - x_{c,i})\log(1 - \hat{x}_{c,i})) \\ & + (1-\beta)\sum_i^{d_z}\mathrm{KL}\left(\mathcal{N}(\hat{\mu}_i, \hat{\sigma}_i), \mathcal{N}(0,1)\right). \end{aligned} \tag{5.13}$$

Here, $\hat{x}_n$ represents the predicted number of objects, $\hat{x}_{r,i}$ represents the i-th predicted regression label, $\hat{x}_{c,i}$ represents the i-th predicted categorical label, d_r represents the number of regression variables, and d_c represents the dimensionality of the categorical data. The relative importance of each of these contributions to the loss function is indicated by β. The total reconstruction loss is given by the first three terms, and the last component is the KL-divergence loss term. The three components of the reconstruction loss are not equally important, and as such they are weighted with numerical factors. The anomaly score of an event is given by the reconstruction loss term (the first three lines of Equation 5.13).

The architecture consists of 3 fully-connected hidden layers for the encoder and decoder, each containing 512, 256 and 128 nodes for the former, 128, 256 and 512 nodes for the latter, and a 13 dimensional latent space. The activation function used between the hidden nodes is the exponential linear unit (ELU) [66].

It is possible to use a VAE as a dimensional reduction technique by passing a point into the encoder and obtaining its latent space representation. In this chapter I explore using the VAE as both a dimensional reduction technique and an anomaly detector. Using a VAE to dimensionally reduce MC events has the benefits of SM events being compressed differently to BSM events (as the VAE is trained on SM events), drawing out more difference between them. Additionally, many of the algorithms detailed above work more effectively in low dimensional spaces, so a significant increase in performance can be expected. The performance of training on 4-vectors is compared to training within

the latent space of this VAE in Section 5.4. If the anomaly scores yielded from each algorithm are not perfectly correlated with each other, there is information to be gained from combining them. In Section 5.3, various combination techniques are defined which are then explored in Section 5.4. I propose that by training algorithms within the latent space of the VAE and then combining the anomaly scores, a better performing anomaly score can be obtained.

5.3 Methodology of Combination Techniques

Now that each algorithm tested in this chapter has been explained, the process can be summarised as such:

1. Define a Variational Autoencoder.
2. Train it on a subset of the background data (80/20 split for training/testing).
3. Pass the remainder of background data + signal through VAE and obtain latent space representations for each event.
4. Train further anomaly detection algorithms on the latent space representations of the background events (Isolation Forest (IF), Gaussian Mixture Model (GMM), Static Autoencoder (AE)) (50/50 split for training/testing).
5. Pass the remaining background and signal events through these algorithms, obtaining 5 measures of anomalousness for each event. (VAE reconstruction loss, IF mean path length, GMM log likelihood, AE reconstruction loss).
6. Normalise each anomaly score to uniform background efficiency.
7. Perform various combinations. Logical AND/OR, Average and Product.
8. Construct ROC curve and compare the area under the curve (AUC), and signal efficiencies at various background efficiencies.

5.3.1 Normalisation of Anomaly Scores

In order to combine these anomaly detection techniques, they must be normalised to uniform background efficiency. This solves an issue of scale as the output from the isolation forest is bounded by $-1 \leq x \leq +1$, whereas the Gaussian mixture model log likelihood is bounded by $0 \leq x \leq \infty$. This also removes shape-dependent effects, for example the autoencoder distribution has a very long tail where the isolation forest distribution does not. For each anomaly score distribution a function $f_i(x)$ is constructed which returns the

background efficiency at a given anomaly score value x for the ith algorithm. Let $g_i(x)$ represent the number of background events with anomaly score *greater* than x for the ith algorithm, and N_{bkg} be the total number of background events. This function $f_i(x)$ is then given by:

$$f_i(x) = \frac{g_i(x)}{N_{bkg}} \tag{5.14}$$

The signal and background datasets are then normalised by computing $f(x)$ for each signal and background anomaly score.

5.3.2 Combination Methods

In this chapter AND, OR, product, and averaging combinations are explored using the normalised representations of the anomaly scores from each algorithm. For a given event, let the anomaly score normalised to uniform background efficiency be x_i for the ith anomaly detection algorithm. The combinations are defined as follows:

- AND: $x^{\text{AND}} = \min(x_i)$
- OR: $x^{\text{OR}} = \max(x_i)$
- Product: $x^{\text{product}} = \prod_i x_i$
- Average: $x^{\text{average}} = \frac{1}{N} \sum_i x_i$

where N is the number of algorithms being used.

It is important to note that the technique of combining algorithms is not guaranteed to always outperform a single algorithm. To demonstrate, let us consider the following example where a signal event is represented as 1 and a background event is represented as 0. Imagine an algorithm (algorithm 1) that incorrectly classifies every background event as signal and vice-versa. Consider a second algorithm (algorithm 2) that correctly classifies every background (signal) event as background (signal). An OR combination of these two algorithms will take the maximum value for each event - meaning every event will be classified as signal. This of course performs worse than algorithm 2. Now lets consider an AND combination, taking the minimum value for every event will classify every event as background. This again performs worse than algorithm 2. This shows that indeed the combination of algorithms is not guaranteed to outperform a single algorithm. This issue is addressed in Chapter 6, where only combinations that improve the performance are taken.

5.4 Results

In the following section, I compare the performance of the aforementioned algorithms when trained on 4-vector components and within the latent space of a VAE. The parameters that are used for training are $(E_T, \phi)_{\text{miss}}$, $(E, p_T, \eta, \phi)_{\text{jets}}$, $(E, p_T, \eta, \phi)_{\text{bjets}}$, $(E, p_T, \eta, \phi)_{\text{leptons}}$, $(E, p_T, \eta, \phi)_{\text{photons}}$. Leptons can be positively or negatively charged electrons or muons. In the first approach (Section 5.4.1), all algorithms are trained on the same input data. In the second approach (Section 5.4.2), the VAE training process is identical, but the remaining algorithms are trained on the latent space representations of events. In both scenarios, the combination methods are employed and yield improved results in some cases. A summary of the considered machine learning algorithms can be found in Table 5.4.

Algorithm	Anomaly-score definition
Isolation forest (IF, Section 5.2.1)	Mean path length (Eq. (5.3))
Gaussian mixture model (GMM, Section 5.2.2)	Log likelihood (Eq. (5.11))
Static autoencoder (AE, Section 5.2.3)	Sliced Wasserstein Distance [62]
Variational autoencoder (VAE, Section 5.2.3)	Reconstruction loss (first three lines of Eq. (5.13))

TABLE 5.4: Summary of the considered ML algorithms and the definition of their anomaly scores.

5.4.1 Results Trained on 4-Vector Components

Figure 5.10 displays ROC curves for algorithms trained on the 4-vector components of background events for the gluino signals detailed in 5.1. ROC curves are computed by taking a number of cuts in the anomaly score variable (or physical variable where one is used) and at each one, calculating the true and false positive rates, denoted ϵ_S and ϵ_B. The true positive rate, ϵ_S, is given by the percentage of signal events to the right of the cut, while the false positive rate, ϵ_B, is given by the percentage of background events to the right of the cut. The black dashed line indicates the point at which the number of background events $B = 100$. The Z score at this point is denoted as

$$Z_B = \frac{S}{\sqrt{S + B + (\sigma_B B)^2}} \tag{5.15}$$

Where B is the number of background events (100), S is the number of signal events, and σ_B is the assumed systematic uncertainty. A background event cut of 100 is chosen in order to ensure that there are enough signal and background montecarlo events present. To begin the analysis, zero systematic uncertainty is assumed ($\sigma_B = 0$) and a reasonable value is introduced in Section 5.4.3.

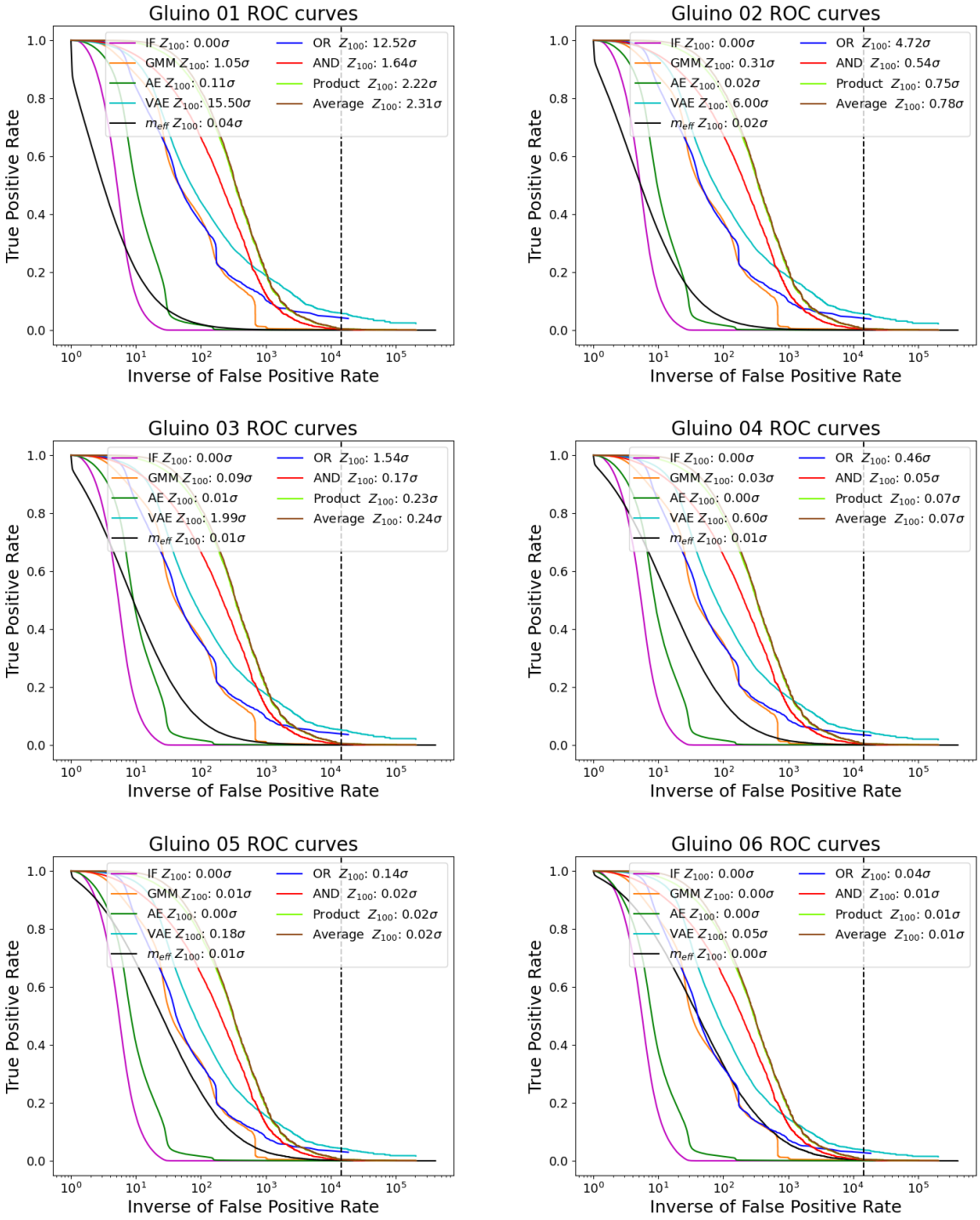


FIGURE 5.10: ROC curves for the gluino signals (Table 5.1) for the algorithms applied on 4-vector representations, with on the horizontal (vertical) axis the inverted false-positive (true-positive) rate. The ROC curves of IF, GMM, AE and VAE (Table 5.4) are shown in pink, orange, dark green and cyan respectively. The effective mass m_{eff} is shown in black, and combinations of the models are shown in blue (OR), red (AND), light-green (Product) and brown (Average). The black dashed line indicates the inverse false-positive rate at which $B = 100$.

In Figure 5.10 it is clear that, when using the Z_{100} metric, the VAE out performs all other algorithms trained on 4-vector components. These algorithms are compared to the effective mass $m_{eff} = E_T^{miss} + \sum_{jets} p_T$, a common discriminating variable in conventional gluino signal searches. While the VAE is by far the strongest discriminator, the OR combination also provides a fair separation between signal and background. The isolation forest especially does not perform well when trained on these 4-vector representations. This can be explained by the fact that dividing up the 4-vector space does not necessarily isolate anomalous events since anomalies generally appear as non-trivial functions of these 4-vector components. The static autoencoder and Gaussian mixture models perform slightly better as they involve defining non-linear functions of the input variables. The AND, sum and product combinations' lacklustre performance is due to the poor results yielded from the isolation forest, Gaussian mixture model, and static autoencoder algorithms in these low background regions. As the gluino mass increases statistical power is lost due to the gluino cross section decreasing.

Figure 5.11 displays the ROC curves for algorithms trained on the 4-vector components of background events for the stop signals detailed in 5.1. These signals are significantly more difficult to isolate than the gluino signals, as the kinematics of stop decays are quite similar to those of top decays when their masses are similar (as is the case for Stop 01). The physical variable used here is $m_T^{b,min} = \sqrt{2p_T^b E_T^{\text{miss}}[1 - \cos\Delta\phi(p_T^b, p_T^{\text{miss}})]}$, which is a common discriminating variable for stop signal searches. These figures show that when using the Z_{100} metric, the OR combination method consistently yields the best results of all the machine learning algorithms. The VAE is consistently very close behind and the GMM also provides fair separation, although decidedly poorer than the performance of the VAE. The poor performance of the isolation forest can be explained in a similar fashion to the case of the gluino signals, while the static autoencoder is essentially a simpler, less flexible VAE so it makes sense that its performance is significantly poorer. Surprisingly, the best results are for the case of Stop 01, the signal that is the most kinematically similar to the dominant background. While the signals become more kinematically distinct from the background as the stop mass increases, the cross section also dramatically decreases, making it infeasible to pull the signal from the background using these techniques. Notice that for the Stop 04 signal, the physical variable $m_T^{b,min}$ outperforms all other algorithms. While the sensitivity is very low, this is an indication that some discriminating information is present in this physical variable that the anomaly detection algorithms have not identified.

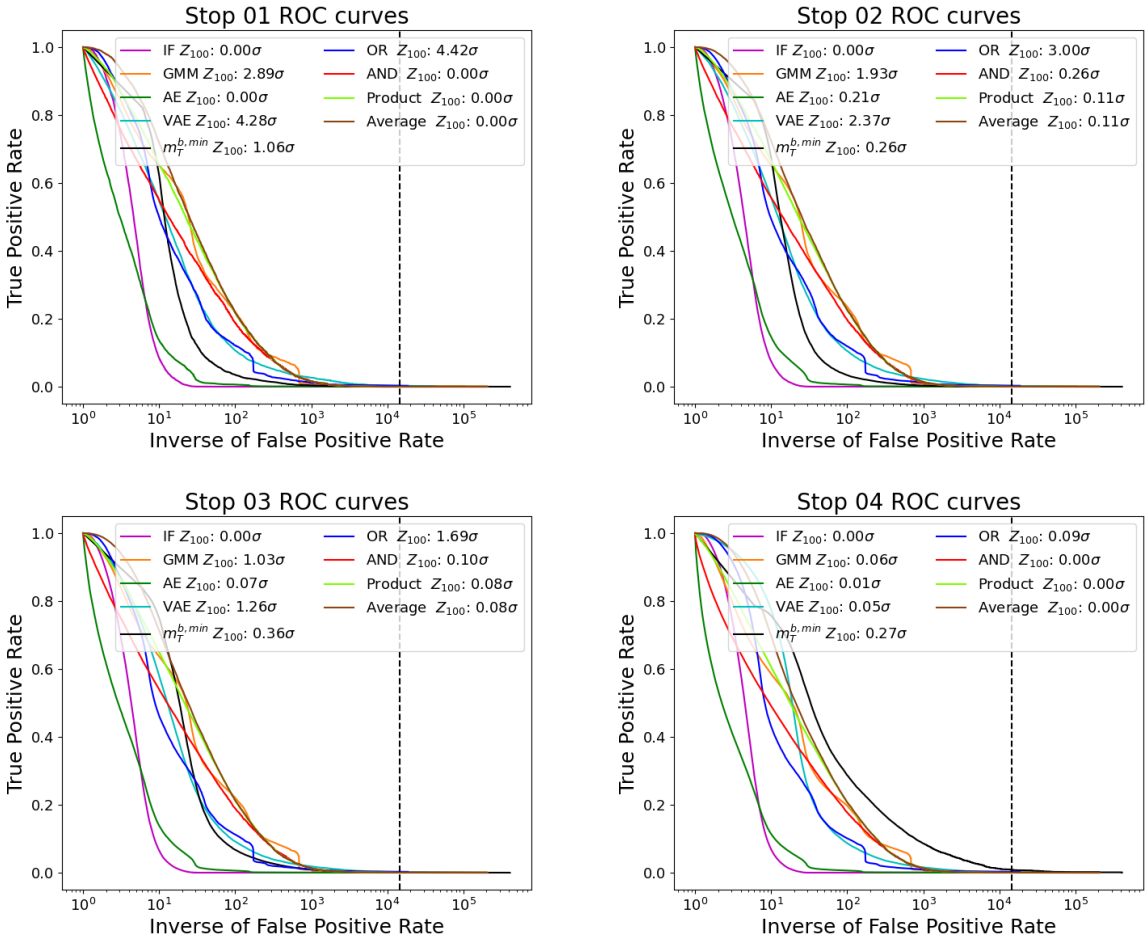


FIGURE 5.11: ROC curves for the stop signals (Table 5.1) for the algorithms applied on 4-vector representations. For further information see Figure 5.10. The physical variable that is used here is $m_{\mathrm{T}}^{\mathrm{b,min}}$.

5.4.2 Results Trained Within the Latent Space of a VAE

Let us now consider the case of training the aforementioned anomaly detection algorithms (IF, GMM, AE) on the 13 dimensional latent space representations of events instead of raw 4-vector components. These latent space variables are non-linear functions of the input variables and can be thought of as containing the "essence" of a given event compressed into 13 variables. The VAE training process remains unchanged from the previous section. The performance of these algorithms is expected to improve in the latent space of the VAE, if for no other reason than the dimensionality of the problem has reduced. However it is to be expected that the VAE will compress anomalous events differently from a typical event, meaning that the differences between signal and background events will be magnified in this new space.

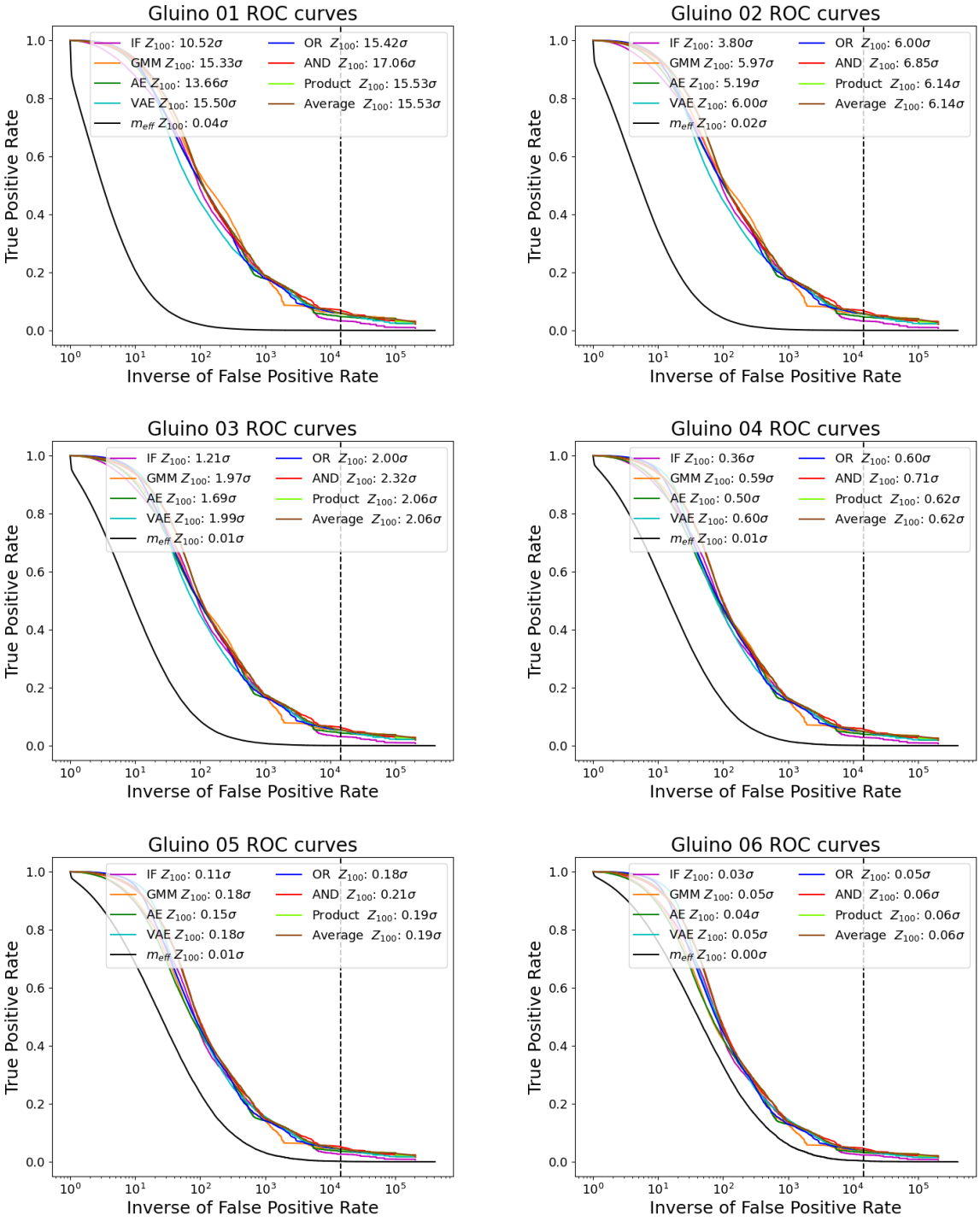


FIGURE 5.12: ROC curves for the gluino signals (Table 5.1) for the algorithms applied on latent space representations. For further information see Figure 5.10.

Figure 5.12 displays the ROC curves for algorithms trained on latent space representations of background events for the gluino signals detailed in Table 5.1. Immediately it becomes clear that the performance of the isolation forest, Gaussian mixture model, and autoencoder have dramatically improved, however they do not outperform the VAE. The

various combination methods all perform at least on par with the VAE, with the AND combination consistently performing the best.

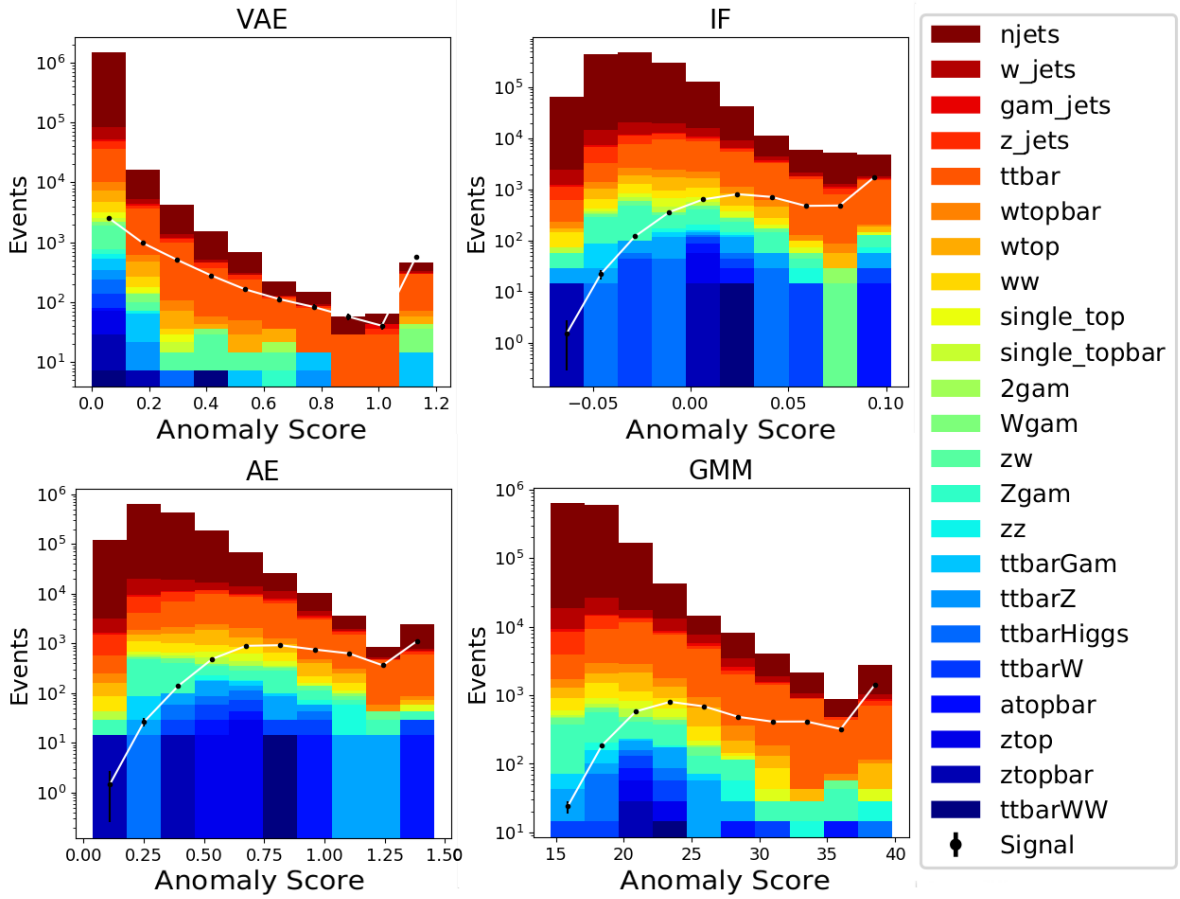


FIGURE 5.13: Anomaly score histograms derived from various algorithms for Gluino 01. The horizontal axis shows the anomaly score, and the histogram counts the number of events normalized to 36 fb^{-1} in each bin. The various colours indicate different backgrounds, while the black data points show the signal.

Figure 5.13 displays histograms of the anomaly score variable for the VAE, isolation forest, autoencoder, and Gaussian mixture model for the gluino 01 signal. The background and signal are plotted separately, with the final bin as an overflow bin to show the performance of each algorithm on its own. These histograms would never be possible to construct in an experiment and are merely tools to observe the trends within each algorithm. These figures reveal that, while the exact shape of the histogram is different for each algorithm, the general trend is the same. The background is clustered in the low anomaly score region, and the signal tends to be clustered more in the higher anomaly score regions.

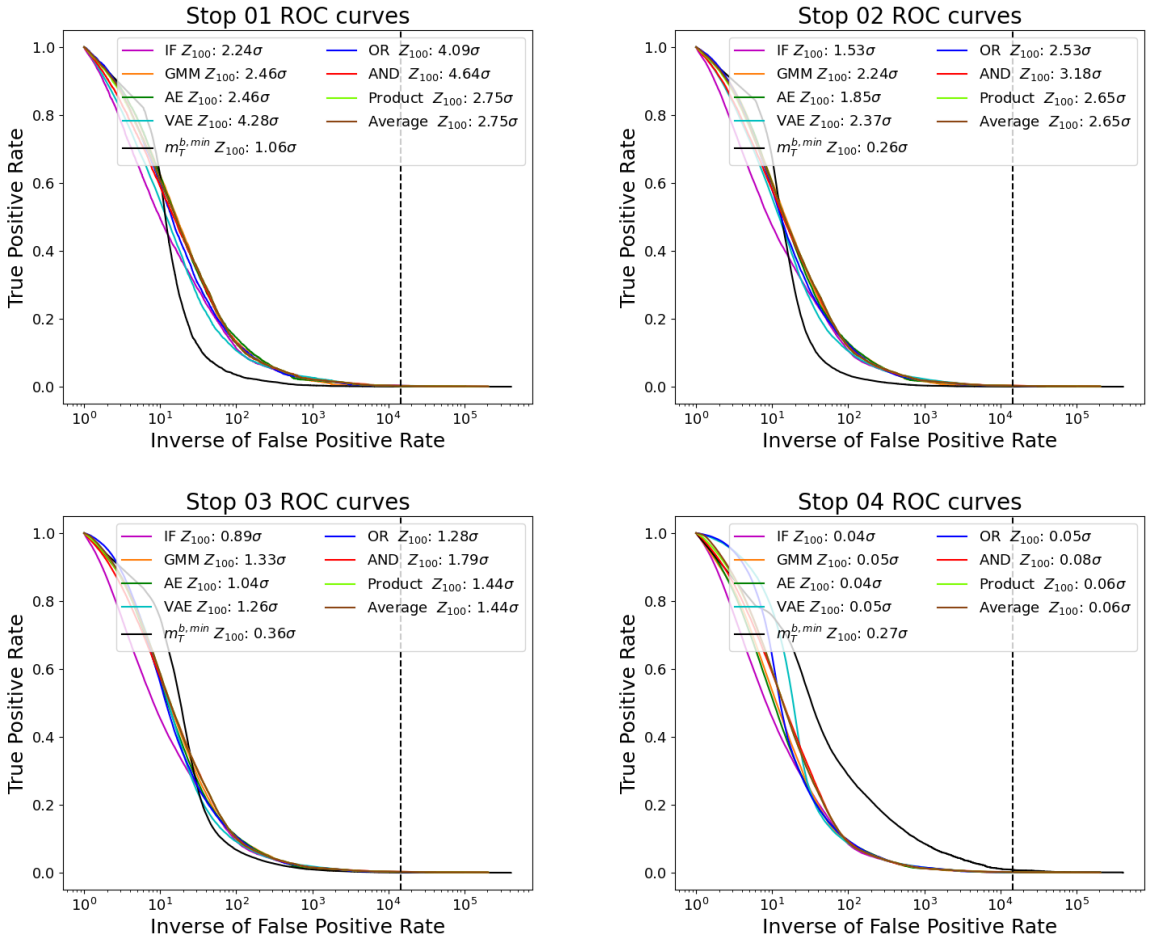


FIGURE 5.14: ROC curves for the stop signals (Table 5.1) for the algorithms applied on latent space representations. Labeling is the same as in Figure 5.10.

Figure 5.14 displays the ROC curves for algorithms trained on latent space representations of background events for the stop signals detailed in Table 5.1. Similar to the gluino results, a significant improvement in performance can be observed for the algorithms trained on latent space representations compared to those trained on 4-vector components, though none of them outperform the VAE on their own. Using the Z_{100} metric, the AND combination proves to be the most effective discriminator, with the VAE following close behind. It is clear that this signal is significantly more difficult to separate from the background which is to be expected, as stop decay can look very similar to top decay in the case where the stop mass is similar to that of the top.

