Searching for new physics signatures in hadronic final states at the Large Hadron Collider with deep learning

A thesis submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

by

Vishal Singh Ngairangbam

(Roll No. 17330040)

Under the guidance of

Dr. Partha Konar

Professor

Theoretical Physics Division

Physical Research Laboratory, Ahmedabad, India.

DISCIPLINE OF PHYSICS

INDIAN INSTITUTE OF TECHNOLOGY GANDHINAGAR

2022

my family
&
teachers

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)

(Name: Vishal Singh Ngairangbam)

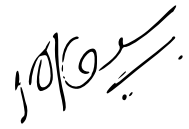
(Roll No: 17330040)

Date: 30/05/2022

Letter of Endorsement

It is certified that the thesis titled "**Searching for new physics signatures in hadronic final states at the Large Hadron Collider with deep learning"** by **Vishal Singh Ngairangbam** (Roll no. **17330040**), has gone through all necessary changes based on all the suggestions made by the three thesis examiners. These corrections and clarifications are minor in nature. Vishal has also attached a point by point reply to reviewers' comments.

I have read this dissertation and in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.



Aug 26, 2022

Dr. Partha Konar (Thesis Supervisor)

Professor, Theoretical Physics Division

Physical Research Laboratory

Ahmedabad, Gujarat - 380 009

Thesis Approval

The thesis titled

"Searching for new physics signatures in hadronic final states at the Large Hadron Collider with deep learning"

by

Vishal Singh Ngairangbam

(Roll No. 17330040)

is approved for the degree of Doctor of Philosophy

Prof. Santosh Kumar Rai Harish-Chandra Research Institute, Prayagraj (External Examiner)	**Prof. Namit Mahajan** Physical Research Laboratory, Ahmedabad (Examiner)	**Prof. Subhendra Mohanty** Physical Research Laboratory, Ahmedabad (Examiner)
Prof. Aveek Sarkar Physical Research Laboratory, Ahmedabad (Examiner)	**Prof. Baradhwaj Coleppa** Indian Institute of Technology, Gandhinagar (Examiner)	**Prof. Partha Konar** Physical Research Laboratory, Ahmedabad (Supervisor)

Date: 12 September 2022 Place: Ahmedabad

List of Publications

I. Publications contributing to this thesis:

1. Influence of QCD parton shower in deep learning invisible Higgs through vector boson fusion.
   P. Konar, and **V. S. Ngairangbam**.
   Phys.Rev.D 105 (2022) 11, 113003 [arxiv:2201.01040]

2. Anomaly detection in high-energy physics using a quantum autoencoder.
   V. S. Ngairangbam, M. Spannowsky, and M. Takeuchi.
   Phys. Rev. D 105, 095004 [arxiv:2112.04958]
   Code availaible at: https://gitlab.com/vishalng/qae_hep

3. Energy-weighted Message Passing: an infra-red and collinear safe graph neural network algorithm.
   P. Konar, **V. S. Ngairangbam**, and M. Spannowsky.
   JHEP 02 (2022) 060 [arxiv:2109.14636]

4. Anomaly detection with Convolutional Graph Neural Networks.
   O. Atkinson, A. Bhardwaj, C. Englert, **V. S. Ngairangbam**, and M. Spannowsky.
   JHEP 08 (2021) 080 [arxiv:2105.07988]

5. Invisible Higgs search through vector boson fusion: a deep learning approach.
   V. S. Ngairangbam, A. Bhardwaj, P. Konar and A. K. Nayak.
   Eur. Phys. J. C 80 (2020) 1055 [arxiv:2008.05434]

II. Other Publications :

1. IRC-safe Graph Autoencoder for an unsupervised anomaly detection.
   O. Atkinson, A. Bhardwaj, C. Englert, Partha Konar, **V. S. Ngairangbam**, and M. Spannowsky.
   Front. Artif. Intell. 5 (2022) 943135 [2204.12231]

2. Deep learning towards Solar flare prediction.
   V. S. Ngairangbam, P. B Domadiya, P. K. Mitra, B. Joshi, R. Bhattacharyya, A. Sarkar, Dhania M B, Manju G, A Kumar, P. Konar
   Published in *"A Compendium of Disruptive & Futuristic Technologies for Space Sector"*, Ed. Dr. T.P. Das & N. Raghu Meetei ISRO Report: Proceedings of the DTDI Technology Conclave - 2021

Acknowledgements

I sincerely thank my supervisor, Prof. Partha Konar, for guiding me in practically every aspect of academic research in the past five years. Not only has he taught me the wonderful area of LHC phenomenology, but he also introduced me to the exciting and multidisciplinary subject of deep learning, motivating and helping me to contribute to an up-and-coming area. He has shaped how I look at the research, encouraging me to persevere in such a competitive and challenging field.

The work completed in this thesis would not have been possible without my collaborators. I thank my collaborators (in alphabetical order): Mr Oliver Atkinson, Dr Akanksha Bhardwaj, Prof. Christoph Englert, Prof. Aruna Kumar Nayak, Prof. Michael Spannowsky, and Dr Michihisa Takeuchi, with whom I have collaborated on exciting projects and took part in various discussions which have shaped the way I look at research in high-energy physics.

I also express my sincere gratitude to my Doctoral Studies Committee: Prof. Namit Mahajan, Prof. Subhendra Mohanty, and Dr Aveek Sarkar. They have always had valuable suggestions about my work, which have brought to my attention many aspects of the work in a generally broader picture, enabling me to form a wider understanding of particle physics. I thank Dr Satyajit Seth for discussing various facets of quantum chromodynamics, helping me understand its more nuanced aspects. These five years of research would not have been possible without the help, support, and guidance of all the theoretical physics division faculty members. I sincerely thank Prof. Dilip Angom, Prof. Srubabati Goswami, Prof. Hiranmaya Mishra, Dr Ketan Patel, and Prof. Navinder Singh for their continuous guidance and motivation throughout my PhD.

I thank my office mates Anupam Ghosh, Dr J. Selvaganapathy, Dr Balbeer Singh, and Sudipta Show for all the time we spent (and not) doing research in the office.

Pursuing such a demanding profession would not have been possible without the love and support of my family. I thank my mother, Dr Premila Chanu, and my father, Dr Narendra Ngairangbam, my sisters, Dr Archana Ngairangbam and Sanatombi Ngairangbam for always supporting me in pursuing research as a viable career path. I thank my wife, Monika, for always supporting me every step of the way and being an unmoving pillar during my PhD. Lastly, nothing prepared me for the love and joy that my daughter Linthoi brought, inspiring me to work harder to achieve my goals in life. I thank my daughter for bringing a newfound joy and enrichment to my life and reminding me that life is not about getting what you want but giving what you can.

(Vishal)

Abstract

The Large Hadron Collider continues to search for signatures of new physics and scrutinise the properties of the Higgs boson. It has accumulated enormous amounts of highly complex high energy collision event data (and will collect even more data in the upcoming high luminosity phase) to pinpoint different parameters with extraordinary precision and constrain classes of new physics models.

The simultaneous development of very powerful deep-learning algorithms presents a genuine opportunity for their application, leveraging their unprecedented power to enhance experimental sensitivities in pursuit of new physics. Unlike many industrial applications, which can be almost entirely data-driven, the application of such deep-learning algorithms to fundamental physics provides unique challenges. The aim is not solely to utilise their superior performance, for instance, in segregating different signals from the background, but also to understand the features they extract, resulting in such an increase. Moreover, rigorous first principle motivation from the underlying physics provides useful priors whose knowledge can be built into architectural design. This thesis investigates some of the applications of deep-learning algorithms to phenomenological searches at the Large Hadron Collider, concentrating mainly on the challenging but abundant hadronic final states.

The Higgs invisible branching ratio, which is pivotal in ruling out many dark-matter-motivated BSM models, is still poorly constrained. The best available limit comes from the vector boson fusion (VBF) channel, which is experimentally challenging. We address the problem of finding the signal as a classification of images as inputs to Convolutional Neural Networks (CNNs) by using the analogy of the detectors to a camera. The energy's spatial distribution essentially forms a picture with the energy deposits as the pixels' values. Using these so-called' Tower Images,' we trained CNNs to identify signal type events from background ones. We improved the upper bounds on the invisible-branching ratio of the Higgs by a factor of three compared to existing methods using the same amount of data.

The differing nature of the QCD radiation pattern is exploited in traditional VBF searches and deep machine learning. Therefore, it is natural to ask: How accurate are the parton-shower models which simulate the predominant radiation patterns? This issue is, in fact, more pronounced in deep learning methods, which look into minute differences in the radiation pattern. The global parton-shower recoil scheme inherently assumes an Initial-Initial colour dipole structure. It fails to correctly produce the wide-angle radiation patterns in a VBF topology with an Initial-Final/Final-Initial dipole structure. Moreover, the next-to-leading order corrections to the tree level VBF process are important in determining the kinematics of the third hardest jet, which would be the dominant information beyond the two-jet system. Therefore, we extend the analysis to explore the robustness of the CNN in identifying VBF Higgs signals to these essential factors

in the signal simulation process. We find that the CNN's performance is more dependent on the recoil scheme when comparing models trained and tested with the same signal simulation. However, the higher perturbative accuracy of a next-to-leading-order matrix element simulation has better stability when tested on models trained on different signal simulations.

The findings of the previous investigation demonstrate that the training of CNNs is dependent on the nature of the data simulation. Infrared and collinear (IRC) safety guarantees the predictability of observables based on hadrons from quantum chromodynamics. It ensures that long-distance non-perturbative hadronic dynamics do not significantly modify observables. In the subsequent study, we devise an IRC safe framework for Graph Neural Networks, a different class of deep-learning algorithms that generalise the favourable properties of CNNs and generalise it to possibly non-Euclidean domains while forgoing the strictly ordered and sparse representation of the tower image. We also found that it performs as well as state-of-the-art IRC unsafe algorithms in identifying boosted hadronic decays of the top quark against QCD jets while being robust to soft and collinear emissions.

With the null results of various well-motivated new physics models, it is important to look at the background-only hypothesis in a model-independent approach. The subsequent studies explore such model-independent anomaly detection techniques.

Although various convolutional autoencoders have been proposed for model-independent anomaly detection methods, it is challenging to design graph autoencoders for inductive graph-based purposes. To utilise the more favourable properties of graph neural networks for unsupervised anomaly detection, we devise a graph autoencoder with the ability to learn inductive graph representations with the help of edge-reconstruction networks. We train the model to efficiently reconstruct the constituent four vectors of large-radius QCD jets in such anomaly detection methods. When trained to reconstruct QCD jets, such models have a high reconstruction error on jets with higher n-prong multiplicities.

In the final study, we explore the power of quantum autoencoders based on variational quantum circuits to detect anomalous signals in a model-independent set-up. Available noisy-intermediate-scale-quantum devices have potential applications in quantum machine learning, where a variational quantum circuit is trained by classical means but utilises the quantum aspects within the circuit implementations. We find that compared to similar bit-based autoencoders with the same inputs, quantum autoencoders have a very efficient training, saturating the test loss reconstruction with as low as ten training samples. Moreover, they perform better than their classical counterparts in a few benchmark signal scenarios.

Contents

List of Abbreviations

2HDM	2-Higgs Doublet Model
AI	Artificial Intelligence
ANN	Artificial Neural Networks
ATLAS	A Toroidal LHC Apparatus
AUC	Area Under the Curve
BR	Branching Ratio
BSM	Beyond the Standard Model
CA	Cambridge Aachen
CAE	Classical Autoencoder
CDF	Collider Detector at Fermilab
CERN	Conseil Européen pour la Recherche Nucléaire
CKM	Cabibbo Kobayashi Maskawa
CM	Centre of Mass
CMS	Compact Muon Solenoid
CNN	Convolutional Neural Network
CNOT	Controlled NOT quantum gate
CP	Charge conjugation Parity
CSWAP	Controlled Swap quantum gate
DGL	Deep Graph Library
DGLAP	Dokshitzer–Gribov–Lipatov–Altarelli–Parisi equations
DIS	Deep Inelastic Scattering
DNN	Deep Neural Networks
EMPN	Energy-weighted Message Passing
EW	Electro-Weak
GeV	GigaelectronVolt
GNN	Graph Neural Network
GPU	Graphics Processing Unit
HEFT	Higgs Effective Field Theory
HELAS	HELicity Amplitude Subroutines
HEP	High Energy Phenomenology
HR	High Resolution
IBM	International Business Machines
IBMQ	IBM Quantum
IF/FI	Initial-Final/Final-Initial
II	Initial-initial

IR	InfraRed
IRC	InfraRed and Collinear
ISR	Initial State Radiation
KLN	Kinoshita–Lee–Nauenberg
LEP	Large Electron-Positron
LHC	Large Hadron Collider
LO	Leading ORder
LR	Low Resolution
MET	Missing Transverse Energy
ML	Machine Learning
MLM	Michelangelo L. Mangano
MLP	MultiLayer Perceptron
MPI	Multi Parton Interaction
MPNN	Message-Passing Neural Network
N3LO	Next-to-next-to-next-to-Leading Order
NISQ	Noisy Intermediate Scale Quantum
NLO	Next-to-Leading Order
NNLL	Next-to-next-to-Leading Log
NNLO	Next-to-next-to-Leading Order
PDF	Parton Distribution Function
PF	Particle-Flow
PGAE	Particle Graph AutoEncoder
QAE	Quantum AutoEncoder
QCD	Quantum ChromoDynamics
QML	Quantum Machine Learning
RMSE	Root-Mean-Square Error
ROC	Receiver Operator Characteristics
SM	Standard Model
UE	Underlying Event
UFO	Universal FeynRules Output
VBF	Vector Boson Fusion

Chapter 1

Introduction

Modern scientific endeavour often involves a huge collaborative effort, bringing experts from various fields together to push the frontier of knowledge and unlock the secrets of the universe. One such enterprise is the Large Hadron Collider (LHC), the world's largest and most powerful particle accelerator, which collides bunches of protons at multi-TeV energies. It studies the structure of matter at the subnuclear length scales. At such length scales, nature is governed by quantum field theoretical predictions based on the Standard Model [1–11] of particle physics.

The Standard Model (SM) is a gauge theory that explains three of the four fundamental interactions and can explain diverse phenomena over a wide length scale. Despite its success, it can still account for only about five per cent [12,13] of the total energy budget of the universe, as it does not predict the existence of dark matter or dark energy. Moreover, there is irrefutable experimental evidence [14, 15] of the neutrinos' non-zero masses, which are exactly massless in the SM. These shortcomings suggest the existence of a larger theory that contains the SM in the correct limit but can explain the experimental observations. After discovering the Higgs boson [16, 17], the last missing piece of the SM, the LHC continues to investigate its various properties and look for signatures of physics beyond the Standard Model (BSM).

With the huge amount of data that the LHC produces, there is a tremendous effort within the experimental and phenomenological communities alike to search for novel ways of looking at the data to maximise the physics output and improve our understanding of the fundamental particles. The analyses are even more complicated because partons, the entities that have a colour charge and undergo the interaction, are not directly observed due to the confinement effects of Quantum Chromodynamics (QCD). Due to the non-abelian nature of the $SU(3)$ group describing the QCD Lagrangian, any coloured parton produced at very high energies ($\sim 10^2$ GeV at LHC) emits even more partons that share the energy of the initial parton. These partons hadronise to form colourless hadrons, which are recorded at the various components of the detectors. Sophisticated reconstruction techniques are then used to map the multitude of hadrons recorded to

a lower-dimensional final state consisting of small classes of reconstructed objects like jets, leptons, and photons to achieve theoretical control from the parton-level kinematics. Therefore, studying various processes at the LHC relies on extensive simulation of the multitude of scales like generation of the hard parton level matrix-element, parton showering, hadronisation, and detector response, which are extensively validated with experimental data.

The particle physics community has long been leading proponents of machine-learning techniques utilising powerful techniques like Boosted Decision Trees [18] (BDTs) and (shallow) Artificial Neural Networks [19,20] (ANNs) to analyse multi dimensional data, leveraging their power to enhance experimental sensitivities. However, most of these analyses used a relatively small number of highly specific variables based on the reconstructed objects and guided by our physics knowledge. With the advent of modern deep-learning algorithms propelled by the wide availability of high-end GPU acceleration capable of processing large datasets, the game has changed completely. Deep-learning algorithms that take very high-dimensional raw information registered at the detectors have been shown to perform exceptionally well (or at least as good as) compared to those based on human-engineered variables. Consequently, there is a considerable increase in using such algorithms at various stages of the analysis.

In this thesis, we will study some phenomenological aspects of applying deep-learning algorithms at the Large Hadron Collider. We concentrate on three facets: the relative power of deep-learning algorithms over physics-specific variables, their relative robustness to elements of theoretical relevance like perturbative accuracy and robustness to soft and collinear emissions, and exploring model-independent methods of learning background-only hypotheses with powerful unsupervised learning methods.

Before we get into the details of the different studies involved, we briefly describe the phenomenological aspects of signal searches in the energy frontier at the LHC in this chapter. In Section 1.1, we give a brief overview of the Standard Model and motivate beyond Standard Model physics. We briefly introduce Quantum Chromodynamics in Section 1.2 and explain its scope at the LHC for accurate prediction and simulation of events. The various components of a detector at the LHC and an outline of event reconstruction are given in Section 1.3. Finally, the summary of the thesis and the subsequent chapters is presented in Section 1.4.

1.1 The Standard Model and beyond

The Standard Model describes three (strong, weak, and electromagnetic interactions) of the four known fundamental forces of nature in a common framework based on the gauge group $SU(3)_C \otimes SU(2)_L \otimes U(1)_Y$. The massive gauge bosons $W^{\pm}$ and Z are given mass via spontaneous symmetry breaking of a scalar Higgs

doublet field. Due to the chiral nature of the $SU(2)_L$ group, the fermions are taken to be massless, and Yukawa couplings with the Higgs field generate their masses. The fermions are divided into quarks and leptons based on their charge under the colour group $SU(3)_C$–quarks are charged under the colour group, while leptons do not carry such a charge. While the $SU(3)_C$ sector, which describes the strong interactions, has a non-perturbative nature at lower energies, the $SU(2)_L \otimes U(1)_Y$, manifests as the electromagnetic interaction and weak interaction at lower energies.