Figure 5.15 displays histograms of the anomaly score variable for the VAE, isolation forest, autoencoder, and Gaussian mixture model for the Stop 01 signal. As with Figure 5.13, the signal and background are plotted separately, and setting the last bin as an overflow bin. Observing the distribution, it becomes clear why the performance is

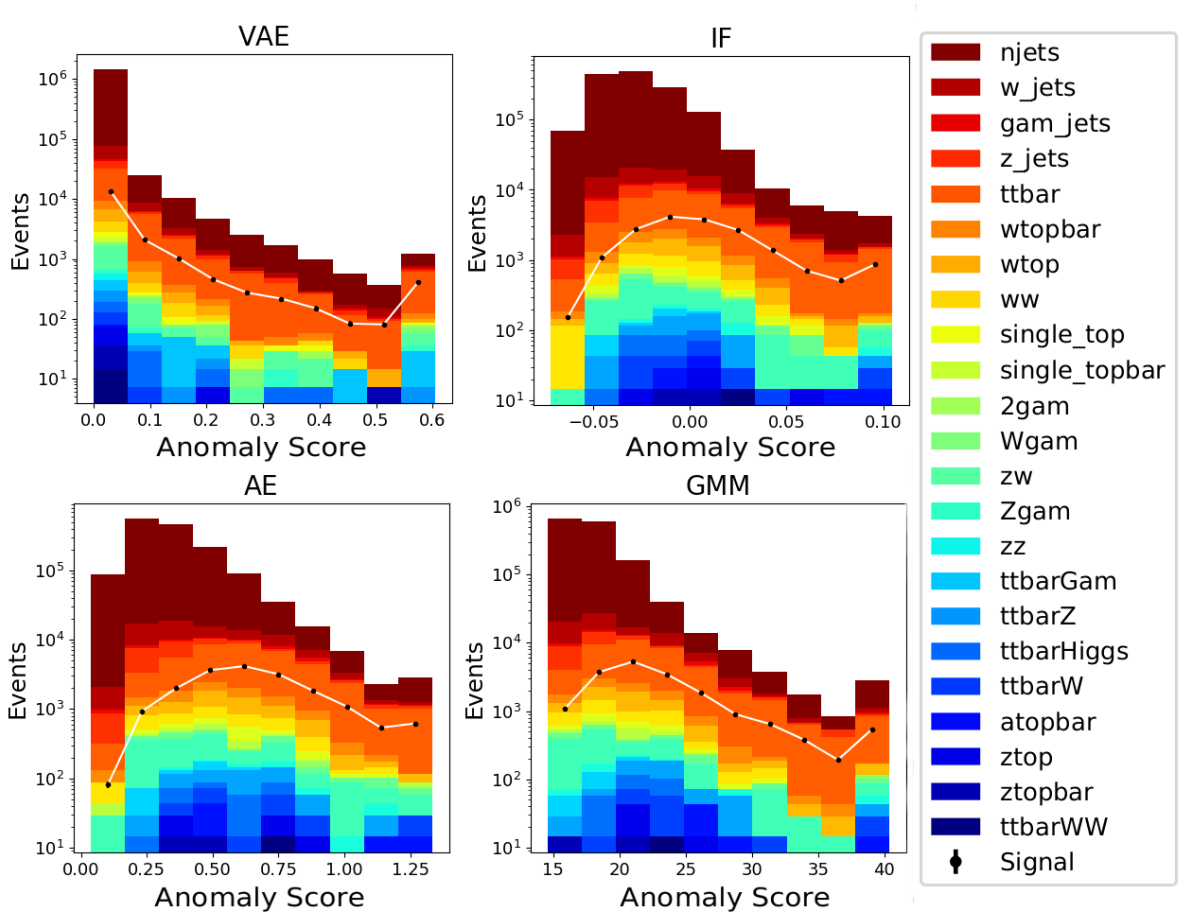


FIGURE 5.15: Anomaly score histograms derived from various algorithms for Stop 01.

worse for this signal, as the shape of the signal histogram is very similar to that of the background, though it is shifted slightly to the right. This reflects what is expected from this scenario, especially for Stop 01 where the stop mass is quite low and the kinematics of the signal are similar to the background.

5.4.3 Summary

Now that the performance of each algorithm on each signal has been examined in detail, let us zoom out and assess these findings. Figure 5.16 displays the Z_{100} values for each algorithm trained on 4-vector components and latent space representations for each signal, as well as the physical variables used for each signal. From this it is clear that the performance of the IF, GMM, and AE algorithms improves quite dramatically when trained on latent space representations. The most effective strategy for all signals tested in this experiment is to train each algorithm on the latent space representations of events and perform an AND combination. Figure 5.17 displays the same results with a 15% assumed systematic uncertainty applied. With these conditions, all signals except for

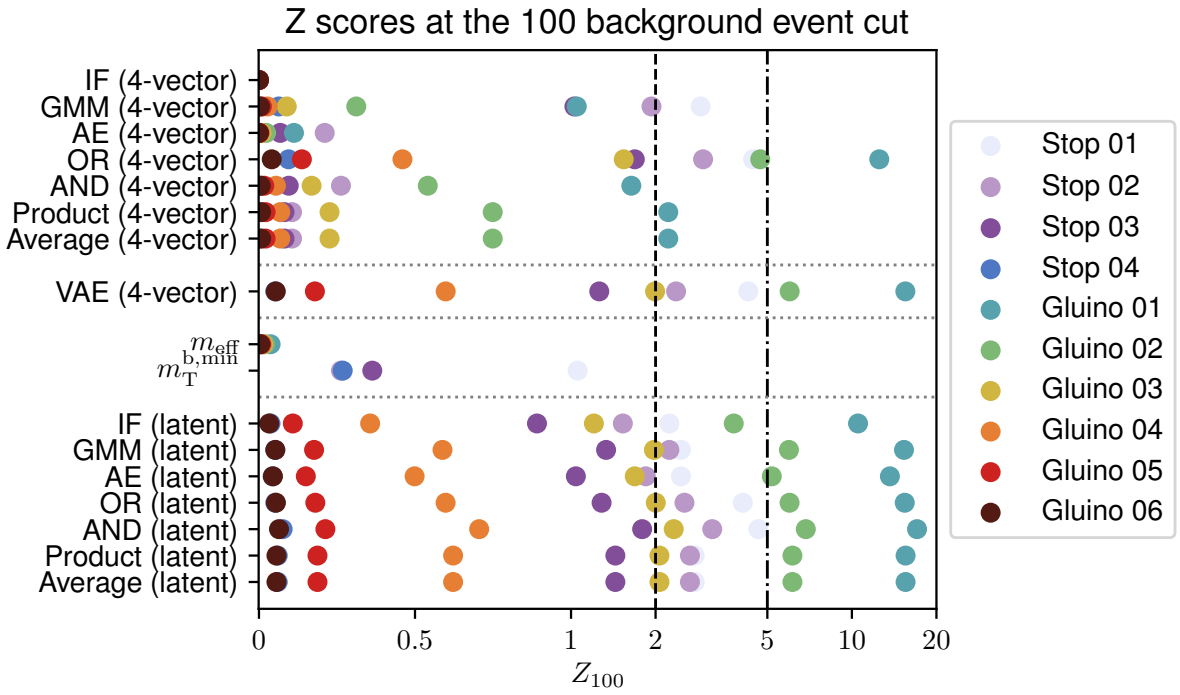


FIGURE 5.16: Z_{100} yielded from various algorithms applied to 4-vector components and latent space representations. See Table 5.1 for the signal definitions, and Table 5.4 for the definitions of the algorithms.

Gluino 01 are below discovery potential using this method, although Stop 01, Stop 02, Gluino 01, and Gluino 02 are above the exclusion limit. This poor discovery potential is not unexpected, as this analysis does not optimise on the signal. In order to gain any discovery potential one must optimise quite heavily on a particular signal, and many of the higher mass signals explored here would be difficult to discover even in a conventional analysis.

However this technique may prove useful as a preliminary step in a conventional analysis. Figure 5.18 displays 2D histograms for various physical variables, and the AND anomaly score for Gluino 01. These figures show that no significant correlation exists between these variables. To further demonstrate this point, Table 5.5 displays the Pearson correlation coefficients between each of these variables. A score of ± 1 indicates perfect positive/negative correlation, while zero implies no correlation. All of the variables in this table are very close to zero, showing that there is indeed minimal correlation. This implies that the anomaly score could be used as the first selection of an LHC analysis, perhaps at the trigger level, though this is beyond the scope of this thesis. This technique would be applicable to a wide variety of BSM physics models, as it starts with very few signal assumptions.

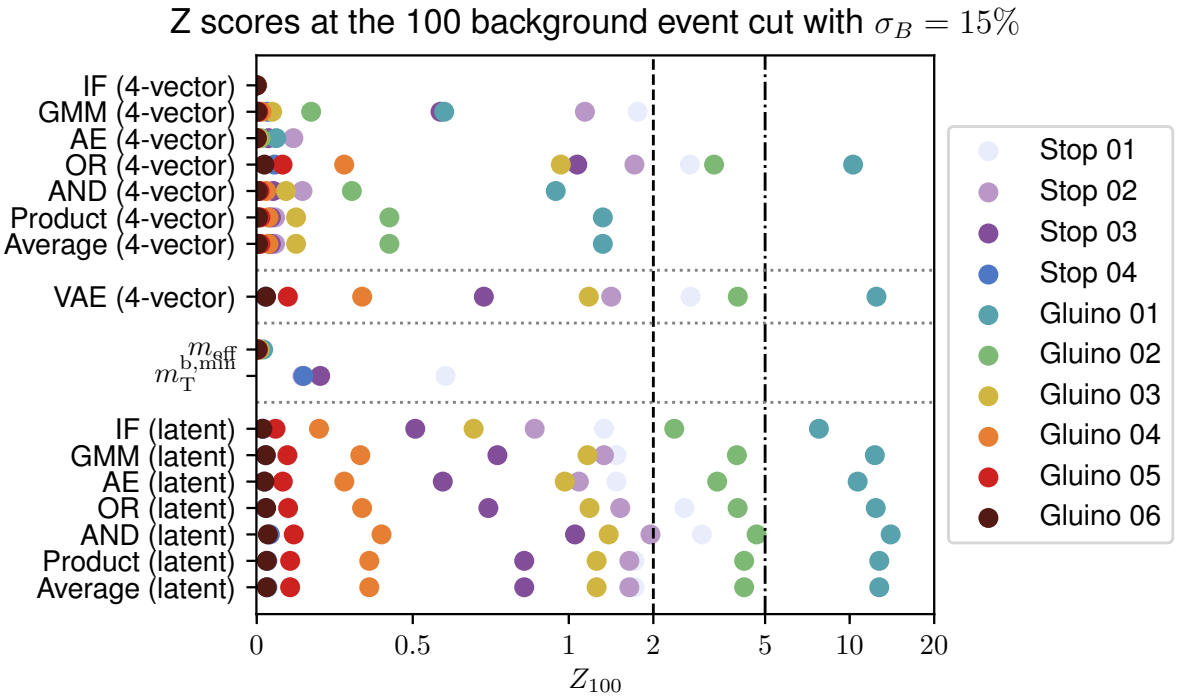


FIGURE 5.17: Z scores yielded from various algorithms applied to 4-vector components and latent space representations with a 15% relative systematic uncertainty applied. See Table 5.1 for the signal definitions, and Table 5.4 for the definitions of the algorithms.

Dataset	E_T^{miss}	H_T	m_{eff}
Background	0.12	0.14	0.15
Gluino 01	0.032	-0.030	-0.017
Gluino 02	0.038	-0.057	-0.039
Gluino 03	0.041	-0.087	-0.063
Gluino 04	0.042	-0.11	-0.084
Gluino 05	0.043	-0.14	-0.11
Gluino 06	0.046	-0.16	-0.12
Stop 01	0.082	-0.0026	0.015
Stop 02	0.13	0.032	0.061
Stop 03	0.096	-0.029	0.0053
Stop 04	0.07	-0.10	-0.056

TABLE 5.5: Pearson correlation coefficients between the AND anomaly score and various physical variables for the background and signal datasets displayed in Figure 5.18. A value of 0 implies no correlation, and a value of ± 1 implies perfect positive/negative correlation. These values suggest minimal correlation between the AND anomaly score and these physical variables.

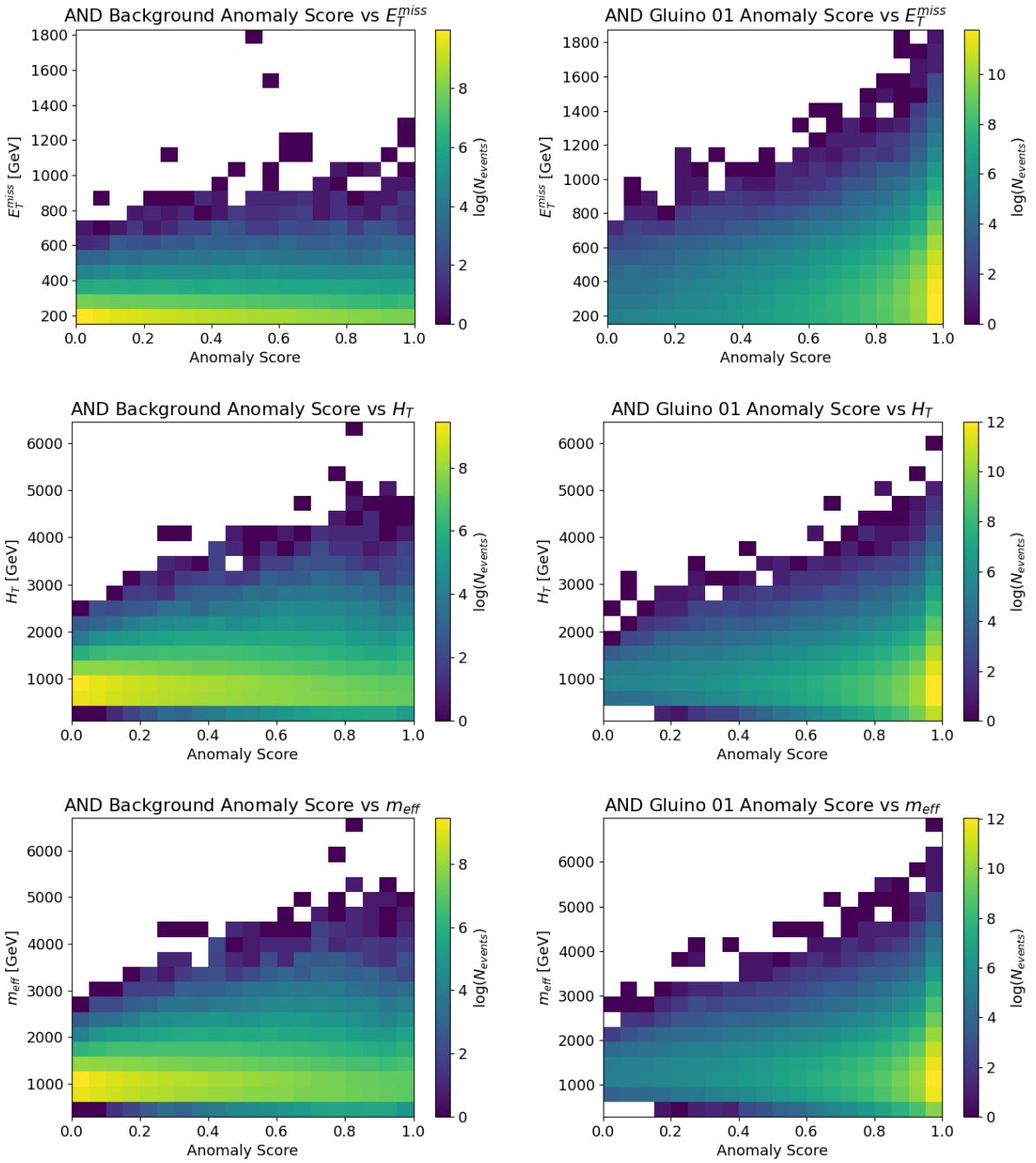


FIGURE 5.18: 2D histograms associated with Gluino 01 for background (left) and signal (right). Various physical variables are plotted on the y-axis, with the anomaly score generated from the AND combination applied in the latent space on the x-axis. The z-axis is $\log N_{\text{EVENTS}}$

5.5 Conclusion

Throughout this chapter I have detailed various machine learning and anomaly detection algorithms and discussed their applications in LHC searches. I have constructed an algorithm that assigns a measure of anomalousness on an event-by-event basis, with low

anomaly score indicating "Standard Model like" and high anomaly score indicating "not Standard Model like". The anomaly detection algorithms explored in this chapter were the isolation forest 5.2.1, Gaussian mixture model 5.2.2, static autoencoder 5.2.3, and the variational autoencoder 5.2.3, with their anomaly scores defined in Table 5.4. The data used to train each of these algorithms is the SM data set published in Ref. [53]. The dataset used to test the performance is a collection of supersymmetric benchmark scenarios, consisting of a series of gluino and stop signals, detailed in Table 5.1. The variational autoencoder was trained on 4-vector components. Each of the other algorithms were trained on both 4-vector components, and on latent space representations of events yielded from the variational autoencoder. In Section 5.4 the performance of a given algorithm is measured by calculating Z_{100}, the Z-score at a 100 background event cut. It is important to note that the conclusions drawn in this chapter do not change significantly when modifying this background cut. Using the Z_{100} metric, it has been shown that by training these algorithms within the latent space of the VAE, a significant improvement in performance is observed. The VAE on its own gives a very strong separation between signal and background in most cases, however I found that by utilizing various combination methods this performance was able to be improved further. The four combination methods explored in this chapter include AND, OR, product, and averaging combinations. When trained on 4-vector components, the OR combination gives the best performance for the stop quark, while the VAE gives the best result for the gluino case. However when trained on latent space components, the AND combination outperforms all other methods for both signals. In both the stop and gluino case these methods are compared to common discriminating physical variables, m_{eff} for the gluino signal, and $m_T^{b,min}$ for the stop signal, and observe that the anomaly detection techniques detailed in this chapter consistently outperform them. To summarise, the most effective method found in this chapter is as follows:

- Train a VAE on 4-vector components of SM background events.
- Train a variety of anomaly detection techniques on the latent space representations of the aforementioned SM background events.
- Normalise the anomaly scores by background efficiency, and perform an AND combination to determine the anomaly score on an event-by-event basis.

While this technique does not have discovery potential on its own, it has the advantage of being signal-model independent. The anomaly score which I have developed is not correlated with other commonly used physical variables such as E_T^{miss}, H_T, and m_{eff}, suggesting that it could be used as an additional variable to perform signal region cuts. I posit that this technique could be viable for use as the first selection in a standard LHC analysis.

6 The Dark Machines Anomaly Score Challenge

This chapter, based on Ref. [67], expands on the comparison of techniques detailed in the previous chapter by applying them, alongside a handful of techniques designed by other groups, to a wide variety of BSM signal scenarios. This was carried out as a challenge organised by the Dark Machines collaboration [68], who seek to answer cutting edge questions about dark matter using the most advanced data science techniques available, especially machine learning. This challenge involved a number of groups who each developed anomaly detection algorithms with the aim of separating a number of BSM signals from the SM background. Each group had access to the same datasets in order to ensure consistency between groups.

6.1 Dataset

The background dataset used here is identical to that which was used in Chapter 5. However all of the signal models are different, including a variety of SUSY and non-SUSY BSM physics models designed to cover different regions of the parameter space. Background and signal events are divided into 4 seperate channels designed to target different types of BSM physics scenarios. Channel 1 focuses on hadronic activity with large missing transverse energy, which is good for mono-jet dark matter signatures and any strongly produced SUSY signals. Channel 2a and 2b require leptons, making them more sensitive to electroweak signals. Channel 3 is the most inclusive, catching most of the signals except some softer electroweak signals. The channels are defined as:

- Channel 1 (2.1×10^5 SM events):

$$H_T \geq 600 \text{ GeV}, \quad E_T^{\text{miss}} \geq 200 \text{ GeV}, \quad E_T^{\text{miss}}/H_T \geq 0.2, \tag{6.1}$$

  with at least four (b)-jets with $p_{\text{T}} > 50$ GeV, and one (b)-jet with $p_{\text{T}} > 200$ GeV.

- Channel 2a (2.0×10^4 SM events):

$$E_T^{\text{miss}} \geq 50 \text{ GeV}, \tag{6.2}$$

and at least 3 muons/electrons with $p_T > 15$ GeV.

- Channel 2b (3.4×10^5 SM events):

$$E_T^{\text{miss}} \geq 50 \text{ GeV}, \quad H_T \geq 50 \text{ GeV}, \tag{6.3}$$

and at least 2 muons/electrons with $p_T > 15$ GeV.

- Channel 3 (8.5×10^6 SM events):

$$H_T \geq 600 \text{ GeV}, \quad E_T^{\text{miss}} > 100 \text{ GeV}. \tag{6.4}$$

6.1.1 Signal Generation

For the signal scenarios, a series of SUSY and non-SUSY BSM physics scenarios are examined. The first four detailed here involve a Z' particle, which is a hypothetical gauge boson arising from extensions to the electroweak symmetry of the standard model.

- The Z' + monojet model [69, 70, 71] contains a 2 TeV Z' which decays fully invisibly to 50 GeV Dirac dark matter. Dirac dark matter is form of dark matter that obeys the Dirac equation [72]. This process is referred to as `monojet_Zp2000.0_DM_50.0` throughout the chapter.

- The Z'+W/Z model [69, 70, 71] also contains a 2 TeV Z', similarly decaying fully invisibly to 50 GeV Dirac dark matter. This process is referred to as `monoV_Zp2000.0_DM_50.0` throughout the chapter.

- The Z' + single top process [69, 70, 71] contains a 200 GeV Z'. This process is referred to as `monotop_200_A` through out the chapter.

- The Z' in lepton-violating $U(1)_{L_\mu - L_\tau}$ [73, 74] process involves a 50 GeV Z' decaying to leptons and neutrinos. There are two processes included, a 3-lepton final state denoted `pp23mt_50`, and a 4-lepton final state denoted `pp24mt_50`.

- The R-parity violating SUSY (denoted $\not{R}$-SUSY) [75, 76] stop-stop process has pair production of 1 TeV supersymmetric stops which decay to leptons and b-quarks. This process is referred to as `stlp_st1000` throughout the chapter.

- The $\not{R}$-SUSY [75, 76] squark-squark process features 1.4 TeV squark pair production. The mass of the neutralino is 800 GeV, and the squarks decay down to jets. This process is referred to as `sqsq1_sq1400_neut800` throughout the chapter.

- The SUSY [77, 78, 79] gluino-gluino process involves the pair production of two gluinos which decay into jets and neutralinos, yielding high missing energy. Two mass benchmarks are examined. The first, denoted `glgl1400_neutralino1100`, consists of 1.4 TeV gluinos and 1.1 TeV neutralinos. The second spectrum, denoted as `glgl1600_neutralino800`, consists of 1.6 TeV gluinos and 800 GeV neutralinos.

- The SUSY [77, 78, 79] stop-stop process consists of pair produced stops decaying to a top quark and a neutralino, yeilding high missing energy. The stop mass is 1 TeV and the neutralino mass is 300 GeV. This is referred to as `stop2b1000_neutralino300` throughout the chapter.

- The SUSY [77, 78, 79] squark-squark process consists of 1.8 TeV squarks decaying to jets and neutralinos, yielding high missing transverse energy. The neutralino mass is 800 GeV. This is referred to as `sqsq_sq1800_neut800` throughout the chapter.

- The SUSY [77, 78, 79] chargino-neutralino processes involve the charged-current production of a chargino and neutralino, with the chargino decaying to a W plus a lightest neutralino (the LSP). Two different mass spectra are considered for this process. The first, denoted as `chaneut_cha200_neut50`, consists of a 200 GeV chargino and a 50 GeV neutralino. The second, denoted as `chaneut_cha250_neut150` consists of a 250 GeV chargino and a 150 GeV neutralino.

- The SUSY [77, 78, 79] chargino-chargino process involves the neutral current pair production of charginos, decaying to a W plus a lightest neutralino. In this case three different mass spectra are considered. The first, denoted as `chacha_cha300_neut140` contains a 300 GeV chargino and a 140 GeV neutralino. The second, denoted as `chacha_cha400_neut60`, has a much higher mass splitting, with a 400 GeV chargino and a 60 GeV neutralino. The last spectrum, denoted as `chacha_cha600_neut200`, is a much heavier scenario, with a 600 GeV chargino and a 200 GeV neutralino.

Each of these scenarios will not necessarily show up in every channel. The BSM models that are present in each channel are summarised in Table 6.1.

6.1.2 Performance Metrics

Each algorithm, as detailed in Section 6.2, will assign an anomaly score on an event-by-event basis. Recall from the previous chapter how a ROC curve is defined. In this chapter a number of metrics are constructed to measure the discriminating power of a given algorithm on a signal. The area under the curve (AUC) is a common metric used to assess the performance of a classification algorithm. An AUC of 1 indicates perfect classification, while an AUC of 0.5 indicates a random guess. An issue with using the AUC

BSM process	Channel 1	Channel 2a	Channel 2b	Channel 3
Z' + monojet	×	×		×
$Z' + W/Z$				×
Z' + single top	×			×
Z' in lepton-violating $U(1)_{L_\mu - L_\tau}$		×	×	
$\not{R}$-SUSY stop-stop	×		×	×
$\not{R}$-SUSY squark-squark	×			×
SUSY gluino-gluino	×	×	×	×
SUSY stop-stop	×			×
SUSY squark-squark	×			×
SUSY chargino-neutralino		×	×	
SUSY chargino-chargino			×	

TABLE 6.1: BSM processes in each channel.

for these purposes is that the primary area of concern is the very low background efficiency regions, where there are very few background events. Therefore the signal efficiency at three low background efficiency working points is also used to assess the performance. The four metrics used in this chapter are:

- Area under the Curve (AUC),
- The signal efficiency at $\epsilon_b = 10^{-2}$,
- The signal efficiency at $\epsilon_b = 10^{-3}$, and
- The signal efficiency at $\epsilon_b = 10^{-4}$.

Additionally combinations of these metrics are used later in this chapter to further explore the performance of each algorithm on the signals.

6.2 Algorithms

The algorithms explored in this project were not all written by me so I will not go into any great detail on them. The techniques explored in these other algorithms include Kernel Density Estimation (KDE) [80], Gaussian Mixture Models (GMMs) [58], flow models [81], (variational) autoencoders [82], and Generative Adversarial Networks (GANs) [83]. The technique I submitted is similar to that which was explained in Section 5.2, but with a few key improvements. Remember, the aforementioned algorithm follows the following steps:

1. Define a VAE architecture.
2. Train it on a subset of the background data.

Parameter	Values
batch size	$[1000, 10000]$
β term	$[1e{-}5, 1e{-}4, 1e{-}3, 0.01, 0.1]$
latent space dimensions	$[4, 13, 20, 30]$

TABLE 6.2: VAE model hyper-parameters.

3. Pass the remainder of background data + signal through the VAE and obtain the latent space representations for each event.

4. Train further anomaly detection algorithms on the latent space representations of the background events.

5. Pass the remaining background and signal events through these algorithms, obtaining multiple measures of anomalousness for each event.

6. Normalise each anomaly score to uniform background efficiency.

7. Perform various combinations (logical and/or, average, and product).

8. Construct a ROC curve and compare the area under the curve (AUC), and signal efficiencies at various background efficiencies.

Previously, the algorithms used in step 4 included the isolation forest (IF), Gaussian mixture model (GMM), and static autoencoder (AE). For the Dark Machines anomaly score challenge, I also introduced the k-means algorithm [84], a simple algorithm that determines anomalousness by fitting a number of "centroids" to the dataset and defining the anomaly score as the distance to the nearest centroid. Subsection 6.2.1 explains the k-means algorithm in detail.

The other change made for the Dark Machines anomaly score challenge is that combinations are done by iterating through every possible combination of algorithms, and picking the one with the highest AUC. Remember that a combination is not necessarily going to be better than the algorithms that it is composed of. By only taking those that improve the AUC, every combination is guaranteed to yield results at least as good as the best performing single algorithm.

In training the VAE for this dataset, a small hyper-parameter scan was conducted. This was done because it's not immediately obvious what VAE architecture will yield the best latent space structure for the training of the other techniques. By testing a number of different architectures it is possible to determine the best setup for anomaly detection within the latent space. A summary of these parameters is detailed in Table 6.2.

6.2.1 k-means Clustering

The k-means clustering algorithm works as an anomaly detection algorithm by first identifying clusters within the training data and then identifying points that lie far from these clusters. The k-means algorithm [84] is a simple clustering algorithm which attempts to partition n points into k clusters. The choice of the number of "centroids" k is an arbitrary one, which must be made on a case-by-case basis. The algorithm is as follows:

1. **Initialization:** The positions of the k centroids are chosen according to the k-means++ algorithm [85]. The algorithm is as follows:

   Take one centroid c_1 chosen uniformly at random from X, where X is the dataset. Then choose the remaining c_i, with probability $\frac{D(x)^2}{\sum_{x \epsilon X} D(x)^2}$ where $D(x)$ is the shortest distance from a point x to the closest centroid that has already been chosen.

2. **Assignment:** Each point is assigned to its nearest centroid. One could use any number of distance metrics for this step, however I have decided to simply use the squared Euclidean distance.

3. **Update:** Recalculate the positions of the centroids as the mean of all points assigned to a given centroid. Steps 2 and 3 are repeated until a tolerance is passed. In this algorithm the tolerance is the Frobenius norm of the difference in the cluster centers of two consecutive iterations. The Frobenius norm of a matrix A is given by $||A||_F = \langle A, A \rangle = \sqrt{\sum_{i=1} \sum_{j=1} |a_{ij}|^2}$. In this experiment the tolerance is set to be 10^{-4}.

The k-means clustering algorithm is very fast and has very low memory requirements. However it can fall into local minima, making it beneficial to run it multiple times. The anomaly score of a given point is then given as the distance to the nearest centroid.

6.3 Results

The algorithms that I submitted for the Dark Machines anomaly score challenge did not perform as well as many of the other algorithms submitted in the challenge. That is not to say that my algorithms do not have their strengths, indeed they do very well in some aspects. In order to focus more on my own work I have reproduced the graphs shown in the official paper using only my own algorithms. In this way this chapter can more closely analyse the strengths and weaknesses of each variation of techniques trained on the latent space representations of events.

Figure 6.1 contains the results using the four metrics discussed in Section 6.1.2. Namely AUC, $\epsilon_S(\epsilon_B = 10^{-2})$, $\epsilon_S(\epsilon_B = 10^{-3})$, and $\epsilon_S(\epsilon_B = 10^{-4})$. The most important

thing to notice about this plot is that not every signal is equally easy for the anomaly detection algorithms to handle. The chargino-neutralino signals with small mass splittings have an AUC close to 0.5, indicating that most algorithms perform no better than randomly guessing. The small mass splitting leads to less energetic decay products, meaning that the anomalous events are not in the tails of distributions and are thus difficult for the algorithms to detect. In contrast, the gluino-neutralino and RPV stop signals have high AUC's for most anomaly detection algorithms. Looking closer, it becomes clear that the algorithms have worse performance on channel 2a than on all other channels. This channel has the most restrictive preselection cuts and so has the least amount of training data, making it difficult for the algorithms to properly learn the background.

6.3.1 Figures of Merit

Let us now define a handful of "figures of merit" which are used to decide which algorithms yield the best performance. There are a number of ways one can use each of the performance metrics to determine the "best" anomaly detection algorithm. There are multiple things one might care about. For example, is an algorithm that performs outstandingly on a few signals better than an algorithm that performs well on many signals? With this in mind a number of figures of merit are defined in order to assess the performance of these algorithms.

- **Top Scorer Method:** Models which have the highest score, the most number of times. This prioritises algorithms which perform outstandingly for a few signals. This is not necessarily the greatest definition of "best", as an algorithm that consistently comes second will not be registered by this technique, while an algorithm that gets the highest score for just two or three signals and zero for the rest would do very well. Figure 6.2 shows the best algorithms applied to all channels/signals using this method. This figure shows that no one algorithm dominates in any two metrics, however it is clear that a few groups of algorithms have risen to the top. The three VAE algorithms with a 20 dimensional latent space and a small β value all do very well. The OR combination also tends to do well. Interestingly, two different applications of the primitive k-means algorithm provide the best AUC and $\epsilon_S(\epsilon_B = 10^{-3})$ results. The k-means algorithm does not excel using any other metric.

- **Top 5 Method:** This is a generalised version of the top scorer method. Using this figure of merit the number of times a given algorithm is within the top 5 scores are counted for a given metric. This softens the issues with the top scorer method, though the same issues still stand. Figure 6.3 shows the best algorithms using this method. Here the results are dominated by the VAE and OR algorithms. Using

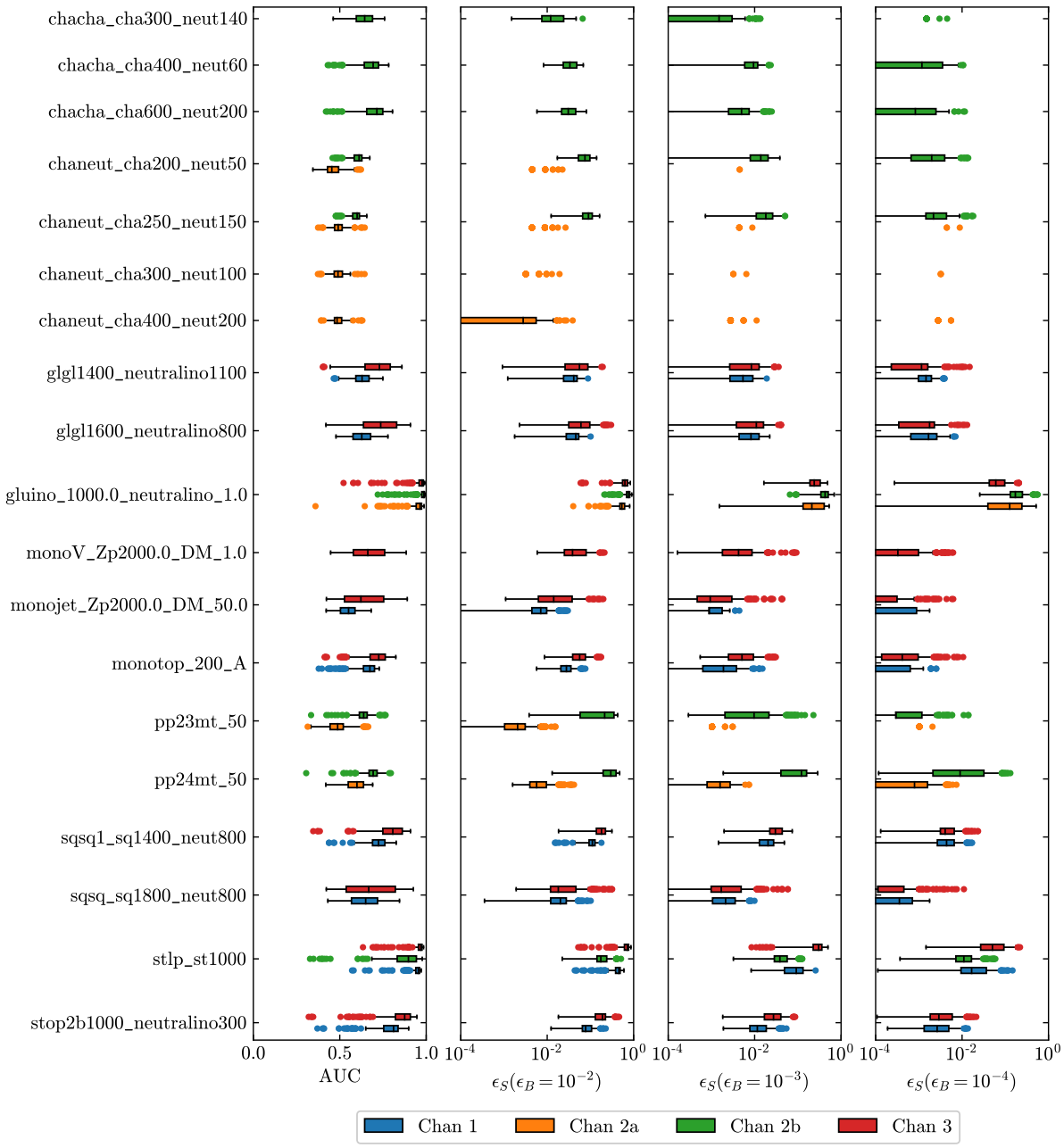


FIGURE 6.1: Box plots for each of the physics signals in the hackathon dataset. These summarize the span of results for the many anomaly detection models trained on background only samples. Channel 2a has the tightest pre-selection cuts, and therefore less data, which leads to the signals looking less anomalous.

this metric, the VAE dominates using the $\epsilon_S(\epsilon_B = 10^{-4})$ metric, and tends to prefer a small latent space and β value. The OR on the other hand, dominates using the AUC, and tends to prefer a larger latent space and small β value.

- **Average Ranking Method:** Taking the average of each rank an algorithm gets favours consistently good scores over a few very high rankings. Figure 6.4 shows the

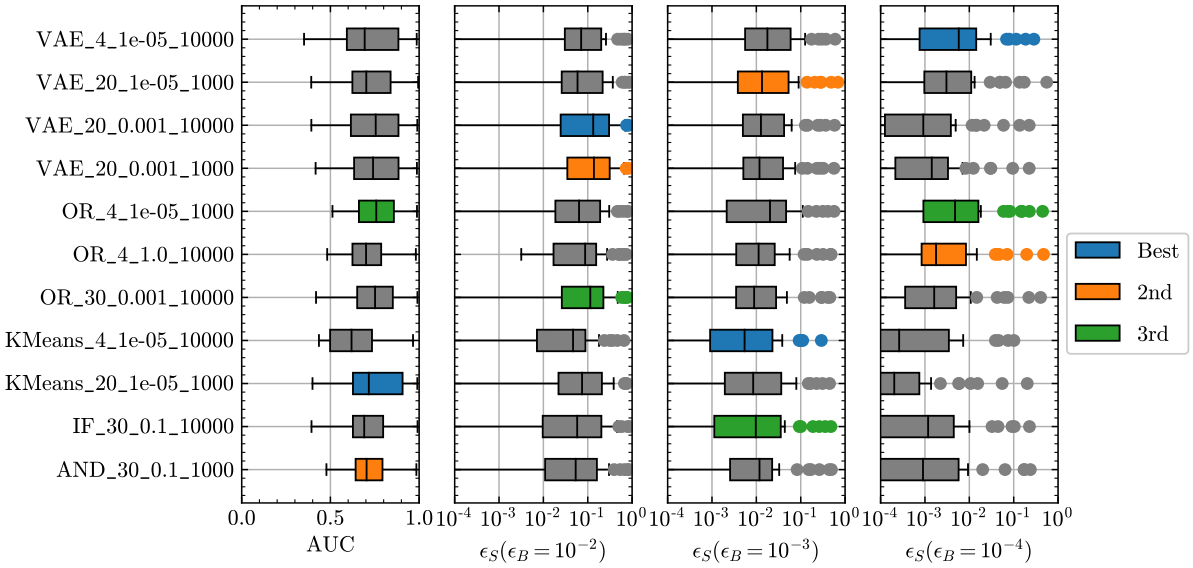


FIGURE 6.2: Box plots summarizing my latent space anomaly detection techniques applied to all of the new physics signals. The colours denote the technique that have the top score the most times.

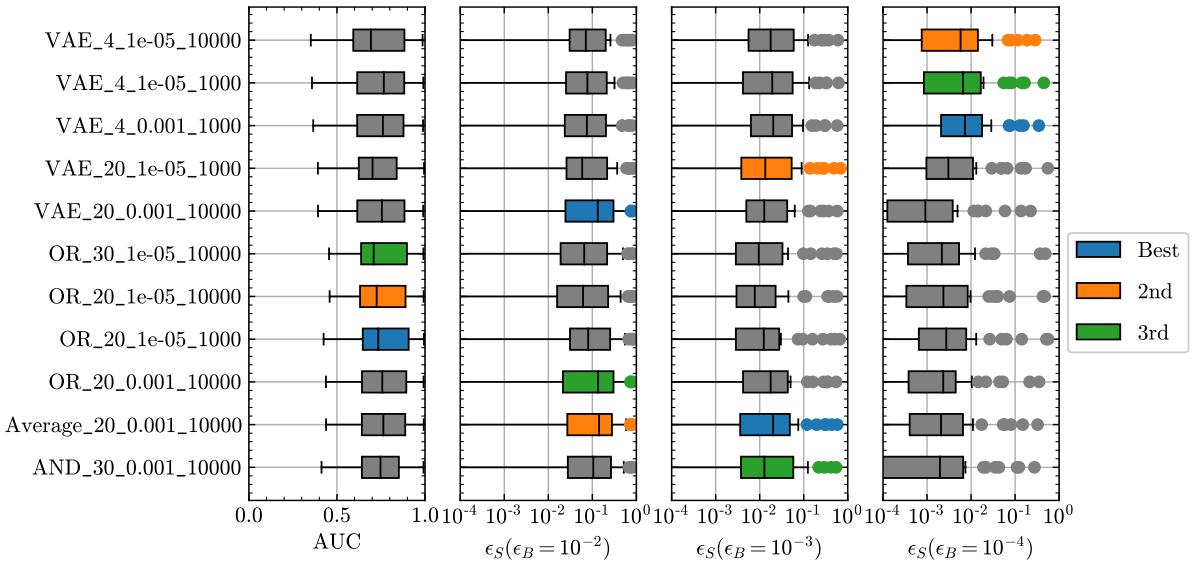


FIGURE 6.3: Box plots summarizing my latent space anomaly detection techniques applied to all of the new physics signals. The colours denote the technique that appear in the top five scores the most times.

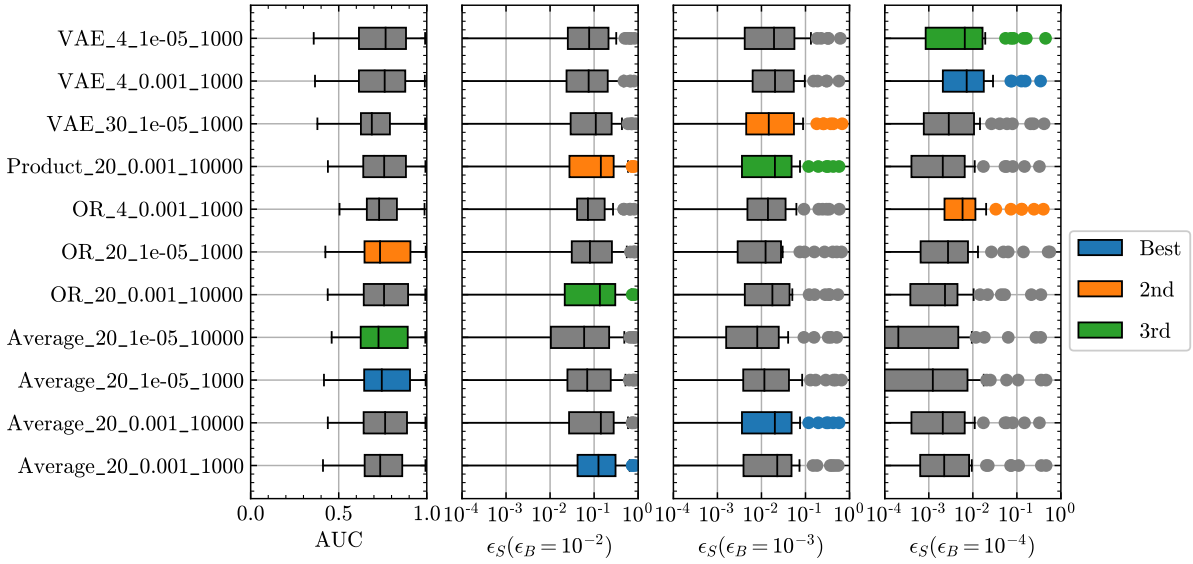


FIGURE 6.4: Box plots summarizing my latent space anomaly detection techniques applied to all of the new physics signals. The colours denote the techniques that have the highest average rankings.

best algorithms using this method. Alongside the VAEs and OR combinations that performed well in the previous two plots, the average combination method with a 20 dimensional latent space and a small β value also does quite well. This algorithm has variations placing first in AUC, $\epsilon_S(\epsilon_B = 10^{-2})$, and $\epsilon_S(\epsilon_B = 10^{-3})$.

- **Highest Mean Score Method:** The previous three figures of merit are based on the rankings of algorithms. Taking the numerical mean of a given metric will perform similarly to the average ranking method but will be subtly different. Figure 6.5 shows the best algorithms using this method. Here, there are fewer VAE algorithms that excel and more combination algorithms. This indicates that the combination algorithms are more consistent than the VAE which, while still powerful, appears to be less reliable across all signal models.

- **E) Highest Median Score Method:** The median score is similar to the mean, however it has more of a preference for algorithms that perform consistently. Figure 6.6 shows the best algorithms using this method. The results using the highest median score method appear similar to the highest mean score method, with combination methods tending to dominate. The two best VAE methods have a very small latent space, and small β values, which is similar to what is observed with the top 5 method.

- **Highest Minimum Score Method:** Algorithms which have the highest performance floor are also interesting. This figure of merit will prioritise algorithms which

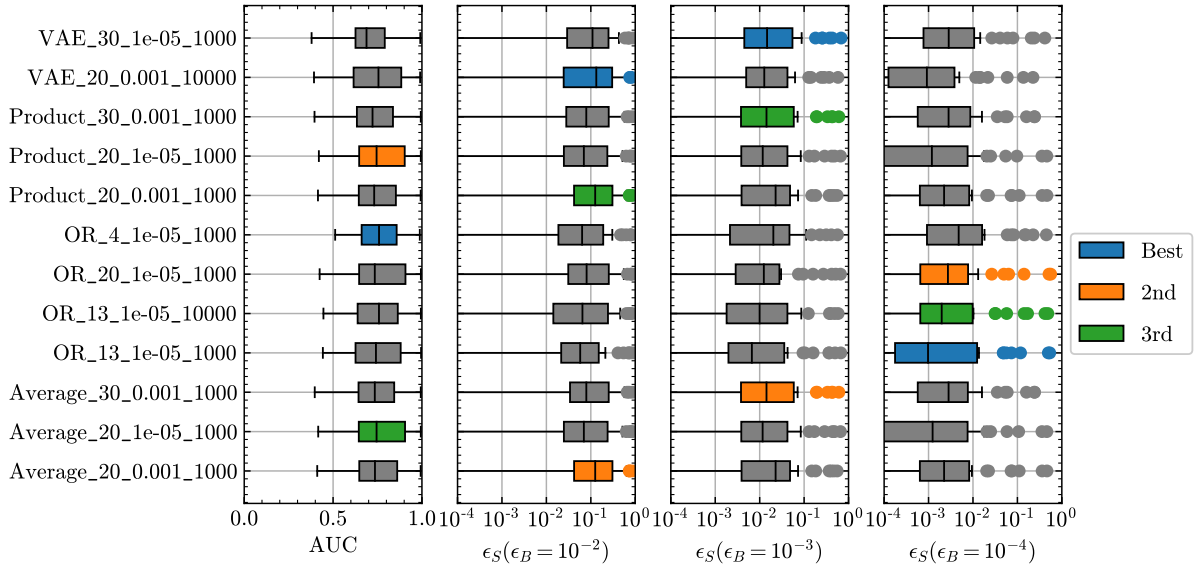


FIGURE 6.5: Box plots summarizing my latent space anomaly detection techniques applied to all of the new physics signals. The colours denote the techniques that have the highest mean scores for each of the figures of merit.

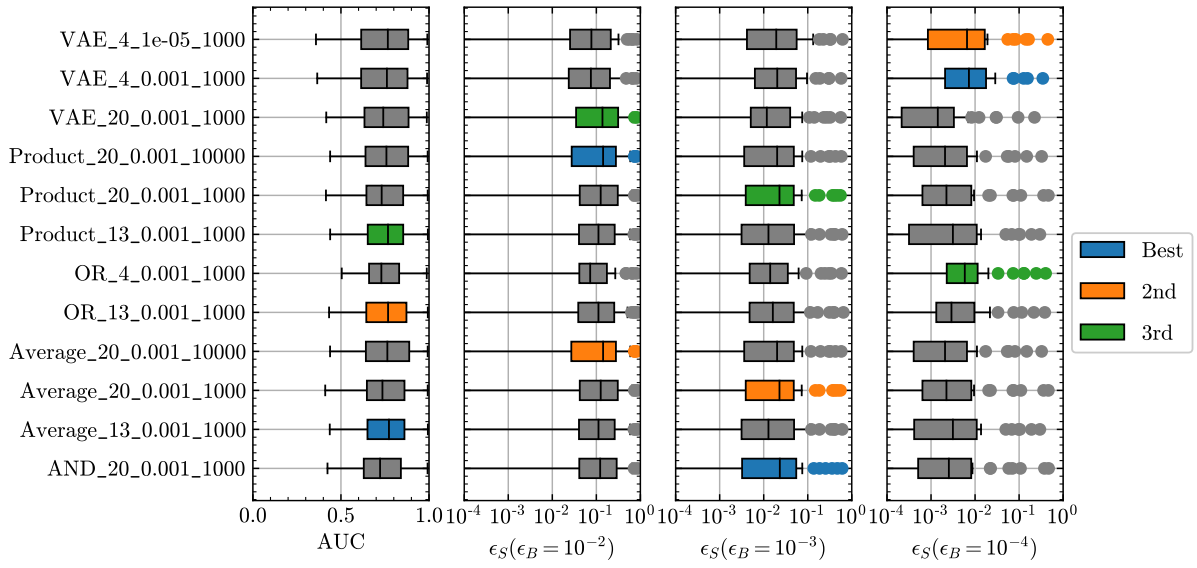


FIGURE 6.6: Box plots summarizing my latent space anomaly detection techniques applied to all of the new physics signals. The colours denote the techniques that have the highest median scores for each of the figures of merit.

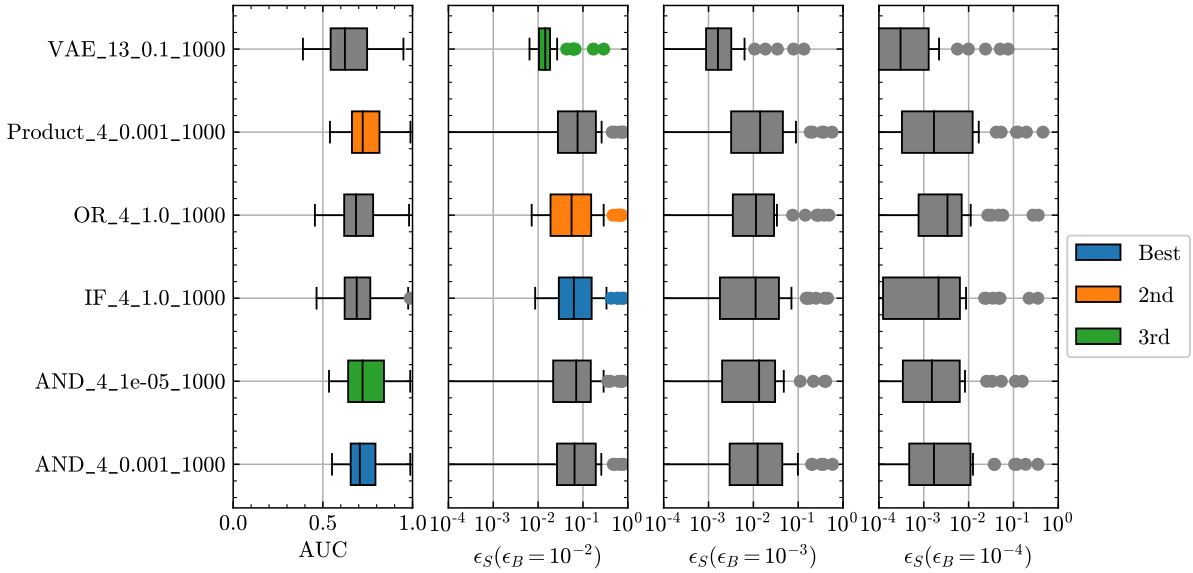


FIGURE 6.7: Box plots summarizing my latent space anomaly detection techniques applied to all of the new physics signals. The colours denote the techniques that have the highest minimum scores for each of the figures of merit. No technique has ϵ_S above 0 for all physics signals for $\epsilon_B = 10^{-3}$ or $\epsilon_B = 10^{-4}$.

do not do excessively poorly on any one signal. Figure 6.7 shows the best algorithms using this method. No algorithm was able to get the metrics $\epsilon_S(\epsilon_B = 10^{-3})$ or $\epsilon_S(\epsilon_B = 10^{-4})$ above zero for all signal models. However for the AUC metric, the AND and product combination methods do best, while the isolation forest, OR combination and VAE do best for $\epsilon_S(\epsilon_B = 10^{-2})$. Interestingly, all of these algorithms except for the VAE use a 4-dimensional latent space, and all have a smaller batch size of 1000.

Summarising these results it is clear that the VAE on its own consistently does quite well with all figures of merit using the ϵ_S metric at various background efficiencies. OR and Average combinations also consistently do well with all figures of merit using all metrics. As for VAE architectures, smaller beta values are generally preferred and larger latent spaces tend to do better.

6.3.2 Significance Improvement

The metric which was considered to be the most useful is dubbed the "significance improvement". At the LHC the chance to discover a given BSM model is dependent upon both the complexity of the signal and its cross section. This metric remains agnostic about the cross section of the process, but instead emphasises how the chance of discovery improves with the anomaly detection algorithms applied. Let us assume that there

are enough events that the background is well modelled by Gaussian statistics, meaning that the standard deviation is equal to the square root of the number of events. Hence the significance of a given new physics signal can be written as

$$\sigma_S = \frac{S}{\sqrt{B}}, \tag{6.5}$$

where S and B are the number of signal and background events respectively. After the anomaly detection algorithm is applied, and a cut is made using the anomaly score, this value changes:

$$\sigma_{\text{cut}} = \frac{\epsilon_S S}{\sqrt{\epsilon_B B}} = \frac{\epsilon_S}{\sqrt{\epsilon_B}}\sigma_S. \tag{6.6}$$

Hence the significance improvement can be defined as

$$\text{SI} = \frac{\epsilon_S}{\sqrt{\epsilon_B}}. \tag{6.7}$$

It's important to note that this metric does not say whether or not a given technique is capable of discovering new physics, as this is still dependant on the cross section of the process. Instead it suggests how much the anomaly detector is able to enhance the statistical purity of the signal over the SM background. The maximum significance improvement for a signal on a given algorithm is defined as the maximum value of SI over the three working points where $\epsilon_B = 10^{-2}, 10^{-3}, 10^{-4}$.

Figure 6.8 shows the maximum significance improvement for all signals and algorithms. The results appear similar to those in Figure 6.1, with the algorithms performing well on the gluino-neutralino and RPV stop signals, and poorly on the low mass splitting chargino-neutralino signals. On channel 2a the algorithms rarely achieve better than unit significance improvement, however it is the case that for all signals except the chargino-neutralino signals, there is a channel for which an algorithm will have a maximum significance improvement greater than 1.

In the final section of this analysis, the total improvement (TI) is defined as the significance improvement over all available channels. This means that if a method has a significance improvement of < 1 for a signal in channel 2a but an improvement > 1 in channel 2b, only the value from channel 2b is considered.

Figures 6.9 and 6.10 show the minimum, median and maximum total improvements for each algorithm across all physics models. This figure shows that the minimum significance improvement is usually zero, although a handful of algorithms rise above this value, notably k-means, VAE and isolation forest. The most useful plot is Median TI vs Max TI, where several clusters can be observed. Notably the VAE on its own looks to be very dependent on its hyperparameters, with a cluster in the high median/maximum region and a cluster in the low median/maximum region. The OR combination also consistently

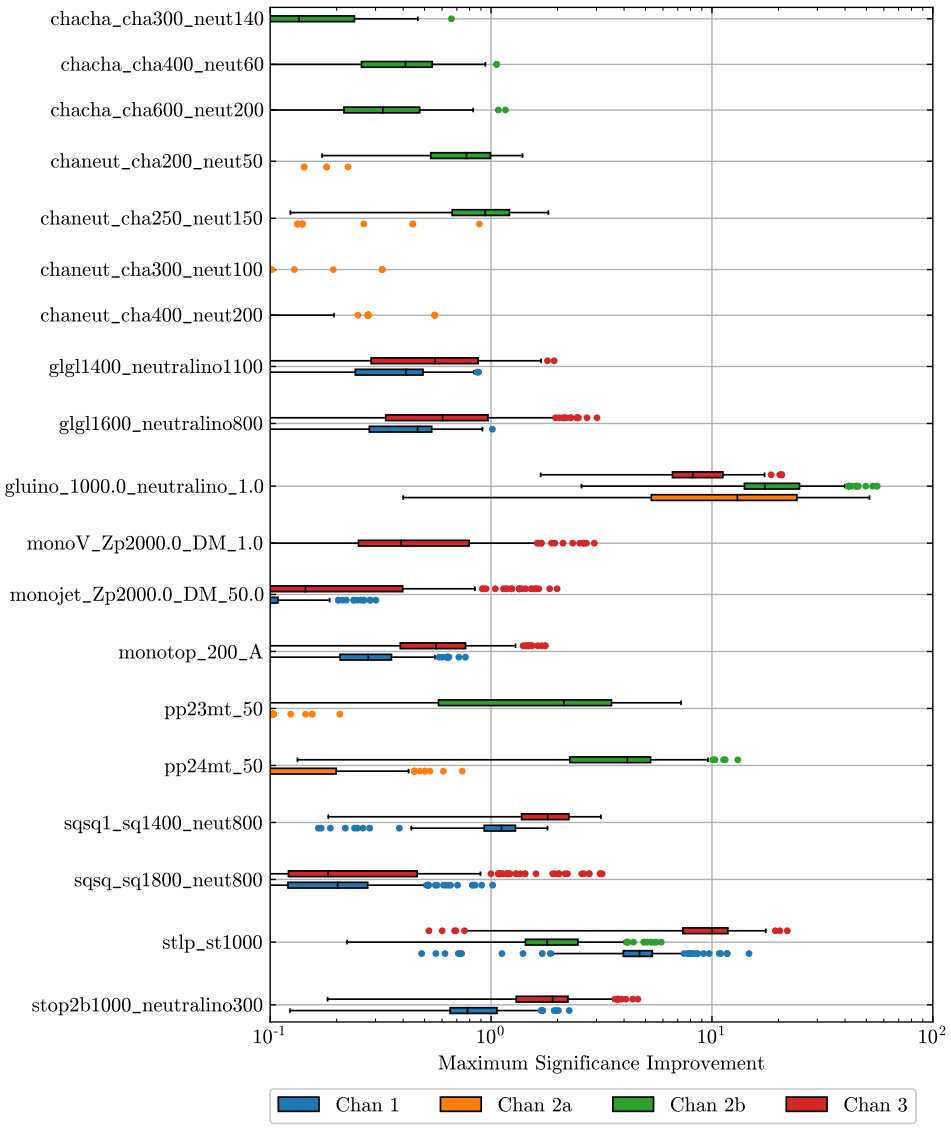


FIGURE 6.8: Box plots for each of the physics signals in the hackathon dataset. These summarize the span of results for the many anomaly detection models trained on background only samples. The SI is defined as $\epsilon_S/\sqrt{\epsilon_B}$. The maximum significance improvement over the three working points ($\epsilon_B = 10^{-2}$, 10^{-3}, and 10^{-4}) are used as the metric for each technique.

performs very well, with high maxima, minima, and medians. When compared with all algorithms submitted in the Dark Machines anomaly score challenge mine scored amongst the highest maximum TI, though they had rather small median TI values.

To determine which algorithm is truly the best with this dataset a cut was placed at 1.0 on the median TI axis, and the remaining algorithms are displayed. This cut is chosen as algorithms with a high median total improvement are expected to perform better on unknown signals. Figure 6.11 shows the 9 best algorithms based on this dataset. Notice that every algorithm, save one, uses a 20 dimensional latent space, and all have

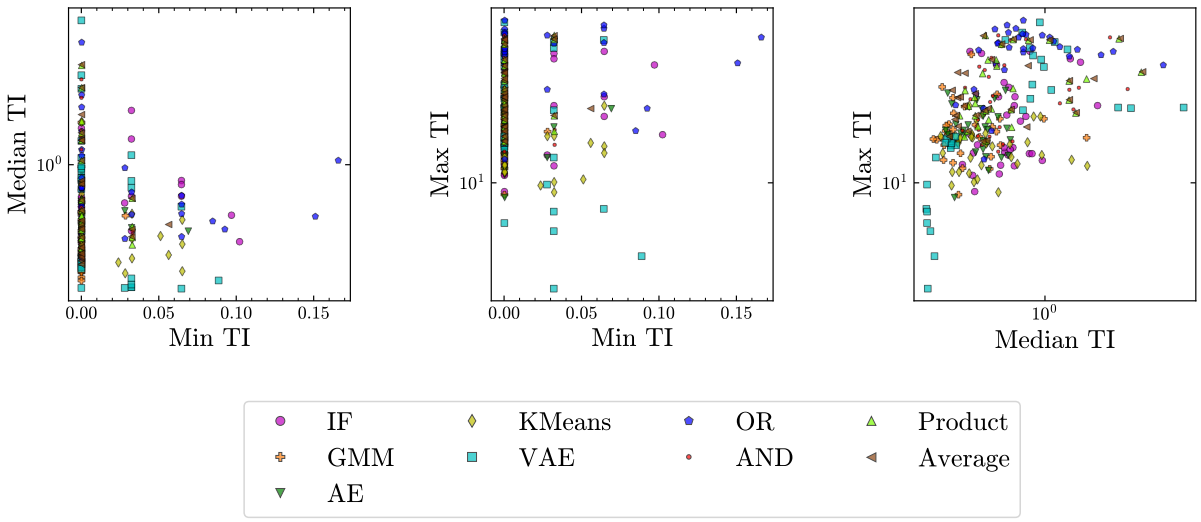


FIGURE 6.9: The minimum, median, and maximum best total improvements for each technique across the physics models. The TI is defined as the maximum signal improvement for a physics model across all signal regions.

a β value of 0.001. The batch size has more variation but generally a larger batch size seems to lead to a higher median total improvement. A 20 dimensional latent space for a 100 dimensional input space makes sense here, as it is large enough to retain important information, but not so large as to cause the algorithms trained within the latent space to suffer from issues with high dimensionality. Using this graph four outstanding algorithms can be identified, each of which have their pros and cons. First, displayed as a yellow pentagon, `VAE_20_0.001_10000`. This algorithm has the highest median TI of the bunch but also has the lowest maximum TI. `Average_20_0.001_1000` and `Product_20_0.001_1000` both have the highest maximum TI of the bunch but the lowest median TI. Finally `OR_20_0.001_10000` sits inbetween these extremes with a high median and maximum total improvement. The results seen here reflect what was observed looking at the various figures of merit defined earlier. The best algorithms tend to be the VAE on its own, and the Average, Product, and OR combinations.