The Standard Model has been tested in multiple experiments in different energies and, to date, has stood up to the experimental scrutiny of its known constituents. The manifestation of the SM's interaction as weak interactions and the long-range electromagnetic in the low energy regime offers significant opportunities to test the prediction of various observables. Many properties of leptons like the anomalous magnetic moment of the electron and the muon and the muon decay lifetime have been measured precisely and show excellent agreement with the predictions of SM originating from higher-order loop corrections. Although the recently updated anomalous muon magnetic moment [21], has a 4.2σ deviation from the SM's expected value, generating a high expectation and euphoria among the HEP community, this deviation has been found after a precision of more than six significant digits, up to which the SM's prediction agrees completely with the experimentally measured value. Other than the precise prediction of various observables, the Standard Model also foreshadowed the existence and properties of undiscovered particles. One such instance is the prediction of the existence of a third family of quarks [22] for the SM to be anomaly free [23,24] after the discovery of the tau lepton [25]. Another instance is the prediction of the top quark's mass range [26] before its discovery from the measurements of the $W^{\pm}$ and Z boson's mass, the forward-backward asymmetry of the Z boson's decay, and low energy observables like the Fermi constant G_F, and the fine structure constant α. This predictability arises since the calculation of parameters in the Lagrangian like the Weinberg-angle $\sin\theta_w$ through these various observables, although the same at tree-level, have different radiative contributions to the underlying process.

Even with this long list of success stories, the consensus among the particle physics community is that the SM is a low energy effective theory. Its shortcomings show up on different fronts, such as the aesthetics of the theory, the inability to accommodate other experimental observations etc. One such inability is the failure to account for observed evidence of dark matter. If dark matter is some fundamental particle that does not interact electromagnetically (and hence dark) but has mass and interacts gravitationally. The Standard Model does not accommodate any such candidate. Although these observations are of astrophysical or cosmological origins, there are many well-motivated extensions of the Standard Model, which could account for the observed abundance of dark matter. The recently discovered Higgs boson could have a sizable amount of interaction with such dark matter particles. These models, known as Higgs-portal [27–31] dark

matter scenarios, could show up at the LHC as a large invisible-branching ratio of the Higgs boson. In chapter 3, we will explore the power of deep-learning methods in constraining the invisible branching ratio of the Higgs, finding them to outperform multivariate and univariate methods by a significant margin.

The matter-antimatter asymmetry of the universe is another empirical evidence that the SM can not explain. In the SM, there is a small amount of CP violation in the quark sector via the CKM matrix, which cannot account for the observed asymmetry. One of the SM's straightforward extensions is expanding the scalar sector–there is no underlying reason why there should only be a single Higgs doublet. A minimal extension is the two-Higgs Doublet Model [32, 33] (2HDM), which could allow for spontaneous CP violation, increasing the amount of CP violation found in the SM. A special case with a natural dark matter candidate is the inert doublet model [34]. However, the minimal model cannot simultaneously explain [35] the observed dark matter abundance and the matter-antimatter asymmetry.

Any new physics model must be able to explain the SM's experimental results, which have been tested at various low-energy and high energy collisions. Therefore, there is already a significant amount of experimental results to constrain various BSM models. Some observables which are highly sensitive to the nature of BSM interactions are the pole masses of the weak bosons. Many BSM models can be constrained by studying the model's contribution to the gauge boson propagators, encapsulated in terms of the S, T, and U parameters [36, 37]. Another important quantity is the ρ-parameter (related to the T-parameter) which is a consequence of the custodial $SU(2)$ global symmetry [38] of the Higgs sector under which the three gauge bosons of the $SU(2)_L$ group transform as a triplet. It is defined as

$$\rho = \frac{m_W^2}{m_Z^2 \cos^2 \theta_w} \quad , \tag{1.1}$$

where m_W and m_Z are the pole masses of the W and Z boson, respectively, and θ_w is the Weinberg angle. In the SM, $\rho = 1$ at the tree level, which gets quantum corrections mainly from the top quark due to its high mass. Note that the equality of $\rho = 1$ is not a consequence of the specific nature of the Higgs field but rather a consequence of the existence of three Goldstone bosons which give mass to the massive gauge bosons. The recent announcement by the CDF collaboration of the precise mass of the W-boson [39], which shows a 7σ deviation from the SM expectations, has considerable implications for these precision observables [40–42].

Many of the proposed models in the literature can show various signatures at the Large Hadron Collider. On the other hand, the absence of such signatures will put tight constraints on the allowed parameter space of these models. Therefore, the LHC, on top of verifying the SM's scalar sector, will also constrain the allowed parameters for the different theoretically motivated extensions of the SM, or hopefully, discover new particles. Doing so requires one to tackle the

huge background that arises from the strong interactions, which present unique situations and require ingenuity to extract significant physics from the observed collision events. The rest of the chapter is dedicated to explaining Quantum Chromodynamics which describes the strong interactions and the essential elements of the Large Hadron Collider's detectors and event reconstruction procedure.

1.2 A brief overview of Quantum Chromodynamics

The Large Hadron Collider collides protons moving at centre-of-mass energies of several TeVs. At such high energies, the degrees of freedom relevant for describing the collisions are in terms of the partons: quarks and gluons that make up the proton and undergo strong interaction. Therefore, we give a brief overview of Quantum Chromodynamics which describes the interaction of the quarks and gluons.

1.2.1 The QCD Lagrangian

Quantum Chromodynamics is a gauge theory based on the group $SU(3)_C$, and the Lagrangian is given as

$$\mathcal{L}_{QCD} = \sum_f \bar{q}_f^a \left(i \not{D}^{ab} - m_f^{ab} \right) q_f^b - \frac{1}{4} F^I_{\mu\nu} \, F_I^{\mu\nu} \quad . \tag{1.2}$$

The indices a and b are the colour indices of the fundamental representation of $SU(3)_C$, the sum over the index f denotes the sum over all active flavours at the energy scale, q_f^a denoting the quark field of flavour f and colour a, $\not{D}^{ab} = \gamma^\mu D_\mu^{ab}$ where we have suppressed the spinor indices. The covariant derivative is given in terms of the gluon fields A_μ^I as

$$D_\mu^{ab} = \delta^{ab} \partial_\mu + i \, g_s \, t^I_{ab} \, A^I_\mu \quad , \tag{1.3}$$

where the summation over the gluon field index I which runs from one to eight in the adjoint representation of the $SU(3)_C$ group, is assumed implicitly. The generators t^I satisfy the commutation relation

$$[t^I, t^J] = i \, f^{IJK} \, t^K \quad , \tag{1.4}$$

with f^{IJK} the structure constants of the $SU(3)$ group. In the fundamental representation, they can be written in terms of the Gell-Mann matrices. The field strength tensor $F^I_{\mu\nu}$ is given as

$$F^I_{\mu\nu} = \partial_\mu \, A^I_\nu - \partial_\nu \, A^I_\mu - g_s \, f^{IJK} \, A^J_\mu \, A^K_\nu \quad . \tag{1.5}$$

Although the strong interactions are non-perturbative in the low energy scale, the phenomena of asymptotic freedom [43, 44] allow the application of perturbative quantum field theory techniques in the high-energy regime.* The evolution of the coupling constant with the scale is described by the renormalisation group equation

$$\beta(\alpha_S) = \mu_R \frac{d\alpha_S(\mu_R)}{d\mu_R} \quad , \tag{1.6}$$

where $\alpha_S(\mu_R) = g_s^2(\mu_R)/4\pi$ and μ_R is the renormalisation scale within the $\overline{\text{MS}}$ subtraction scheme and is of the same order as the scale of the interaction, and $\beta(\alpha_s)$ is the QCD β-function which has a perturbative expansion. At 1-loop, the solution is [45]

$$\alpha_S(\mu_R) = \frac{6\pi}{33 - 2\,n_f} \frac{1}{\log \frac{\mu_R}{\Lambda_{QCD}}} \quad , \tag{1.7}$$

n_f is the number of quark flavours, and Λ_{QCD} is the Landau pole of QCD. Note that the equation is valid for $\mu_R > \Lambda_{QCD}$, as the Landau pole denotes the energy scale below which the theory becomes non-perturbative.† Since we have six quark flavours in nature, $\alpha_S(\mu_R)$ decreases with increasing energy, and hence calculations with perturbative techniques can be relied upon for high enough energies. The β-function is known for up to five loops [46] in literature, and the Landau pole is of the order of the inverse of the classical proton radius $1/\Lambda_{QCD} \sim 10^{-15}$ m $\sim 1/200$ MeV.

1.2.2 Hard scattering cross sections

In a proton-proton collision at the LHC, the cross-section of a final state **F** containing a fixed number of partonic species is given by,

$$\sigma(\mathbf{PP} \to \mathbf{F}) = \int dx_1\, dx_2 \sum_{i,j} f_i(x_1, \mu_F)\, f_j(x_2, \mu_F)\, \hat{\sigma}(ij \to \mathbf{F}|\mu_F, \mu_R) \quad , \tag{1.8}$$

where i and j are the flavour of the partons, f_i and f_j are the parton distribution functions (PDFs) of the parton species i and j, whose momentum are given in terms of the proton momentum fractions x_1 and x_2 as $p_i = x_i\, P_1$ and $p_j = x_2\, P_2$, with P_1 and P_2 being the momenta of the protons undergoing the hard interaction. The sum is over all possible combinations of parton species, which can give the final state **F**. The PDFs are universal, process independent, and encode the non-perturbative physics inside the proton. However, their evolution with the energy scale can be described with perturbative methods and is encapsulated in the DGLAP equations [47–50], allowing for their extensive vali-

*In fact, the discovery of asymptotic freedom in non-abelian gauge theories instigated the adoption and verification of quantum chromodynamics [10, 11] as the fundamental theory describing strong interactions.

†It does not mean that QCD is not valid at and below the Landau pole, but simply that perturbative techniques fail to capture the phenomena accurately.

dation from experiments at different energy scales. The parton level cross-section $\hat{\sigma}$ is process-specific and depends on the specific nature of the interaction of the final state partons to the proton's constituents, and has well-defined perturbative expansions in α_S. There is an implicit assumption of collinear factorisation in eq 1.8, emissions with transverse momenta below μ_F are encoded in the PDFs while those above are evaluated perturbatively in $\hat{\sigma}$. Although generally assumed to be true for inclusive observables, factorisation has been proved for a very small subset of processes [51].

1.2.3 Soft and collinear divergences

The presence of disparate energy scales in any hard scattering process at the LHC prohibits the reliability of perturbative calculations for any finite order. This unreliability is closely connected to the zero-mass of the gluons, as any coloured parton can emit an infinite number of soft or collinear gluons without taking any noticeable momentum fraction, which manifests as divergences in the differential cross-section in these regions of phase space. However, since any detector has finite energy and angular resolution, any additional emissions beyond the resolving power will be distinguishable from those without such an emission. Therefore, we must include the sum of all these degenerate states when calculating physically observable quantities. At the same time, one would like the predictions to be not dependent on the particular resolution of the detector installations. The requirement of infra-red and collinear (IRC) safety on observables provides a handle to control these unphysical divergences by making them calculable with perturbative methods enabling their prediction from theory. It constrains the observable's nature when a particle emits additional particles in the soft and collinear regions. If a particle q from a final state of n partons undergoes a splitting $q \to r + s$, with $p_q = p_r + p_s$, an IRC safe observable $\mathcal{O}_n$ satisfies

$$\begin{aligned} &\mathcal{O}_{n+1}(p_a, ..., p_b, p_r, p_s, p_c, ...) \to \mathcal{O}_n(p_a, ..., p_b, p_q, p_c, ...) \quad \text{as} \quad z_r \to 0 \quad , \\ &\mathcal{O}_{n+1}(p_a, ..., p_b, p_r, p_s, p_c, ...) \to \mathcal{O}_n(p_a, ..., p_b, p_q, p_c, ...) \quad \text{as} \quad \Delta_{rs} \to 0 \quad , \end{aligned} \tag{1.9}$$

where z_r is the relative hardness of p_r, and Δ_{rs} is the angle between $\vec{p}_r$ and $\vec{p}_s$. For hadron colliders, the relative hardness is defined as $z_r = p_T^r/(\sum_{i=1}^n p_T^i)$, with p_T^i denoting the transverse (perpendicular plane to the collision axis) momentum of particle i. The divergences of these real emissions cancel exactly with virtual correction for inclusive observables, under the KLN theorem [52, 53], provided they are infra-red and collinear safe (IRC safe). However, most observables of interest in hadronic environments are of the exclusive type, and there are potentially large logarithmic terms arising from the unbalanced cancellation of the real and virtual corrections at any finite order in perturbation theory. Consequently, an all-order resummation is needed for a useful prediction of exclusive observables. These calculations are highly non-trivial, and only a restricted set of such

observables (for instance, see [54–57]) have been calculated for hadronic collisions. Moreover, there are inherently IRC unsafe but useful observables like track-based observables [58, 59] or the ratio of angularities [60]. Therefore, a major portion of the phenomenological program relies on numerical procedures to approximate the leading behaviours of the evolution of partons from the hard interaction scale down to a scale closer to Λ_{QCD}.

1.2.4 Simulating QCD at different scales

For the extensive phenomenological program at the LHC, decades of research have resulted in comprehensive open-source simulation programs, extensively validated on experimental data. Here, we present a brief overview of the available packages used extensively in the studies conducted in this thesis. The procedure of predicting hadronic final states originating from initial protons requires a detailed examination of QCD at different length scales. These may be divided into:

- the initial process-dependent hard-scattering partonic cross-section
- the evolution of the hard partons via emission of new partons sharing the total energy, encapsulated in parton shower generators and
- the hadronisation of the showered partons to form colourless hadrons. This also includes simulating the interaction between spectator partons between the two colliding protons, commonly referred to as multi-parton interactions (MPI), and is of a non-perturbative origin.

While the first two situations can be solved using perturbative methods based on an underlying Lagrangian, the simulation of the hadronisation effects and the multi-parton interactions is based on parametrised models tuned with data. The contribution of multi-parton interactions to the underlying event, which additionally consists of beam-beam interactions, presents irreducible smearing of various perturbative calculations and is a high source of systematic uncertainties when studying hadronic final state objects.

We will mainly rely on `MadGraph5_aMC@NLO` [61] for the generation of parton level differential cross-section, with `POWHEG-BOX` [62–65] used to generate next-to-leading order events for the fourth chapter. The former is a meta-code capable of generating `HELAS` subroutines [66] to evaluate tree-level matrix elements of any $2 \to n$ processes. It can take any model file implemented within the `Universal FeynRules Output` (UFO) [67] format. The model output is obtained using the `FeynRules` [68, 69] Mathematica package, which outputs the UFO model for any generic theory based on their Lagrangian. For the parton showering of the hard partons down to a scale near Λ_{QCD}, and the hadronisation of the showered partons to form colourless hadrons and the multi-parton interactions, we will use `Pythia8` [70]. These tools facilitate the widespread phenomenology of various BSM models at the LHC.

1.3 The Large Hadron Collider

The idea of a hadron collider at CERN was conceived during project discussions for the Large Electron-Positron Collider (LEP), where it was proposed that the tunnel for the LEP be made large to accommodate a future superconducting proton collider, whose purpose would be to look for the Higgs boson and other heavier particles predicted by BSM theories. Higher energies are achievable for protons since the maximum feasible centre-of-mass (CM) energy for a circular collider is determined by the power lost via synchrotron radiation–the radiation emitted due to the circular acceleration of charged particles. In the relativistic regime, electromagnetic power radiated by a charged particle of charge e, moving in a circular orbit of radius R, is given by Schwinger's formula [71],

$$P \propto \frac{1}{R^2}\left[\frac{E}{m}\right]^4 \quad ,$$

where E and m represent the energy and the rest mass of the particle, respectively. For the same radius, a heavier particle emits much less energy–the proton, which weighs 10^3 times more than the electron, will radiate approximately 10^{-12} times less power. Hence, protons can generally be accelerated to higher CM energies in circular colliders. This higher energy coupled with the relatively large strong coupling constant makes hadronic colliders highly suitable for discovering previously undiscovered particles. However, the hadronic environment, unlike lepton colliders, reduces the maximum achievable precision. In summary, hadronic colliders are discovery machines, while lepton colliders are precision machines.

The Large Hadron Collider is a 27 km long synchrotron capable of accelerating protons to multi-TeV energies. It consists of two circular tunnels where protons and heavy ions can be accelerated in opposite directions to very high energies. Six experiments are installed at the LHC, specialising in exotic physics scenarios, with more additions planned for the future. At the energy frontier, the two detectors of interest are:

- ATLAS: A Toroidal LHC ApparatuS, a general-purpose detector designed to study Higgs boson and physics beyond the standard model.

- CMS: Compact Muon Solenoid, also a general-purpose detector with the same goals as ATLAS

The two detectors are designed with important technical dissimilarities, reflected in different systematics to facilitate the independent verification of physics discovered at the LHC. Describing these differences is beyond the scope of phenomenological studies. However, we will depict the important parts of the detectors and their uses in this section and briefly describe the event reconstruction and coordinates systems used to study hard-scattering events.

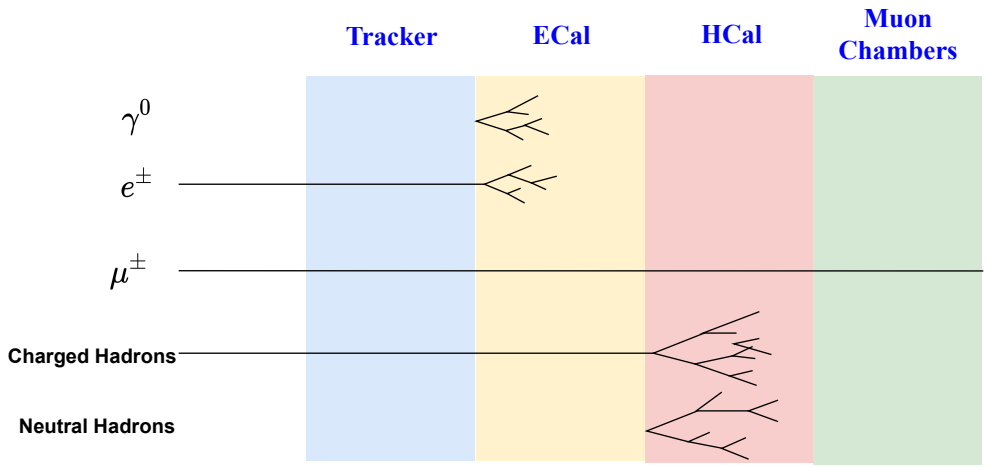


Figure 1.1: The signature of different particles in the components of a hermetic detector are shown in the figure.

1.3.1 Parts of a Detector

The general-purpose detectors consist of several parts designed to measure some properties of particles produced in a collision. They are hermetic detectors capable of measuring all particles (except neutrinos) in the full range of the solid angle from the collision point. It has many subcomponents arranged in a standard order to facilitate this coverage. These parts, in order of proximity to the collision point, are:

1. **Tracker:** The tracker lies in the innermost chambers of the detectors. Its purpose is to distinguish charged particles like the electron, which interacts with the medium and leaves noticeable signatures from neutral particles like photons. A solenoidal magnetic field is applied to the tracking chamber, which, via the Lorentz force generated, gives the ability to measure the momentum and distinguish positive and negatively charged particles. A precise determination of the momenta of the highly-energetic particles produced at the LHC requires very high magnetic fields to facilitate noticeable changes in their trajectory. Consequently, the CMS utilises a magnetic field of around 4 Tesla, while it is around 2 Tesla for the ATLAS detector.

2. **Electromagnetic Calorimeter (ECal):** The electromagnetic calorimeter encloses the tracking chamber, which measures the energy of particles that loses their energy primarily via electromagnetic interactions. Such particles produce electromagnetic showerings in the calorimeters, generally of a lower depth that finishes before reaching the hadronic calorimeter.

3. **Hadronic Calorimeter (HCal):** The hadronic calorimeters measure the energy of all strongly interacting particles that pass through the electromagnetic calorimeter. Such particles interact primarily with the nucleus and produce a nuclear showering profile. These showering shapes are much

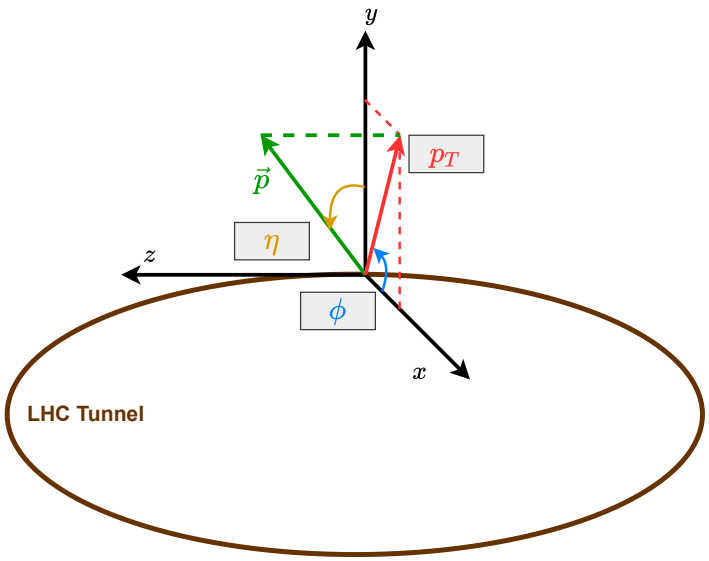


Figure 1.2: The figure shows the coordinate system used for measuring the momenta of particles registered at the general purpose detectors at LHC.

longer and have more shape fluctuations, resulting in larger hadronic calorimeters to allow for complete showerings and a lower resolution than the electromagnetic calorimeters.

4. **Muon chambers:** Muons can pass through all the previous layers unaffected as it is much heavier than the electron and interacts weakly with nuclear matter. The hadronic calorimeters are enclosed by muon chambers which measure them by a tracking system. Only muons can register in the muon chambers out of all the particles currently known in the Standard Model.

A depiction of the signatures of different particles at the general-purpose detectors is shown in figure 1.1. All charged particles show tracks in the tracker through which we can measure their momentum precisely. Light particles like electrons, positrons, and photons deposit most of their energy in the electromagnetic calorimeter and seldom reach the hadronic calorimeter. Neutral pions π^0, which predominantly decays to two photons, also deposit their energy at the electromagnetic calorimeter. Although some light mesons like charged-Kaons, and pions will lose some energy via electromagnetic showering in the electromagnetic calorimeter, most of their energy is deposited at the hadronic calorimeters.

1.3.2 Hadron Collider Coordinates

The inability to determine the longitudinal boosts of the collision frame resulting from our inability to ascertain the partonic centre-of-mass $\sqrt{\hat{s}}$ motivates the use of longitudinal boost invariant quantities in describing the kinematics of particles at hadron colliders. A schematic representation of the coordinate system used to

describe the observed particles at the detectors is shown in figure 1.2. The three quantities used to describe the momentum vector $\vec{p}$: the transverse momentum p_T, the azimuthal angle ϕ, and the pseudorapidity η are shown within grey boxes. Out of these, the first two are invariant under longitudinal boosts. However, the pseudorapidity defined in terms of the polar angle θ between the direction of $\vec{p}$ and the z-axis as

$$\eta = -\ln \tan\frac{\theta}{2} \quad , \tag{1.10}$$

depends on the p_z component of the momentum. For massless particles, it is equal to the rapidity y defined as

$$y = -\frac{1}{2}\ln\frac{E - p_z}{E + p_z} \quad . \tag{1.11}$$

Since, for *massless* particles $|\vec{p}| = E$, we $\cos\theta = p_Z/E$, which gives,

$$\frac{E - p_z}{E + p_z} = \frac{1 - \cos\theta}{1 + \cos\theta} = \tan\frac{\theta}{2} \quad .$$

The difference between rapidities is longitudinal boost invariant and useful for analysis of final state particles at hadron colliders. Except for jets, the very high energy of the LHC makes most reconstructed objects like electrons and muons; and raw detected particles practically massless. Hence, pseudorapidity makes it practical to compare theoretical calculations (based primarily on rapidity) and experimental measurements. Another important quantity is the angular separation of two objects in the $\eta - \phi$ plane,

$$\Delta R = \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2} \tag{1.12}$$

1.3.3 Event Reconstruction

As we have seen in Section 1.2, there is a multitude of particles produced in any hard-scattering event. The objects of theoretical interest are the partons and their kinematics. Complex event reconstruction techniques facilitate such a map from the very high dimensional final state measured at the detectors to a lower-dimensional final state. Here, we present the essential aspects of event reconstruction. Although experimental analyses require complex simulation of the interaction of the hadronised particles with matter, we will only present the qualitative aspects of the reconstruction taken care of in parametrised detector simulation software like `Delphes3` [72].

Any hard parton produced that does not interact strongly but electromagnetically will generally recoil against other partonic species and be highly segregated and radiate much lesser than a coloured parton. Therefore, various isolation criteria are imposed on the neighbourhood of detected particles to identify electrons, photons and muons. A straightforward definition used in Delphes3 for a particle

P is,

$$I(P) = \frac{\sum_{i \in \mathcal{S}} p_T(i)}{p_T(P)} \quad , \tag{1.13}$$

where the set $\mathcal{S}$ is defined as all the particles (measured in some detector subcomponent) within a radius $R : \Delta R_{iP} < R$, with a threshold $p_T(i) > p_T^{min}$ on their transverse momentum. A lower value of I denotes better isolation and an upper bound I_0 is set on I to identify the particle P. Therefore, the three parameters of interest in determining the isolation criteria of a type of particle are R, p_T^{min} and I_0. The various reconstructed objects which are obtained after processing the measurements of the different detector components are summarised in the following:

- **Electrons:** Electrons and positrons are charged particles which will leave tracks and deposit all of their energies in the electromagnetic calorimeter. Their reconstruction, therefore, involves determining their isolation in the tracker as well as the electromagnetic calorimeter, with no deposits in the hadronic calorimeters.

- **Photons:** Photons will leave no tracks but deposit all their energy in the electromagnetic calorimeter. Therefore, an isolation criterion of deposits on the electromagnetic calorimeter with no signatures in any other component reconstructs them efficiently.

- **Muons:** Muons and anti-muons will leave tracks in the trackers, not deposit any discernible energy in the calorimeters, and reach the muon chambers mostly unaffected. Therefore, the presence of isolated tracks in the inner tracker and the muon chambers with suppressed activity in the calorimeters will identify muons.

- **Jets:** Strongly interacting energetic particles manifest themselves as collimated sprays of various charged and neutral particles, which show signatures in most detector components. They are reconstructed into composite objects called jets consisting of collimated multiparticle final states. Their definition, which is discussed in the next chapter, has undergone extensive investigation to facilitate a comparison between theoretical predictions and experimental measurements. The third-generation particles in the SM, like tau leptons and bottom quarks, decay to the lower mass particles with a finite decay length (although the taus decay directly to mesons or lighter leptons, bottom quarks first hadronise to form B-hadrons which then decay to lighter hadrons). Such heavier states can be identified by observing the displacement of the decay particles from the initial collision point.

- **Long-lived Particles:** In the Standard Model, only neutrinos or muons can reach the muon chambers. However, in many BSM models, quasi-stable neutral particles can decay to SM particles with a long enough lifetime

to reach the muon chambers. Such long lived particles' decay to charged particles will manifest as tracks originating within the muon chambers and, therefore, can be used as an identifying signature.

- **Missing transverse energy:** Those particles that do not undergo strong interaction and are electrically neutral like neutrinos will not show any signatures anywhere in the detector. Their presence in the final state can be inferred from a momentum mismatch in the vector sum of all detected particles. Since the partonic longitudinal boost along the collision axis is indeterminable, we can only measure the momentum mismatch in the transverse plane. Thus, the missing transverse energy (MET) is a two-vector defined in the transverse plane as,

$$\vec{\not{E}}_T = -\sum_{i \in \text{visible}} \vec{P}_T \quad , \tag{1.14}$$

where the sum over particles is generally taken over all the detected particles in the calorimeters.

Although we have described the nominal aspects of event reconstruction, many aspects of experimental importance have been ignored. In the following paragraphs, we briefly describe the qualitative aspects of some crucial experimental factors.

The first important aspect is that our discussions focused on reconstructing events after being recorded into storage– the so-called "offline" event reconstruction. The LHC collides bunches of protons which collide almost every twenty-five nanoseconds. Most of these collisions are uninteresting and produce events within known sectors of the Standard Model. The result of all these collisions cannot be stored due to the hardware's limited processing and storage capabilities, and dedicated online tiered trigger systems select interesting events recorded for later analysis. The intricate design of such online triggers [73–76] achieves unbiased event selection for interesting processes.

Another important aspect is the presence of pileup–the collision of more than a pair of protons from opposite beams. These secondary collisions are unavoidable, and multiple collisions happen at each bunch crossing. The collision point of two protons shows in the detector as vertices where clusters of tracks originate. Out of these, the one with the highest constituent energy is the primary vertex. Most of the charged particles arising from secondary collisions can be determined from the track information if the collision point is resolvable from the primary vertex. The subtraction of pileup deposits from neutral particles is vital in assessing the accuracy of theoretical predictions of jets from QCD. Hence, it is of major phenomenological and experimental interest [77–82].

1.4 Outline of thesis

This thesis investigates the use of deep-learning algorithms for identifying signatures of new physics at the LHC, concentrating on three aspects of phenomenological relevance:

- The first aspect explores the power of deep learning to signature specific searches of new physics. We show that deep-learning algorithms like Convolutional Neural Networks can outperform traditional univariate or multivariate analyses of high-level variables in identifying processes with unique radiation patterns like vector-boson fusion.

- In taking the raw detector level inputs, the precise determination of an algorithm's working based on first principle analysis is often lost, leading to such algorithms showing a higher systematic uncertainty. The second aspect studies the dependence of deep-learning algorithms on the specifics of the simulation, like the parton-recoil scheme and perturbative accuracy of the hard simulations. We also devise procedures to make the output of deep-learning algorithms robust to soft and collinear emissions.

- Due to the absence of well-motivated new-physics scenarios at the LHC, model-independent search techniques are increasingly important so that we do not miss out on possible new physics. Unsupervised deep-learning methods provide powerful ways of learning background-only distributions, which could mine the presence of new physics signals in a very large phase space volume. The third aspect explores anomaly detection techniques based on Graph Neural Networks and Variation Quantum Circuits.

Before going into the details of the study conducted, we introduce the basic concepts related to the theoretical basis of machine learning with artificial neural networks in the second chapter. The different types of neural networks and their current use in LHC phenomenology are also explained in this chapter.

In the third chapter, we look at the ability of Convolutional Neural Networks (CNNs), taking the full calorimeter information as an image (tower-image) to identify invisible decays of the Higgs boson produced via the Vector Boson Fusion (VBF) process, which is the most sensitive channel for constraining the invisible branching ratio of the Higgs boson, important in various Higgs-portal dark matter models. CNNs using the tower image outperform ANNs based on high-level variables or the shape-analysis of variables like the invariant mass of the dijet system or their pseudorapidity separation.

Although CNNs show the best performance in identifying the invisible Higgs signal, using the raw calorimeter information to train the network presents the possibility of the CNN picking up subtle aspects of the simulation. One such aspect is the inability of a global-recoil scheme in the parton shower algorithm

to describe the wide-angle radiation in VBF events correctly. In the fourth chapter, we scrutinise the effect of using the more physically accurate dipole-recoil scheme for the VBF signal on the network's performance, finding that it is more important than the perturbative accuracy of the parton-level matrix-element simulation. We find that even though the training accuracy is highly affected by the type of simulation used, the validation accuracy of the most physically accurate simulation: the ones with a next-to-leading perturbative parton-shower accuracy coupled with a dipole-recoil parton shower, has a validation accuracy relatively unaffected by the dataset used to train a CNN.

The results of the fourth chapter show that the learning of deep neural networks is heavily dependent on the input data. In the fifth chapter, we devise an infra-red and collinear safe graph neural network algorithm to improve the response of deep-neural networks to imperfect modelling of hadronisation dynamics. Graph Neural Networks are a different deep-learning algorithm that takes the favourable properties of CNNs and generalises them to higher dimensional non-Euclidean spaces and forgoes the sparse representation found in calorimeter images. We apply such an IRC safe network to the problem of jet identification on publicly available datasets and find that their performance is comparable to current state-of-the-art but IRC unsafe algorithms.

All studies in the previous chapters concentrated on signal specific discrimination, and the model was specifically trained to maximise the signal discrimination. Such analyses fall into the wider purview of model-specific searches and learn the decision boundary between the signal and background distributions. With all current model-specific searches for physics beyond the Standard Model yielding negative results, it is paramount that we explore all possible avenues, increasing the importance of model-unspecific investigations. Modern deep-learning algorithms provide powerful tools to learn background only distribution in a large phase space volume. In the sixth chapter, we devise a graph autoencoder based on the message-passing paradigm, capable of inductively learning the substructure of QCD jets. As a benchmark on possible signals, we find that the loss function is a capable discriminant in identifying various n-prong jets.

Quantum computing promises to revolutionise various aspects of simulation and data analysis. One such area of interest is Quantum Machine Learning, using parametrised variational quantum circuits with tunable unitaries for machine learning purposes. In the seventh chapter, we explore the power of quantum autoencoders for anomaly detection at the LHC. Compared to similarly expressive bit-based autoencoders, they converge much faster with minuscule datasets and outperform them on benchmark signal scenarios.

In the eighth chapter, we summarise the findings of the thesis and elucidate future lines of investigation into the problems tackled in the thesis.

Chapter 2

Methodology

In the previous chapter, we have described the underlying importance of theoretical QCD predictions and precise experimental measurements to facilitate the discovery of new physics at the LHC. This chapter discusses the theoretical basis of analysing hadronic final states at colliders–jets, with current definitions satisfying the principle of IRC safety, which facilitates an accurate prediction of jet cross-sections order by order in perturbation theory. The very high energy of the LHC also motivates looking within such jets. Most heavy particles in the SM like the Higgs, top-quark, or the vector bosons and any appropriately coupled BSM particles predominantly decay to quarks due to their higher (colour) multiplicity. Although at low boosts, these decays are indecipherable due to the massive backgrounds originating from QCD jets, due to the high energy of the LHC, one can find a statistically significant amount of events at the high-momentum tail where the markedly different energy patterns from QCD jets can discriminate and enhance searches. To exploit such differences, made possible by the better granularity of the installed detectors, a vibrant field of jet substructure has emerged in the past decade, which also planted the entry of modern deep-learning techniques in the form of classifying jet images.

In the Section 2.1, we describe the definition of jets and provide elements of modern jet substructure techniques and bridge the connection to visual recognition techniques. Section 2.2 describes the rigours of artificial neural networks in the supervised and unsupervised settings with autoencoders as a specific example, along with a basic introduction to the optimisation procedure. Having laid down the necessary groundwork, we examine the details of the deep-learning algorithms employed in the thesis in Section 2.3 and summarise in Section 2.4.

2.1 Jets

Jets are ubiquitous final states at any hadronic collider. Therefore, the analysis of any probable physics at the LHC requires detailed knowledge of their kinematics and cross-sections. Due to their importance both in theory and experiments,

jet definitions have undergone a lot of scrutiny [83–91] and discussions [92] to facilitate their theoretical calculability and ease of experimental calibration and measurements. Jet definitions can be broadly divided into cone algorithms and sequential recombination algorithms. The former is motivated by a top-down approach considering that low energy effects do not significantly alter the hard energy flow of the event. In contrast, the latter follows a bottom-up approach and sequentially recombines measured particles via a metric generally following the structure of QCD splittings.

A prime example of a cone algorithm is the Sterman-Weinberg jet definitions for e^+e^- collisions, where an event is classified as a two jet event if one could find an energy fraction $1-\epsilon$ of the entire event in two back-to-back cones of solid half-angle δ. Although the Sterman-Weinberg jet is infra-red and collinear safe, its generalisation to hadronic colliders is challenging primarily due to the presence of final state particles not participating in the hard interaction, thereby making the concept of total energy nonsensical. This issue results in the arbitrariness of cone placements and overlaps for a higher multiplicity of jets. Iterative procedures with initial seeded placement of cones have been widely used in hadron colliders, which were found to be IRC unsafe in retrospect.