The main Dark Machines anomaly score challenge paper goes into detail on a hidden dataset, testing the performance of the best algorithms on this secret dataset. Only algorithms which had a median TI ≥ 2 were accepted for testing using this hidden dataset as they were assumed to be the most generalisable to other BSM physics scenarios. The algorithm of mine with the highest median TI is `VAE_20_0.001_10000`, which has a value slightly below 2.

The algorithms that scored the best in the main paper are both detailed in Ref. [86]. The first of which is a spline autoregressive flow model which is a probabilistic model that applies a series of transformations to a simple prior distribution. This model can

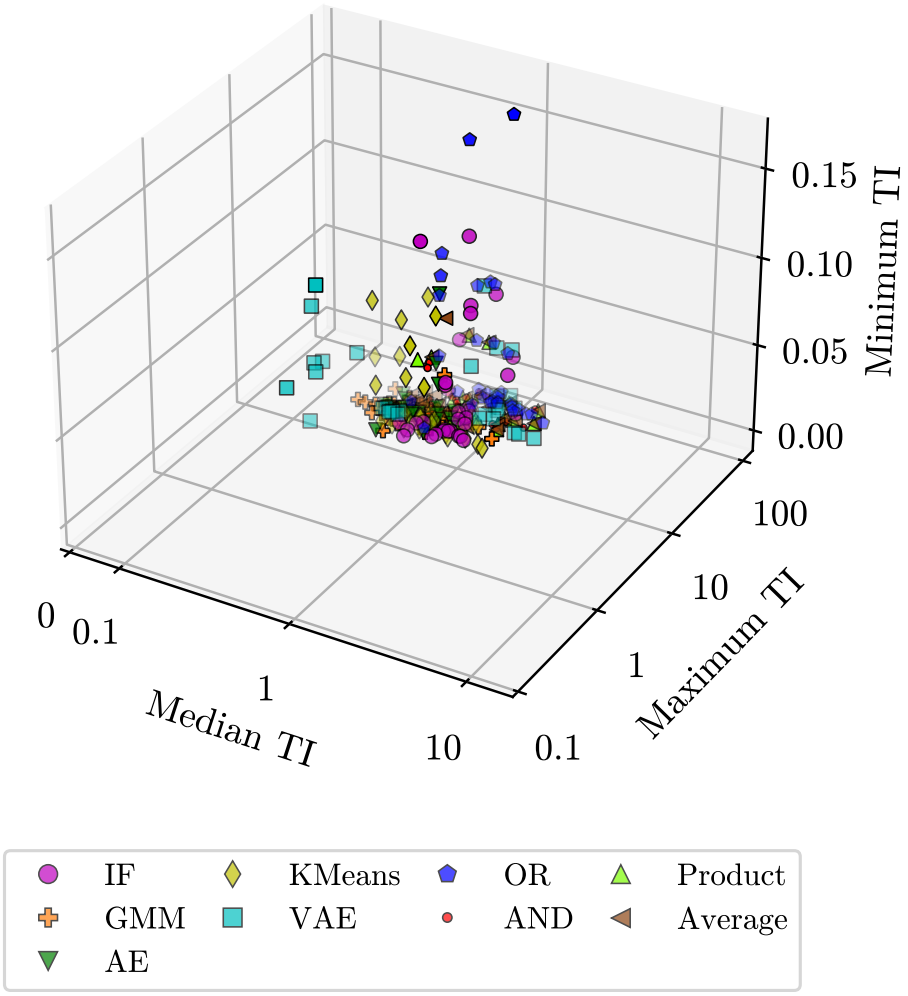


FIGURE 6.10: The minimum, median, and maximum best total improvements for each technique across the physics models.

then be evaluated to obtain the likelihood of a given event. The second high performing algorithm is a combination of the aforementioned spline autoregressive flow model and a deep support vector data description (SVDD) model. A deep SVDD model is a neural network which transforms an input to a vector of constant numbers. These two algorithms, when combined using the various combination methods outlined in Section 5.3, yielded consistently high median total improvements. Many of the highest performing neural network algorithms detailed in the main paper use a constant target vector in the loss function, performing no reconstruction and only compressing the input to a target constant. This is a major difference in the behaviour of my algorithms compared to the highest performing algorithms.

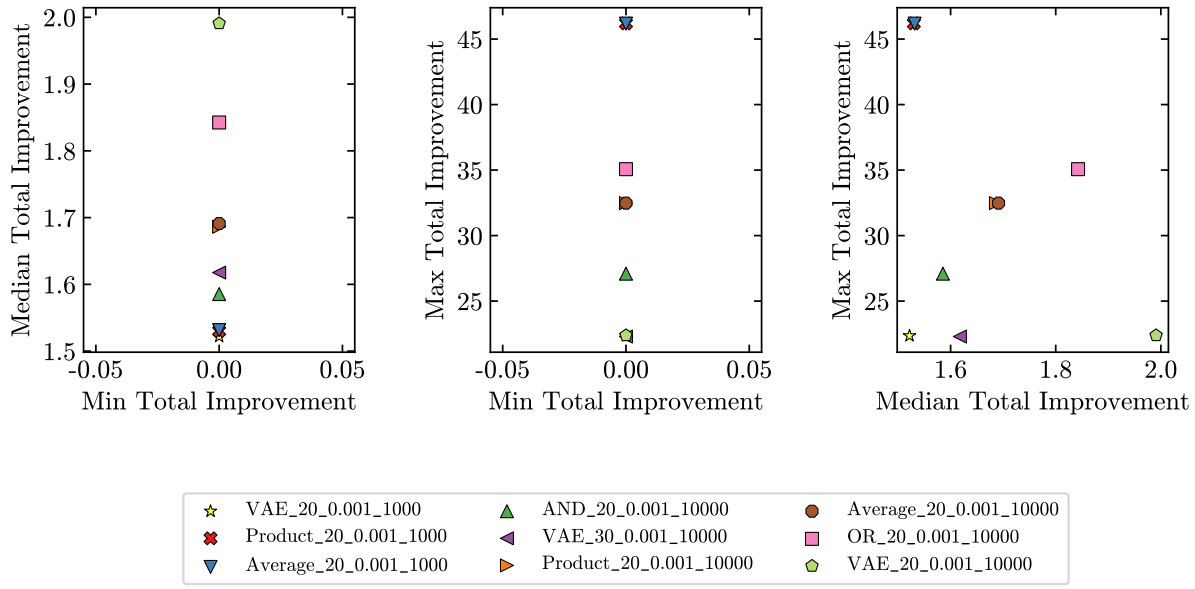


FIGURE 6.11: The minimum, median, and maximum best total improvements for the best techniques across the physics models. The TI is defined as the maximum signal improvement for a physics model across all signal regions.

6.4 Conclusion

In this chapter, I described a handful of benchmark datasets for studies of new physics detection at the LHC. The datasets are divided into 4 channels designed to target interesting regions of the parameter space. These datasets are used not only in order to compare a variety of machine learning methods, but also to more deeply understand the strengths and weaknesses of my own anomaly detection techniques. I defined several metrics to assess the performance of these algorithms, the AUC, and the signal efficiencies at background efficiencies of 10^{-2}, 10^{-3}, and 10^{-4}. In addition to these metrics, I used the significance improvement, defined as $\frac{\epsilon_S}{\sqrt{\epsilon_B}}$, which is a measure of how much a given anomaly score algorithm is able to improve the discovery potential of a signal. The results of all algorithms submitted to the anomaly score challenge can be found in the official paper [67]. In the results section of this chapter I focused on the performance of my own algorithms and came to the conclusion that a VAE with a latent space of 20, a β term of 0.001, and a batch size of 10000 gave the best median total improvement. OR, Average (with a batch size of 1000), and Product combinations of algorithms trained on latent space representations of events generated by this same architecture were also among the highest scoring. While these combination techniques had lower median total improvement scores, they had higher maximum total improvement scores.

7 Improving Optimisation Through Dimensional Reduction

In the previous two chapters, I have explored anomaly detection as a way of performing searches in a model agnostic fashion. While these techniques are certainly promising, the discriminatory power is low and so it is not likely they will discover new physics on their own. In this chapter I investigate supervised analyses, and explore a novel way to broaden the range of new physics being considered when performing a supervised analysis.

Extensions to the SM come with a number of free parameters, many of which are relatively unconstrained even after considering the existing LHC data from Run 2. This means that any choice of BSM model introduces a large number of parameters to be constrained. Searches for SUSY are typically optimised on simplified models, where a sparticle pair is produced and decays with an assumed decay process. Analyses are then tuned by simulating benchmark signal models from within specific planes of parameter values, fixing the remaining parameters. For example it's common to use the plane of two sparticle masses, assuming fixed branching ratios and fixed masses for other sparticles within the simplified model. Many planes are explored, fixing the other parameters at specific values, with the assumption being that after exploring enough of these planes, most of the viable models will have been covered. In reality each of these planes represents a vanishingly thin slice of the total high dimensional parameter space. In this chapter, I explore dimensional reduction of the parameters of the electroweakino sector of the MSSM to 2-D. By creating an invertible map from the original model parameters to a 2-D plane, one can easily identify benchmark points and regions of this low dimensional parameter space on which to optimise. This 2-D plane captures the full phenomenology of the original parameter space rather than being some slice through the high dimensional space which misses the bulk of interesting models.

7.1 The Electroweakino Sector of the MSSM

The electroweakino sector of the MSSM, detailed in Section 2.3.1, is a parameter space of great interest for BSM searches. The masses of the neutralinos and charginos are expected to be fairly light, and so it is possible that they are within reach at modern collider experiments. However, recent searches have turned up nothing, and the range

of excluded masses is rapidly growing. Recall from Section 2.3.2 that in the MSSM, the superpartners of the electroweak gauge bosons and Higgs bosons mix to form the electroweakinos, which consist of four Majorana fermions and two Dirac fermions. These are referred to as the neutralinos, denoted $\widetilde{\chi}_i^0$, for $i = 1, 2, 3, 4$, and the charginos, denoted $\widetilde{\chi}_j^{\pm}$, for $j = 1, 2$. The mass matrices that mix these states contain four parameters, denoted M_1, M_2, μ, and $\tan\beta$. Recall that the electroweak Lagrangian density includes the terms:

$$\mathcal{L}_{\text{EWino}} = -\frac{1}{2}(\psi^0)^T M_N \psi^0 - \frac{1}{2}(\psi^{\pm})^T M_C \psi^{\pm} + c.c. \tag{7.1}$$

where ψ^0 and $\psi^{\pm}$ are the neutral and charged Higgsinos, winos and binos defined as

$$\psi^0 = \left[\widetilde{B}, \widetilde{W}^0, \widetilde{H}_d^0, \widetilde{H}_u^0\right], \tag{7.2}$$

$$\psi^{\pm} = \left[\widetilde{W}^+, \widetilde{H}_u^+, \widetilde{W}^-, \widetilde{H}_d^-\right]. \tag{7.3}$$

The neutralino and chargino mass matrices are denoted as M_N and M_C. These can be written as

$$M_N = \begin{bmatrix} M_1 & 0 & -\frac{1}{2}g_Y v\cos\beta & \frac{1}{2}g_Y v\sin\beta \\ 0 & M_2 & \frac{1}{2}g_w v\cos\beta & -\frac{1}{2}g_w v\sin\beta \\ -\frac{1}{2}g_Y v\cos\beta & \frac{1}{2}g_w v\cos\beta & 0 & -\mu \\ \frac{1}{2}g_Y v\sin\beta & -\frac{1}{2}g_w v\sin\beta & -\mu & 0 \end{bmatrix}, \tag{7.4}$$

$$M_C = \begin{bmatrix} 0 & 0 & M_2 & \frac{g_w v\cos\beta}{\sqrt{2}} \\ 0 & 0 & \frac{g_w v\sin\beta}{\sqrt{2}} & \mu \\ M_2 & \frac{g_w v\sin\beta}{\sqrt{2}} & 0 & 0 \\ \frac{g_w v\cos\beta}{\sqrt{2}} & \mu & 0 & 0 \end{bmatrix}, \tag{7.5}$$

where g_w and g_Y are defined as the $SU(2)$ and $U(1)_Y$ gauge couplings, and v as the electroweak VEV. These values are fixed from experimental data. Hence the free parameters governing the masses of the electroweakinos are M_1, M_2, μ, and $\tan\beta$, where $\tan\beta$ is defined as the ratio of the aforementioned $\sin\beta$ and $\cos\beta$ terms. As you can see, these parameters govern the mixing of the Higgsino, wino and binos which in turn define the behaviours of the electroweakinos.

The dataset used in this chapter comes from a global fit of the EWMSSM performed by the GAMBIT collaboration [87]. It consists of many MSSM models with differing values of M_1, M_2, μ, and $\tan\beta$ along with an associated log likelihood based on the searches for electroweakinos at the LEP and LHC colliders, plus constraints on the invisible widths of the Z and SM-like Higgs boson. The full scan of the parameter space and implemented searches are detailed in Ref. [88]. This global fit contains up-to-date results for 2017, which means that all results are for 36 fb^{-1} of integrated luminosity at a centre of mass energy

of 13 TeV. This chapter is therefore a historical test case for this new approach rather than a current example. The global fit assumed that all other sparticles of the MSSM are heavy and decoupled. Specifically, all mass parameters except for those relevant to the electroweakino sector are set to 3 TeV, the pseudo-scalar Higgs mass and gluino mass parameters are set to 5 TeV, and all trilinear couplings are set to zero. In order to ensure that the models used in this analysis are suitably unexcluded, only points within the 3σ contour of these global fit results are selected. Throughout this chapter all models within this 3σ contour are treated with equal weight. Now that the dataset is understood, let us consider the dimensional reduction tool. Importantly, the algorithm must be invertible. This means it must be possible to take a given point in the dimensionally reduced space and return to the original parameter space. For this, a variational autoencoder is a perfect fit.

7.1.1 Preparation of the Dataset

In preparation for training the VAE, the dataset must be prepared in a few important ways. First, each model parameter is scaled between 0 and 1 such that, for $x \in [M_1, M_2, \mu, \tan\beta]$,

$$\xi_x \equiv \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}}, \tag{7.6}$$

where x_{min} and x_{max} are the maximum and minimum values of x across the whole dataset. These minima and maxima are detailed in Table 7.1.

Parameter	Minimum	Maximum
M_1 [GeV]	-2000	2000
M_2 [GeV]	0	2000
μ [GeV]	-2000	2000
$\tan\beta$	1	70

TABLE 7.1: Maximum and minimum values of each parameter before scaling.

When training a neural network it is important to sample the entire space. Ideally the distribution of each variable would be totally flat, however this is almost never the case. To this end variables which are over-represented in certain regions of the parameter space must be "unskewed". In this dataset, the variable M_2 is skewed negative, meaning there are more points with low M_2 than high M_2. In order to reduce the skew of a variable, it is common take the square root, natural log, or inverse of the value. In preparation for training, all variables are normalised between 0 and 1, and so to simply avoid any numerical errors, the square root of ξ_{M_2} is taken. This provides a significant improvement

in the quality of the VAE over simply leaving the value skewed. Figure 7.1 displays a histogram of ξ_{M_2} values before and after taking the square root.

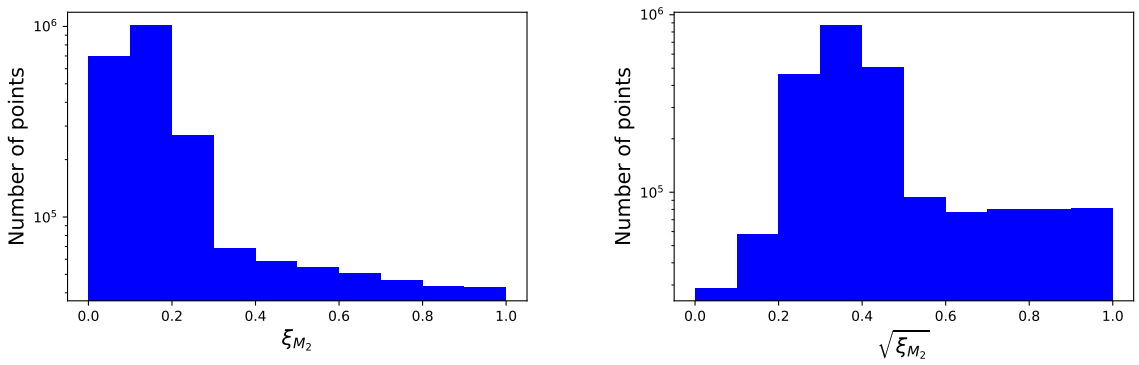


FIGURE 7.1: Values of ξ_{M_2} displayed as a histogram before (left) and after (right) taking the square root in order to unskew the distribution. Notice how the histogram on the right is more flat than that on the left.

7.2 VAE Training on GAMBIT Global Fit Results

In the process of developing the VAE used in this chapter, a number of hyperparameters are tested and the best are used for the remainder of the chapter. These include:

- The number of hidden layers in the encoder/decoder ranging from 3-5 with numbers of nodes between 4 and 256.
- The activation function for each of the nodes, where the hyperbolic tangent (tanh), rectified linear unit (ReLU), and the sigmoid linear unit are tested.
- Various loss functions: mean-squared error, absolute error, and the β-VAE loss function with $\beta = [10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}]$.

The hyperparameters chosen are those that give both the lowest mean squared error and the highest Pearson correlation coefficients between the input and output of the VAE. The architecture of this VAE is defined as: a 4 dimensional input layer, an encoder with 5 layers containing 100, 100, 50, 25, and 10 nodes, a 2 dimensional latent space, a decoder with 5 layers containing 10, 25, 50, 100, and 100 nodes, and finally a 4 dimensional output layer. Each hidden layer uses a a tanh activation function, while the input and output layers use linear activation functions. The loss function comparing a single point x to its reconstructed counterpart y is defined as

$$\mathcal{L} = (1-\beta)(x-y)^2 + \beta \sum_{i}^{d} \mathrm{KL}(\mathcal{N}(\hat{\mu}_\mathrm{i}, \hat{\sigma}_\mathrm{i}), \mathcal{N}(0,1)), \tag{7.7}$$

where μ_i and σ_i are the mean and standard deviations of the ith Gaussian within the d dimensional latent space. The β term is set to 10^{-6} and balances the relative importance of the mean squared error term and the KL divergence term. For these purposes, a small value of β gives the best reconstruction quality, while still ensuring regularisation of the latent space. The dataset is split into a training and testing set with an 80/20 ratio. The network is then trained with an 80/20 training/validation split. Training runs over 10 iterations of 10,000 epochs with the Adam optimiser, beginning with a learning rate of 10^{-3}, and halving each iteration. An iteration ends when 10,000 epochs have passed, or when the performance on the validation set has not improved in 50 epochs. The network is saved only when the loss of the network evaluated on the validation set decreases in order to avoid overfitting.

In order to assess the performance of the variational autoencoder, and to ensure no overfitting has occurred, the testing set is utilised. The input value of each parameter can be compared with its reconstructed value in order to assess the quality of reconstruction. Figure 7.2 plots a heatmap of each input parameter with its reconstructed value, where the z axis is $\log_{10}(N_{points})$. A heatmap is chosen here in order to more accurately represent the quality of reconstruction with the number of points present. Perfect reconstruction will yield a perfectly straight line $y = x$. The aforementioned figure shows mostly straight lines for each variable, apart from M_1 which appears more staggered. However one can more robustly assess the reconstruction quality by examining various metrics. The Pearson correlation coefficient between each input parameter x and its reconstructed value $\hat{x}$ indicates the degree of correlation. It can be defined as

$$\rho = \frac{\sum_i^N (x_i - \mu_x)(\hat{x} - \mu_{\hat{x}})}{\sqrt{\sum_i^N (x_i - \mu_x)^2}\sqrt{\sum_i^N (\hat{x} - \mu_{\hat{x}})^2}}, \tag{7.8}$$

for N data points where μ_x is the mean of the parameter x. A value of ± 1 indicates perfect positive/negative correlation, while a value of 0 indicates no correlation whatsoever. The mean squared error, defined as

$$\epsilon_{msq} = \frac{1}{N}\sum_i^N (x_i - \hat{x}_i)^2, \tag{7.9}$$

for N data points where x is the input parameter and $\hat{x}$ is its reconstructed value also gives an indication of the reconstruction quality, with smaller values indicating better correlation. Table 7.2 displays the Pearson correlation coefficients and mean squared error for each parameter. It is clear to see that M_1, despite appearing to have poor reconstruction quality in Figure 7.2, in fact has a high Pearson correlation coefficient and low mean squared error implying that it is acceptable for use. It is possible that an unskewing process similar to what was done with ξ_{M_2} would improve the reconstruction

of ξ_{M_1}.

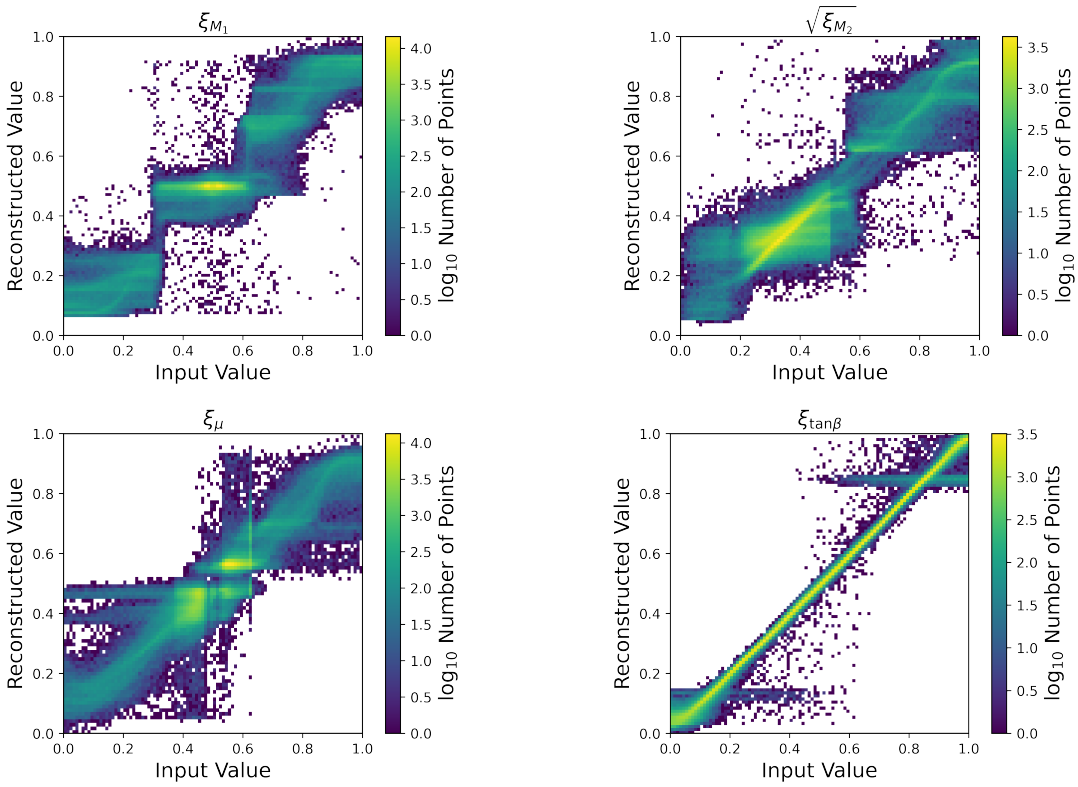


FIGURE 7.2: Input vs output from VAE for each parameter ξ_{M_1}, $\sqrt{\xi_{M_2}}$, ξ_{μ}, and $\xi_{\tan\beta}$. A perfect straight line $y = x$ is desirable.

Parameter	Pearson Correlation Coefficient	Mean Squared Error
ξ_{M_1}	0.947	4.36×10^{-3}
$\sqrt{\xi_{M_2}}$	0.936	4.37×10^{-3}
ξ_{μ}	0.927	3.64×10^{-3}
$\xi_{\tan\beta}$	0.998	3.49×10^{-4}

TABLE 7.2: Pearson correlation coefficients between the input and reconstructed parameter. A value of ± 1 implies perfect positive/negative correlation, while a value of 0 implies no correlation whatsoever. Note that the correlation coefficient for each parameter is > 0.9, and so reconstruction quality is generally high for all parameters.

7.3 Visualisation of the Latent Space

After the network is trained, the 4-dimensional EWMSSM model parameters can be passed through the VAE, and mapped to the $\theta_1 - \theta_2$ plane, where (θ_1, θ_2) are the latent space variables of the VAE. Figure 7.3 displays colour maps of each of the EWMSSM

input variables within the $\theta_1 - \theta_2$ plane. This lets one view the structure of the latent space and observe relationships between the parameters. One can see that each input variable has continuous regions of similar colour, confirming the regularisation of the latent space. Points that were close together in the original space remain close together in the latent space, indicating that the latent space is preserving the nature of each model. Note that in order to aid in the visualisation of these variables, approximately 3% of the most poorly reconstructed models are discarded from each of the following scatter plots. The reconstruction metric is defined as

$$\alpha = \sum_{i}^{n} (x_i - \hat{x}_i)^2, \tag{7.10}$$

where n refers to each input parameter. Models with $\alpha > 0.05$ are discarded purely for the purpose of visualisation of the latent space.

The latent space has some interesting features to note. Firstly, the entire $\theta_1 - \theta_2$ plane is not covered by the testing set in the original 4-D space. This could be solved by using a larger dataset with a wider selection of models to train and test the VAE. Notice that M_1 and M_2 appear to be split into parallel bands of similar values, while μ tends to have smaller values on the right and larger values on the left. $\tan\beta$ is notable for its symmetry of values across the space.

Other quantities can be shown on the z-axis to gain insight into the structure of the latent space. Figure 7.4 shows the models within the latent space with the neutralino and chargino particle masses on the z-axis. The highlighted benchmark points are used in Section 7.4. Due to regularisation of the latent space from the KL term, models which the VAE has deemed to be similar occupy similar spaces within the VAE. This allows one to observe areas of the parameter space which may be accessible to analyses. Areas with very high particle masses are likely to be outside of the reach of current detector experiments.

Searches in the electroweak sector of the MSSM regularly examine pair production of the lightest chargino $\widetilde{\chi}_1^{\pm}$ and the second neutralino $\widetilde{\chi}_2^0$ with subsequent decay to a pair of LSPs $\widetilde{\chi}_1^0$ and leptons via intermittent W and Z bosons, leading to a final state containing 3 leptons. The cross sections for each process, calculated in `PROSPINO` [89], can be used to identify regions within the latent space with high amounts of a given process. Figure 7.5 (left) displays the latent space with the number of $\widetilde{\chi}_1^{\pm}$ $\widetilde{\chi}_2^0$ events at 36 fb^{-1} as a colour map. It is clear that there are a number of pockets containing high numbers of events, however as stated previously, a 3-lepton final state is of particular interest in this case. By calculating the branching ratios of $\widetilde{\chi}_1^{\pm} \to W^{\pm}\widetilde{\chi}_1^0$ and $\widetilde{\chi}_2^0 \to Z\widetilde{\chi}_1^0$ in `SUSYHIT` [90], an upper bound on the production of 3-lepton events can be estimated for each MSSM model at 36fb^{-1}. Note that the number of 3-lepton events observed at the LHC will certainly

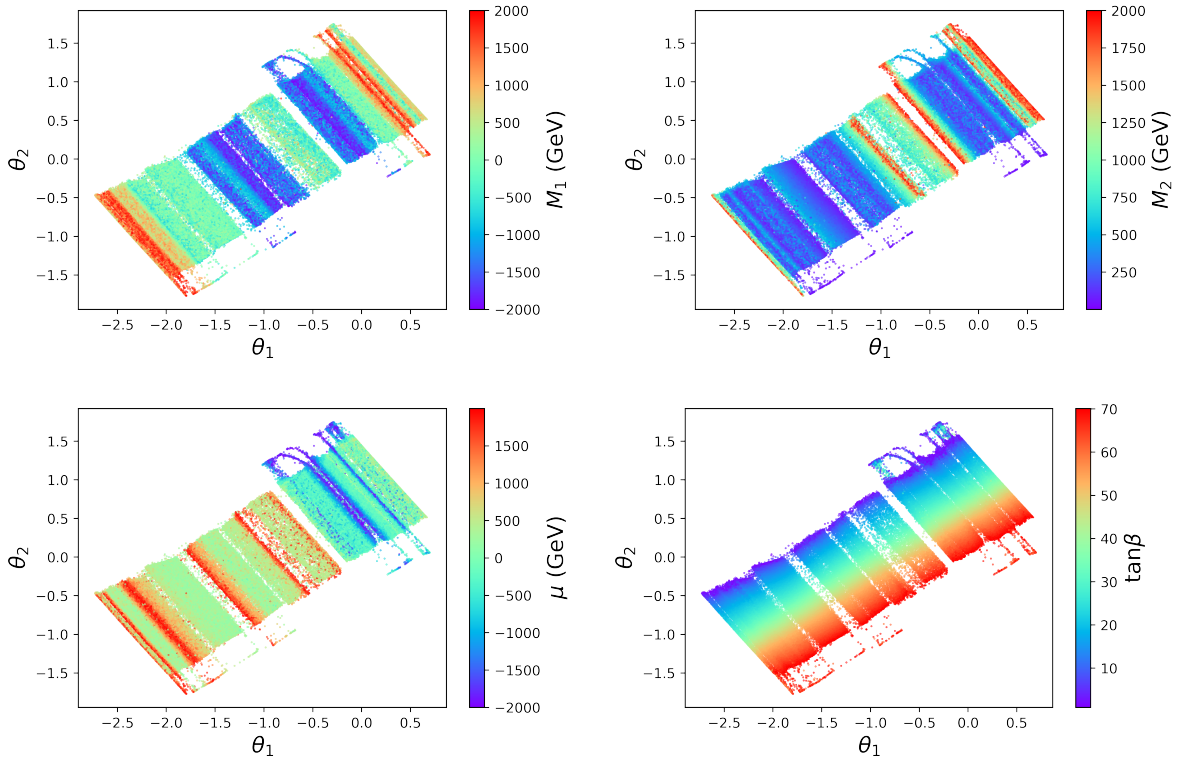


FIGURE 7.3: Values of the transformed EWMSSM input variables M_1 (top left), ξ_{M_2} (top right), μ (bottom left) and $\tan\beta$ (bottom right) in the $\theta_1 - \theta_2$ plane.

be lower, as these values do not account for detector effects. Models with small mass splittings $m_{\widetilde{\chi}_1^\pm} - m_{\widetilde{\chi}_1^0} < 15$ GeV or $m_{\widetilde{\chi}_2^0} - m_{\widetilde{\chi}_1^0} < 15$ GeV have their branching ratios set to 0. This is done so that the highlighted models contain leptons with high enough p_T to be registered by the detector. Figure 7.5 (right) displays the latent space with the upper bound on the number of 3-lepton events at 36 fb^{-1} as a colour map. Notice that the points yielding the most 3-lepton events are generally clustered together, showing that the VAE is correctly clustering models with similar behaviours.