Cone algorithms are particularly advantageous for experimental calibrations due to the geometric nature of their definitions. Seedless cone algorithms which are IRC safe have been formulated [93], whose implementation takes up a lot of memory and computational power. With the advent of the anti-k_t algorithm, a sequential recombination algorithm giving ideal cone-like jets and its fast implementation in the `FastJet` [94] package, most analyses have shifted to using the anti-k_t algorithm at the LHC. Therefore, we describe the modern variant of sequential recombination algorithms, namely the generalised k_t family of algorithms. Other than the anti-k_t algorithm being a member, its different variants like the k_t and the Cambridge-Aachen (CA) algorithms, which have well-defined connections to the splitting structure of QCD are important in the study of jet-substructure, which we will also touch upon briefly.

2.1.1 Generalised k_t algorithms

The generalised k_t algorithms for hadron colliders are described in terms of the distance metric d_{ij} and a jet radius R_{jet} as

$$d_{ij} = \min(p_{Ti}^{2p}, p_{Tj}^{2p}) \frac{\Delta R_{ij}^2}{R_{jet}^2} \quad , \quad d_{iB} = p_{Ti}^{2p} \quad , \tag{2.1}$$

where $p = 1$ for the k_t algorithm [87, 89], $p = 0$ for the Cambridge-Aachen (CA) algorithm [90, 95, 96], and $p = -1$ for the anti-k_t algorithm [91]. Note that the distance ΔR_{ij} is defined as $\Delta R_{ij} = \sqrt{(y_i - y_j)^2 + (\phi_i - \phi_j)^2}$, with y being the

rapidity.* For a final state consisting of four vectors $\{p_1, p_2, ...p_N\}$, the algorithm† to form the jet goes as follows.

1. Evaluate all possible d_{ij} and d_{iB}.
2. From all d_{ij} and d_{iB} if the minimum is some d_{ij} combine i and j via a recombination scheme to form k, and replace i and j in the set with k, and go to step 1. Unless otherwise stated, we will always use the E-scheme where the combined particle is formed by taking the sum of the four-vectors of the two particles, $p_k = p_i + p_j$.
3. remove i from the set of particles if the minimum is d_{iB} and declare it as as a jet, and return step 1.
4. Terminate when no particles remain.

Since every particle gets assigned to a jet, a minimum criterion is often applied to the transverse momentum of the obtained jets to ignore contributions from soft particles.

We can now understand the reason behind calling the distance parameter R_{jet} as the jet's radius from the second step: it puts a limit on any particle j, which can combine with i at each stage. Dividing d_{ij} by d_{iB} we have,

$$\frac{d_{ij}}{d_{iB}} = \frac{\min(p_{Ti}^{2p}, p_{Tj}^{2p})}{p_{Ti}^{2p}} \frac{\Delta R_{ij}^2}{R_{jet}^2} \quad .$$

Since all values are strictly positive, we have for $\Delta R_{ij} > R_{jet}$,

$$\frac{d_{ij}}{d_{iB}} > \frac{\min(p_{Ti}^{2p}, p_{Tj}^{2p})}{p_{Ti}^{2p}} \quad . \tag{2.2}$$

It is straightforward to see that for $p = 0$, any particle j outside R_{jet} will not be combined with i since we get $d_{ij} > d_{iB}$. The same line of argument follows when $p_{Ti} \leq p_{Tj}$ for $p = 1$, and $p_{Ti} \geq p_{Tj}$ when $p = -1$. For $p = 1$ and $p_{Ti} > p_{Tj}$, we have

$$\frac{p_{Tj}^2}{p_{Ti}^2} < 1 \implies \frac{\min(p_{Ti}^2, p_{Tj}^2)}{p_{Tj}^2} = \frac{p_{Tj}^2}{p_{Ti}^2} < 1 \quad .$$

Since, $a > b \wedge b < c \implies a > c$, we have $d_{ij} > d_{iB}$. Similarly for $p = -1$ and $p_{Ti} < p_{Tj}$ we get $d_{ij} < d_{iB}$, since

$$\frac{p_{Ti}^2}{p_{Tj}^2} < 1 \implies \frac{\min(p_{Ti}^{-2}, p_{Tj}^{-2})}{p_{Tj}^{-2}} = \frac{p_{Ti}^2}{p_{Tj}^2} < 1 \quad .$$

*We cannot use pseudorapidity since a combination of two massless particles p_i, and p_j (in the E-scheme described thereafter) would yield a particle $p_k = p_i + p_k$ of mass 2 $p_i.p_j$ which need not be zero, hence making the two quantities unequal.

†We describe the inclusive variant widely used currently, although there is an exclusive variant with a slightly different algorithmic evolution.

Therefore, for all three cases, the parameter R_{jet} determine the distance beyond i for which any particle j will not get combined into a jet.

Although R_{jet} is analogous to a radius parameter in cone algorithms, jets formed via the k_t and CA algorithms are irregularly shaped in the $y - \phi$ plane. This irregularity is due to the nature of the evolution of the combinatorial sequence determined by d_{ij}. For the k_t algorithm, soft particles get combined first and grow to the harder regions leading to unstable jet axes as one evolves further. The CA algorithm, oblivious to the p_T, starts from the nearest pairs and gradually moves to combine faraway particles. Although not as unstable as the k_t algorithm, it still produces irregularly shaped jets, as the jet axis can change dynamically as the jet grows. Note that the k_t measure d_{ij} is proportional to the square inverse of the splitting probability for a parton to split into two particles i and j in the collinear regions when either one of them is soft. On the other hand, the CA measure goes from combining particles with small angular separations to larger angular scales and thereby can look at multiple angular scales of emission in an event or a jet.

Unlike the k_t and CA algorithms, the anti-k_t algorithm gives almost conical jets in the $y - \phi$ plane since the combination starts from the hardest particles and gradually collects softer particles, leading to a very stable jet axis as one adds more particles. Its disadvantage is that the recombination sequence bears no connections with QCD. When needed, one can define anti-k_t jets and then recluster its constituents with k_t or CA algorithm to infer the underlying QCD evolution of the jet.

2.1.2 Jet substructure

Before the Large Hadron Collider, analysing hadronic final states at all colliders was almost exclusively done by taking the jets as single entities and examining their kinematic configurations with other jets or reconstructed objects like leptons and photons. However, the very high centre-of-mass energy of the collisions can result in the production of a significant number of electroweak scale particles in the boosted regime. Their decay captured within wider radius jets will generally have different energy patterns and evolution of the emitted particles than QCD jets. These features can be used as discriminants to enhance searches in the boosted regime. A significant example is the s-channel production of a vector boson and a Higgs boson via Higgs-strahlung processes via the mass-drop algorithm [97], which led to the observation [98, 99] of the Higgs' decay to a pair of bottom quarks. Due to the larger multiplicity of quarks, most heavy particles in the SM and any new BSM particle with democratic couplings to the quarks and leptons will decay predominantly to quarks. Moreover, the LHC can directly probe various multi-TeV scale particles hypothesised in different BSM models. These heavy particles' decay will naturally produce highly boosted SM particles. Therefore, looking into the substructure of large-radius jets provide novel ways

of looking at various SM and BSM physics at the LHC.

Since the literature is dynamic and extensive, we will concentrate on the essential ideas necessary to motivate the complexity of the subject. We describe the general idea behind prong finders which generally decluster a jet and try to find splittings with not too asymmetrical energy divisions, and generic observables designed to exploit the different radiation patterns arising from QCD jets and signal jets. The effect of underlying events (UE), which is important even for small radius jets, is even more pronounced for large-radius jets since their contribution grows with R_{jet} [100]. We also discuss some basic UE and pileup removal techniques which increase the performance of the prong finders and observables.

Prong finders and jet-shapes

The first proposal of using jet-substructure techniques, i.e. the mass-drop tagger, falls into a broader class of methods called prong-finders. These techniques use the dominance of mostly soft gluon emissions from the primaeval parton for QCD jets against the almost symmetrical parton-level decay dynamics of a heavy particle like the Higgs' decay to two bottom quarks. As an example, we discuss the modified Mass-Drop tagger [101], which has a small modification over the original mass-drop leading to better logarithmic behaviour. The algorithm follows the declustering sequence of the angular ordered CA algorithm with two parameters μ and y_{cut}, and proceeds as:

1. Undo the last clustering of the jet j to form two subjets and label them as j_1 and j_2 such that $m_{j_1} > m_{j_2}$.

2. If there is a significant mass-drop $m_{j_1} < \mu\ m_j$ and a fairly symmetric splitting
$$y = \frac{\min(p_{Tj_1}^2, p_{Tj_2}^2)\ \Delta R_{j_1 j_2}}{m_j^2} > y_{cut} \quad ,$$
then j is a tagged jet

3. If the above condition is not satisfied, replace j with the one from j_1 or j_2 which has a larger transverse mass $m_{j_i}^2 + p_{Tj_i}^2$ and start from the first step. If j_i is a single particle the jet is not tagged, and discarded.

The mass-drop tagger and the modified one were specifically designed to tag the Higgs' decay to a pair of bottom quarks. The principle for a prong finder to tag a heavy state decaying to more than two particles [102,103] have similar procedures for multiple declustering of the branches. Thus, such methods can be generalised to tag the decay of a heavy particle into some n number of strongly interacting particles. These n-particle decays give rise to the so-called n-prong jet signature, and a large portion of the literature on jet substructure revolves around finding ways to distinguish different pronged jets.

A complementary approach to tagging any generic n-prong hadronic decays of heavy particles is to study the *jet-shape* [104]–observables that are sensitive to the energy distribution of particles within a jet. Such observables are closely connected to event-shape observables in e^+e^- colliders. They need to satisfy infra-red and collinear safety to reduce sensitivity to low-energy and long-range physics. A standard example is the N-subjettiness [105] variable adapted from the inclusive event shape observable N-jettiness [106] at hadron collider environments. It requires defining N axes within the jet via some reclustering procedure. A straightforward choice is to take the N exclusive subjets obtained with the k_t algorithm. Once the axes are obtained, it is defined as

$$\tau_N = \frac{1}{d_0} \sum_k p_{Tk} \ \min\{\Delta R_{k1}, \Delta R_{k2}, ...\Delta R_{kN}\} \quad . \tag{2.3}$$

The distances $\Delta R_{ki} = \sqrt{(\eta_k - \eta_i)^2 + (\phi_k - \phi_i)^2}$ are calculated for each particle k with each of the candidate subjets, and the sum is over all particles. The normalisation factor is defined as $d_0 = \sum_k \ p_{Tk} \ R_{jet}$. It is evident that when $\tau_N \approx 0$, most of the radiation within the jet is aligned with the axes; therefore it will have N (or fewer) hard subjets. If $\tau_N >> 0$, then the N subjet axes fail to capture all the radiation within the jet pointing towards the jet having at least $N+1$ pronged hard subjets. Since a single N is not enough to ascertain the exact behaviour of the jets, often the N-subjettiness ratios $\tau_{N+1,N} = \tau_{N+1}/\tau_N$, which have more discriminatory power, are used in analyses. However, since the ratio is not continuous for $\tau_N \to 0$, it is IRC unsafe.[‡] Moreover, the naive selection of N exclusive subjets is sensitive to the recoild from emissions of soft particles outside the jet leading to complications in predicting their all-order behaviour. The winner-takes-all recombination scheme [107] guarantees that the axes are recoil-free.

Another important variable of interest which does not need any subjet reclustering are energy-correlation functions [108]. For hadron colliders, it is defined as

$$\mathrm{ECF}(N,\beta) = \sum_{i_1<i_2<....i_{N-1}<i_N \in J} \left(\prod_{a=1}^{N} z_{i_a}\right) \left(\prod_{b=1}^{N-1} \prod_{c=b+1}^{N} \Delta R_{i_b i_c}^{\beta}\right) \quad , \tag{2.4}$$

where $z_i = p_{Ti}/(\sum_{j\in J} p_{Tj})$. Clearly, $\mathrm{ECF}(N,\beta) = 0$ for any jet with less than N particles and IRC safe for any $\beta > 0$. Note that the angular exponent β introduces non-linearity in the Euclidean distances ΔR_{ij} on the rapidity-azimuth plane. Similar to the N-subjettiness, the ratio

$$r_N^{\beta} = \frac{\mathrm{ECF}(N+1,\beta)}{\mathrm{ECF}(N,\beta)} \quad , \tag{2.5}$$

[‡]They are, however, calculable with perturbative methods [60] as Sudakov factors exponentially suppress the singular regions and hence follow "Sudakov safety".

is helpful in discriminating $N+1$-prong jets from other jets of lower multiplicities.

Jet groomers

The preceding discussions have focussed on ways to discriminate different N-prong jets, assuming that the constituents follow the structure of the parton evolution described via perturbative QCD. Various non-perturbative effects and experimental conditions like underlying events and pileup inhibit the direct application of such techniques. There is a need to reduce their impact before applying the earlier methods to the large-radius jets. Jet grooming methods reduce the contribution from unwanted soft radiation keeping mostly the hard part that is likely to have originated from hard partons so that the jets have a better susceptibility to tagging techniques. They can be broadly divided into

- **Filtering:** It was originally proposed in reference [97] to sharpen the mass peak of the Higgs boson against the contribution of UE after applying the mass-drop tagger. The jet is reclustered again with the CA algorithm on a smaller angular R_{filt}, and only the hardest n_{filt} subjets are kept for analysing the mass distribution. Knowledge of the N-prong structure of the signal jet is required to apply this technique, where n_{filt} is generally taken to be $N+1$ to account for additional gluon radiation from the decayed partons.

- **Pruning:** It was proposed [109,110] as a bottom-up approach to removing soft and wide-angle radiation regardless of the prong structure of the jet. The original jet constituents are reclustered with the k_t or CA algorithm, and the procedure follows its declustering sequence. For each declustering $j \to j_1, j_2$ with $p_{Tj_1} > p_{Tj_2}$, we define $z = \min(p_{Tj_1}, p_{Tj_2})/p_{Tj}$ and $\Delta R_{j_1 j_2}$, and compare it with z_{cut} and $D_{cut} = m_J/p_{TJ}$, with m_J and p_{TJ} being the original jet's mass and transverse momentum respectively. If $z < z_{cut}$ and $\Delta R_{j_1 j_2} > D_{cut}$ remove j_2 otherwise keep both j_1 and j_2.

- **Trimming:** It was proposed in reference [111] as a generic method of removing contamination from non-relevant emissions like ISR, UE and pileup. The procedure involves reclustering the constituents of the jet with an algorithm with radius R_{trim} and keeping all those subjets with transverse momentum higher than a fraction f_{trim} of the original jet's transverse momenta, i.e. a given reclustered subjet s is removed if $p_{Ts} < f_{trim}\, p_{TJ}$, where p_{TJ} is the transverse momentum of the original jet.

The methods discussed here have undergone intensive scrutiny of their various experimental and theoretical properties. More improved versions are available that are currently being used by the experimental collaborations. The current landscape of various new techniques, including machine-learning methods, is presented in recent reviews [112,113] and lecture notes [114].

Jet images

Alongside the development of theoretical tools to study the structure of large-radius jets, there has been a growing impetus for using modern deep-learning techniques to identify hadronic decays of boosted heavy particles. We will describe the technical details of such algorithms based on Artificial Neural Networks in the next section. Here, we give a brief account of the benefits and drawbacks of representing large-radius jets as images [115–117] over the more conventional jet-substructure analysis techniques. These image-based representations were the first forrays of modern deep-learning methods into mainstream particle physics phenomenology, resulting in a dynamic and exciting area of enquiry.

Any quantity sampled on a two-dimensional grid can be represented as an image (of one channel). The pixel values denote the quantity's value in a particular bin of the x and the y-axes. Therefore, the jet constituents registered as calorimeter readings in the $\eta - \phi$ plane can be represented as a *"jet image"*. Even though the calorimeter cells measure the total energy, one would use its transverse component E_T for a hadronic environment. Convolutional Neural Networks (CNNs) trained to identify the decays of the W-boson [117], or the top quark [118], against jets of QCD origin have found a significant gain in performance compared to those using jet substructure methods described in the preceding sub-sections. These gains persist even in underlying events and pileup [119]. Despite the huge improvements, there is a general apprehension about these methods as they are poorly understood theoretically. Practically these translate to higher systematic uncertainties [120] in experimental applications. One might argue that the present scenario is similar to jet substructure techniques' situation upon their initial introduction and general acceptance after intensive scrutiny of their properties within QCD. However, an analytic understanding of these techniques is highly challenging [121], although numerical results [122, 123] point towards favourable behaviour against phenomenologically relevant factors.

2.2 Artificial Neural Networks

In the preceding section, we have seen the importance of examining the hadronic final states of processes to look for signatures of new physics and gave a brief account of Convolutional Neural Networks' power in segregating boosted hadronic decays of heavy particles from an overwhelming QCD background by looking at jet images. These networks fall within a generally larger class of models called Artificial Neural Networks (ANNs). ANNs are statistical learning models inspired by the biological structure of neurons [124, 125]. They are highly expressive statistical models capable of approximating a general class of function between two well-behaved spaces [126–129] and form the backbone of the modern artificial intelligence (AI) and deep-learning revolution. Before discussing specialised and deep architectures like Convolutional Neural Networks, in this section, we will

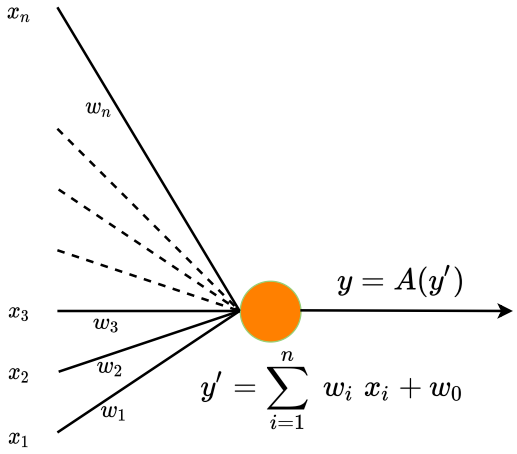


Figure 2.1: The figure shows a single perceptron (or a node) taking an n dimensional vector x_i. It consists of n weights w_i and a bias term w_0, via which we evaluate y'. A non-linear activation function $A(y')$ is applied to get the output y of the perceptron.

introduce the most basic networks: "Multilayer Perceptrons" (MLPs), their optimisation procedure, and some use cases that will be of interest in the subsequent chapters.

2.2.1 Multilayer Perceptrons

The fundamental building blocks of MLPs are perceptrons or colloquially known as nodes,[§] which take a vector $\mathbf{x}$ as input and apply an *affine* function $y' = \sum_{i=1}^{n} w_i x_i + w_0$, with $\{w_0, w_1, ...w_n\}$ the tunable parameters and $\{x_1, x_2, ...x_n\}$ the components of the input vector $\mathbf{x}$. To introduce non-linearities (required for better expressivity), one applies a non-linear function to get the output of the perceptron as $y = A(y')$. A single perceptron is diagrammatically shown in figure 2.1. The generalisation to an MLP requires two steps:

- increasing the dimensions of the output y by promoting it to a generic m dimensional vector $\mathbf{y}$
- introducing functional compositions to account for hidden layers between the input $\mathbf{x}$ and the output $\mathbf{y}$.

For the first step, we have each component of $\mathbf{y}$ as

$$y_j = A\left(\sum_{i=1}^{n} w_{ji}\ x_i + w_{j0}\right) \quad , \tag{2.6}$$

[§]Depending on the context, we will use nodes to mean either a perceptron or a graph's node in the subsequent discussions and following chapters.

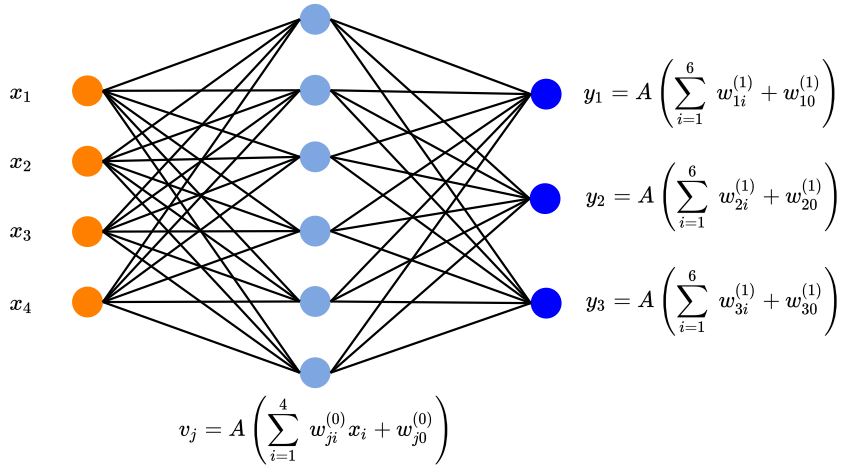


Figure 2.2: The figure shows a multilayer perceptron for a four dimensional input and three dimensional output with a single hidden layer with six nodes.

where w_{ji} is the component of an $m \times n$ dimensional weight matrix, and w_{j0} are the components of an m-dimensional bias vector. Note that such a generalisation introduces the concept of an input layer that takes the input vector $\mathbf{x}$ and an output layer that gives the output vector $\mathbf{y}$. Here, the input layer is directly connected to the output layer. These architectures do not yet possess the ability to universally approximate any function $f : \mathbb{R}^n \to \mathbb{R}^m$ with a compact domain and range, which require the second part of the generalisation.

To increase the expressive power of neural networks, one introduces hidden layers between the input and output layers, which, mathematically, are functional compositions of hidden vectors $\mathbf{v}^{(h)}$ between the input and output vectors, with h denoting the layer index. The input vectors feed into a hidden layer with outputs

$$v_i^{(1)} = A\left(\sum_{i=1}^{n} w_{ji}^{(0)}\, x_i + w_{j0}^{(0)}\right) \quad .$$

These $v_i^{(1)}$ can then produce the output y_i by replacing x_i in eq. 2.6, or feed into subsequent hidden layers. Note that the dimension of the vector $\mathbf{v}^{(1)}$ is not fixed and can be chosen to be an arbitrarily large but finite value. Such architectures with a single hidden layer with arbitrary width and some non-linear and non-polynomial activation possess the universal approximation property [127]. The generalisation to any MLP with K hidden layers is

$$v_i^{(k)} = A\left(\sum_{i=1} w_{ji}^{(k-1)}\, v_i^{(k-1)} + w_{j0}^{(k-1)}\right) \quad , \tag{2.7}$$

where k is the layer index with $v_i^{(0)} = x_i$ and $v_i^{(K+1)} = y_i$. Note that although we have used a generic notation A to denote the activation function, there is freedom in choosing the activation function for any layer. A graphical representation of

an MLP with a single hidden layer is shown in figure 2.2.

2.2.2 Optimisation

Having discussed the structure of MLPs, we now turn our attention to the optimisation procedure of the weights. The most common form of training is by reducing a loss function $L(\mathbf{y}_0, \mathbf{y}_t)$, which quantifies a faithful distance between the network output $\mathbf{y}_0$ and a target vector $\mathbf{y}_t$. Note that these target vectors can be the input vectors $\mathbf{x}$ as well, as we have not currently specified the nature of the learning process. If the loss function is differentiable, one can use gradient descent in the space of weights to reach an optimal position. In this section, we will describe the basics of back-propagation, taking a relatively simple approach, and discuss some practical aspects when training a neural network for inductive learning purposes.

Gradient descent and back propagation

To understand gradient descent, we concentrate on a simple linear regression between a dependent variable y and an independent variable x, with the sampled data consisting of ordered pairs $\{(x, y_t)_\alpha\}$, with $\alpha \in \{1, 2, ...N_{samples}\}$, being the sample index and $N_{samples}$ the data size. Let $y_0 = w_1 x + w_0$ and $L = (y_0 - y_t)^2$ denote the loss function. The procedure of finding the optimal weights w_i^0 via gradient descent involves an iterative update of each weight w_i as

$$w_i \leftarrow w_i - \gamma \frac{\partial \langle L \rangle}{\partial w_i} \quad , \tag{2.8}$$

where

$$\langle L \rangle = \frac{1}{N_{samples}} \sum_{\alpha=1}^{N_{samples}} L(y_0, y_t)|_{(x,y_t)_\alpha} \quad . \tag{2.9}$$

The constant factor γ is called the *learning rate* and determines the rate and the precision of the converged position. The process is shown geometrically in figure 2.3 by projecting on a single weight w_i with a quadratic loss function. In such a simple scenario with a two-dimensional weight space, the loss function has a global minimum accessible from every point in $(w_0, w_1) \in \mathbb{R}^2$. This very special circumstance is not generally required and seldom satisfied for the training of complicated neural networks. However, the loss function should be bounded from below for gradient descent to work.

The generalisation of gradient descent to a multi-layer feedforward network requires the implementation of a back-propagation algorithm [130]. It is a consequence of the functional compositions that the input vector $\mathbf{x}$ goes through to evaluate the output $\mathbf{y}$. For a single hidden layer network, we have

$$\mathbf{y}_0 = f_1(\mathbf{w}^{(1)}, \mathbf{v}^{(1)}) = f_1(\mathbf{w}^{(1)}, f_0(\mathbf{w}^{(0)}, \mathbf{x})) \quad , \tag{2.10}$$

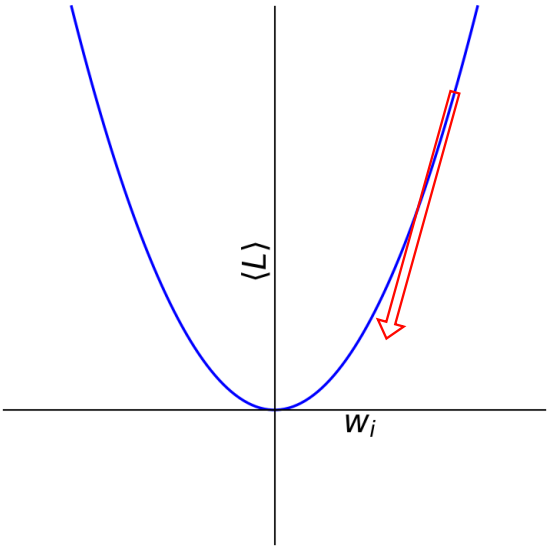


Figure 2.3: A geometrical representation of gradient descent for a quadratic loss function with a weight w_i.

where the vector functions $f_i : \mathbb{R}^{d_i} \to \mathbb{R}^{d_{i+1}}$ map between the relevant spaces, with definite dimensions d_i and d_{i+1}, and $\mathbf{w}^{(h)}$ represent the weights and the biases of the h^{th} layer. Therefore the loss function is of the form

$$L = L(\mathbf{y}_0, \mathbf{y}_t) = L(f_1(\mathbf{w}^{(1)}, f_0(\mathbf{w}^{(0)}, \mathbf{x})), \mathbf{y}_t) \quad .$$

We have the gradient descent for the weights $w_{ij}^{(h)}$ as

$$w_{ij}^{(h)} \leftarrow w_{ij}^{(h)} - \gamma \frac{\partial \langle L \rangle}{\partial w_{ij}^{(h)}} \quad . \tag{2.11}$$

The derivative of the loss function with respect to $w_{ij}^{(1)}$ and $w_{ij}^{(0)}$ are of the form

$$\begin{aligned}
\frac{\partial \langle L \rangle}{\partial w_{ij}^{(1)}} &= \frac{\partial \langle L \rangle}{\partial \mathbf{y}_0} \frac{\partial \mathbf{y}_0}{\partial w_{ij}^{(1)}} \quad , \\
\frac{\partial \langle L \rangle}{\partial w_{ij}^{(0)}} &= \frac{\partial \langle L \rangle}{\partial \mathbf{y}_0} \frac{\partial \mathbf{y}_0}{\partial \mathbf{v}^{(1)}} \frac{\partial \mathbf{v}^{(1)}}{\partial w_{ij}^{(0)}} = \sum_{k,l} \frac{\partial \langle L \rangle}{\partial w_{kl}^{(1)}} \frac{\partial w_{kl}^{(1)}}{\partial \mathbf{v}^{(1)}} \frac{\partial \mathbf{v}^{(1)}}{\partial w_{ij}^{(0)}} \quad .
\end{aligned}$$

Clearly, the second relation generalises for any hidden layer h and $h-1$ as

$$\frac{\partial \langle L \rangle}{\partial w_{ij}^{(h-1)}} = \sum_{k,l} \frac{\partial \langle L \rangle}{\partial w_{kl}^{(h)}} \frac{\partial w_{kl}^{(h)}}{\partial \mathbf{v}^{(h)}} \frac{\partial \mathbf{v}^{(h)}}{\partial w_{ij}^{(h-1)}} \quad . \tag{2.12}$$

The evaluation of the weight updates (eq. 7.3) of the hidden layers can be effi-

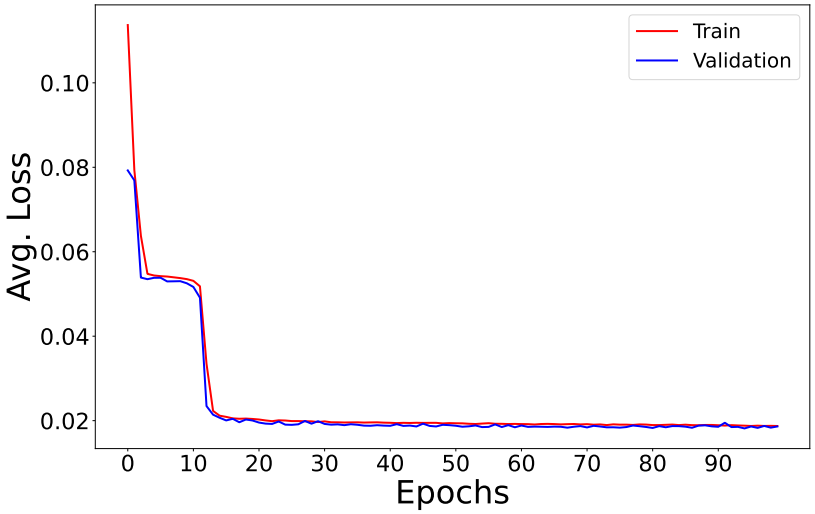


Figure 2.4: The figure shows the history of a neural network training for hundred epochs.

ciently implemented with a backward pass algorithm following eq 2.12. Such an evaluation of the gradient updates is known as the *back-propagation* algorithm and is used to train deep neural networks efficiently.

Practical considerations

In implementing back-propagation for the studies conducted in this thesis through deep-learning packages in python, we will generally use improved versions of gradient descent with velocity and momentum terms, which have better convergence properties over the vanilla gradient descent. While the discussions till now have been based on the gradient of the averaged loss function as defined in eq. 2.9, the number of training samples $N_{samples}$ is generally very large (of the order of a hundred thousand and sometimes millions). Therefore, it is computationally prohibitive to use the total dataset for updating the gradients. The training is done with small batches sampled uniformly from the training data with a fixed size $N_{batch} << N_{samples}$ to circumvent this issue. An epoch consists of a single pass over all training samples. Due to the large parameter space and dataset, the loss reaches an asymptotically small value[¶] after a large number of epochs.

For inductive purposes, since the model should be able to generalise to unseen data following the same underlying distribution, the training process involves a validation step with a separate dataset not used to update the gradients. After each training epoch, the model's performance is evaluated for this validation dataset. Comparing the averaged training loss to the average validation loss gives us an estimate of the network's generalisation capability. This feature is

[¶]This is, of course, dependent on the efficiency of the architecture to capture the data's underlying properties, which otherwise would not converge.

captured in the average training and validation loss plot against the number of epochs called the training history. An example is shown in figure 2.4, which ideally has an overlapping validation and training curve. More often than not, one would find that the training curve is lower than the validation curve, which indicates a degree of overfitting–the network has learnt specifics of the training data not present in the validation data. However, such a difference is generally tolerated as long as the validation curve continues to reduce. An extreme case of overfitting would occur when the training curve continues to decrease while the validation curve starts increasing after reaching a minimum point. The training is generally stopped once such behaviour is found in the training history.

The value of the learning rate γ is another crucial aspect when training a neural network. A larger value would generally have faster convergence but a higher loss value in the converged plateau of the training history. On the other hand, a lower learning rate would converge slowly but will reach a lower plateau. A reduce-on-plateau condition generally combines the favourable behaviour of both these situations by starting with a relatively larger learning rate but reducing it every time on reaching a plateau.

Another important aspect of neural network training and inference is its uncertainties arising from the finite training size and noise associated with the data-taking process. Although the statistical uncertainties associated with the training size reduce with increasing data size, noise in the training data translates as a source of systematic uncertainty in the training process. Bayesian Neural Networks [131, 132] can estimate such uncertainties by providing per-sample uncertainty estimates and have been studied in the context of LHC [133, 134]. Another way of assessing the uncertainties widely used by the various LHC collaborations [135, 136] is by using bootstrapping [137] methods. In the simplest scenario, the statistical uncertainties are estimated from multiple datasets (called bootstrap samples) generated by random sampling (with replacement) from the nominal test dataset.

In the following sections, we will look at some specific applications of ANNs, concentrating on supervised classification and unsupervised anomaly detection techniques which are used in the remainder of the thesis. We note that these are but a small part of their applicability, and there are different scenarios like the generation of events [138–147], simulation of detector response [148–152], and pileup mitigation [153–155] which generally proceeds through adversarial training [156], normalising flows [157], or regression methods.

2.2.3 Supervised Classification

One of the most prevalent uses of neural networks is in various classification tasks like pattern recognition and image classification. Such classification scenarios naturally exist in multiple areas of LHC. Therefore, it is not surprising that there is a major focus on applying neural networks to such classification tasks. We

discuss the general formulation of such classification tasks in this section.

In supervised classification, we have a dataset of two tuples

$$\mathcal{D} = \{(\mathbf{x}, \mathbf{y}_t)_\alpha\} \quad |\mathcal{D}| = N_{samples} \quad ,$$

with $\mathbf{x}$ the input vector to the network, $\mathbf{y}_t$ giving the class information, and α the sample index. The most popular method of embedding the class information in a vector is the *one-hot* encoding. In this method, $\mathbf{y}_t$ has the same dimensions as the number of classes N_{class}, with zero entries in all but one dimension. The position of the non-zero element gives the class of each sample, and all samples belonging to the same class have the non-zero component at the same index. The output of the network $\mathbf{y}_0$, therefore, is made to correspond to an N_{class} dimensional vector whose components $y_{0,i}$ follows the required probability normalisation

$$\sum_{i=0}^{N_{class}} y_{0,i} = 1 \quad .$$

Although this is the only essential requirement on the output, simple activation functions like a linearly normalised vector $\mathbf{y}_0$ with components

$$y_{0,i} = \frac{\hat{y}_{0,1}}{\sum_{i=1}^{N_{class}} \hat{y}_{0,i}} \quad ,$$

with $\hat{y}_{0,i}$ the components of the network output before normalisation is highly sensitive to outliers in the data. The `SoftMax` activation function, which is a multidimensional generalisation of the sigmoid function $\sigma(z) = 1/(1 + e^{-z})$, renders the output insensitive to outliers in the data and is, therefore, the preferred choice of output activation for classification tasks. It is given as,

$$y_{0,i} = \frac{e^{-\hat{y}_{0,i}}}{\sum_{j=1}^{N_{class}} e^{-\hat{y}_{0,j}}} \quad . \tag{2.13}$$

Note that it has $N_{class} - 1$ degrees of freedom due to the normalisation condition.