Since searches for electroweakino models typically target these simplified models, an interesting prospect is to examine unexcluded models with behaviour different to that of a simplified model. The most straightforward way to examine this is to calculate the proportion of the production cross section that does not arise from processes typically examined in simplified model scenarios. These "simplified" processes include $\widetilde{\chi}_1^0\ \widetilde{\chi}_1^\pm$, $\widetilde{\chi}_1^\pm\ \widetilde{\chi}_1^\mp$, and $\widetilde{\chi}_2^0\ \widetilde{\chi}_1^\pm$ pair production processes. Figure 7.6 shows the number of non-simplified events, as well as the relative proportion of non-simplified events defined as

$$\frac{\sum_i \sigma_i - (\sigma_{\widetilde{\chi}_1^0\widetilde{\chi}_1^\pm} + \sigma_{\widetilde{\chi}_1^\pm\widetilde{\chi}_1^\mp} + \sigma_{\widetilde{\chi}_2^0\widetilde{\chi}_1^\pm} + \sigma_{\widetilde{\chi}_1^0\widetilde{\chi}_1^0})}{\sum_i \sigma_i}, \tag{7.11}$$

where i sums over all production cross sections. $\sigma_{\widetilde{\chi}_1^0\widetilde{\chi}_1^\pm}$ is removed as monojet searches

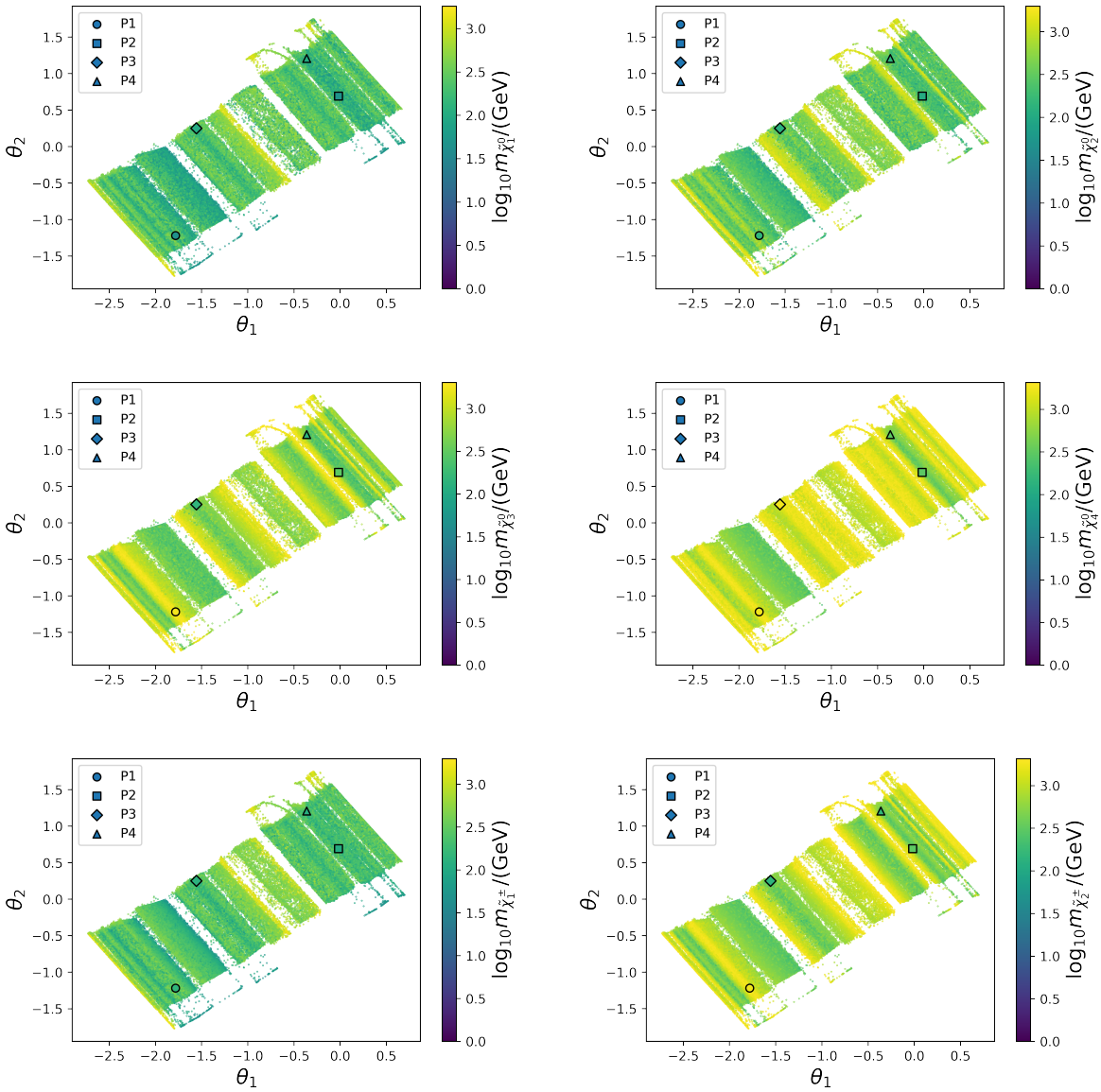


FIGURE 7.4: The electroweakino masses in the $\theta_1 - \theta_2$ plane. The positions of the on/off shell models and the non-simplified model are highlighted, where their colours in each plane correspond to the mass of that models electroweakino.

are weakly constraining on SUSY models. Choosing a benchmark point based on this information will prioritise SUSY scenarios that are totally unlike those that are typically examined in simplified model scenarios.

7.4 Optimisation of Analyses in the Latent Space

In order to demonstrate the approach of optimising a search strategy on parameters other than the fundamental SUSY parameters, I construct three separate analyses for four

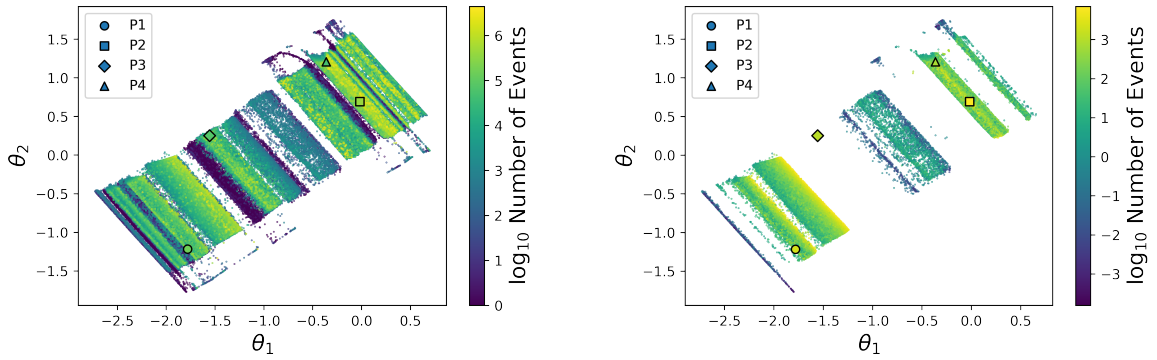


FIGURE 7.5: Left: latent-space representations of points in the $\theta_1 - \theta_2$ plane with the number of $\widetilde{\chi}_1^{\pm}\widetilde{\chi}_2^0$ events at 36 fb^{-1} as a colour map. Right: latent-space representations of points with the upper bound on the number of 3-lepton events from $\widetilde{\chi}_1^{\pm}\widetilde{\chi}_2^0$ production at 36 fb^{-1} as a colour map.

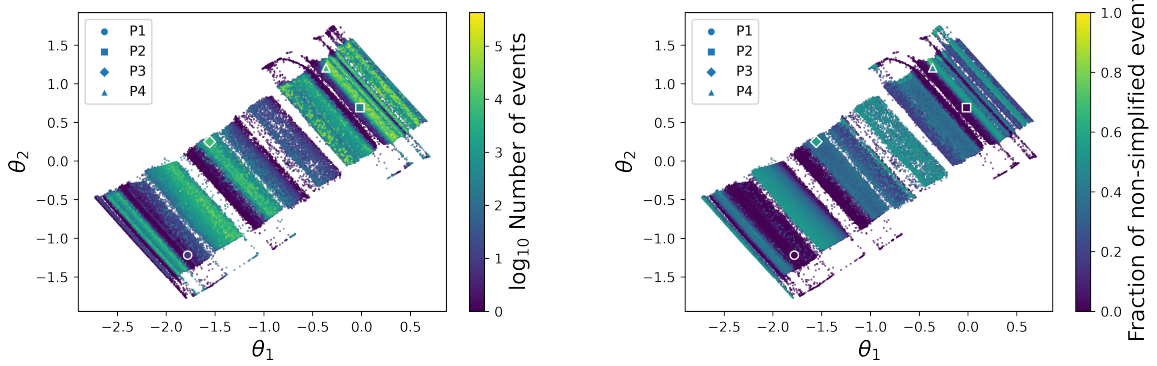


FIGURE 7.6: The number (left) and proportion (right) of non-simplified events at 36 fb^{-1} in the $\theta_1 - \theta_2$ plane.

separate unexcluded benchmark points[1]. These benchmark points are chosen based on the visualisations done in Section 7.3 and are shown on Figures 7.4 to 7.6. The first two search strategies target direct pair production of the lightest chargino $\widetilde{\chi}_1^{\pm}$ and the second neutralino $\widetilde{\chi}_2^0$ decaying to a pair of LSPs $\widetilde{\chi}_1^0$ and leptons via an intermittent W and Z boson. The two models producing an abundance of 3-lepton events via on and off shell WZ mediated decays (Figure 7.7) can be summarised as:

- **P1:** On-shell WZ mediated decay. $\widetilde{\chi}_1^{\pm} \to W^{\pm}\widetilde{\chi}_1^0$ and $\widetilde{\chi}_2^0 \to Z\widetilde{\chi}_1^0$ both with 100% branching ratio. In this case $\Delta m(\widetilde{\chi}_1^{\pm}, \widetilde{\chi}_1^0) \geq m_W$ and $\Delta m(\widetilde{\chi}_2^0, \widetilde{\chi}_1^0) \geq m_Z$.
- **P2:** Off-shell WZ mediated decay. Same as above, but with $\Delta m(\widetilde{\chi}_1^{\pm}, \widetilde{\chi}_1^0) < m_W$ and $\Delta m(\widetilde{\chi}_2^0, \widetilde{\chi}_1^0) < m_Z$.

[1]These benchmark points are excluded by some individual analyses, as the selection is made based on the *overall* likelihood generated by GAMBIT. This will be fixed in future work.

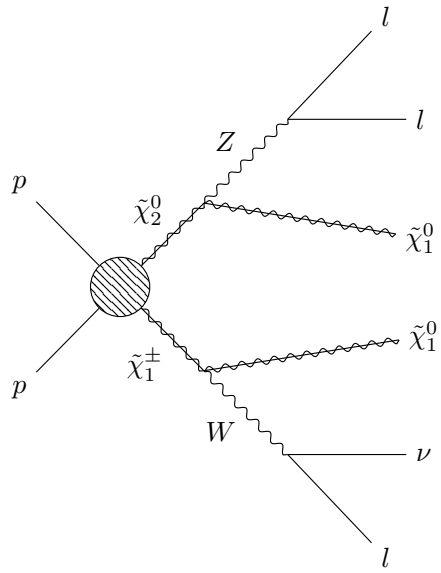


FIGURE 7.7: The primary $\tilde{\chi}_2^0$ $\tilde{\chi}_1^\pm$ decay mode of interest, yielding a 3-lepton final state. Note that 3-lepton final states can be reached via a $\tilde{\chi}_1^\pm$ decaying into a $\tilde{\chi}_2^0$ plus a W boson or vice versa given the correct mass differences.

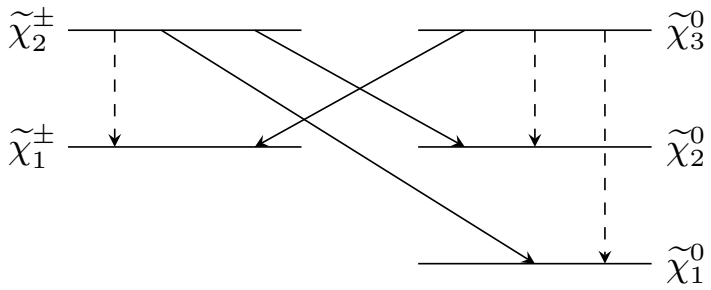


FIGURE 7.8: $\tilde{\chi}_3^0$ $\tilde{\chi}_2^\pm$ decays into lower mass charginos/neutralinos. A solid arrow indicates a W boson, while a dashed arrow indicates a Z boson.

The third analysis targets two non-simplified models, **P3** and **P4**, and specifies a 2-lepton final state with high missing energy. **P3** features primarily $\tilde{\chi}_3^0$ $\tilde{\chi}_2^\pm$, and $\tilde{\chi}_1^0$ $\tilde{\chi}_2^0$ pair production. The process $\tilde{\chi}_3^0$ $\tilde{\chi}_2^\pm$ has many decay paths yielding 2-lepton final states, the predominant one involving $\tilde{\chi}_3^0 \to Z\tilde{\chi}_1^0$ where the Z boson decays to 2 leptons and the $\tilde{\chi}_2^\pm$ decays hadronically. Figure 7.8 shows all of the ways these particles can decay via W/Z bosons, where a solid line indicates a W boson, and a dashed line indicates a Z boson. The production of 2-lepton final states from $\tilde{\chi}_1^0$ $\tilde{\chi}_2^0$ pair production can only come from $\tilde{\chi}_2^0 \to Z\tilde{\chi}_1^0$ where the Z decays to 2 leptons. **P4** features primarily $\tilde{\chi}_3^0$ $\tilde{\chi}_2^\pm$, $\tilde{\chi}_3^0$ $\tilde{\chi}_1^\pm$, and $\tilde{\chi}_1^0$ $\tilde{\chi}_2^0$ production. The only significant difference here is the inclusion of the $\tilde{\chi}_3^0$ $\tilde{\chi}_1^\pm$ process, which can decay to two leptons in a number of ways via off-shell W/Z bosons, similar to the decay of $\tilde{\chi}_2^\pm$, $\tilde{\chi}_3^0$, displayed in Figure 7.8.

Benchmark point	M_1 (GeV)	M_2 (GeV)	μ (GeV)	$\tan\beta$	σ (pb)	N_{tot} ($N_{36\ \text{fb}^{-1}}$)
P1	-46	130	1752	61	$\widetilde{\chi}_1^{\pm}\widetilde{\chi}_2^0$: 10.7	1696957 (384654)
P2	48	102	-275	40	$\widetilde{\chi}_1^{\pm}\widetilde{\chi}_2^0$: 27.0	6980683 (972988)
P3	-1748	97	-71	1.1	$\widetilde{\chi}_2^{\pm}\widetilde{\chi}_3^0$: 10.1 $\widetilde{\chi}_1^0\widetilde{\chi}_2^0$: 6.52	997283 (363692) 497914 (234587)
P4	308	94	-80	1.6	$\widetilde{\chi}_2^{\pm}\widetilde{\chi}_3^0$: 8.45 $\widetilde{\chi}_1^0\widetilde{\chi}_2^0$: 4.10 $\widetilde{\chi}_1^{\pm}\widetilde{\chi}_3^0$: 2.17	498525 (304168) 497861 (147627) 497208 (78136)

TABLE 7.3: Summary of signal processes which are optimised on in this analysis. Details include the values of M_1, M_2, μ, and $\tan\beta$, the production cross section at $\sqrt{s} = 13$ TeV, the number of Monte Carlo events that were generated, and the number of events expected in 36 fb^{-1} of LHC data.

7.4.1 Generation of Events

In this analysis I use the same background dataset, first detailed in Chapter 5 and described in detail in Ref [53]. To recap, these events were generated at leading order for a 13 TeV LHC centre of mass energy with two extra jets. Events for each signal model were generated in `Madgraph` [34] with generator cuts set as follows:

- Minimal transverse momentum of the jets $p_T^j > 20$ GeV, and their rapidity restricted to be $|\eta^j| < 2.8$;
- Minimal transverse momentum of photons $p_T^\gamma > 20$ GeV and generated up to a maximum rapidity of $|\eta^\gamma| < 2.37$;
- Minimal lepton (electron e and muon μ) transverse momentum $p_T^l > 15$ GeV and with the rapidity window $|\eta^l| < 2.7$;

`Pythia8.2` was used for showering with up to two additional jets with MLM matching. `Delphes 3` was used for detector simulation with the same modified ATLAS detector card as the background simulation. A summary of each BSM model is provided in Table 7.3, including the total number of events generated, and the number of events at 36 fb^{-1}. Each event is weighted such that the sum of weights adds up to the total cross section of the process. Table 7.4 shows the masses of each electroweakino in GeV for each benchmark point.

7.4.2 Definition of Analyses

The cuts placed on the on/off-shell signals **P1** and **P2**, loosely based on Refs. [91, 92], are detailed in Table 7.5. Requirements common to the on/off-shell analyses include a 3-lepton selection, and lepton trigger requirements. Electrons must have $p_T^e > 18$ GeV and $\eta^e < 2.47$, while muons must have $p_T^\mu > 14.7$ and $\eta^\mu < 2.5$. In order to increase the likelihood

Benchmark point	$m_{\widetilde{\chi}_1^0}$	$m_{\widetilde{\chi}_2^0}$	$m_{\widetilde{\chi}_3^0}$	$m_{\widetilde{\chi}_4^0}$	$m_{\widetilde{\chi}_1^\pm}$	$m_{\widetilde{\chi}_2^\pm}$
P1	46.18	142.4	1766	1767	142.5	1769
P2	46.84	104.2	293.3	302.1	104.4	306.9
P3	74.18	104.6	138.6	1738	105.8	138.6
P4	81.06	118.0	135.8	3122	106.0	142.0

TABLE 7.4: Masses of the electroweakinos in GeV for each examined signal processes.

that one pair of leptons originated from a Z boson, all events are required to have one same-flavour opposite-charge-sign (SFOS) lepton pair. The other lepton is assigned to the W boson. The invariant mass, defined as $m_{12} = \sqrt{(E_{T_1} + E_{T_2}) - (p_{T_1} + p_{T_2})}$ for two leptons labelled 1 and 2 with energies E_i and transverse momenta p_{T_i}, is used to determine which lepton pair originated from the Z boson in the event that there are multiple candidates.

The cuts placed on the non-simplified signals **P3** and **P4**, loosely based on [93], are detailed in Table 7.6. The trigger level requirements placed on lepton p_T and η are the same as for the 3-lepton selection, with the only difference being that 2 leptons are required in the final state.

On-Shell Analysis

The on-shell analysis (for **P1**) targets decays with mass splittings near or above the Z mass, $\Delta m \geq m_Z$. First, the two highest momentum leptons are required to have p_T greater than 25 and 20 GeV respectively. Leptons are assigned as originating from either the Z or W boson. Two leptons are assigned to the Z boson by selecting the same flavour opposite charge sign (SFOS) pair, and the remaining lepton is assigned to the W boson. In the event that there are multiple candidates for one of the SFOS leptons, the lepton pair with invariant mass closest to the Z mass is selected to be the SFOS pair. The variable m_T is constructed using the lepton originating from the W boson and the missing transverse energy. m_T has a Jacobian peak in the WZ background which drops off at $m_T \approxeq m_W$ while the signal distribution is more flat. ΔR_{SFOS}, defined as $\Delta R_{12} = \sqrt{(\eta_1 - \eta_2)^2 + (\phi_1 - \phi_2)^2}$ where objects 1 & 2 are same-flavour opposite-charge-sign lepton pairs, is used to restrict the angular separation between the SFOS lepton pair and reduce the ZW background. The total hadronic activity is limited with a cut on H_T, defined as the sum of jet p_T.

Off-Shell Analysis

The off-shell analysis (for **P2**) targets decays with mass splittings less than the Z mass, $\Delta m < m_Z$. Similar to the on-shell analysis, two leptons are assigned as originating from the Z boson by selecting the same-flavour opposite-charge-sign pair. In the event that

Variable	On-shell requirement	Off-shell requirement
n_{lep}	$= 3$	
p_T^e [GeV]	> 18	
η^e	< 2.47	
p_T^μ [GeV]	> 14.7	
η^μ	< 2.5	
n_{SFOS}	$= 1$	
$p_T^{l_1}$, $p_T^{l_2}$ [GeV]	> 25, 20	$-$
m_T [GeV]	> 90	< 110
H_T [Gev]	< 75	$-$
m_{ll}^{min} [GeV]	$-$	> 10
$\lvert m_{3l} - m_z \rvert$ [GeV]	$-$	< 190 ($l_W = e$ only)
$\text{min}\Delta\text{R}_{\text{3l}}$	$-$	$\in [0.2, 1.1]$
ΔR_{SFOS}	> 0.2	$\in [0.2, 1.1]$

TABLE 7.5: Summary of selection criteria for the on- and off-shell W/Z selection. "$-$" indicates no requirement is applied for a given variable in the corresponding region.

there are multiple candidates, the lepton pair with the minimum invariant mass is chosen. In this analysis m_T is required to be small, as the signal is more SM-like than the on-shell signal and predominantly resides in the low m_T region. The variable $\text{min}\Delta\text{R}_{\text{3l}}$ is defined as the minimum ΔR between all lepton pairs and is used to further restrict the angular distribution of the leptons and reduce the ZW background. ΔR_{SFOS} is used in a similar fashion. The variable $|m_{3l} - m_Z|$, defined as the difference between the trilepton mass and the Z boson mass is used when the lepton originating from the W boson is an electron to ensure that the trilepton mass is not too far from the Z mass. This further reduces backgrounds involving Z bosons. Finally, a lower bound is placed on the invariant mass of the SFOS lepton pair m_{ll}^{min} to further reduce the ZW background.

Non-Simplified Analysis

The non-simplified analysis (for **P3** and **P4**) is done in a 2-lepton final state with high missing transverse energy. H_T is required to be low in order to reduce backgrounds with high hadronic activity. The variable m_{T2} is restricted in order to reduce backgrounds involving Z bosons, and finally the angular separation between the two leptons ΔR_{ll} is required to be low in order to reduce the WW background.

7.4.3 Results

With the chosen cuts placed, the number of remaining signal and background events are compared for each analysis. The binomial significance, denoted Z_{bi}, is calculated using

Variable	Non-simplified requirement
n_{lep}	$= 2$
p_T^e [GeV]	> 18
η^e	< 2.47
p_T^μ [GeV]	> 14.7
η^μ	< 2.5
E_T^{miss} [GeV]	> 100
m_{T2} [GeV]	> 240
H_T [Gev]	< 100
ΔR_{ll}	< 0.6

TABLE 7.6: Summary of selection criteria for the non-simplified 2-lepton selection.

the `RooStats` framework within `ROOT 6.24.02` [94]. This value tests between signal-plus-background and background only hypotheses where signal and background events are drawn from a Poisson distribution. A value of $Z_{bi} = 1.64$ indicates a 95% confidence level, which is what is needed in order to safely exclude a signal. This calculation includes both the statistical uncertainty on the number of Monte Carlo events as well as an assumed systematic uncertainty of 15%. Using this metric, all four signal models are able to be excluded at a 95% confidence level. Table 7.7 shows the signal models, the numbers of signal and background events, and the Z_{bi} score.

Process ID	N_{sig} (N_{sig}^{MC})	N_{bkg} (N_{bkg}^{MC})	Z_{bi}
P1	62.47 (281)	111.6 (31)	1.78
P2	34.98 (256)	43.2 (12)	1.67
P3	38.07 (99)	46.8 (13)	2.19
P4	34.66 (96)	46.8 (13)	2.00

TABLE 7.7: Summary of analysis results. N_{sig} and N_{bkg} denote the number of signal and background events at 36 fb^{-1} respectively, while N_{sig}^{MC} and N_{bkg}^{MC} refer to the number of signal and background Monte Carlo events. The significance quoted was calculated using the `RooStats` stats framework within `ROOT 6.24.02`. Using this metric, a Z score of 1.64 corresponds to a 95% confidence level.

Thus it has been demonstrated that by using the latent space to identify models unexcluded by standard analysis techniques, and tweaking existing analyses, it is possible to exclude as of yet unexcluded models. While it may have been possible to exclude **P1** and **P2** using the standard approach, it is highly unlikely that **P3** or **P4** would have been excluded in this way, due to their distinctly non-simplified model like behaviour.

7.5 Conclusion

In this chapter, I have constructed a variational autoencoder designed to compress 4-dimensional EWMSSM model parameters to 2-dimensional latent space representations. Within this 2-D plane, interesting regions not yet excluded by current experiments have been examined. Various colour maps have been applied, including an upper bound on the number of events for a 3-lepton final state yielded from the commonly examined $\widetilde{\chi}_1^{\pm}\widetilde{\chi}_2^0$ pair production process. Additionally, non-simplified electroweak SUSY models have been examined by summing production cross sections for complex processes, not usually studied by conventional analyses. Using this information, I identified four models of interest and constructed analyses to exclude all four at the 95% confidence level.

The use of variational autoencoders for dimensional reduction of model parameters to a 2-dimensional plane, combined with global fit results allows one to be be sensitive to a broader range of phenomenology than is present in a simplified model. This technique has the potential to raise the sensitivity of LHC searches for supersymmetry by not imposing restrictions on the original parameter space. This technique is very easy to generalise to non-SUSY applications.

8 Optimisation Algorithms for High Dimensional Particle Physics Models

Throughout the previous chapters I have explored techniques involving BSM physics theories formulated using models with various free parameters. Let us now explore methods to identify what values of these free parameters are most realistic given what is known about the state of modern particle physics. Comparing these model predictions to experimental data can constrain the values of these free parameters and converge towards a theory that matches experimental observations as closely as possible. It is not uncommon for a BSM theory to have $\mathcal{O}(100)$ free parameters, and as such it is incredibly difficult to tune a given model to fit experimental data. A natural question to ask is how does one find the best set of parameters? In this chapter I explore a number of optimisation algorithms in high dimensional spaces and examine their performance with a number of metrics.

The likelihood function [95] refers to the probability of observing a set of experimental data given a set of model parameters. If one is considering data from multiple experiments, the likelihood function may be taken to be the product of likelihoods for each experiment. This likelihood function is rarely ever analytically known, and is often calculated using non-differentiable experimental simulations, so one must use derivative-free numerical methods to locate optima. Likelihood functions often contain multiple local optima so, especially in a high dimensional space, it is very difficult to locate a global optimum. Therefore, in this chapter, local optimisers commonly used in physics are neglected and instead a focus is placed on global methods. The baseline approach that each algorithm is measured against is randomly sampling the space a number of times and picking the highest likelihood value. This approach is extremely inefficient, since as the dimensionality of the space increases, the number of samples that must be taken in order to maintain the same point density increases exponentially. Additionally, high likelihood regions of physics parameter space generally occupy a very small percentage of the overall space, so each random sample will likely fall within uninteresting regions of the space.

Based off of my work in Ref. [96], this chapter explores a wide range of optimisation techniques which have not yet had mainstream use in particle physics applications. Working with members of the Dark Machines collaboration, each technique is run by a different member, aiming to identify the optima of a set of analytic test functions, and later, a

12-dimensional particle astrophysics test problem based on the MSSM. Each member is required to use a common test framework in order to ensure that the only differences in results are due to the performance of each algorithm. As this chapter contains work done by a collaboration, I will focus especially on my own contributions to the project.

8.1 Definition of Comparison Test Functions

Two different tests for each algorithm are explored in this chapter. The first test involves comparing the performance of each algorithm on a handful of analytic test functions, where the solution is known. Each test function is chosen to represent a different challenge to sampling algorithms that one might expect to see in a physics scenario. The second test applies each algorithm to a real physics example in order to determine whether the results seen in the analytic test scenarios generalise to realistic physics applications.

8.1.1 Analytic Test Functions

The four test functions are hidden from the parties writing and running each sampling function in order not to introduce biases towards any global optimum. In cases where the optimum naturally sits at a value which might be easily guessed by the initial conditions of a given sampling algorithm (such as 0), this optimum is shifted slightly. For all of the following equations, the number of dimensions is written as n.

Analytic Function 1

The equation for the first test function is written as

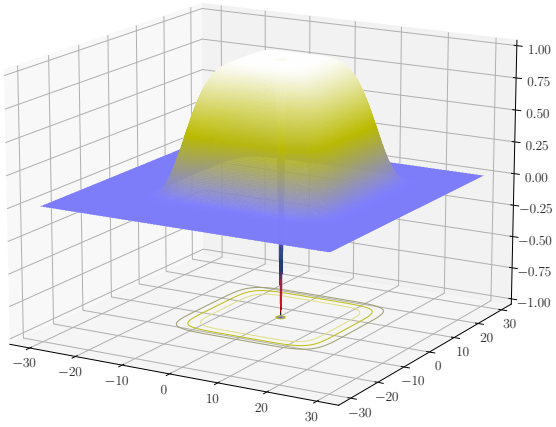
$$f(\mathbf{x}) = \exp\left(-\sum_{i=1}^{n}((x_i-2)/15)^6\right) - 2\exp\left(-\sum_{i=1}^{n}(x_i-2)^2\right)\prod_{i=1}^{n}\cos^2(x_i-2), \tag{8.1}$$

and is displayed in Figure 8.1a with $n = 2$. This function has a global minimum at $\mathbf{2}$, where it reaches a value of -1. This function is expected to be difficult to optimise, as the minimum is surrounded by a region of high function values. The domain of points that this function is searched over is $[-30, 30]^n$.

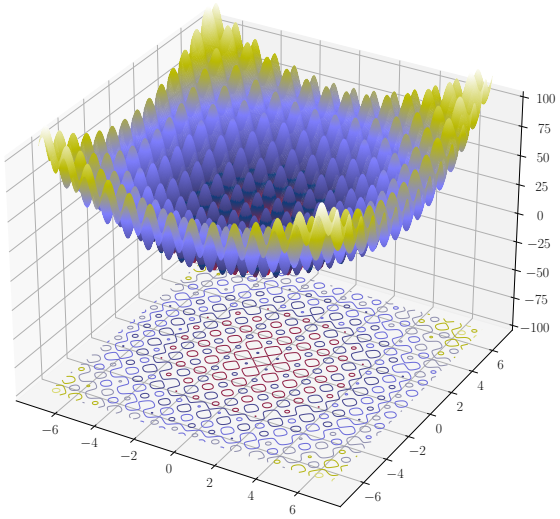
Analytic Function 2

The equation for the second test function is written as

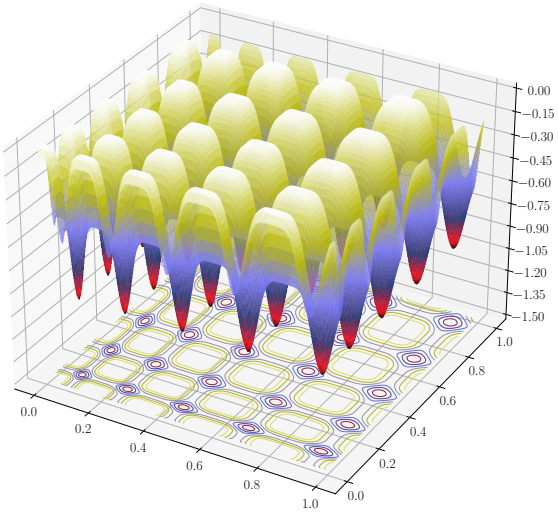
$$f(\mathbf{x}) = \sum_{i=1}^{n}\left[(x_i+0.23)^2 - 10\cos(2\pi(x_i+0.23)) + 10\right], \tag{8.2}$$



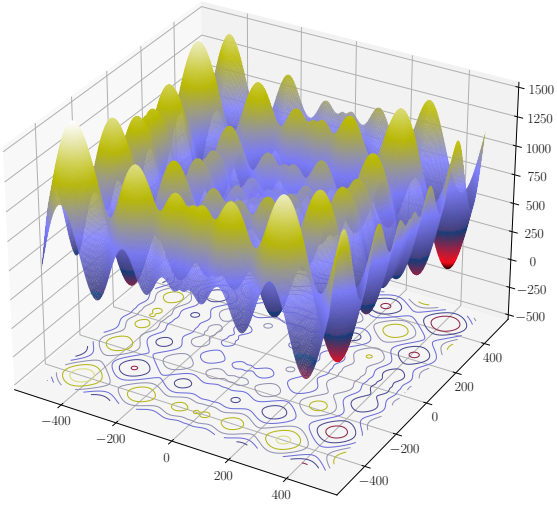
(A) Analytic Function 1 (eq. 8.1) — global minimum at **2**.



(B) Analytic Function 2 (eq. 8.2) — global minimum at $-\mathbf{0.23}$.



(C) Analytic Function 3 (eq. 8.3) — many degenerate minima.



(D) Analytic Function 4 (eq. 8.4) — global minimum at about **421**.

FIGURE 8.1: Visualisation of the explored analytic functions from Section 8.1 in 2-dimensional form.

and is displayed in Figure 8.1b with $n = 2$. This function has many local minima, with the global minimum at $\mathbf{-0.23}$. The challenge with this function will be avoiding the many local minima and identifying the correct global minimum. The domain of points that this function is searched over is $[-7, 7]^n$.

Analytic Function 3

The equation for the third test function is written as

$$f(\mathbf{x}) = -\frac{1}{n}\sum_{i=1}^{n} \sin^6\Big(5\pi(x_i^{\frac{3}{4}} - 0.05)\Big), \tag{8.3}$$

and is displayed in Figure 8.1c with $n = 2$. This function has many global minima, with a value of -1. This should be a fairly easy minimum to identify. The domain of points that this function is searched over is $[0, 1]^n$.

Analytic Function 4

The equation for the fourth test function is written as

$$f(\mathbf{x}) = 418.9829n - \sum_{i=1}^{n} x_i \sin\Big(\sqrt{|x_i|}\Big), \tag{8.4}$$

and is displayed in Figure 8.1d with $n = 2$. This function has a large domain, and is irregularly shaped which makes it difficult to locate the global minimum. The global minimum is approximately $\mathbf{420.968746}$, where the function is equal to 0. The domain of points that this function is searched over is $[-500, 500]^n$.

8.1.2 Particle Astrophysics Test Problem

Once the optimisation algorithms have been evaluated on the analytic test functions, each is applied to a realistic particle astrophysics problem. A recent global fit of a supersymmetric theory performed by the GAMBIT collaboration in Ref. [97] is used, and a fast interpolation of the likelihood function $\mathcal{L}$ is obtained. This likelihood function was originally quite computationally expensive to evaluate, hence the requirement for a fast interpolation function. This model adds a number of parameters to the Standard Model which must be explored in order to find regions of the parameter space which have strong agreement with current experimental results. In frequentist statistics, this is typically done by maximising the likelihood function. In this case the function chosen to be minimised is $-\log\mathcal{L}$. The original fit done by GAMBIT explored a 7-parameter phenomenological version of the Minimal Supersymmetric Standard Model (MSSM7). The 7 parameters, detailed in Section 2.3.2, that are involved in this model are the soft masses

$M_2, m^2_{\tilde{f}}, m^2_{H_u}, m^2_{H_d}$, the trilinear couplings for the third generation of quarks A_{u_3}, A_{d_3}, and $\tan\beta$. The input scale Q is chosen to be 1 TeV, and the sign of μ is chosen to be positive. These mass parameters are all defined at the Q common scale whereas $\tan\beta$ is defined at m_Z. The original fit also includes a variety of nuisance parameters. These are the strong coupling constant, the top quark mass, the local dark matter density, and the nuclear matrix elements for the strange, up and down quarks. In total, the global fit was performed with 12 parameters.

The fast interpolation of the likelihood function was done using a deep neural network, as proposed in Refs. [53, 98]. Approximately 2.3×10^7 samples taken from the global fit were used to train the network. The network consists of 4 hidden layers, each containing 20 fully connected nodes with a SELU activation function. The data, consisting of samples taken from a global fit done with GAMBIT, has been normalised to a Gaussian distribution, and is split into training/testing sets with a ratio of 90%/10%. The batch size is 1024, and the learning rate of the Adam optimiser is halved and early stopping is applied whenever the loss function (mean absolute error) stops improving for a handful of iterations in order to speed up training.

Figure 8.2 displays the reconstruction quality of the neural network. It is clear that the predicted log-likelihood is very well-correlated with the true log-likelihood. It's important to note that the reconstruction is not required to be perfect, and that this serves as a suitable proxy for such a difficult likelihood function as would typically be encountered in a particle astrophysics application. Now that the suitability of this neural network in approximating the likelihood function for this BSM model has been validated, let us take a look at a handful of sampling algorithms that will be tested on these problems. It should be noted that the original paper contains more algorithms than what are covered here. This chapter instead focuses on covering the algorithms that I personally worked on.

8.2 Optimisation Algorithms and Framework

The three algorithms that I helped to write, debug, and run are known as Bayesian Optimisation [99] (GPyOpt), Trust Region Bayesian Optimisation [100] (TuRBO), and Differential Evolution [101] (Diver). A number of other algorithms were also tested alongside these three, namely Particle Swarm Optimisation [102], Covariance Matrix Adaptation Evolution Strategy [103], Grey Wolf Optimisation [104], PyGMO Artificial Bee Colony [105], Gaussian Particle Filter [106], and AMPGO [107].

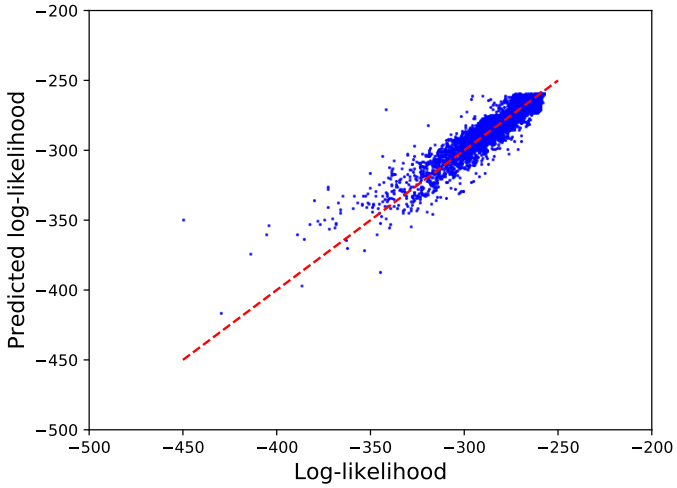


FIGURE 8.2: A scatter plot of the log-likelihood yeilded from the neural network compared to the true value of the log-likelihood. One expects to see a straight line $y = x$ (displayed in red) for perfect prediction. The high degree of correlation indicates that the neural network is adequately predicting the log-likelihood for a given point.

8.2.1 Bayesian Optimisation (GPyOpt)

Bayesian Optimisation [99] techniques attempt to find the optimum value $\mathbf{x}^*$ of an objective function $f(\mathbf{x})$ using the minimum number of function evaluations. This is especially useful when each function evaluation is expensive to calculate. The first step of the algorithm is to approximate the objective function $f(\mathbf{x})$ with a probabilistic regression model, known as the surrogate model, which is able to to predict the value of unseen samples in order to guide the decision of which samples to evaluate next. The surrogate model is initially trained on a set of random samples of the objective function. These samples can also be chosen by any other sampling algorithm. This surrogate model must be probabilistic, as such, popular choices are Gaussian processes or probabilistic ensembles. This surrogate model is continuously updated with each further sample of the objective function. After a given number of samples, a good estimate of what the objective function looks like is obtained. The acquisition function $\alpha(\mathbf{x})$ takes the latest posterior of the surrogate model and indicates where to sample next. This function must be easy to evaluate so that one can do cheap samples of the surrogate model, rather than computationally expensive evaluations of the objective function. In order to avoid getting stuck in local optima, the acquisition function must trade off "exploration" and "exploitation". Exploration involves exploring new regions of the parameter space in order to identify locations of interest, whereas exploitation involves further investigating these locations of interest and delving into a given local optimum. This procedure has the advantage of requiring relatively few function evaluations in order to find the optimum value $\mathbf{x}^*$ of $f(\mathbf{x})$, however

more computation is required for each subsequent sample.

For an illustrative example, let us use a Gaussian process to define the surrogate model. A Gaussian process is a stochastic process defined such that every finite linear combination of its variables is normally distributed. This can be written as $f(\mathbf{x}) \approx GP(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$. Here $\mu(\mathbf{x})$ is the prior function which can be assumed to be 0 without loss of generality. $k(\mathbf{x}, \mathbf{x}')$ is the kernel. Let us use the radial basis function for k, also known as the square exponential kernel, defined as:

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\lambda^2}\right), \text{ where } \lambda > 0. \tag{8.5}$$

The next step is to choose an acquisition function. There are a number of commonly used functions, each of which have trade-offs in exploration and exploitation. A number of popular choices are listed here, defining $\mu(\mathbf{x})$, and $\sigma(\mathbf{x})$ as the predicted mean and standard deviation for a given input $\mathbf{x}$. f^* is defined as the current best value found by the algorithm, and ψ is a parameter that controls the relative weight of exploration and exploitation. Φ and ϕ are the cumulative distribution function (CDF) and probability density function (PDF) of the standard normal distribution respectively.

- Maximum probability of improvement (MPI):

$$\alpha_{MPI}(\mathbf{x}) = \Phi(\gamma(\mathbf{x})), \text{ where } \gamma(\mathbf{x}) = \frac{\mu(\mathbf{x}) - f^* - \psi}{\sigma(\mathbf{x})} \tag{8.6}$$

- Expected improvement (EI):

$$\alpha_{EI}(\mathbf{x}) = (\mu(\mathbf{x}) - f^*)\Phi(\gamma(\mathbf{x})) + \sigma(\mathbf{x})\psi(\gamma(\mathbf{x})) \tag{8.7}$$

- Upper confidence bound (UCB):

$$\alpha_{UCB}(\mathbf{x}) = \mu(\mathbf{x}) - \psi\sigma(\mathbf{x}) \tag{8.8}$$

The acquisition function is sampled using some sampling algorithm, and the sample with the highest acquisition score is chosen to be evaluated by the optimisation function. After evaluation, this new sample becomes the next training point, the surrogate model is updated, and the next sample is selected. This iterative process is repeated until some stopping point is reached, be that after a certain number of evaluations, or after an acquisition function threshold is passed.

8.2.2 Trust Region Bayesian Optimisation (TuRBO)

The standard Bayesian Optimisation algorithm has poor scaling in high-dimensional applications. This is due to the acquisition function becoming difficult to optimise in high-dimensional spaces as it requires the same number of dimensions as in the input space. An emphasis on exploration over exploitation, and the assumption that surrogate models are homogeneous also contribute to the poor scaling of this technique. The surrogate model also uses a constant length scale λ and signal variance σ, which is often ineffective for non-trivial high dimensional functions. Samples in high dimensional spaces tend to be far apart, which leads to high uncertainty in the surrogate model, causing the acquisition function to focus more on exploration than exploitation.

The trust region Bayesian optimisation (TuRBO) algorithm [100] improves on the standard version by fitting a set of local models and determining how best to sample them in order to find the global optimum $\mathbf{x}^*$ in the most efficient manner. Multiple independent Gaussian processes are employed to perform simultaneous local optimisations. This allows for heterogeneous modelling of the objective function, while retaining all of the benefits of Bayesian optimisation. Each Gaussian process is assigned a "Trust Region", which is a hyper-rectangle centred at the best found solution at each iteration. The parameter λ controls the base side length L of each Gaussian process with the relation:

$$L_i = \frac{\lambda_i L}{\left(\prod_{j=1}^{d} \lambda_j\right)^{1/d}}. \tag{8.9}$$

The set of points that can be selected by the acquisition function is limited to points within this trust region, meaning that one can control the trade-off between exploration (larger L) and exploitation (smaller L). Points are selected from each trust region using a greedy Thompson Sampling approach, where the ith point x_i is drawn using

$$x_i = \min_{l} \min_{x \in \text{trust region}} f_l, \tag{8.10}$$

where f_l is a sample from the lth Gaussian process.

8.2.3 Differential Evolution (Diver)

The Diver algorithm uses a form of Differential Evolution [101] in order to sample high dimensional spaces. Differential Evolution (DE) is a population based heuristic optimisation strategy which belongs to the broader class of evolutionary algorithms. DE involves evolving a population of "target vectors" $\mathbf{X}_i^g$, of particular points in the parameter space

for a number of generations. Here i denotes the ith individual, and g refers to the generation of the population. The initial generation is typically randomly selected from within the parameter space.

A single generation is evolved from one to the next via three steps: mutation, crossover, and selection. Let us look at the particular variant that is used by Diver known as λjDE, or rand-to-best/1/bin where some selected parameters are optimised as if they were dimensions of the parameter space. The first two parts of the name "rand-to-best/1" refer to the mutation strategy (best solution, plus some randomly selected points, and a single difference vector). The third part of the name "bin" refers to the cross over strategy (binomial).

The mutation step is done by choosing a target vector $\mathbf{X}_i$, and constructing one or more vectors $\mathbf{V}_i$ referred to as "donor vectors". The donor vectors are a set of vectors that are drawn from in the construction of "trial vectors" which are denoted $\mathbf{U}_i$. In the rand-to-best/1/bin algorithm, $\mathbf{V}_i$ is given by:

$$\mathbf{V}_i = \lambda \mathbf{X}_{best} + (1 - \lambda)\mathbf{X}_{r1} + F(\mathbf{X}_{r2} - \mathbf{X}_{r3}), \tag{8.11}$$

where $\mathbf{X}_{r1}, \mathbf{X}_{r2}$, and $\mathbf{X}_{r3}$ are three unique randomly chosen vectors from the current generation. F and λ are two free parameters which are tuned as if they were dimensions of the parameter space in the λjDE variant of the algorithm.

Crossover involves the construction of a trial vector $\mathbf{U}_i$, by selecting parameter values from either the target vector $\mathbf{X}_i$, or one of the donor vectors $\mathbf{V}_i$. This selection process is controlled by one final parameter C_r, which, just as with F and λ, is tuned as if it were a dimension of the parameter space. For each parameter, a random number is chosen between 0 and 1. If that number is greater than C_r, that parameter value is taken from the target vector, otherwise it is taken from the donor vector. At the end of this process, a single parameter of $\mathbf{U}_i$ is chosen at random and replaced with the corresponding component of $\mathbf{V}_i$. This is done to ensure that $\mathbf{U}_i \neq \mathbf{X}_i$.

The final step, selection, is done by comparing the trial vector $\mathbf{U}_i$ and the target vector $\mathbf{X}_i$. Whichever vector returns the best value of the objective function is kept for the next generation.

This project explores two different differential evolution algorithms, Diver and PyGMO. PyGMO is a Python package that implements iDE and jDE differential evolution algorithms, which are simplified versions of the λjDE algorithm described here. These algorithms have fewer tunable parameters and so are expected to be less flexible.

8.2.4 Particle Swarm Optimisation

The following algorithms I did not have a hand in, but are important to understand in order to interpret results in Section 8.3. Particle Swarm Optimisation [102] is a population-based evolutionary algorithm that does not use derivatives. A number of parameter samples, collectively referred to as "the swarm", are taken and each particle within the swarm is given a velocity. The positions of each particle are used to update the velocities of all others, with each particle moving along their velocity vector after each generation. The velocity of particle i in generation g is denoted

$$\boldsymbol{v}_i^{g+1} = \omega \boldsymbol{v}_i^g + \phi_1 r_1 (\boldsymbol{x}_{i,pb} - \boldsymbol{x}_i^g) + \phi_2 r_2 (\boldsymbol{x}_{gb} - \boldsymbol{x}_i^g), \tag{8.12}$$

such that the updated position vector $\boldsymbol{x}_i^{g+1} = \boldsymbol{x}_i^g + \boldsymbol{v}_i^g$. Here r_1 and r_2 are uniform random numbers between 0 and 1, $x_{i,pb}$ is the ith particles best-fit position across all generations, x_{gb} is the global best fit across all particles and generations, and ω, ϕ_1, and ϕ_2 are free parameters. This chapter uses a particle swarm optimisation algorithm referred to as j-Swarm, where ω, ϕ_1, and ϕ_2 are dynamically optimised over the course of a run.

8.2.5 Covariance Matrix Adaptation Evolution Strategy

The Covariance Matrix Adaptation Evolution Strategy [103] henceforth referred to as CMA-ES, is another evolutionary optimisation algorithm. From an initial sample $x^{(0)}$, a set of λ new points, referred to as a population, are sampled from a multivariate normal distribution about $x^{(0)}$ with covariance matrix $(\sigma^{(g)})^2 \boldsymbol{C}^{(g)}$. Here g refers to the generation number. The optimistation function is evaluated at all λ points and the best μ points are used to calculate the next generation given by

$$\boldsymbol{x}^{(g+1)} = \sum_{j=1}^{\mu} w_j \boldsymbol{x}_j^{(g)}, \tag{8.13}$$

where $w_j > 0$ are weights summing to 1, with j being the sorted index running from best to worst point. The step size $\sigma^{(g)}$ and the matrix $\boldsymbol{C}^{(g)}$ are updated after each generation in order to maximise the probability of the new generation improving upon the old. This update is given by

$$\begin{aligned}
\mathbf{p}_c^{(g+1)} &= (1 - c_c)\mathbf{p}_c^{(g)} + \sqrt{c_c(2 - c_c)\mu_{\text{eff}}}\, \frac{\mathbf{x}^{(g+1)} - \mathbf{x}^{(g)}}{\sigma^{(g)}}, \\
\mathbf{C}^{(g+1)} &= (1 - c_{\text{cov}})\mathbf{C}^{(g)} + \frac{c_{\text{cov}}}{\mu_{\text{cov}}} \mathbf{p}_c^{(g+1)} \mathbf{p}_c^{(g+1)^T} \\
&\quad + c_{\text{cov}} \left(1 - \frac{1}{\mu_{\text{cov}}}\right) \sum_{j=1}^{\mu} w_j \left(\frac{\mathbf{x}_j^{(g+1)} - \mathbf{x}^{(g)}}{\sigma^{(g)}}\right) \left(\frac{\mathbf{x}_j^{(g+1)} - \mathbf{x}^{(g)}}{\sigma^{(g)}}\right)^T,
\end{aligned} \tag{8.14}$$

where $\mathbf{p}_c^{(g)}$ is a cumulative path which stores information from previous generations, $c_c < 1$ is the learning rate for the cumulative path, $c_{cov} < 1$ is the learning rate for the covariance matrix, and $\mu_{cov} \geq 1$ controls the ratio between the cumulation and rank-μ updates. $\mu_{eff} = \left(\sum_{j=1}^{\mu} w_j^2\right)^{-1}$ is the parameter representing the effective selection mass. The cumulation update adapts the matrix to the large scale gradient of the optimisation function, while the rank$-\mu$ update adapts to the local gradient of the optimisation function.

The chosen update to the step size at each generation is based on the absolute length of the cumulative path. If many steps are taken in the same direction, the length is assumed to be larger than if steps are taken in different directions. The update is given by

$$\begin{aligned} \mathbf{p}_\sigma^{(g+1)} &= (1 - c_\sigma)\mathbf{p}^{(g)} + \sqrt{c_\sigma(2 - c_\sigma)\mu_{\text{eff}}}\, \mathbf{C}^{(g)^{-1/2}} \left(\frac{\mathbf{x}^{(g+1)} - \mathbf{x}^{(g)}}{\sigma^{(g)}}\right) \\ \sigma^{(g+1)} &= \exp\left\{\left(\frac{c_\sigma}{d_\sigma}\left[\frac{|\mathbf{p}_\sigma^{(g+1)}|}{|\mathcal{N}(0, \mathbf{I})|}\right]\right)\right\} \end{aligned} \tag{8.15}$$

with $\mathbf{C}^{(g)^{-1/2}} = \mathbf{B}^{(g)}\mathbf{D}^{(g)^{-1}}\mathbf{B}^{(g)^T}$ where $\mathbf{C}^{(g)} = \mathbf{B}^{(g)}\mathbf{D}^{(g)^2}\mathbf{B}^{(g)^T}$ is the eigenvalue decomposition of $\mathbf{C}^{(g)}$. The learning rate c_σ and the damping rate d_σ both control the adaptation speed.

The tunable parameters of the CMA-ES algorithm are entirely independent from the objective function and depend almost exclusively on the dimensionality of the parameter space. The implementation in this chapter is from the pycma package.

8.2.6 Grey Wolf Optimisation

The Grey Wolf Optimisation algorithm [104] is a swarm intelligence algorithm drawing inspiration from the behaviour of packs of grey wolves. Each search agent is assigned one of four categories: α, β, δ, or ω. The 1st, 2nd, and 3rd best (fittest) solutions are assigned to α, β, and δ, while the remainder are assigned to ω. During optimisation, searching is led by α, while β and δ have a smaller influence. Each search agent is initialised to random positions in the feature space and are assigned categories based on their fitness. The positions of each agent are then updated with each positional update containing both stochastic elements and influence from the positions of the α, β, and δ agents. The roles of each agent are set at the initialisation of the algorithm and are not updated. Let us define two vectors used in the update process:

$$\vec{A}_i = 2\vec{a} \cdot \vec{r}_{1,i} - \vec{a} \tag{8.16}$$

$$\vec{C}_i = 2\vec{r}_{2,i}. \tag{8.17}$$

Where $\vec{A}_i$ denotes the ith unique generation of the vector. These vectors have a number of parameters equal to that of the dimensionality of the feature space. $\vec{r}_{(1,2),i}$ are vectors of random numbers between $[0, 1]$, and $\vec{a}$ has components which decrease linearly from 2 to 0 over each iteration. Let us now define three more vectors to capture the relationship between the position of a given agent relative to the three fittest agents:

$$D_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}|, \tag{8.18}$$
$$D_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}|, \tag{8.19}$$
$$D_\gamma = |\vec{C}_3 \cdot \vec{X}_\gamma - \vec{X}|. \tag{8.20}$$
$$\tag{8.21}$$

Each agent is updated to their new positions given by

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}, \tag{8.22}$$

where t indicates the current iteration, and

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot D_\alpha \tag{8.23}$$
$$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot D_\beta \tag{8.24}$$
$$\vec{X}_3 = \vec{X}_\gamma - \vec{A}_3 \cdot D_\gamma. \tag{8.25}$$
$$\tag{8.26}$$

These updates are designed to have the α, β, and γ agents encircle the optima, while the remaining ω agents randomly search around this position. This process is continued until either a maximum number of iterations is reached, or some condition of fitness on the optimal solution is met. The only hyperparameters present in this algorithm are the number of agents, which should be at least 4.

8.2.7 PyGMO Artifical Bee Colony

The Artificial Bee Colony algorithm [105] is another swarm intelligence algorithm inspired by the behaviour of honey bees searching for food sources. The version used in this chapter comes from the PyGMO package. The Artificial Bee Colony algorithm keeps track of SN active points referred to as x_s, with s running from 1 to SN. The initial position of each point is uniformly initialised, and the objective function at each point is evaluated. Returning to the bee analogy, these initial points can be thought of as food sources, and the value of the objective function can be thought of as the food gain from a given source. This algorithm runs in an iterative cycle until a certain number of iterations have passed.

Each iteration is broken into three steps. First each active data point x_i is moved for each dimension j towards another randomly selected datapoint x_k with its new position given by

$$v_{i,j} = x_{i,j} + \phi_{ij}(x_{i,j} - x_{k,j}), \tag{8.27}$$

where ϕ_{ij} is a random number between 0 and 1 drawn from a uniform distribution. For the newly proposed point v_i the objective function is evaluated, and upon finding an improved value, the point x_i is updated to v_i. If no improved value is found, the original point x_i is kept. The number of failures is kept and is used to give up on points which are not updated for many iterations. This is done in the second step, where these inactive points are reinitialised uniformly in the parameter space. The third and final step of each iteration is to use Equation 8.27 to update the "active" points. A point is has a chance to be deemed active based on the probability function

$$p_i = \frac{\text{fitness}_i}{\sum_j \text{fitness}_j}, \tag{8.28}$$

where fitness_i is given by

$$\text{fitness}_i = \begin{cases} (1 + f_i)^{-1}, & f_i \geq 0 \\ 1 + |f|, & f_i < 0, \end{cases} \tag{8.29}$$

and f_i is the value of the objective function at x_i. Once again, as in the first step, only updates that improve the fitness are taken. In this phase the update attempts are also kept track of in order to remove inactive points. One of the major strengths of this algorithm is its automated balance of exploration/exploitation. As the number of iterations increases, the swarm agents will move closer to the best fit, as updates only occur when the fitness increases. The two parameters that govern this algorithm are the number of iterations, improving resolution, and the number of times to run the algorithm, improving reliability.

8.2.8 Gaussian Particle Filter

The Gaussian Particle Filter [106] is a scanning algorithm that begins by taking an initial number of randomly selected points. The number, range, and sampling prior are all defined by the user. Each of these points acts as a seed to define a multi-dimensional Gaussian distribution from which new points are drawn. The number of points drawn is proportional to the value of the objective function at the seed. The width of the Gaussians steadily decrease over the course of the run, the rate of which is controlled by the "width decay" parameter. In each iteration N data points are sampled from M Gaussians and the objective function is evaluated at each point. These samples are combined with a

fraction of the best data points which form the seeds for the Gaussians. This fraction is referred to as the survival rate. From the surviving set, the M best data points are chosen to seed the next iteration of Gaussians. The version of this algorithm employed in this chapter explores both a logarithmic and uniform prior, as well as a variety of width decays and survival rates.

8.2.9 AMPGO

AMPGO or Adaptive Memory Programming for Global Optimisation [107] is a global optimisation algorithm that consists of three basic steps. First, a number of points are chosen from the parameter space using a uniform distribution. Next, a local solver is then used to find the local optima about each point. Finally, a method known as Tabu Tunnelling [108] is used to locate another point with equal or better fitness to the original local optimum. From this new point the local optimiser is rerun. This iterative process is repeated until some stopping condition is met. This algorithm relies heavily on being able to find points with equal or greater fitness to the local optima, and since the possible tunnelling directions are infinite, this can be quite the challenge. Very narrow global optima, such as in Equation 8.1, are difficult to find, as the probability of tunnelling into them is increasingly small. High dimensional parameter spaces are also a great difficulty for this algorithm as when the dimensionality increases, the volume to tunnel through increases exponentially. Given these difficulties it is unlikely that this algorithm will be able to perform well in high dimensional particle astrophysics problems.

8.2.10 Algorithm Parameters

Each optimisation algorithm explored in this chapter has a number of hyper-parameters to explore. These can broadly be divided into 4 different categories.

- **Convergence Parameters:** These parameters control the point at which an algorithm halts. A stricter convergence condition will mean that the optimum found will likely be closer to the true value, however it will also likely require more function evaluations.
- **Resolution Parameters:** These parameters affect the resolution at which the function is searched. A higher resolution will mean the likelihood function is searched with more detail around points of interest, however it will also require a higher number of function evaluations.
- **Hint Parameters:** These parameters give the algorithm hints as to the location of the optimum. For example, algorithms where one can choose the starting point will benefit greatly from starting as near as possible to the global optimum.

Parameter	Explored values	Type
AMPGO		
Number of sampled points	2000, 5000, 10000, 20000	Resolution
CMA-ES		
Function tolerance	10^{-11}, 10^{-7}, 10^{-4}, 10^{-1}	Convergence
Population size (λ)	20, 50, 100, 500	Resolution
Diver		
Threshold for convergence	10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}	Convergence
Population size	2000, 5000, 10000, 20000	Resolution
Parameter update scheme	λjDE	-
Gaussian Particle Filter		
Width decay	0.90, 0.95, 0.99	Convergence
Logarithmic sampling	`True`, `False`	Hint
Survival rate	0.2, 0.5	Reliability
Initial gaussian width	2	Reliability
GPyOpt		
Threshold for Convergence	10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}	Convergence
Particle Swarm Optimisation		
Threshold for convergence	10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}	Convergence
Population size	2000, 5000, 10000, 20000	Resolution
Adaptive ϕ	`True`	Reliability
Adaptive ω	`True`	Reliability
PyGMO Artificial Bee Colony		
Generations	100, 250, 500, 750	Resolution
Maximum number of tries	10, 50, 100	Reliability
PyGMO Differential Evolution		
Generations	100, 250, 500, 750	Resolution
Parameter update scheme	iDE, jDE	-
PyGMO Grey Wolf Optimisation		
Generations	10, 50, 100, 1000	Resolution
Random Sampling		
Number of points	10, 50, 100, 500, 1000, 5000, 10000, 50000, 100000, 500000, 1000000	Resolution
Trust Region Bayesian Optimisation (TuRBO)		
Max #evaluations / iteration	64, 100	Convergence

TABLE 8.1: A grouping of the various parameters of each optimisation technique into the categories described in the main text. The explored values for these parameters can be found in the second column. Note that the algorithms I personally worked on are Diver, GPyOpt, and TuRBO.

- **Reliability Parameters:** These are parameters that control the robustness of a given algorithm.

The best choice of hyper-parameters is not known and will depend on the function being sampled. In order to find the best values possible, a handful of different choices are

considered. While each algorithm does not necessarily have one of each of the 4 types of hyper-parameter detailed previously, most have at least a convergence and a resolution parameter. Table 8.1 details each algorithm and their parameters which are explored in the next section.

8.2.11 High-Dimensional Sampling Framework

All tests run for this chapter were performed within an open-source Python package written specifically for the project. The High Dimensional Sampling framework (HDS) can be found at `https://github.com/DarkMachines/high-dimensional-sampling/`, along with a detailed technical introduction to the package. The full code is published under the MIT license. This package is used to ensure that the only difference in the performance of each algorithm comes from the workings of the algorithm itself and not from quirks in how the operator runs said algorithm. This package also makes the results easily reproducible, and automates as much of the experiment as possible while minimising loss of configurability. The output of the HDS framework is standardised in order to make the comparison of algorithms as easy as possible.

8.3 Results

In this section I compare the best found optimum of each hidden function described in Section 8.1 yielded from each sampling algorithm. Each of these problems is being treated as a minimisation problem, searching for the lowest value of some underlying function.