The preferred loss function for training supervised classification networks is the cross-entropy loss. The cross-entropy between two probability distributions $\mathbf{y}_0(\mathbf{x})$ and $\mathbf{y}_t(\mathbf{x})$ is defined as,

$$L = -\sum_{\mathbf{x}\in\mathcal{B}} \sum_i y_{t,i}(\mathbf{x}) \ \ln(y_{0,i}(\mathbf{x})) \quad , \tag{2.14}$$

where the distributions are functions of the feature-vector $\mathbf{x}$, and $\mathcal{B}$ denote the batch of training data. It is a measure of how well a modelled distribution $\mathbf{y}_0$, corresponding to the network output, resembles the true distribution of $\mathbf{y}_t$, the true values provided during training. For a fixed true-distribution $\mathbf{y}_t$ with a sample space $\mathcal{X}$, minimising the cross-entropy essentially minimizes the KL-

divergence [158],

$$D_{KL}(\mathbf{y}_t||\mathbf{y}_0) = \sum_{\mathbf{x}\in\mathcal{X}} \mathbf{y}_t(\mathbf{x})\ \ln(\mathbf{y}_t(\mathbf{x})) - \sum_{\mathbf{x}\in\mathcal{X}} \mathbf{y}_t(\mathbf{x})\ \ln(\mathbf{y}_0(\mathbf{x})) \quad ,$$

which is a measure of the similarity between two distributions and becomes zero iff they are identical.

2.2.4 Unsupervised learning

We have laid down the basics of supervised classification with neural networks in the preceding discussions. Although such techniques are highly effective, their application requires at least one signal hypothesis through which the training proceeds to distinguish them from background processes. Since all model-dependent searches at the LHC have returned null results, we require broad-ranging model-independent investigations that would look for hidden clues in a large phase space volume. These searches where we look into the background-only hypothesis without having any particular signal in mind can be accommodated within unsupervised learning techniques. Broadly speaking, while supervised learning looks to find the best boundary between at least two overlapping[‖] probability distributions in the underlying space of $\mathbf{x}$, unsupervised methods try to learn the distribution themselves.

We can distinguish between the two by taking a simple example of binary classification of a hypothetical signal distribution $p_S(\mathbf{x})$ and background $p_B(\mathbf{x})$, where $\mathbf{x} \in \mathcal{X}$ is the input vector to an ML model, and $\mathcal{X}$ is the underlying space. If $\mathcal{S} \subseteq \mathcal{X}$ and $\mathcal{B} \subseteq \mathcal{X}$ denote the support of the distributions $p_S(\mathbf{x})$ and $p_B(\mathbf{x})$, respectively, then $\mathcal{S} \cap \mathcal{B} \neq \emptyset$. A supervised classification model $f(\Theta, \mathbf{x})$ amounts to finding an optimal point Θ_0 in the weight space, such that the function $f(\Theta_0, \mathbf{x})$ approximates some monotonic function of the likelihood ratio for the training dataset, which is the optimal classifier via the Neyman-Pearson lemma [159]. Therefore, the network output in such a case is not a simple projection of the probabilities $p_S(\mathbf{x})$ or $p_B(\mathbf{x})$.

On the other hand, an unsupervised learning model tries to learn the underlying distribution

$$p(\mathbf{x}) = \omega_S\ p_S(\mathbf{x}) + \omega_B\ p_B(\mathbf{x})$$

in the sample space $\mathcal{X}$, where ω_S and ω_B are weight factors determined by their relative occurrence. Since the cross-section of the background would be orders of magnitude higher than most probable signals, which will determine their relative weights, we have

$$\omega_B >> \omega_S \implies p(\mathbf{x}) \approx p_B(\mathbf{x}) \quad .$$

[‖] The problem of statistical inference does not arise when we have probability distributions with non-overlapping support and can be classified into different classes with perfect efficiency, by looking at the values of the observables $\mathbf{x}$ in $\mathcal{X}$.

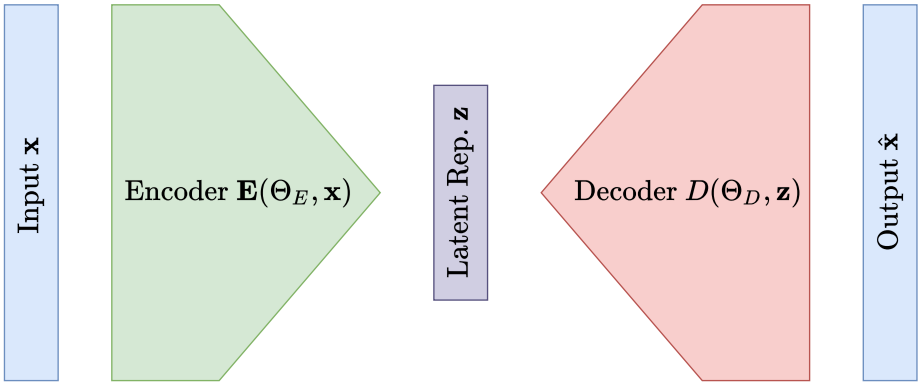


Figure 2.5: The figure shows a schematic representation of an autoencoder. The encoder (shown in green) maps the input vectors $\mathbf{x}$ to a latent representation $\mathbf{z}$ of reduced dimensionality, while the decoder (shown in light red) maps it back to the reconstructed vector $\hat{\mathbf{x}}$ of the same dimensions as the output. These steps can be achieved by specific architectures dependent on the data representation, like convolutional architectures and graph neural networks. We explore graph autoencoders and quantum autoencoders in this thesis.

Therefore, one can train on the background data in unsupervised learning to a good approximation. This is in stark contrast to supervised learning, where one takes a balanced dataset to train classification models to ensure that the optimisation procedure democratically picks up both classes' features. An unsupervised learning model $f(\Theta, \mathbf{x})$ tries to learn a representation of the underlying distribution $p(\mathbf{x})$.

2.2.5 Anomaly detection with autoencoders

Autoencoders [160] are neural networks utilised in various applications of unsupervised learning. They learn to map input vectors $\mathbf{x}$ to a *compressed* latent vector $\mathbf{z}$ via an encoder. This latent vector feeds into a decoder that reconstructs the inputs. Denoting the encoder and decoder networks as $\mathbf{E}(\Theta_E, \mathbf{x})$ and $\mathbf{D}(\Theta_D, \mathbf{z})$ with Θ_E and Θ_D denoting the learnable parameters of the respective network, we have

$$\mathbf{z} = \mathbf{E}(\Theta_E, \mathbf{x}) \quad , \quad \hat{\mathbf{x}} = \mathbf{D}(\Theta_D, \mathbf{z}) \quad , \tag{2.15}$$

where $\hat{\mathbf{x}}$ denotes the reconstructed output vector. The whole network is trained via gradient descent to reduce a faithful distance L between the reconstructed output $\hat{\mathbf{x}}$ and the input vector $\mathbf{x}$. For instance, L can be the root-mean-square error (RMSE),

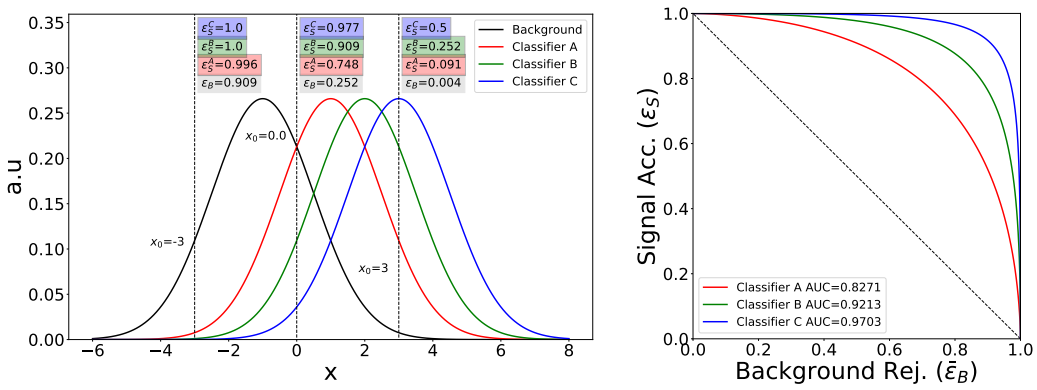
$$L(\mathbf{x}, \hat{\mathbf{x}}) = \sqrt{\frac{\sum_{i=1}^{i=n} (\hat{x}_i - x_i)^2}{n}} \quad , \tag{2.16}$$

where $\hat{x}_i$ and x_i are the i^{th} component of the reconstructed and input vectors, respectively, and n is their dimension. A faithful encoding should have an optimal

latent dimension $k < n$, with k being the intrinsic dimension of the data set. This *dimensionality reduction* is crucial in many applications of autoencoders, which otherwise learn trivial mappings to reconstruct the output vectors $\hat{x}$. Unsupervised learning deals with learning probability distributions, and properly trained autoencoders are excellent for many applications. A schematic representation of an autoencoder is shown in figure 2.5. As described in the preceding section, the loss function $L : \mathcal{W} \times \mathcal{X} \to \mathbb{R}$, where $\mathcal{W}$ denote the weight space $(\Theta_E, \Theta_D) \in \mathcal{W}$, is a one-dimensional projection of the probability distribution $p(\mathbf{x})$.

One popular usage of autoencoders in collider physics is anomaly detection [161–170]. In various scenarios at the LHC, the background processes' contributions are orders of magnitude larger than most viable signals. However, a plethora of possible signal scenarios exist that could be realised in nature, making it unlikely that the signal-specific reconstruction techniques of supervised learning methods comprehensively cover all possible scenarios. This motivates unsupervised anomaly detection techniques, wherein a statistical model learns the probability distribution of the background to classify any data not belonging to it as anomalous (signal) data. Using an autoencoder as an anomaly detector, we train it to reconstruct the background data faithfully. Many signals have a higher intrinsic dimension than background data due to their increased complexity. Hence, they incur higher reconstruction losses. Thus, the loss function can be used as a discriminant to look for anomalous events.

2.2.6 Performance metrics



(a) (b)

Figure 2.6: The figure shows the relation between the separation of the distribution of two classes (left) and the corresponding ROC curve (right) for a binary classifier. As the separation increases, the distance of the ROC curve increases from the black line, which indicates completely overlapping distributions.

In the preceding discussions, we have reviewed most of the essential ingredients

of machine learning with artificial neural networks. Before discussing specialised architectures in the next section, we briefly present the performance metrics used to compare various machine learning models. Since the main emphasis of the thesis would be on unsupervised or supervised segregation of signal and background events, we will describe those metrics used to compare binary classification models. Thus, we will exclusively work with one-dimensional probability distributions. Note, however, that such a distribution is obtained as an output of a model $f(\Theta_0, \mathbf{x})$ by mapping a very high dimensional input $\mathbf{x}$ where the optimal parameters Θ_0 maximally address the particular aim of the training, and hence the inference. Moreover, since we have the class information from the Monte-Carlo simulations, we will work with metrics that assume such knowledge.

Let $p_S(y)$ and $p_B(y)$ denote the normalised probability distribution of a single variable y. In a supervised model y can be the binary classification score, while for anomaly detection with autoencoders, it can be the loss function. Regardless of the nature of y, we define the signal acceptance and background acceptance based on the signal rich region in $y \in \mathbb{R}$ by evaluating the median values $\tilde{y}_S$ and $\tilde{y}_B$, such that $\int_{-\infty}^{\tilde{y}_X} dy\, p_X(y) = 0.5$, for p_S and p_B. If $\tilde{y}_B < \tilde{y}_S$, then we define the signal acceptance $\epsilon_S = f_S(T_0)$ and the background acceptance $\epsilon_B = f_B(T_0)$ as

$$f_S(T_0) = \int_{-\infty}^{T_0} dx\, p_S(x) \quad , \quad f_B(T_0) = \int_{-\infty}^{T_0} dx\, p_B(x) \quad . \tag{2.17}$$

When $\tilde{y}_B > \tilde{y}_S$, one would change the limits from T_0 to infinity. The signal and background acceptance quantify the fraction of selected signal and background, respectively, for a threshold T_0 on the value of observable y, while the ordering of the median values determines which side of the distribution one should keep. Therefore, the background rejection $\bar{\epsilon}_B$ is simply defined as

$$\bar{\epsilon}_B = \bar{f}_B(T_0) = 1 - f_B(T_0) \quad .$$

One would ideally want the signal acceptance to be always larger than the background acceptance for all thresholds. If $\epsilon_S < \epsilon_B$, for all values of thresholds, it implies an incorrect identification of the signal rich regions, and one would get a correct ordering ($\epsilon_S > \epsilon_B$), once it has been fixed. However, an undesirable situation would be when $\epsilon_S \approx \epsilon_B$ for all threshold values, which would imply that the two probability distributions are virtually overlapping and the classifier is not able to distinguish between the two. For a fixed signal and background, a better classifier would have a higher signal acceptance for the same background acceptance (or rejection).

As one can see, the dependence of the signal acceptance ϵ_S on the background acceptance or rejection quantifies the ability of a classifier to segregate the signal from the background. This dependence can be diagrammatically shown in a plot between the signal acceptance and the background acceptance or rejection by

using the threshold to connect the two quantities of interest, i.e.

$$\epsilon_S = f_S(T_0) = f_S(f_B^{-1}(\epsilon_B)) = f_S(\bar{f}_B^{-1}(\bar{\epsilon}_B)) \quad . \tag{2.18}$$

For historical reasons, such a plot between the signal acceptance and the background acceptance or rejection is known as a Receiver-Operator-Characteristics (ROC) curve. The area under the ROC curve (AUC) is an integrated quantity useful in comparing various classifiers. An AUC of 0.5 would mean $\epsilon_S = \epsilon_B$ for all values of thresholds, while a value of 1.0 would mean $\epsilon_S = 1$ and $\epsilon_B = 0.$ for all values of thresholds. Therefore, a classifier with a higher value of AUC performs better than those that have a lower AUC.

A diagrammatic representation of three hypothetical classifiers A, B, and C for the same signal and background is shown in figure 2.6. For a simple comparison, we take the background distribution of all three classifiers to coincide, while the separation increases from classifiers A to B, and B to C, as shown in figure 2.6(a). The ROC curve of these three classifiers shown in figure 7.6, reflects the increase in performance, with the curve closest to the upper top corner performing the best, which is evident also from the value of the AUC.

2.3 Deep-learning on high-dimensional raw data

In the previous section, we have discussed the essential idea behind the architecture and training of multilayer perceptrons. Such multilayer perceptrons have a long history in particle physics phenomenology, well before the LHC era, along with other shallow machine learning techniques like boosted decision trees. The major difference between these algorithms and the current influx of deep-learning algorithms is in the dimensionality of the input vector $\mathbf{x}$ and, consequently, the design of the architectures used to handle such high-dimensional data. First, we discuss the differences and similarities between such shallow machine learning techniques and the modern deep-learning methods. We then describe the details of Convolutional Neural Networks and Graph Neural Networks, the algorithms used in the studies presented in the subsequent chapters.

2.3.1 Looking at high-dimensional phase space

It is well known that multivariate analyses of different physically constructed variables outperform a traditional cut-based approach. The reason ascribed to such an increase is the formulation of non-linear cut boundaries in the multidimensional space spanned by the various quantities in the input vector. The variables used in such multivariate analyses are highly specific and based on physical intuition for the particular type of signal and background. Such domain-specific variables (like the dijet mass), which are obtained after significant processing of raw data (particle four-vectors in the present case), are called *high-level* vari-

ables. In contrast, the unprocessed quantities are called *low-level* variables. The underlying cause of a performance gain from a cut-based approach to a high-level multivariate approach is similar, in principle, to the improvements found in employing deep-learning algorithms with low-level high-dimensional data over a multivariate analysis of high-level features. In other words, with multivariate methods, one can directly look at the multidimensional distribution of the various high-level features. Similarly, deep-learning algorithms can look into the very high dimensional phase space of the particles measured by the detector and pick up the underlying features directly.

The gain in performance with deep-learning algorithms is made possible mainly by two factors: (1) the highly improved versions of gradient descent optimisation currently available and (2) the exploitation of inherent features in the data by appropriate representations and designing architectures that can exploit the underlying features efficiently. The first point is crucial since the power of neural networks to approximate functions, although known through various universal approximation theorems [126–128], most, if not all, are existence-theorems stating that such approximators exist, without any hints of obtaining or designing a practical approximator. Highly improved gradient descent algorithms like `Adam` [171] or `Nadam` [172] ensure a relatively fast training of neural networks with a huge parameter space.

In statistical learning terminology, the design of architecture and data representation to bring out particular features in the data is known as building "inductive biases". Such inductive biases help effectively approximate functions by favouring certain minima over others or even restricting the nature of the obtainable minima itself during the gradient descent optimisation. As an example, many jet-shape observables which discriminate between different n-prong jets are some p_T weighted non-linear functions of the Euclidean distance ΔR_{ij} in the $\eta-\phi$ plane. Therefore, this inherent Euclidean structure in the jets can be aptly represented as jet images and efficiently extracted with convolutional architectures. In general, such designs are advantageous in fundamental physics because we know the properties of the underlying distribution even though analytic expressions are not always feasible, which can be exploited by building physical biases and symmetries into the architecture and the data representations. Many recent works [173–183] have started to explore such directions with exciting results and new insights into the workings of neural networks.

The power of deep-learning algorithms to efficiently extract features from data directly is unparalleled. However, we rely on an extensive simulation program through various Monte-Carlo procedures to study the properties of neural networks. Although their actual realisation would be based on a complex validation process with experimental data, there is a possibility of the algorithms learning features of the imperfect simulations. In any analysis, including multivariate or a cut-based approach, we have to invariably deal with the imperfections in the simulations in the data from the underlying theory predictions, which trans-

lates as a source of systematic uncertainty in the result. Modern deep-learning algorithms that take low-level inputs suffer from larger systematics [184] since they look into more subtle differences in the data. However, their relative gain in performance compensates for the higher uncertainties, although it would be favourable to reduce the uncertainties for precision analyses with more data in the future LHC runs. One such source of uncertainty is the relative stability of the network output to soft and collinear emissions, which we will work upon in chapter 5.

In the remainder of this chapter, we discuss the inbuilt inductive biases and the data representation for two important types of architectures and explain their widespread applicability at LHC, including a brief outline of the available public packages for implementing these algorithms.