8.3.1 Analytic Test Functions

The analytic functions, detailed in Section 8.1.1, are explored in 2, 3, 5, and 7 dimensional examples in order to observe how the performance of each algorithm changes with different dimensionalities. This will give insight into the performance on the particle astrophysics problem, which is modelled using a 12 dimensional likelihood function. It is to be expected that as the dimensionality of the problem increases, the performance of each algorithm will decrease. Each algorithm is run for a variety of hyper-parameters, detailed in Table 8.1, in order to see which set of hyper-parameters works best in each scenario. For the algorithms I worked on, the very best set of these hyper-parameters for each hidden function is detailed in Section 8.4.

Figure 8.3 displays the accuracy with which each algorithm is able to identify the global minimum for the hidden function shown in Figure 8.1a in various dimensions. Each circle represents a single run of the algorithm with different hyper-parameters. The size of the circle is proportional to $\log_{10}$ of the total number of function evaluations, and

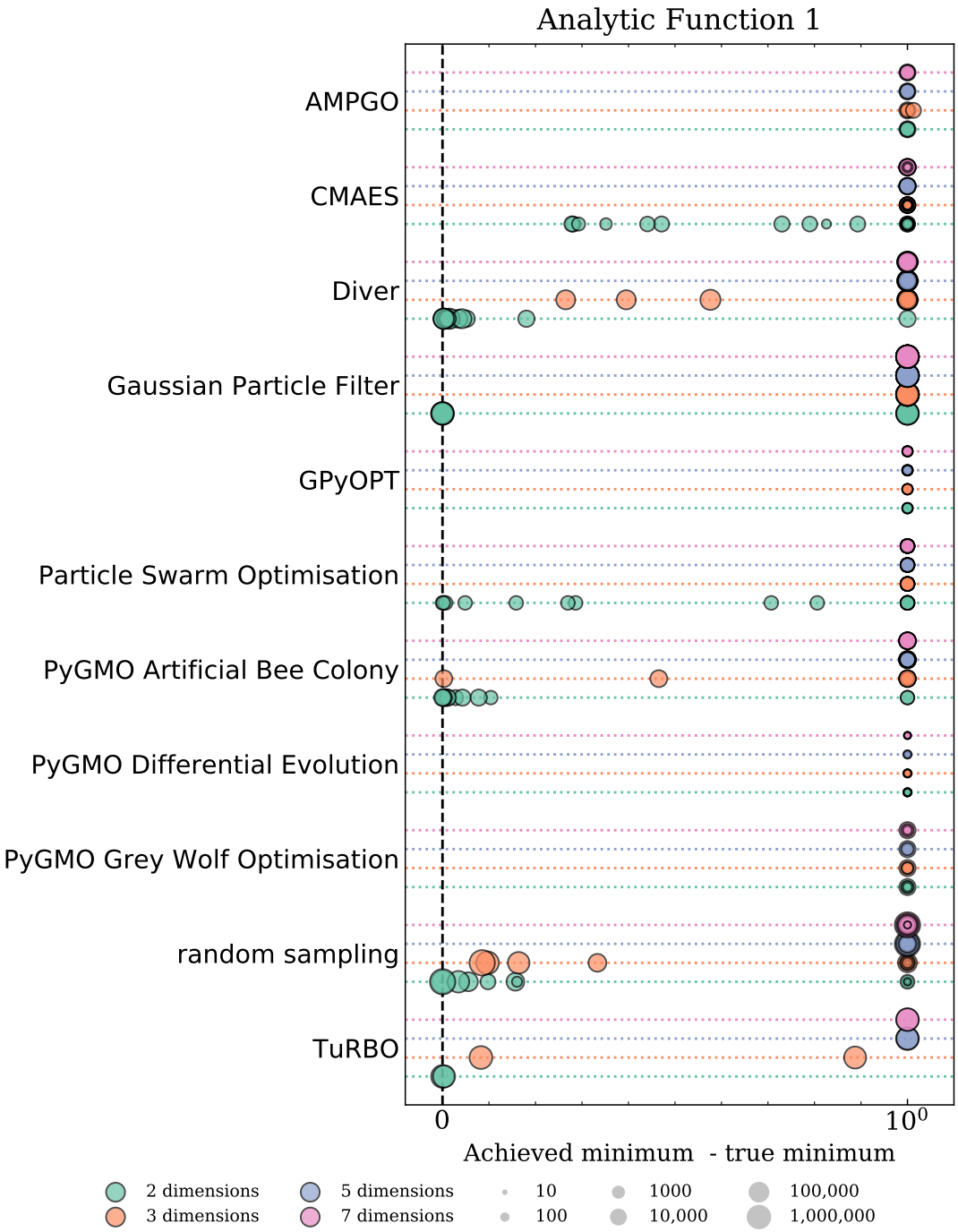


FIGURE 8.3: Results from different optimisation algorithms on the analytic function in Equation 8.1. The results are shown as semi-opaque circles, of which the area increases logarithmically with the number of function evaluations needed to obtain that specific result. The four horizontal lines for each algorithm belong to the four explored dimensionalities, from top to bottom 7-dimensional (pink), 5-dimensional (purple), 3-dimensional (orange) and 2-dimensional (green). The horizontal axis shows the difference between the (known) log-likelihood at the global minimum and that at the found minimum.

each line/colour corresponds to a different number of dimensions. The x-axis displays the true minimum subtracted from the achieved minimum. This means that the very best possible score here is 0, where the achieved minimum is equal to the true minimum. Figure 8.3 shows that some algorithms never get close to the true minimum, even in low dimensional scenarios. In ≥ 3 dimensions, no algorithm manages to find the global minimum. It is shown in Figure 8.1a that the global minimum lies in a very sharp pit surrounded by local maxima, making this minimum very difficult to find, even in low dimensions. The best performing algorithm is PyGMO Artificial Bee Colony, which finds the global minimum in both 2 and 3 dimensions with relatively few function evaluations. The worst performing algorithms are PyGMO Grey Wolf Optimisation, Gaussian Particle Filter, AMPGO, GPyOpt, and PyGMO Differential Evolution. For the PyGMO implementation of differential evolution, this poor performance could be due to the relatively small number of function evaluations, and a different choice of hyper-parameters might yield better performance. This is supported by the fact that Diver performs quite well, though is not able to find the minimum in 3D. In 3D, all algorithms except for TuRBO and PyGMO Artificial Bee Colony are outperformed by simple random sampling, which is not terribly surprising due to the nature of this problem running counter to the way a lot of these algorithms search for global minima. GPyOpt's poor performance is easily explained due to the way the algorithm operates. The sharply spiked local minimum is exactly the sort of feature that a Bayesian optimisation algorithm is likely to miss due to its relatively small number of function evaluations, and concentration of said evaluations in regions of the parameter space that the algorithm has deemed interesting. TuRBO on the other hand is able to find the global minimum in 2D and get quite close in 3D. This is due to it partitioning the space into separate regions. One of these regions is small enough to contain the global minima as an obvious value to choose over the plateau of false minima around the outside of the function.

Figure 8.4 shows the accuracy with which each algorithm is able to identify the global minimum for the hidden function shown in Figure 8.1b in various dimensions. Almost every algorithm is able to identify the global minimum, and outperform random sampling which is an indication that these algorithms are in fact an effective means of identifying global minima for functions that resemble hidden function 2. GPyOPT succeeds in 2 dimensions and is outperformed by TuRBO in all cases except for the 7-dimensional case, though neither is able to find the global minimum. TuRBO does, however, require many more function evaluations. Diver is able to find the global minimum in all dimensions, however it uses quite a few function evaluations to do so. The performance of PyGMO Differential Evolution suggests that it is possible that Diver could achieve similar results with fewer function evaluations. The final feature of note is that the results of each algorithm consistently get worse in higher dimensions, as expected.

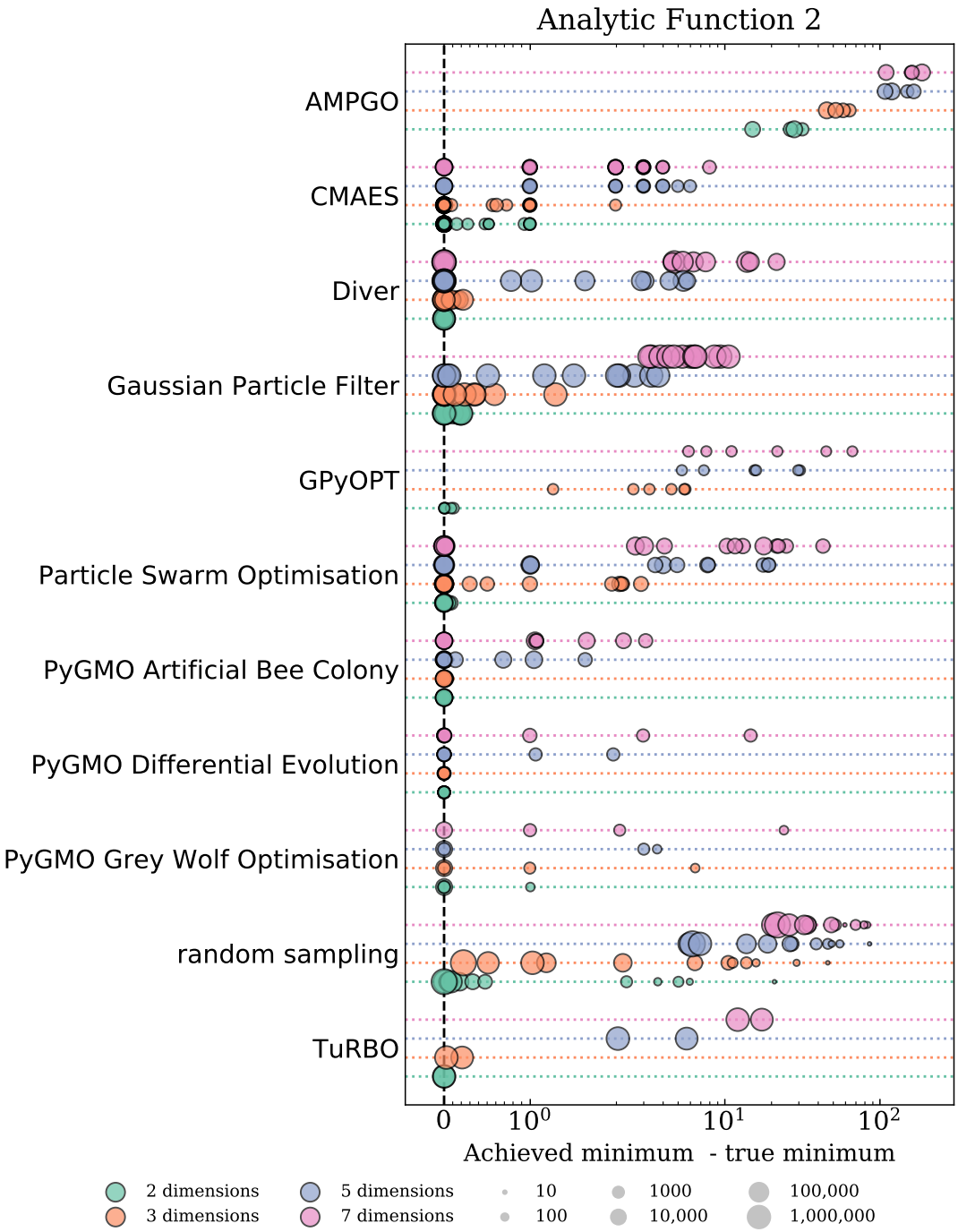


FIGURE 8.4: Results from different optimisation algorithms on the analytic function in Equation 8.2. The results are shown as semi-opaque circles, of which the area increases logarithmically with the number of function evaluations needed to obtain that specific result. The four horizontal lines for each algorithm belong to the four explored dimensionalities, from top to bottom 7-dimensional (pink), 5-dimensional (purple), 3-dimensional (orange) and 2-dimensional (green). The horizontal axis shows the difference between the (known) log-likelihood at the global minimum and that at the found minimum.

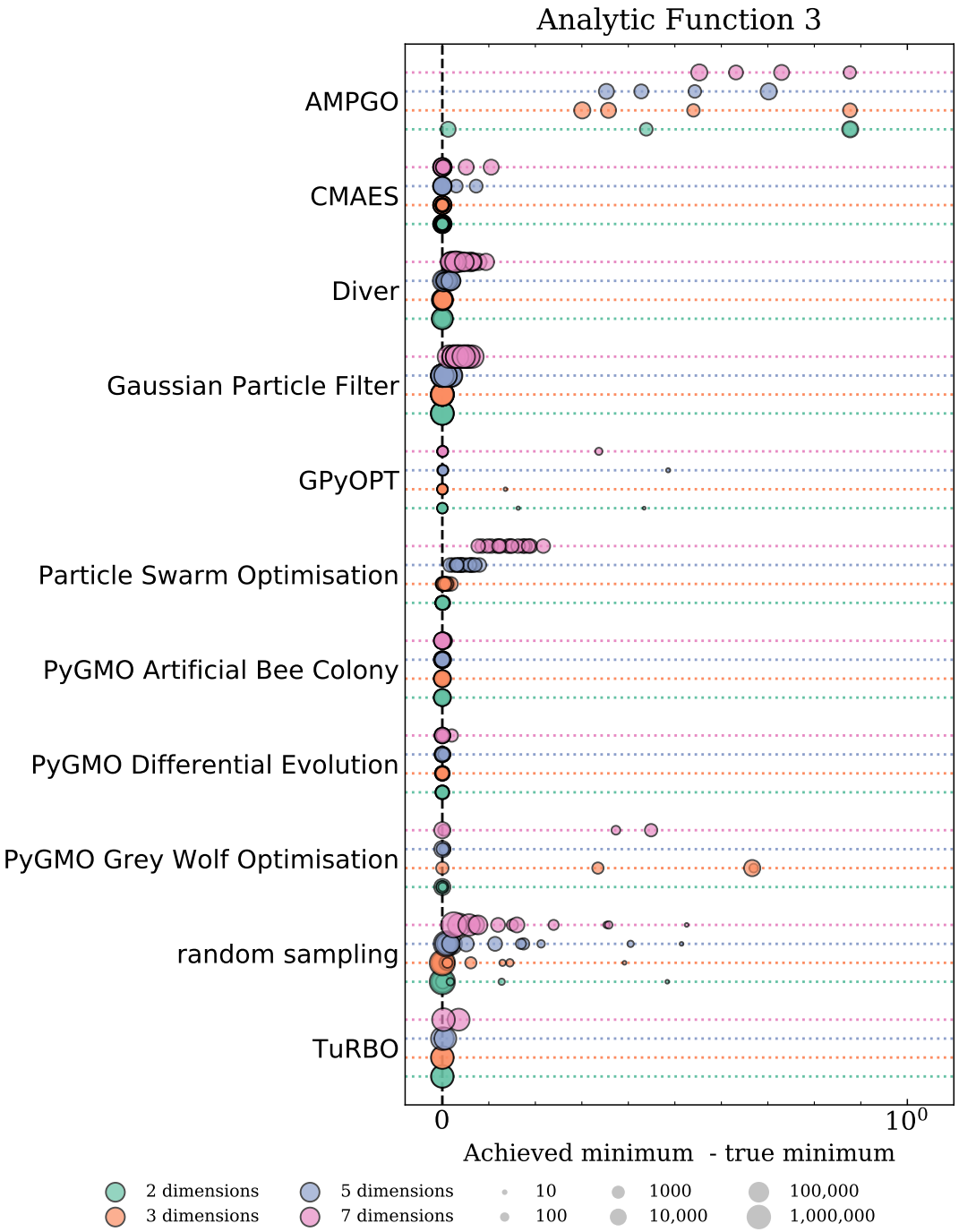


FIGURE 8.5: Results from different optimisation algorithms on the analytic function in Equation 8.3. The results are shown as semi-opaque circles, of which the area increases logarithmically with the number of function evaluations needed to obtain that specific result. The four horizontal lines for each algorithm belong to the four explored dimensionalities, from top to bottom 7-dimensional (pink), 5-dimensional (purple), 3-dimensional (orange) and 2-dimensional (green). The horizontal axis shows the difference between the (known) log-likelihood at the global minimum and that at the found minimum.

Figure 8.5 shows the accuracy with which each algorithm is able to identify the global minima for the hidden function shown in Figure 8.1c in various dimensions. The performance on hidden function 3 is much better than on any of the previous functions. Every algorithm (except AMPGO) is able to find the global minimum in 2 and 3 dimensions, and most are able to outperform random sampling. This is indicative of the fact that hidden function 3 has many identical global minima, and most algorithms are able to find one of them. GPyOPT gives the best performance here, finding the global minimum in all dimensions with very few function evaluations. TuRBO similarly finds the global minimum in all dimensions, albeit with more function evaluations due to it needing to perform redundant evaluations in many separate optimisations. Diver is able to find the global minima in all dimensions except the 7-dimensional case. However the PyGMO implementation of differential evolution outperforms Diver, finding the global minima in all dimensions with less function evaluations.

Figure 8.6 shows the accuracy with which each algorithm is able to identify the global minimum for the hidden function shown in Figure 8.1d in various dimensions. Hidden function 4 has many local minima, but only one global minimum, making it a little more difficult than hidden functions 2 and 3. Once again, AMPGO has the poorest performance, performing worse than random sampling. PyGMO Grey Wolf Optimisation also performs similarly poorly. PyGMO Differential Evolution performs the best here, getting the correct value in all dimensionalities. Diver is also able to get the correct values, albeit with higher numbers of function evaluations, though this may change with a different selection of hyper-parameters. Again, Bayesian optimisation algorithms perform poorly in higher dimensions, although TuRBO performs considerably better than GPyOPT in > 5 dimensions.

Table 8.2 presents a summary of the 11 algorithms on these 4 analytic functions. PyGMO Artificial Bee Colony appears to be the best overall algorithm, performing well on all functions except hidden function 1 where all algorithms performed quite poorly. AMPGO is consistently worse than random sampling. The Bayesian optimisation algorithms generally perform well when there are not hidden global minima, however this can be mitigated by adding latin hypercube sampling, as in the TuRBO algorithm. Both differential evolution algorithms are consistently quite strong in both the PyGMO and Diver implementations, while CMA-ES also shows fair performance. Particle Swarm Optimisation consistently requires a high number of evaluations, and is not consistent across all hidden functions. Finally, Gaussian Particle Filter struggles in high dimensions, and is very sensitive to the choice of hyper-parameters. In Section 8.4 I go into detail on the best hyper-parameters for each of the algorithms that I personally worked on.

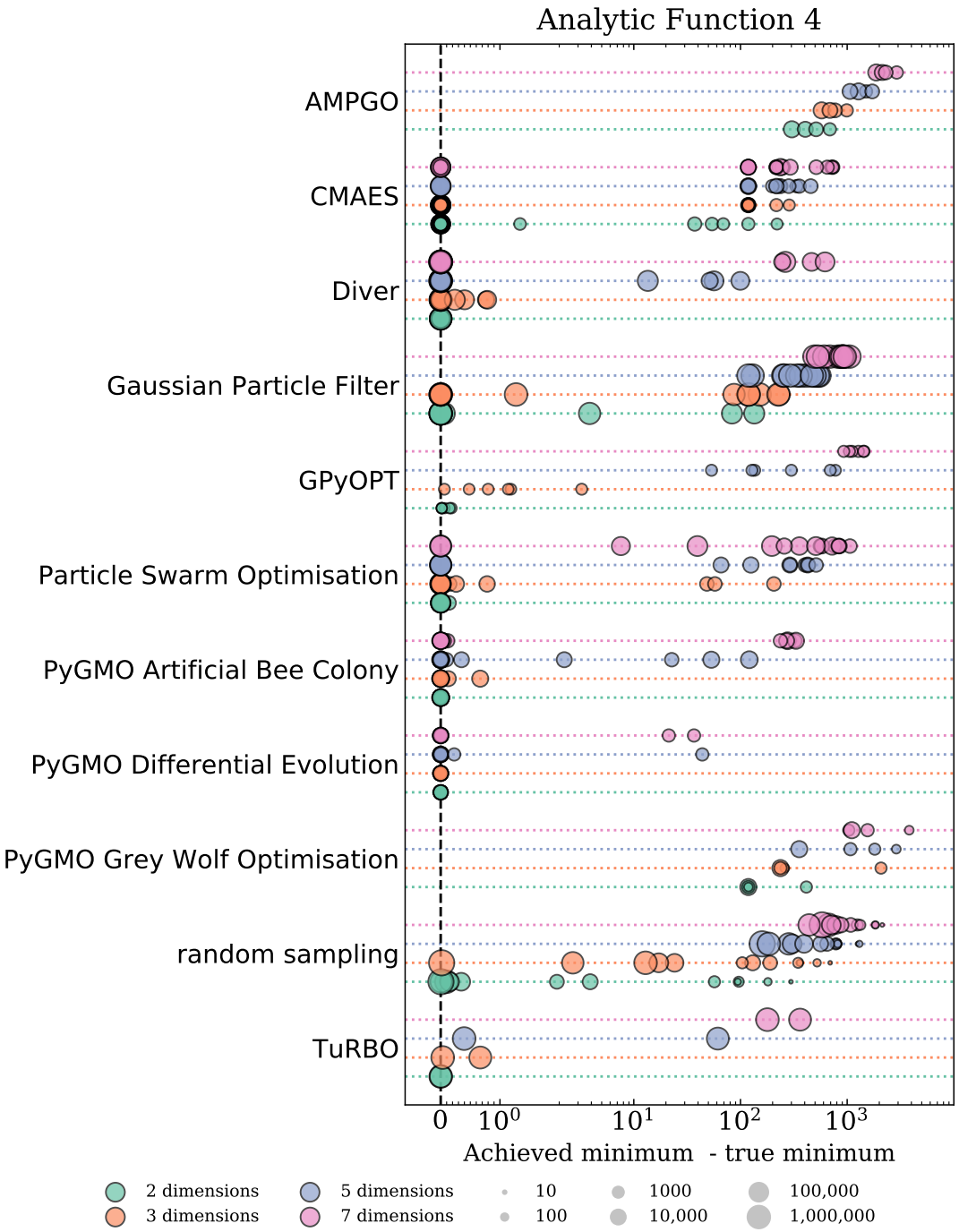


FIGURE 8.6: Results from different optimisation algorithms on the analytic function in Equation 8.4. The results are shown as semi-opaque circles, of which the area increases logarithmically with the number of function evaluations needed to obtain that specific result. The four horizontal lines for each algorithm belong to the four explored dimensionalities, from top to bottom 7-dimensional (pink), 5-dimensional (purple), 3-dimensional (orange) and 2-dimensional (green). The horizontal axis shows the difference between the (known) log-likelihood at the global minimum and that at the found minimum.

	Finding sharp minimum	**Finding global minimum**	**Performance with high dimensions**	**Average number of evaluations**
AMPGO	bad	bad	bad	low
CMA-ES	very low dimensions	good	good	medium
Diver	low dimensions	good	good	high
Gaussian Particle Filter	very low dimensions	low dimensions	highly configuration and function dependent	high
GPyOpt	bad	low dimensions	function dependent	low
Particle Swarm Optimisation	very low dimensions	good	configuration dependent	medium
PyGMO Artificial Bee Colony	low dimensions	good	good	medium
PyGMO Differential Evolution	bad	good	good	low
PyGMO Grey Wolf Optimisation	bad	bad	function dependent	medium
TuRBO	low dimensions	moderate dimensions	function dependent	high
Random Sampling	low dimensions	low dimensions	function dependent	high

TABLE 8.2: Summary of optimisation algorithm performance. Analytic Function 1 can be used to evaluate the performance of algorithms for finding a sharp minimum. Analytic Function 4 can be used to compare the performance of algorithms for finding a global minimum. The performance of algorithms for increasing number of dimensions and average number of evaluations was compared for all four analytic functions. The descriptors here are qualitative assessments based on the performance of a given algorithm in Figures 8.3-8.6 compared to the other algorithms presented. These are order of magnitude assessments rather than rigorous statistics.

8.3.2 Particle Astrophysics Test Problem

Figure 8.7 displays the results for each algorithm for the MSSM7 particle astrophysics test problem detailed in Section 8.1.2. It is immediately clear that Diver by far outperforms all other algorithms, including the PyGMO implementation of differential evolution, albeit with more function evaluations. The reason for this extraordinary performance from Diver may be that the neural network used to create the fast interpolation of the likelihood function was trained on samples taken from Diver. The training data was created totally independently from any optimisation presented here, however it is possible that the neural network encodes patterns that are naturally explored by the Diver algorithm and not others. However, while the other algorithms were not able to find the global minimum of this function, many of them outperformed randomly sampling the space. The two Bayesian optimisation algorithms that I worked on performed poorly compared to other algorithms tested here, but performed comparably to random sampling. This is explained by the high dimensionality of the problem. In the analytic test function examples it was shown that as the dimensionality increased, the performance of the Bayesian optimisation algorithms was especially affected. AMPGO is the only algorithm of the group which performs consistently worse than random sampling, as was generally seen in the analytic functions.

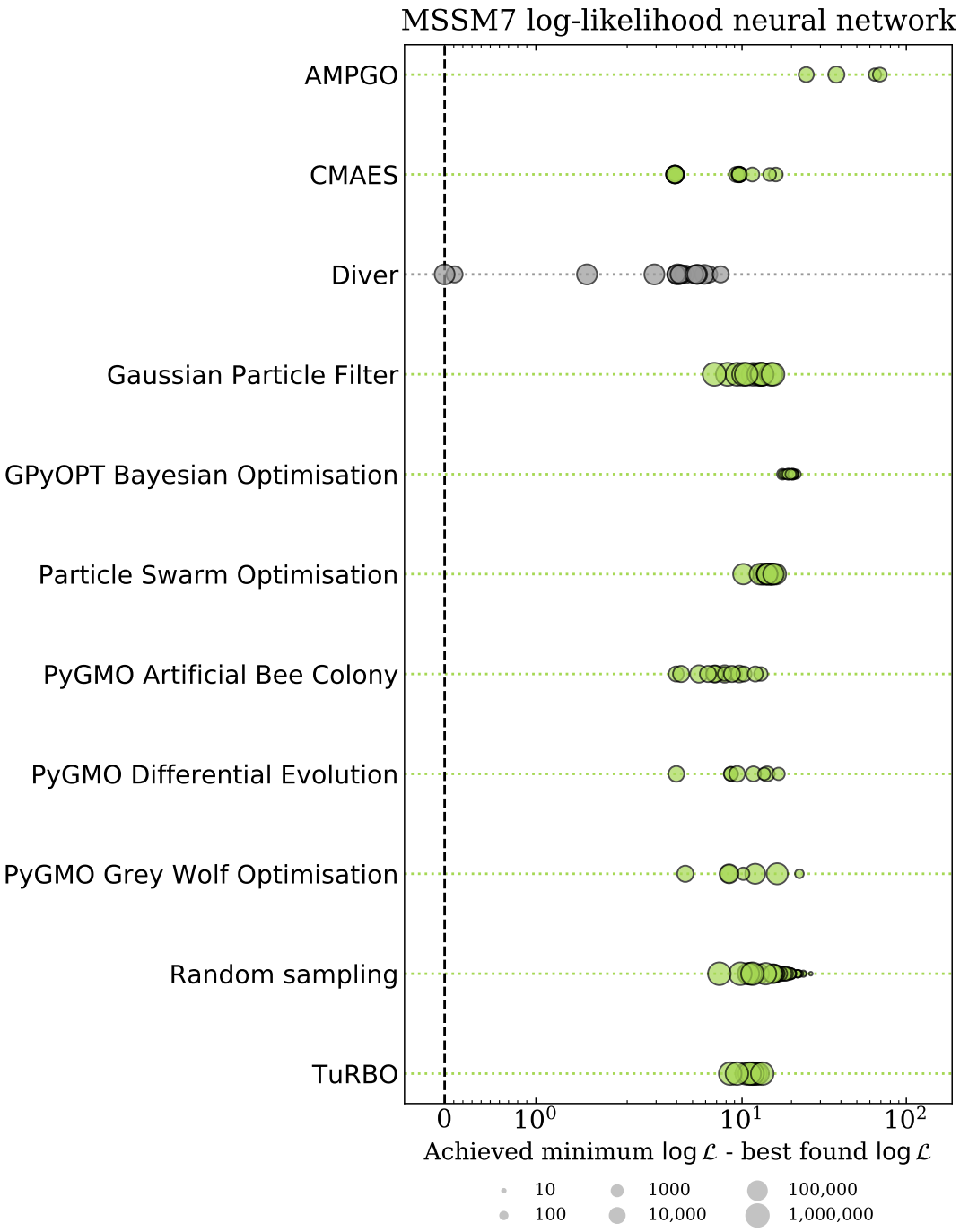


FIGURE 8.7: Results from different optimisation algorithms on the neural network approximation of the MSSM7 log-likelihood described in Section 8.1.2. The results are shown as semi-opaque circles, of which the area increases logarithmically with the number of function evaluations needed to obtain that specific result. The horizontal axis shows the difference between the log-likelihood at the found minimum and the deepest minimum found by any algorithm for any settings. To emphasise the possible bias in the test function towards Diver, the results for that algorithm are coloured differently.

8.4 Best Found Results and Parameter Settings

In this section I briefly review the performance, and optimal hyper-parameters for each of the algorithms I worked on applied to the various analytic functions and the MSSM7 particle astrophysics problem. Tables 8.3-8.6 display the minima found by each algorithm, as well as the parameters that were used to find this optimal value, and the number of function evaluations used to find said value. Here the best result is that which has the lowest function value. If multiple samples found the same value, the result that required the lowest number of function of evaluations is taken. Of note is that Diver tends to work best for the smallest value of the convergence threshold, `convthresh`=0.0001. Notable exceptions being for the 3-D case of analytic function 1, and the 3-D and 7-D cases of analytic function 3, where a convergence threshold of 0.1 performed best. It also tends to prefer a higher resolution value, the notable exception being analytic function 4, where it successfully found the minimum with a rather low resolution value in all dimensions. GPyOPT generally works best for smaller convergence thresholds, notable exceptions being the the 3-D case of analytic function 3, and the 3-D case of analytic function 4, where it successfully finds the minimum with a rather high value of `eps`. This is the only such case, as in all other cases where GPyOPT locates the correct minimum, `eps` was ≤ 0.0001. GPyOPT also consistently uses far fewer function evaluations than the others, as expected. Generally the number of evaluations is $\mathcal{O}(100)$ rather than $> \mathcal{O}(10^5)$ as for the others. TuRBO unsurprisingly tends to work best for the highest number of allowed function evaluations, `max_eval`=100.

Table 8.7 displays the minima found by each algorithm, as well as the parameters that were used to find this optimal value, and the number of function evaluations used to find said value. The trends observed on the analytic functions are consistent with what is observed in this case. Diver performs best with a small convergence threshold, and a high resolution. GPyOPT works best with a small convergence parameter and uses a very small number of function evaluations, though it does not find a very good minimum. TuRBO performs best with a higher number of maximum function evaluations.

Algorithm	dim	Parameters	min	N_{eval}
Diver	2	convthresh=0.0001, np=5000	-0.998	65000
	3	convthresh=0.1, np=10000	-0.735	110000
	5	convthresh=0.001, np=2000	0.0	22000
	7	convthresh=0.1, np=2000	0.0	22000
GPyOPT	2	eps=0.1	0.0	511
	3	eps=0.0001	0.0	425
	5	eps=0.01	0.0	345
	7	eps=0.001	0.0	456
TuRBO	2	max_eval=100	-1.0	1001876
	3	max_eval=100	-0.917	1052097
	5	max_eval=64	0.0	650000
	7	max_eval=64	0.0	650000

TABLE 8.3: Best obtained result for Analytic Function 1 (Equation 8.1). The 'best' result is the result with the lowest found function value. If multiple samples found the same value, the result with the lowest number of needed function evaluations is shown.

Algorithm	dim	Parameters	min	N_{eval}
Diver	2	convthresh=0.0001, np=10000	0.0	380000
	3	convthresh=0.0001, np=20000	0.0	1040000
	5	convthresh=0.0001, np=20000	0.0	1600000
	7	convthresh=0.0001, np=20000	0.0	2000000
GPyOPT	2	eps=0.0001	0.002	754
	3	eps=0.0001	1.265	693
	5	eps=1e-06	5.284	587
	7	eps=0.001	5.829	797
TuRBO	2	max_eval=100	0.0	1000584
	3	max_eval=100	0.027	1050660
	5	max_eval=100	2.044	1050025
	7	max_eval=100	12.101	1050007

TABLE 8.4: Best obtained result for Analytic Function 2 (Equation 8.2). The 'best' result is the result with the lowest found function value. If multiple samples found the same value, the result with the lowest number of needed function evaluations is shown.

Algorithm	dim	Parameters	min	N_{eval}
Diver	2	convthresh=0.0001, np=10000	-1.0	210000
	3	convthresh=0.1, np=20000	-1.0	220000
	5	convthresh=0.0001, np=20000	-0.998	420000
	7	convthresh=0.1, np=10000	-0.984	110000
GPyOPT	2	eps=1e-06	-1.0	636
	3	eps=0.01	-1.0	623
	5	eps=1e-06	-1.0	635
	7	eps=1e-05	-1.0	675
TuRBO	2	max_eval=64	-1.0	700000
	3	max_eval=100	-1.0	1050981
	5	max_eval=100	-1.0	1050025
	7	max_eval=100	-0.998	1050000

TABLE 8.5: Best obtained result for Analytic Function 3 (Equation 8.3). The 'best' result is the result with the lowest found function value. If multiple samples found the same value, the result with the lowest number of needed function evaluations is shown.

Algorithm	dim	Parameters	min	N_{eval}
Diver	2	convthresh=0.0001, np=2000	-0.0	64000
	3	convthresh=0.0001, np=2000	-0.0	88000
	5	convthresh=0.0001, np=5000	-0.0	315000
	7	convthresh=0.0001, np=5000	-0.0	365000
GPyOPT	2	eps=0.0001	0.017	651
	3	eps=0.01	0.062	852
	5	eps=1e-05	53.532	954
	7	eps=0.0001	928.87	1011
TuRBO	2	max_eval=100	0.0	1000636
	3	max_eval=100	0.034	1050318
	5	max_eval=100	0.395	1050010
	7	max_eval=100	178.766	1050000

TABLE 8.6: Best obtained result for Analytic Function 4 (Equation 8.4). The 'best' result is the result with the lowest found function value. If multiple samples found the same value, the result with the lowest number of needed function evaluations is shown.

Algorithm	**Parameters**	**min**	$\mathbf{N}_{\text{eval}}$
Diver	convthresh=0.0001, np=20000	238.214	200000
GPyOPT	eps=0.0001	255.827	684
TuRBO	max_evals=100	246.688	1000000

TABLE 8.7: Best obtained result for the approximation of the 12-dimensional MSSM7 log-likelihood described in Section 8.1.2. The "best" result is the result with the lowest found function value. If multiple samples found the same value, the result with the lowest number of needed function evaluations is shown.

8.5 Conclusion

In this chapter a variety of optimisation algorithms were explored in an effort to find new algorithms for use in particle astrophysics problems. Many of these algorithms have not been used in a particle astrophysics context before, and so exploring their applicability to such a task is a novel endeavour. Each algorithm was tested on a handful of hidden analytic test functions, detailed in Section 8.1.1, and then applied to a particle astrophysics problem, detailed in Section 8.1.2. The algorithms which I worked on are two Bayesian optimisation algorithms called GPyOPT, and TuRBO, as well as a differential evolution algorithm called Diver. The other algorithms that were investigated are PyGMO Differential Evolution, Particle Swarm Optimisation, Covariance Matrix Adaptation Evolution Strategy, Grey Wolf Optimisation, PyGMO Artificial Bee Colony, Gaussian Particle Filter, and AMPGO. Each of these algorithms has a publicly available software implementation. For each algorithm the hyper parameters were characterised as controlling either the convergence of the algorithm, the resolution of the algorithm, providing hints, or improving reliability. Each algorithm was then run with a variety of these hyper-parameters on a handful of hidden analytic test functions, and on a custom implementation of the MSSM7 likelihood function. While it cannot definitively be said that any given algorithm was the "best", we can come to some interesting conclusions.

- Algorithms that perform consistently well on the analytic test functions do not necessarily generalise well to the MSSM7 likelihood function. For example, the PyGMO Artificial Bee Colony consistently performed very well on the analytic functions, however it was outperformed by both differential evolution implementations, notably Diver. However this may be due to an implicit bias towards Diver inherent in the training of the neural network.

- Both differential evolution algorithms performed consistently well across all examples.

- AMPGO consistently performed poorly on every test example, being outperformed by simple random sampling in almost all cases.

- Both Bayesian optimisation algorithms perform well for functions with many global minima, however they struggle in cases with very sharply peaked minima, or with many local minima. High dimensional functions also prove difficult for both Bayesian optimisation algorithms to navigate. TuRBO consistently performs better than GPyOPT as expected due to it performing many optimisations in different hyper-rectangles. This comes at the price of requiring many more function evaluations.

Many of the algorithms detailed here show strong results in both the analytic test functions and the particle astrophysics problem. These results suggest that the use of these algorithms in future real world particle astrophysics problems is well-motivated.

9 Summary

Throughout this thesis I have explored the state of modern particle physics, collider experiments, machine learning, unsupervised anomaly detection, high dimensional optimisation, and dimensional reduction. Modern collider experiments have been very successful in excluding regions of the BSM parameter space but have not found any strong evidence of physics beyond the Standard Model. Using cutting edge data analysis algorithms I have developed a number of techniques to aid in the discovery of new physics, from identifying interesting regions of parameter space, to visualisation of MSSM models.

In Chapter 5 I present an unsupervised anomaly detection algorithm designed to distinguish an anomalous BSM signal from the SM background. In Figure 5.17 I confirm that training algorithms in the latent space of a VAE dramatically improves the performance, and that by combining the anomaly scores, one can draw out better discriminating power. In Chapter 6 I further hone my algorithm, and test it on a wide variety of SUSY and non-SUSY BSM signals. I perform a detailed analysis of different hyperparameters for the VAE using many metrics, and identify a set of hyperparameters which consistently yield the best results across all tested signals. These results are displayed in Figure 6.11.

In Chapter 7 I project 4-dimensional MSSM model parameters onto a 2-D plane in order to optimise on a more representative selection of models. I pick four unexcluded models, two based on the simplified models that are typically examined at the LHC, and two non-simplified models which are only visible due to the representative nature of the model selection. From there, I construct analyses able to exclude each one, showing that this method can aid in illuminating interesting regions of the parameter space without constraining the models to hyperplanes within the total parameter space.

In Chapter 8 I test a number of high dimensional optimisation algorithms on a series of analytic test functions as well as a particle astrophysics problem. In Table 8.2 I present the strengths and weaknesses of each algorithm for a variety of criteria. For the algorithms that I personally worked on, I present a detailed analysis of the results in Tables 8.3 to 8.7.

Modern particle physics has entered an interesting period where the next major discovery will likely come from a theory for which the details are not known. In this thesis I have presented model agnostic techniques which are able to provide strong discriminating power with very minimal signal assumptions. I have explored high dimensional sampling techniques which have been shown to be able to identify BSM models fitting closest with experimental observations. Finally, I have tested dimensional reduction techniques which

allow one to capture the entire behaviour of the parameter space, and examine BSM models in a way that is more representative of the entire parameter space. These novel data analysis techniques have been shown to be valid tools for use in the search for physics beyond the Standard Model.

Bibliography

[1] Peskin, Michael E and Schroeder, Daniel V. *An Introduction To Quantum Field Theory (Frontiers in Physics)*. Westview Press Incorporated, 1995.

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

[3] Ian JR Aitchison and Anthony JG Hey. *Gauge Theories in Particle Physics: A Practical Introduction, -2 Volume set*. Taylor & Francis, 2012.

[4] Fei Gao, Chong-yao Chen, and Yu-xin Liu. *Colour Confinement: a Dynamical Phenomenon of QCD*. 2018. DOI: 10.48550/ARXIV.1802.08184. URL: https://arxiv.org/abs/1802.08184.

[5] James Clerk Maxwell. *A treatise on electricity and magnetism*. Vol. 1. Clarendon press, 1873.

[6] H David Politzer. "Reliable perturbative results for strong interactions?" In: *Physical Review Letters* 30.26 (1973), p. 1346.

[7] David J Gross and Frank Wilczek. "Ultraviolet behavior of non-abelian gauge theories". In: *Physical Review Letters* 30.26 (1973), p. 1343.

[8] Steven Weinberg. "A model of leptons". In: *Physical review letters* 19.21 (1967), p. 1264.

[9] Langacker, Paul. *The Standard Model and beyond*. CRC press, 2017.

[10] Martin, Stephen P. "A supersymmetry primer". In: *Perspectives on supersymmetry II*. World Scientific, 2010, 1–153.

[11] Serguei Chatrchyan et al. "Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC". In: *Physics Letters B* 716.1 (2012), pp. 30–61.

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

[13] Katherine Garrett and Gintaras Duda. "Dark matter: A primer". In: *Advances in Astronomy* 2011 (2011).

[14] Edvige Corbelli and Paolo Salucci. "The extended rotation curve and the dark matter halo of M33". In: *Monthly Notices of the Royal Astronomical Society* 311.2 (2000), pp. 441–447.

[15] Peter Schneider. "Gravitational lensing statistics". In: *Gravitational Lenses*. Springer, 1992, pp. 196–208.

[16] Priyamvada Natarajan et al. "Mapping substructure in the HST Frontier Fields cluster lenses and in cosmological simulations". In: *Monthly Notices of the Royal Astronomical Society* 468.2 (2017), pp. 1962–1980.

[17] Massimo Giovannini. *A primer on the physics of the cosmic microwave background*. World Scientific, 2008.

[18] Lars Bergström and Ariel Goobar. *The Cosmic Microwave Background Radiation and Growth of Structure*. Springer, Berlin, Heidelberg, 2004.

[19] E. Aprile et al. "The XENON1T dark matter experiment". In: *The European Physical Journal C* 77.12 (2017). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-017-5326-3. URL: http://dx.doi.org/10.1140/epjc/s10052-017-5326-3.

[20] R. Adam et al. "Planck2015 results". In: *Astronomy & Astrophysics* 594 (2016), A1. ISSN: 1432-0746. DOI: 10.1051/0004-6361/201527101. URL: http://dx.doi.org/10.1051/0004-6361/201527101.

[21] The ATLAS Collaboration. *SUSY June 2021 Summary Plot Update*. https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-PHYS-PUB-2021-019/. 2021.

[22] *SUSY Summary Plots March 2022*. Tech. rep. Geneva: CERN, 2022. URL: http://cds.cern.ch/record/2805985.

[23] M.C. Carmona-Benitez et al. "First Results of the LUX Dark Matter Experiment". In: *Nuclear and Particle Physics Proceedings* 273-275 (2016). 37th International Conference on High Energy Physics (ICHEP), pp. 309–313. ISSN: 2405-6014. DOI: https://doi.org/10.1016/j.nuclphysbps.2015.09.043. URL: https://www.sciencedirect.com/science/article/pii/S2405601415005325.

[24] E. Aprile et al. "First Dark Matter Results from the XENON100 Experiment". In: *Phys. Rev. Lett.* 105 (13 2010), p. 131302. DOI: 10.1103/PhysRevLett.105.131302. URL: https://link.aps.org/doi/10.1103/PhysRevLett.105.131302.

[25] P. Picozza et al. "PAMELA – A payload for antimatter matter exploration and light-nuclei astrophysics". In: *Astroparticle Physics* 27.4 (2007), pp. 296–315. ISSN: 0927-6505. DOI: https://doi.org/10.1016/j.astropartphys.2006.12.002. URL: https://www.sciencedirect.com/science/article/pii/S0927650506001861.

[26] F. Barao. "AMS—Alpha Magnetic Spectrometer on the International Space Station". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 535.1 (2004). Proceedings of the 10th International Vienna Conference on Instrumentation, pp. 134–138. ISSN: 0168-9002. DOI: https://doi.org/10.1016/j.nima.2004.07.196. URL: https://www.sciencedirect.com/science/article/pii/S0168900204015888.

[27] M.G. Aartsen et al. "The IceCube Neutrino Observatory: instrumentation and online systems". In: *Journal of Instrumentation* 12.03 (2017), P03012–P03012. ISSN: 1748-0221. DOI: 10.1088/1748-0221/12/03/p03012. URL: http://dx.doi.org/10.1088/1748-0221/12/03/P03012.

[28] M. Ackermann et al. "THE FERMI LARGE AREA TELESCOPE ON ORBIT: EVENT CLASSIFICATION, INSTRUMENT RESPONSE FUNCTIONS, AND CALIBRATION". In: *The Astrophysical Journal Supplement Series* 203.1 (2012), p. 4. ISSN: 1538-4365. DOI: 10.1088/0067-0049/203/1/4. URL: http://dx.doi.org/10.1088/0067-0049/203/1/4.

[29] Lyndon Evans and Philip Bryant. "LHC Machine". In: *Journal of Instrumentation* 3.08 (2008), S08001–S08001. DOI: 10.1088/1748-0221/3/08/s08001. URL: https://doi.org/10.1088/1748-0221/3/08/s08001.

[30] The ATLAS Collaboration et al. "The ATLAS Experiment at the CERN Large Hadron Collider". In: *Journal of Instrumentation* 3.08 (2008), S08003–S08003. DOI: 10.1088/1748-0221/3/08/s08003. URL: https://doi.org/10.1088%2F1748-0221%2F3%2F08%2Fs08003.

[31] CMS Collaboration et al. *The CMS experiment at the CERN LHC*. 2008.

[32] Kenneth Aamodt et al. "The ALICE experiment at the CERN LHC". In: *Journal of Instrumentation* 3.08 (2008), S08002.

[33] A Augusto Alves Jr et al. "The LHCb detector at the LHC". In: *Journal of instrumentation* 3.08 (2008), S08005.

[34] J. Alwall et al. "The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations". In: *JHEP* 07 (2014), p. 079. DOI: 10.1007/JHEP07(2014)079. arXiv: 1405.0301 [hep-ph].

[35] Torbjörn Sjöstrand et al. "An Introduction to PYTHIA 8.2". In: *Comput. Phys. Commun.* 191 (2015), pp. 159–177. DOI: 10.1016/j.cpc.2015.01.024. arXiv: 1410.3012 [hep-ph].

[36] J. de Favereau et al. "DELPHES 3, A modular framework for fast simulation of a generic collider experiment". In: *JHEP* 02 (2014), p. 057. DOI: 10.1007/JHEP02(2014)057. arXiv: 1307.6346 [hep-ex].

[37] *Standard Model Summary Plots Spring 2020*. Tech. rep. Geneva: CERN, 2020. URL: https://cds.cern.ch/record/2718937.

[38] Christopher G Lester and Alan J Barr. "MTGEN: Mass scale measurements in pair-production at colliders". In: *Journal of High Energy Physics* 2007.12 (2007), p. 102.

[39] David JC MacKay, David JC Mac Kay, et al. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[40] Tom M Mitchell et al. *Machine learning*. 1997.

[41] John R Koza et al. "Automated design of both the topology and sizing of analog electrical circuits using genetic programming". In: *Artificial Intelligence in Design'96*. Springer, 1996, pp. 151–170.

[42] Jaime G Carbonell, Ryszard S Michalski, and Tom M Mitchell. "An overview of machine learning". In: *Machine learning* (1983), pp. 3–23.

[43] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. 1999.

[44] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. URL: http://yann.lecun.com/exdb/mnist/.

[45] Kishan G Mehrotra, Chilukuri K Mohan, and HuaMing Huang. *Anomaly detection principles and algorithms*. Vol. 1. Springer, 2017.

[46] John J Hopfield. "Neural networks and physical systems with emergent collective computational abilities". In: *Proceedings of the national academy of sciences* 79.8 (1982), pp. 2554–2558.

[47] Pascal Vincent et al. "Extracting and Composing Robust Features with Denoising Autoencoders". In: *Proceedings of the 25th International Conference on Machine Learning*. ICML '08. Helsinki, Finland: Association for Computing Machinery, 2008, 1096–1103. ISBN: 9781605582054. DOI: 10.1145/1390156.1390294. URL: https://doi.org/10.1145/1390156.1390294.

[48] Diederik P Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: (2013). cite arxiv:1312.6114. URL: http://arxiv.org/abs/1312.6114.

[49] S. Kullback and R. A. Leibler. "On Information and Sufficiency". In: *The Annals of Mathematical Statistics* 22.1 (1951), pp. 79 –86. DOI: 10.1214/aoms/1177729694. URL: https://doi.org/10.1214/aoms/1177729694.

[50] Melissa van Beekveld et al. "Combining outlier analysis algorithms to identify new physics at the LHC". In: *Journal of High Energy Physics* 2021.9 (2021), pp. 1–33.

[51] Gentleman, R and Carey, VJ. "Unsupervised machine learning". In: *Bioconductor Case Studies*. Springer, 2008, 137–157.

[52] G. Aad et al. "Dijet Resonance Search with Weak Supervision Using $\sqrt{s} = 13$ TeV pp Collisions in the ATLAS Detector". In: *Phys. Rev. Lett.* 125 (13 2020), p. 131801. DOI: 10.1103/PhysRevLett.125.131801. URL: https://link.aps.org/doi/10.1103/PhysRevLett.125.131801.

[53] G. Brooijmans et al. "Les Houches 2019 Physics at TeV Colliders: New Physics Working Group Report". In: *11th Les Houches Workshop on Physics at TeV Colliders: PhysTeV Les Houches*. Feb. 2020. arXiv: 2002.12220 [hep-ph].

[54] Andy Buckley et al. "LHAPDF6: parton density access in the LHC precision era". In: *Eur. Phys. J. C* 75 (2015), p. 132. DOI: 10.1140/epjc/s10052-015-3318-8. arXiv: 1412.7420 [hep-ph].

[55] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. "FastJet User Manual". In: *Eur. Phys. J.* C72 (2012), p. 1896. DOI: 10.1140/epjc/s10052-012-1896-2. arXiv: 1111.6097 [hep-ph].

[56] "Expected performance of the ATLAS b-tagging algorithms in Run-2". In: (July 2015).

[57] Liu, Fei Tony and Ting, Kai Ming and Zhou, Zhi-Hua. "Isolation forest". In: *2008 Eighth IEEE International Conference on Data Mining*. IEEE. 2008, 413–422.

[58] Douglas Reynolds. "Gaussian Mixture Models". In: *Encyclopedia of Biometrics*. Ed. by Stan Z. Li and Anil Jain. Boston, MA: Springer US, 2009, pp. 659–663. ISBN: 978-0-387-73003-5. DOI: 10.1007/978-0-387-73003-5_196. URL: https://doi.org/10.1007/978-0-387-73003-5_196.

[59] Edward James McShane. "Jensen's inequality". In: *Bulletin of the American Mathematical Society* 43.8 (1937), pp. 521–527.

[60] Mayu Sakurada and Takehisa Yairi. "Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction". In: *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*. MLSDA'14. Gold Coast, Australia QLD, Australia: Association for Computing Machinery, 2014, 4–11. ISBN: 9781450331593. DOI: 10.1145/2689746.2689747. URL: https://doi.org/10.1145/2689746.2689747.

[61] Jan Hajer et al. "Novelty detection meets collider physics". In: *arXiv preprint arXiv:1807.10261* (2018).

[62] Soheil Kolouri et al. "Sliced-Wasserstein autoencoder: an embarrassingly simple generative model". In: *arXiv preprint arXiv:1804.01947* (2018).

[63] Olmo Cerri et al. "Variational Autoencoders for New Physics Mining at the Large Hadron Collider". In: *JHEP* 05 (2019), p. 036. DOI: 10.1007/JHEP05(2019)036. arXiv: 1811.10276 [hep-ex].

[64] Solomon Kullback. *Information Theory and Statistics*. New York: Wiley, 1959.

[65] Kevin P. Murphy. *Machine learning : a probabilistic perspective*. Cambridge, Mass. [u.a.]: MIT Press, 2013. ISBN: 9780262018029 0262018020. URL: https://www.amazon.com/Machine-Learning-Probabilistic-Perspective-Computation/dp/0262018020/ref=sr_1_2?ie=UTF8&qid=1336857747&sr=8-2.

[66] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. *Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)*. 2016. arXiv: 1511.07289 [cs.LG].

[67] Thea Aarrestad et al. "The Dark Machines Anomaly Score Challenge: Benchmark Data and Model Independent Event Classification for the Large Hadron Collider". In: *SciPost Physics* 12.1 (Jan. 2022). DOI: 10.21468/SciPostPhys.12.1.043.

[68] The Dark Machines Collaboration. *Dark Machines*. URL: https://darkmachines.org/.

[69] Lorenzo Basso et al. "Phenomenology of the minimal B-L extension of the Standard model: Z' and neutrinos". In: *Phys. Rev. D* 80 (2009), p. 055030. DOI: 10.1103/PhysRevD.80.055030. arXiv: 0812.4313 [hep-ph].

[70] Frank F. Deppisch, Wei Liu, and Manimala Mitra. "Long-lived Heavy Neutrinos from Higgs Decays". In: *JHEP* 08 (2018), p. 181. DOI: 10.1007/JHEP08(2018)181. arXiv: 1804.04075 [hep-ph].

[71] S. Amrith et al. "LHC Constraints on a $B - L$ Gauge Model using Contur". In: *JHEP* 05 (2019), p. 154. DOI: 10.1007/JHEP05(2019)154. arXiv: 1811.11452 [hep-ph].

[72] Paul Adrien Maurice Dirac. "The quantum theory of the electron". In: *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 117.778 (1928), pp. 610–624.

[73] X. G. He et al. "New-Z' phenomenology". In: *Phys. Rev. D* 43 (1 1991), R22–R24. DOI: 10.1103/PhysRevD.43.R22. URL: https://link.aps.org/doi/10.1103/PhysRevD.43.R22.

[74] Xiao-Gang He et al. "Simplest Z' model". In: *Phys. Rev. D* 44 (7 1991), pp. 2118–2132. DOI: 10.1103/PhysRevD.44.2118. URL: https://link.aps.org/doi/10.1103/PhysRevD.44.2118.

[75] R. Barbier et al. "R-parity violating supersymmetry". In: *Phys. Rept.* 420 (2005), pp. 1–202. DOI: 10.1016/j.physrep.2005.08.006. arXiv: hep-ph/0406039.

[76] Benjamin Fuks. "Beyond the Minimal Supersymmetric Standard Model: from theory to phenomenology". In: *Int. J. Mod. Phys. A* 27 (2012), p. 1230007. DOI: 10.1142/S0217751X12300074. arXiv: 1202.4769 [hep-ph].

[77] Janusz Rosiek. "Complete set of Feynman rules for the minimal supersymmetric extension of the standard model". In: *Phys. Rev. D* 41 (11 1990), pp. 3464–3501. DOI: 10.1103/PhysRevD.41.3464. URL: https://link.aps.org/doi/10.1103/PhysRevD.41.3464.

[78] H.E. Haber and G.L. Kane. "The search for supersymmetry: Probing physics beyond the standard model". In: *Physics Reports* 117.2 (1985), pp. 75–263. ISSN: 0370-1573. DOI: https://doi.org/10.1016/0370-1573(85)90051-1. URL: https://www.sciencedirect.com/science/article/pii/0370157385900511.

[79] H.P. Nilles. "Supersymmetry, supergravity and particle physics". In: *Physics Reports* 110.1 (1984), pp. 1–162. ISSN: 0370-1573. DOI: https://doi.org/10.1016/0370-1573(84)90008-5. URL: https://www.sciencedirect.com/science/article/pii/0370157384900085.

[80] Murray Rosenblatt. "Remarks on Some Nonparametric Estimates of a Density Function". In: *The Annals of Mathematical Statistics* 27.3 (1956), pp. 832 –837. DOI: 10.1214/aoms/1177728190. URL: https://doi.org/10.1214/aoms/1177728190.

[81] Rob Verheyen and Bob Stienen. *Phase Space Sampling and Inference from Weighted Events with Autoregressive Flows*. 2020. arXiv: 2011.13445 [hep-ph].

[82] Mark A Kramer. "Nonlinear principal component analysis using autoassociative neural networks". In: *AIChE journal* 37.2 (1991), pp. 233–243.

[83] Ian J. Goodfellow et al. *Generative Adversarial Networks*. 2014. arXiv: 1406.2661 [stat.ML].

[84] James MacQueen et al. "Some methods for classification and analysis of multivariate observations". In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297.

[85] David Arthur and Sergei Vassilvitskii. *k-means++: The advantages of careful seeding*. Tech. rep. Stanford, 2006.

[86] Sascha Caron, Luc Hendriks, and Rob Verheyen. "Rare and Different: Anomaly Scores from a combination of likelihood and out-of-distribution models to detect new physics at the LHC". In: *SciPost Physics* 12.2 (2022). DOI: 10.21468/scipostphys.12.2.077. URL: https://doi.org/10.21468%2Fscipostphys.12.2.077.

[87] Peter Athron et al. "GAMBIT: the global and modular beyond-the-standard-model inference tool". In: *The European Physical Journal C* 77.11 (2017). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-017-5321-8. URL: http://dx.doi.org/10.1140/epjc/s10052-017-5321-8.

[88] Peter Athron et al. "Combined collider constraints on neutralinos and charginos". In: *Eur. Phys. J. C* 79.5 (2019), p. 395. DOI: 10.1140/epjc/s10052-019-6837-x. arXiv: 1809.02097 [hep-ph].

[89] W. Beenakker, R. Hopker, and M. Spira. *PROSPINO: A Program for the Production of Supersymmetric Particles in Next-to-leading Order QCD*. Tech. rep. 12 pages, latex, no figures, Complete postscript file and FORTRAN source codes available from http://wwwcn.cern.ch/ mspira/prospino/. 1996. URL: https://cds.cern.ch/record/314229.

[90] A. Djouadi, M. M. Muhlleitner, and M. Spira. "Decays of Supersymmetric Particles: the program SUSY-HIT (SUspect-SdecaY-Hdecay-InTerface)". In: (2006). DOI: 10.48550/ARXIV.HEP-PH/0609292. URL: https://arxiv.org/abs/hep-ph/0609292.

[91] G Aad et al. "Search for chargino–neutralino pair production in final states with three leptons and missing transverse momentum in $\sqrt{s} = 13$ TeV pp collisions with the ATLAS detector". In: *The European Physical Journal C* 81.12 (2021), pp. 1–55.

[92] *Search for physics beyond the standard model in final states with two or three soft leptons and missing transverse momentum in proton-proton collisions at 13 TeV*. Tech. rep. Geneva: CERN, 2021. URL: https://cds.cern.ch/record/2758359.

[93] G. Aad et al. "Search for charginos and neutralinos in final states with two boosted hadronically decaying bosons and missing transverse momentum in *pp* collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector". In: *Phys. Rev. D* 104 (11 2021), p. 112010. DOI: 10.1103/PhysRevD.104.112010. URL: https://link.aps.org/doi/10.1103/PhysRevD.104.112010.

[94] Rene Brun et al. *root-project/root: v6.24/02*. Version v6-24-02. Aug. 2019. DOI: 10.5281/zenodo.3895860. URL: https://doi.org/10.5281/zenodo.3895860.

[95] Robert D. Cousins. *What is the likelihood function, and how is it used in particle physics?* 2020. arXiv: 2010.00356 [physics.data-an].

[96] Csaba Balázs et al. "A comparison of optimisation algorithms for high-dimensional particle and astrophysics applications". In: *Journal of High Energy Physics* 2021.5 (2021), pp. 1–46.

[97] Peter Athron et al. "A global fit of the MSSM with GAMBIT". In: *The European Physical Journal C* 77.12 (2017). ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-017-5196-8. URL: http://dx.doi.org/10.1140/epjc/s10052-017-5196-8.

[98] A. Buckley, A. Shilton, and M.J. White. "Fast supersymmetry phenomenology at the Large Hadron Collider using machine learning techniques". In: *Computer Physics Communications* 183.4 (2012), 960–970. ISSN: 0010-4655. DOI: 10.1016/j.cpc.2011.12.026. URL: http://dx.doi.org/10.1016/j.cpc.2011.12.026.

[99] Jonas Močkus. "On Bayesian methods for seeking the extremum". In: *Optimization techniques IFIP technical conference*. Springer. 1975, pp. 400–404.

[100] David Eriksson et al. "Scalable global optimization via local bayesian optimization". In: *Advances in Neural Information Processing Systems* 32 (2019).

[101] Rainer Storn. "On the usage of differential evolution for function optimization". In: *Proceedings of north american fuzzy information processing*. Ieee. 1996, pp. 519–523.

[102] Mohammad Reza Bonyadi and Zbigniew Michalewicz. "Particle swarm optimization for single objective continuous space problems: a review". In: *Evolutionary computation* 25.1 (2017), pp. 1–54.

[103] Nikolaus Hansen and Anne Auger. "Principled design of continuous stochastic search: From theory to practice". In: *Theory and principled methods for the design of metaheuristics*. Springer, 2014, pp. 145–180.

[104] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. "Grey Wolf Optimizer". In: *Advances in Engineering Software* 69 (2014), pp. 46–61. ISSN: 0965-9978. DOI: https://doi.org/10.1016/j.advengsoft.2013.12.007. URL: https://www.sciencedirect.com/science/article/pii/S0965997813001853.

[105] Dervis Karaboga et al. *An idea based on honey bee swarm for numerical optimization*. Tech. rep. Technical report-tr06, Erciyes university, engineering faculty, 2005.

[106] J.H. Kotecha and P.M. Djuric. "Gaussian particle filtering". In: *IEEE Transactions on Signal Processing* 51.10 (2003), pp. 2592–2601. DOI: 10.1109/TSP.2003.816758.

[107] Leon Lasdon et al. “Adaptive memory programming for constrained global optimization”. In: *Computers & Operations Research* 37.8 (2010). Operations Research and Data Mining in Biological Systems, pp. 1500–1509. ISSN: 0305-0548. DOI: https://doi.org/10.1016/j.cor.2009.11.006. URL: https://www.sciencedirect.com/science/article/pii/S0305054809002937.

[108] Yangkun Xia et al. “Tabu search algorithm for the distance-constrained vehicle routing problem with split deliveries by order”. In: *PloS one* 13.5 (2018), e0195457.