2.3.2 Convolutional Neural Networks

Convolutional Neural Networks [185] are deep-learning models inspired by the brain's visual cortex [186]. They are powerful models capable of image classification and segmentation and have been widely used in various fields (see references [187, 188] for recent reviews). Here, we present its basic structure and the inductive biases that one implicitly assumes when using Convolutional Neural Networks (CNNs). We will exclusively work with two-dimensional images since most uses of CNNs are restricted to the two-dimensional case. The performance of CNNs can be attributed to two steps:

- **Convolution on local regions:** the network executes convolution operation with several filters with significantly smaller dimensions than the image, which is shared for the whole image, followed by a non-linear activation
- **Pooling:** a downsampling operation which progressively reduces the dimensionality of the data by taking a summary statistics of a local region forward to the next layer.

We will outline the basics of these two steps and the biases they generate in the following paragraphs. To understand the properties of convolution operation within perturbative QCD (which will be discussed in the next subsection), we will discuss them in the continuum limit by neglecting the activation function and bias terms. Doing so simplifies the expressions through which we can infer interesting behaviours in the soft and collinear regions of the multi-particle phase space.

Convolution operation: A two-dimensional image is some $N_r \times N_c$ matrix of real-valued elements $F_{ij} \in \mathbb{R}$, with N_r and N_c denoting the number of rows and columns, respectively. Relevant to the nature of calorimetric measurements, F_{ij}

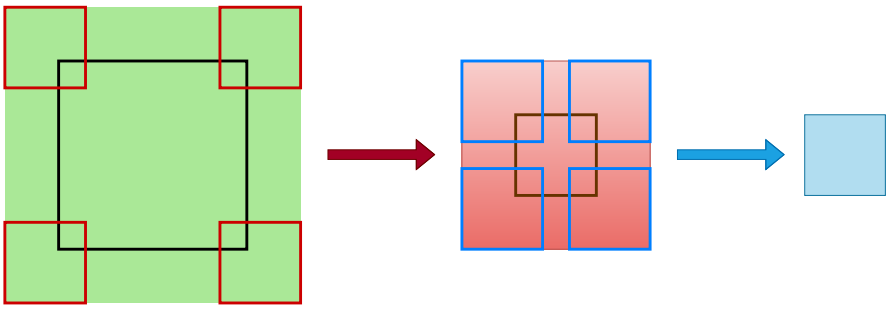


Figure 2.7: The figure shows a diagrammatic representation of the effective region (shown as a solid greed box) in the input image for a second convolution on the first feature map. The result of the convolution on the green region of the image with the filters (shown as hollow red squares) produces a part of the feature map shown as a solid red box. Similarly, a second convolution on the region of the feature map with the different filters of the same size produces a region of a new feature map. Therefore, the second filters learn the features corresponding to a much larger area (determined by the relative size of the filters) in the original image.

can be mathematically defined as the binned value of an observable of bin size $(\Delta x, \Delta y)$

$$F_{i,j} = F(x_i, y_i) \coloneqq \int_{x_i-\Delta x/2}^{x_i+\Delta x/2} dx \int_{y_i-\Delta y/2}^{y_i+\Delta y/2} dy\ f(x,y) \quad , \tag{2.19}$$

where $f(x,y)$ denotes the underlying continuous function for the observable. Let (x_0, y_0) be the central point in the image. It is the central value of the central bin when N_c and N_r are odd, while it is the upper boundary of the $(N_r/2)^{th}$ or $(N_c/2)^{th}$ bin in case they are even. The region of the two-dimensional Euclidean plane $\mathbb{R}^2$ where we sample the image can be written as,

$$\mathcal{F}(x_0, y_0) = \left[x_0 - N_c\frac{\Delta x}{2}, x_0 + N_c\frac{\Delta x}{2}\right] \times \left[y_0 - N_r\frac{\Delta y}{2}, y_0 + N_r\frac{\Delta y}{2}\right] \subset \mathbb{R}^2 \ . \tag{2.20}$$

It is straightforward to describe the convolution operation[**] with a filter $w^a_{i,j}$ with size $n_r \times n_c$ with a denoting the filter index as

$$G^a_{k,l} = \mathbf{A}\left(\sum_{i,j} \omega^a_{i,j}\ F_{k-i,l-j} + w^a_0\right) \quad , \tag{2.21}$$

where $\mathbf{A}$ is the activation function. The matrices G^a_{ij} are commonly referred to as *feature maps*. Neglecting the bias term w^a_0 and taking a linear activation, this

[**]Introducing strides overcomplicates matters since, we will be solely using convolution operations with single stride.

can be written in the continuous form as

$$g^a(x,y) = \int_{x-n_c\frac{\Delta x}{2}}^{x+n_c\frac{\Delta x}{2}} dx' \int_{y-n_r\frac{\Delta y}{2}}^{y+n_r\frac{\Delta y}{2}} dy'\; w^a(x',y') f(x-x', y-y') \quad . \tag{2.22}$$

Let us denote the region of the integration in $\mathbb{R}^2$ as

$$\mathcal{R}(x,y) = \left[x - n_c\frac{\Delta x}{2}, x + n_c\frac{\Delta x}{2}\right] \times \left[y - n_r\frac{\Delta y}{2}, y + n_r\frac{\Delta y}{2}\right] \subset \tilde{\mathcal{F}}(x_0,y_0) \subset \mathbb{R}^2 \quad , \tag{2.23}$$

which is a fixed neighbourhood of a finite area around (x,y). The region $\tilde{\mathcal{F}}(x_0,y_0)$ is an expansion of $\mathcal{F}(x_0,y_0)$ given as

$$\begin{aligned}\tilde{\mathcal{F}}(x_0,y_0) = &\left[x_0 - (N_c+n_c)\frac{\Delta x}{2}, x_0 + (N_c+n_c)\frac{\Delta x}{2}\right] \\ &\times \left[y_0 - (N_r+n_r)\frac{\Delta y}{2}, y_0 + (N_r+n_r)\frac{\Delta y}{2}\right] \subset \mathbb{R}^2 \ ,\end{aligned}$$

and we assume that $f(x,y) = 0$ when (x,y) falls outside of $\mathcal{F}(x_0,y_0)$, colloquially referred to as zero-padding. Note that $g^a(x,y)$ is defined for every point in $\mathcal{F}(x_0,y_0)$ and not $\tilde{\mathcal{F}}(x_0,y_0)$. It is now easier to infer the biases that eq. 2.22 places on the function $g^a(x,y)$, without loss of generality in the discrete case. In analogy to G^a_{ij}, we will refer to $g^a(x,y)$ as the feature function in the following discussion.

Local connectivity: The region of integration $\mathcal{R}(x,y)$ implies that the feature function $g^a(x,y)$ is dependent on the values of $f(x,y)$ in a small neighbourhood determined by the filter size. Such a quality is known as local connectivity, and it effectively decouples the amalgamation of information from $f(x,y)$ into $g^a(x,y)$ into a local scale, determined by the filter size.

Parameter sharing: The sharing of the filter by the whole image imposes a periodic boundary condition on the functions $w^a(x,y)$ as

$$w^a(x,y) = w^a(x + n_c\Delta x, y) \quad , \quad w^a(x,y) = w^a(x, y + n_r\Delta y) \quad .$$

This condition translates to the inherent assumption that learning a function $w^a(x,y)$ amounts to extracting features common to all regions of size $\mathcal{R}(x,y)$ in $f(x,y)$.

Separation of Scales: This results as a consequence of the local connectivity and the sequential application of convolution operations on the feature function $g^a(x,y)$. We can write down the convolution operation of $g^a(x,y)$ with new sets

of filters $u^b(x, y)$

$$h^{(b,a)}(x, y) = \int_{\mathcal{R}(x,y)} ds'\ u^b(x', y')\ g^a(x - x', y - y') \quad , \tag{2.24}$$

where $ds' = dx'dy'$. For simplicity, we have taken the size of the new filters to be the same as $w^a(x, y)$, making $\mathcal{R}$ the region of integration in $\mathbb{R}^2$. However, the new feature functions $h^{(b,a)}(x, y)$ is dependent on the input image $f(x, y)$ of a larger area than $\mathcal{R}$. The reason behind this lies in the integration of the feature map $g^a(x, y)$ to determine $h^{(b,a)}(x, y)$. The feature map $g^a(x, y)$ at the extremal points of the region, say $(x_b, y_b) = (x - n_c\frac{\Delta x}{2}, y - n_r\frac{\Delta t}{2})$ to determine $h^{(b,a)}$ is dependent on the value of the input image $f(x, y)$ in the neighbourhood $\mathcal{R}(x_b, y_b)$ (c.f. eq 2.22). This is true for any (x_b, y_b) at the boundary of $\mathcal{R}(x, y)$, which is figuratively shown in figure 2.7, where the effective area in the input image for the output $h^{(b,a)}(x, y)$ in the blue square is given by the red square. Therefore, the successive application of convolution operation separates the problem of extracting features to different length scales in the input image $f(x, y)$. The effective range of correlations that a filter can extract increases as one increases the number of convolution operations.

Pooling: The success of CNNs depends on another important operation which downsamples a feature map G^a_{ij}. Such operations, called pooling operations, replace the value of the element G^a_{ij} by a function of its neighbouring values. The size of the neighbour is called the pool size, and popular choices include taking the maximum, average or sum of the neighbourhood. A pooling operation reduces the dimensionality by taking only a summary over a larger area. A more aggressive dimensionality reduction can be achieved if we pool over the image with more than a single stride. In the continuous limit, one can write a max-pooling operation as

$$p^a(x, y) = \max_{(x',y')\in\mathcal{R}_p(x,y)} g^a(x', y') \quad , \tag{2.25}$$

while an average pooling operation can be written as,

$$p^a(x, y) = \frac{\int_{\mathcal{R}_p(x,y)} dx'dy'\ g^a(x', y')}{\int_{\mathcal{R}_p(x,y)} dx'dy'} \quad . \tag{2.26}$$

The neighbourhood $\mathcal{R}_p(x, y)$ is defined in analogy to $\mathcal{R}(x, y)$ (eq. 2.23) for a pool size of say $m_r \times m_c$.

2.3.2.1 Calorimeter Images

In the preceding discussions, we have laid down the basic concept of convolutional architecture and its inductive biases. Here, we define calorimeter images and look at some of their properties from the perspective of perturbative QCD. A calorimeter image in some connected region in the (η, ϕ) plane can be defined in

analogy to eq 2.19 as

$$F_{i,j} = F(\eta_i, \phi_i) \coloneqq \int_{\eta_i - \frac{\Delta\eta}{2}}^{\eta_i + \frac{\Delta\eta}{2}} d\eta \int_{\phi_i - \frac{\Delta\phi}{2}}^{\phi_i + \frac{\Delta\phi}{2}} d\phi \; p_T(\eta, \phi) \quad , \tag{2.27}$$

which is sampled with a resolution $(\Delta\eta, \Delta\phi)$. The convolution operation now becomes,

$$g^a(\eta, \phi) = \int_{\mathcal{R}(\eta,\phi)} d\eta' d\phi' \; w^a(\eta', \phi') \; p_T(\eta - \eta', \phi - \phi') \quad . \tag{2.28}$$

From this expression, one can infer that the feature functions $g^a(\eta, \phi)$ for any generic filter $w^a(\eta, \phi)$ is not sensitive to the soft radiation when $p_T \to 0$. Since CNNs are connectionist models where all following layers are sequentially dependent on $g^a(\eta, \phi)$, the output will therefore be robust to soft emissions. This has been numerically verified in reference [123].

For exactly collinear emissions, the calorimeter resolution $(\Delta\eta, \Delta\phi)$ provides a natural cutoff. However, one would not want the output to be highly sensitive to the experimental resolutions. The pooling operation whose output is tolerant to small deformations in the data should make the network output less susceptible to small angle (but not exactly collinear) emissions. A max-pooling operation will be more sensitive to such effects since emissions from the hardest particles determining the maximum value of $g^a(\eta, \phi)$ in the neighbourhood will change the maximum value. However, an average pooling operation (or a sum pooling) would be more resilient since the p_T sum would not be significantly affected in a particle's neighbourhood with such a splitting. A CNN is not completely impervious to such effects since a splitting that changes the value of η and ϕ of the daughter particles will cause a change in the value of the filter $w^a(\eta, \phi)$ while evaluating $g^a(\eta, \phi)$ with eq. 2.28.

The other properties of CNNs like local connectivity, parameter sharing, and scale separation are good approximations of QCD behaviour. The parton shower structure in the collinear limits mandates that particles are closely related to other particles in their immediate vicinity. Although wide-angle soft gluon emissions are important in determining the colour flow in an event, they are sensitive to non-perturbative multi-parton interactions. They are not as well-controlled as the collinear regions, and one would not want the algorithm to be highly susceptible to such effects for general searches. The universality of parton emissions makes it favourable to share the weights to pick up their features irrespective of the position. Moreover, identifying hard prongs within jets or jets within an event requires looking at different scales in the image, which is naturally done with sequential convolutions. Due to their favourable biases in identifying QCD radiation patterns, CNNs have been applied to various supervised [117–120, 122, 167, 189, 190] and unsupervised [167, 169, 170, 191] jet-tagging tasks and signal event classification [192–196] scenarios.

2.3.2.2 Drawbacks of CNNs in LHC phenomenology

We have seen in the preceding discussions that CNNs and images have inductive biases innately suitable for differentiating QCD radiation patterns. However, they have some disadvantages in identifying various other factors of QCD and event kinematics beyond the Euclidean nature that it presumes. This section will highlight some of the issues that limit the practical application of CNNs to LHC phenomenology.

The first disadvantage is the sparsity of the calorimeter images, which makes their use waste a lot of computational power. The number of non-zero pixels in a calorimeter image of $N \times N$ dimensions is approximately $O(N)$. For a typical jet-image size of 32×32, only about 3% of the total pixels are non-zero. This fraction decreases as one increases the image size. This also prohibits the generalisation of convolutional architectures to higher dimensions as the sparsity will reduce as one increases the dimensions.

The next drawback is the generalisation of convolution operation to non-Euclidean domains. The generality of Riemannian manifolds makes it highly non-trivial to have a common framework of convolutional operations with non-Euclidean metric signatures [197]. The situation is even more intricate in the case of high-energy physics, where the underlying manifold is pseudo-Riemannian and is formally non-compact.

The nature of the image representation prohibits an efficient representation of heterogeneous data, which is naturally obtained at LHC, whether low-level from the different components of the detector or the various classes of reconstructed objects. Moreover, a Euclidean binning assumes an ordered structure and fixed dimensions, which is not the natural representation of the obtained data. The raw or reconstructed data can have a variable number of particles or reconstructed objects, respectively, from event to event and is essentially an unordered set with only their interrelations being important.

2.3.3 Graph Neural Networks

In the discussions above, we have seen that although CNNs can differentiate Euclidean patterns in the QCD radiation patterns in the lego plane, they cannot capture many other aspects of high-energy collisions like the non-Euclidean nature of spacetime. Graph Neural Networks [198–200] (GNNs) are hierarchical neural networks [201] that take the favourable inductive biases like local connectivity (thereby separating the scales with sequential application) and parameter-sharing of CNNs to data sampled from an underlying metric with possibly any non-Euclidean metric. GNNs consist of several subnetworks organised so that the output respects properties of graph-structured data like permutation-invariance of the nodes. In machine-learning literature, two distinct subareas use GNNs. One consists of learning the structure of graphs (generally very large) and classifying

the nodes or edges. Such problems are usually transductive [202], where the model tries to extend knowledge from labelled data to unlabelled regions. The other area of more immediate interest in LHC physics uses GNNs [203] to extract features from point clouds [204–206].

The evolution of a typical QCD event from high to low energies is well understood over a vast range of energy scales, as demonstrated by the successful application of QCD shower Monte Carlo programmes to the modelling of collider data (see, e.g. [207]). This evolution also motivates the application of Graph Neural Networks (GNNs) [208, 209] to QCD phenomenology, as recently done in Refs. [210], and exploit the Lund-plane representation of splittings [211, 212]. GNNs have also been studied in various scenarios [155, 213–216] at the LHC. Moreover, they have also shown promising performances for use in real-time triggers [217].

GNNs have been studied for jet classification in supervised [210, 218–222] as well as unsupervised [168, 223] scenarios and have state-of-the-art performance [219] compared to other still excellent architectures [224] like Convolutional Neural Networks (CNNs), Deep-sets, and Recurrent Neural Networks (RNNs). The better performance originates in GNNs having an inductive bias more appropriate for jet substructure and collider physics. These biases include generalising the Euclidean bias of CNNs to higher dimensional non-Euclidean spaces [225], enhancing the feature extraction in the deep-sets [204–206] framework by including local structures [203], and generalising RNNs [226, 227] to undirected cyclic graphs [198]. It has received wide attention in recent years and has been concisely described in the message-passing neural network (MPNN) formalism [199]. Thus, we will examine the MPNN formalism concentrating on the problem of inductive graph classification. Before explaining MPNNs, we will briefly describe the basic properties of graphs focussing on those neccesary to understand their construction from point clouds, and a better understanding of information flow in MPNNs.

2.3.3.1 Point clouds to graphs

Point clouds are sets of data points sampled from an underlying metric space. It is the most general and abstract way to define data and hence can represent data of virtually any nature. For instance, the calorimeter hits would be represented by a set of $\mathcal{S} = \{p_1, p_2, ..., p_{N_{hits}}\}$ with the four vectors of the impacts as its elements. Note that since $\mathcal{S}$ is a set, the ordering of the particles does not matter. Moreover, we can now have different sets for each detector component. At their outset, models capable of processing point clouds should be capable of handling sets of variable cardinality, and their output should not depend on the order of the data. The second point is a practical consequence since computing models will invariably take data in some particular order–to preserve the underlying property of sets, we make the model's output invariant to permutations in the order of the

constituents.

Models which process point clouds do so by looking at the topology and the geometry of the point cloud. Simpler models [204–206] take these points to output their collective property. However, many tasks in machine-learning and collider physics generally involve complex correlations between the constituents of the sets. As a concrete example, to study any process, we look at the relative positions of the final state particles like the rapidity gap $\Delta\eta_{jj}$ for vector-boson fusion processes or the interparticle distance ΔR_{ij} for various substructure observables. Therefore, such inter particle correlations can be efficiently abstracted by constructing a graph out of the constituents. Thus, for our case, graphs are point clouds with the addition of edges and edge features, which capture the inter-component relations.

A graph $G(\mathcal{S}, \mathcal{E})$ is defined on a set of nodes $\mathcal{S}$ with edges $\mathcal{E} \subseteq \mathcal{S} \times \mathcal{S} = \{(i,j) \mid i,j \in \mathcal{S}\}$ consisting of an ordered pair of elements in $\mathcal{S}$. The nodes can have a representation $\mathbf{h}_i \in \mathcal{M}\ \forall\ i \in \mathcal{S}$, in some metric space $\mathcal{M}$, where $\mathbf{h}_i$ is the feature of node i. In the context of LHC phenomenology, this metric-space is the union of the timelike and lightlike regions of the *Lorentz manifold* with the Minkowski metric or some other metric in flat spacetime. It can also include other information like charge, detector component etc. When learning from a point cloud, the edge set $\mathcal{E}$ is not provided *a priori* and is constructed with an algorithm defined on the node features. Well-known examples exist in the point cloud literature [228], the most famous one being the k-nearest neighbour (k-NN) graph [203, 219]. The edges can also have a representation $\mathbf{e}_{ij}$ in some space $\mathcal{X}$. In our context, these can be quantities like mass, directional separation, or the generalised k_t measures [91], which are derived from the node features themselves.

A *walk* is a sequence of edges that joins a sequence of nodes; for instance,

$$W = ((i,j),(j,l),(l,m),(m,i),(i,j))$$

is a walk with the edge (i,j) repeated twice. On a graph, all possible walks of length L would indicate all possible flow of information between the nodes under L iterative application of message-passing operations. If all the edges are distinct, a walk is called a *trail*. A *path* is a trail with no repeated nodes. The *distance* between two nodes is given by the number of edges of their shortest possible path. Considering a jet graph after L message-passing operations, any two nodes with a distance less than or equal to L would have information about each other encoded in the updated node features. A *graph cycle* is a trail where the first and the last node corresponds to the same node, with all other nodes distinct. A *cyclic graph* has at least one graph cycle. If a graph has no graph cycles, it is called an *acyclic graph*. A connected acyclic graph is called a *tree*. QCD splittings are naturally described by a tree [211, 229].

A *simple graph* is one where two distinct nodes can have at most one connection, and there are no self-loops. A graph is *undirected* if we do not distinguish

an edge based on the order of the nodes it connects; instead of the edge being defined by an ordered pair, we define it by an unordered pair. Constructing a directed graph will inherently have richer structural information of the underlying space on a point cloud. If a graph is simple, it can be equivalently represented in terms of *neighbourhood sets* $\mathcal{N}(i)$ in place of the edge set $\mathcal{E}$. For a directed graph, $\mathcal{N}(i)$ is defined for each node i, as the set containing all the nodes with incoming connections to i. We can allow for self-loops if we take the *closed neighbourhood* where i is also a part of the set $\mathcal{N}[i] \ni i$. The l-hop neighbourhood of a node i is the set of all nodes with distances from i, less than, or equal to l.

2.3.3.2 Message-passing Neural Networks

Modern deep neural networks (DNNs) generally have a two-stage architecture: a specialised feature extraction section, followed by a generic dense architecture processing the extracted information further. Message-passing neural networks (MPNNs) are specialised feature extraction modules that act on graphs with node features and edge features. A message-passing operation takes a graph with node features $\mathbf{h}_i^{(l)}$ as input and updates it to $\mathbf{h}^{(l+1)}$, with a two-step process:

1. **Message passing:** We define a learnable function $\Phi^{(l)}$ with trainable parameters,[††] which takes as input the node features $\mathbf{h}_i^{(l)}$ and $\mathbf{h}_j^{(l)}$, connected by the edge[‡‡] (i, j) and returns the message ${}^i\mathbf{m}_j^{(l)}$,

$$ {}^i\mathbf{m}_j^{(l)} = \Phi^{(l)}(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}) \quad . \tag{2.29} $$

   The message is calculated for all edges present in the graph. We choose to use the notation ${}^i\mathbf{m}_j^{(l)}$ instead of a homogenous subscript to make it evident that the message or the function $\Phi^{(l)}$ need not be symmetric with respect to the source node j, and the destination node i.

2. **Node readout:** The nodes are updated with new features $\mathbf{h}_i^{(l+1)}$ by applying a permutation invariant function $\square^{local}$ to all incoming messages

$$ \mathbf{h}_i^{(l)} \to \mathbf{h}_i^{(l+1)} = \square^{local}_{j \in \mathcal{N}(i)} {}^i\mathbf{m}_j^{(l)} \quad . \tag{2.30} $$

   Note that the updated features $\mathbf{h}_i^{(l+1)}$, contain the *local neighbourhood* information of i.

As one can see, the process of constructing a graph itself is a strong inductive bias in any GNN since the neighbourhood set $\mathcal{N}(i)$ constrains the local connectivity of any nodes i. Therefore, an MPNN taking as the complete graph K_n (each node is connected to every other node) as inputs will have no local connectivity.

[††] This can be a multilayer perceptron, but it can be a collection of sub networks arranged in some particular way in general.

[‡‡] It can also take the corresponding edge feature $\mathbf{e}_{ij}^{(l)}$ if present.

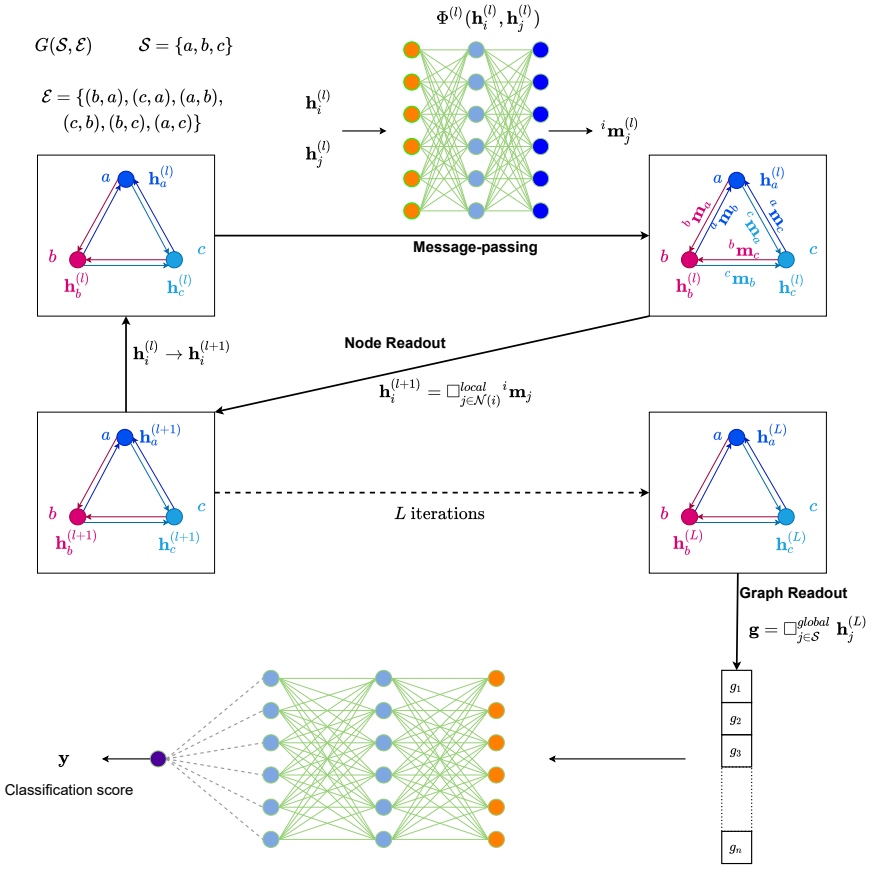


Figure 2.8: A diagramatic representation of a *message-passing neural network*(MPNN) for *graph-classification*. We are given a graph $G(\mathcal{S},\mathcal{E})$ with nodes $\mathcal{S}$ and edge set $\mathcal{E}$. Each node i has a feature vector $\mathbf{h}_i^{(l)}$. The first step called the *message-passing* involves evaluating the message ${}^i\mathbf{m}_j^{(l)}$ for each edge in (j,i) via a DNN $\Phi^{(l)}$ shared for all edges. The different MPNN proposed in literature has structural differences in how $\Phi^{(l)}$ takes the inputs, which could include edge features as well. The second step, called the node readout, updates the features of each node to $\mathbf{h}_i^{(l+1)}$ with a permutation invariant function $\square^{local}$ acting on all incoming messages. After L iterations, a graph readout function $\square^{global}$ on the final node features $\mathbf{h}_i^{(L)}$, gives fixed length n-dimensional graph representation $\mathbf{g}$. This is fed into a downstream neural network which outputs the graph classification score $\mathbf{y}$.

This global connectivity is similar in spirit to using a filter size of the same dimensions as the image in the case of CNNs. The learnable parameters are shared all over the graph since the message-function $\Phi^{(l)}(\mathbf{h}_i^{(l)},\mathbf{h}_j^{(l)})$ is shared for all edges in the graph. The sequential application of message-passing operation coupled with the local connectivity ensures a separation of scales similar to the case of CNNs. In the following, we discuss the scale separation and other emergent features in MPPNs in detail.

The message-passing operation can be repeated any number of times. Each iteration leads to a gradual increase in the neighbourhood information held within

the node features. On a *static graph* where the neighbourhood sets $\mathcal{N}(i)$ or equivalently the edge set $\mathcal{E}$ remain unchanged, the node features contain information of the L-hop neighbourhood when applied L times. Thus, the number of message-passing operations applied L, is a crucial hyperparameter in any GNN. It determines the scale at which the final node features $\mathbf{h}_i^{(L)}$, capture the *local structures* within the graph. The number L is restrained by the high computational cost of applying message passing operations, thereby reducing expressive power for the classification of large graphs. Even for jet graphs, we have a relatively large number of nodes, and hence, the information gets restricted to a local scale, intrinsically determined by L. For instance, in a two-prong W tagging case, if L is lesser than the length of the path between the two hard subjets, which would vary for each jet graph, the message-passing functions $\Phi^{(l)}$ would not see this feature for jets with several soft particles $n_{soft} > L$, between the two subjets. Nevertheless, a precisely determined graph construction algorithm would probabilistically give graphs with very low $< n_{soft} > \approx 0$. To avoid this limitation in the message-passing step, *dynamic* graphs are used to gather information from different scales, with the possibility of learning correlations from the entire graph after one dynamic iteration. However, for the same L, dynamic MPNNs will have a higher computational cost because of the added graph construction after each message-passing operation. Moreover, it may not always be desirable to mix information at the message-passing stage as the graph representation would still have the global features intact.

Note that the number of nodes in a graph can vary. For graph classification, a permutation invariant graph readout function $\square^{global}$ is applied to these node features, giving a fixed-length *global representation* of the graph

$$\mathbf{g} = \square_{i \in G}^{global} \mathbf{h}_i^{(L)}. \tag{2.31}$$

In all instances, a graph readout serves similar purposes to the node readout, with the only difference being the scale of the operation. The graph representation is fed into a densely connected network, which outputs a classification score for the whole graph. The steps of an MPNN for graph classification, which we have discussed, are shown diagrammatically in Figure 2.8.

2.3.4 Deep-learning libraries

The applications of modern deep-learning in industry and fundamental research are propelled by the wide availability of easy-to-code python libraries with backends having GPU acceleration capabilities. Such libraries accelerate the implementation of complex deep-learning modules and their training by implementing automatic differentiation libraries in GPUs for the back-propagation algorithm while leaving the essential aspects of architecture design highly flexible. Here, we summarise the various libraries used extensively in the studies conducted for this

thesis.

`TensorFlow` [230] and `PyTorch` [231] are two of the most popular deep-learning libraries which provide a base for other high-end packages. Initially, Tensorflow was based on a static computational graph for executing the models, making it problematic to implement models that inherently take variable-length inputs. Recent versions have moved to dynamic computational graph definitions, similar to PyTorch. The `Keras` [232] package provides high-level abstractions to TensorFlow's model definition and training implementations, which now comes prebuilt as a module within TensorFlow. On the other hand, although default classes in PyTorch, like "`torch.nn.module`", provides high-level model implementation, training implementation is still rather involved compared to Keras. However, the design of PyTorch is generically pythonic and less complicated compared to their equivalent TensorFlow implementations.

Due to TensorFlow's initial static computational design, the generic implementation of models like graph neural networks with variable length inputs have been popularly based on PyTorch. `Deep Graph Library` [233] is a high-level python package for implementing graph neural networks. Although it has optional backends to both TensorFlow and PyTorch, we will primarily use the PyTorch version, which is more mature. `Pytorch Geometric` [234] is a high-level extension of PyTorch with easy-to-use abstractions for handling and designing new graph neural network architectures. It follows the same design principles as PyTorch and is much more pythonic than DGL.

2.4 Summary

This chapter has laid down the basics of jet substructure techniques and Artificial Neural Networks. Although deep-learning algorithms are extremely powerful, we have seen that this excellent performance results from an intricate design of neural network architectures and exploiting underlying features in data. In stark contrast to practical applications of such algorithms in the industry, problems in particle physics phenomenology consist of studying the nature of fundamental interactions which have well-understood behaviour but intractable probability distributions. Therefore, it is essential to scrutinise such deep-learning algorithms in detail, including, but not limited to, their resilience to imperfect simulations, interpretability, and possibly an understanding of their convergence properties from a first principle analysis. We will analyse their applicability and performance in some scenarios of interest in the following chapters.

Chapter 3

Probing invisible VBF Higgs decay with CNNs

In this chapter, we study the capability of CNNs to identify vector boson fusion signatures. Vector-boson fusion (VBF) production of color singlet particles provide a unique signature in hadron colliders. First studied in reference [235–237], they are characterised by the presence of two hard jets in the forward regions with a large rapidity gap, and a relative absence of hadronic activity in the central regions, when the singlet particle decays non-hadronically. For illustration, the left panel of figure 3.1 shows an event of a Higgs produced in VBF channel decaying invisibly in a simplistic tower geometry, while the same event is mapped in a flattened (η, ϕ) plane by rolling out the ϕ-axis, with the height of the bars corresponding to the magnitude of the transverse projection of calorimeter energy deposits in each pixel. In order to highlight the differences with non-VBF processes, it is instructive to show one such example in figure 3.2. This is a representative event from $Z(\nu\bar{\nu}) + jets$ background, where the jets arise from QCD vertices, which inherently has a much higher hadronic activity in the central regions between the two leading jets. Even though the rapidity gap vanishes when the singlet particle decays hadronically, the absence of color connection between the two forward jets and the central region persists and has been used in the experimental analysis [238], in searches of the Higgs boson decaying to bottom quarks. The VBF process was proposed as the most important mechanism for heavy Higgs searches [239] thanks to a much slower fall in cross-section compared to the s-channel mediated process. Usefulness for intermediate to light mass scalar was also subsequently realised [240] due to its unique signature at the collider. VBF process holds great importance to measure Higgs coupling with gauge bosons and fermions as it allows independent observations of Higgs decay like $h^0 \to WW$ [241], $h^0 \to \tau\tau$ [242]. Therefore, it also plays a vital role in determining anomalous coupling to vector boson [243, 244] or the CP properties of the Higgs [245, 246]. Its clean features make it the most sensitive channel for searching invisible decay of the Higgs boson [247] and in search for physics beyond

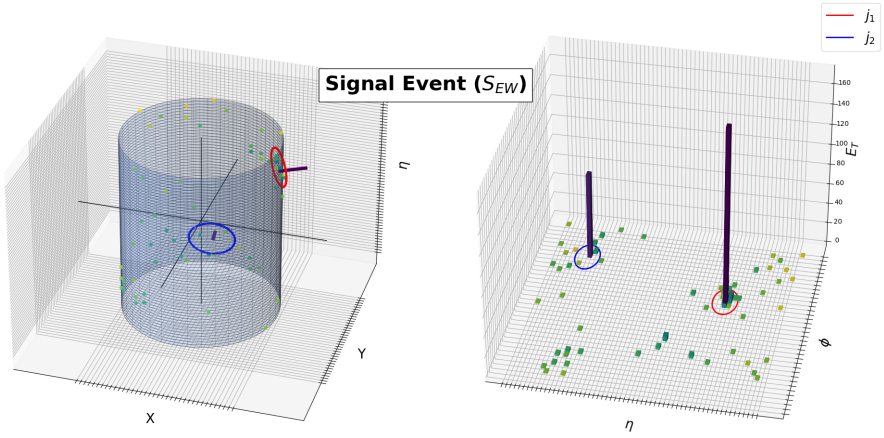


Figure 3.1: The figure shows a 3D depiction of a prototype signal event originated from an electroweak VBF Higgs production in a naive detector geometry in left plot. The same event is flattened in a convenient $\eta - \phi$ plane in right plot, where the transverse projection of calorimeter energy deposits in different pixels are drawn. Two reconstructed primary jets are shown with color circles, and corresponding transverse energy deposits are visible from height of the bars.

the standard model [248–250].

The rest of the chapter is organised as follows. In Section 3.1, we discuss the Higgs production mechanism via the VBF channel and different SM backgrounds contributing to this process. We also discuss the generation of simulated data consistent with the VBF search strategy. In Section 3.2, we describe the details of the data representation used in the present study. Here, different classes of high-level variables are also defined. Preprocessing methods of feature spaces are addressed in Section 3.3. We discuss the seven different neural network architecture and its performance in Section 3.4. The results, interpreted in terms of expected bounds on the invisible branching ratio, for all the architectures are presented in Section 7.4. There, we also discuss the impact of pileup on the result of our analysis. Finally, we close our discussion with the summary and conclusion in the Section 3.6.

3.1 Vector Boson Fusion production of Higgs and analysis set-up

VBF production of the SM Higgs has the second-highest production cross-section after gluon-fusion at the LHC. Loop induced Higgs production and decay depend on the presence of contributing particles and different modifiers in fermions and gauge boson coupling with the scalar. Hence, both production cross-section and decay branching ratios are modified in the presence of new physics. In this present work, we consider the production of SM like Higgs boson and constrain its invisible

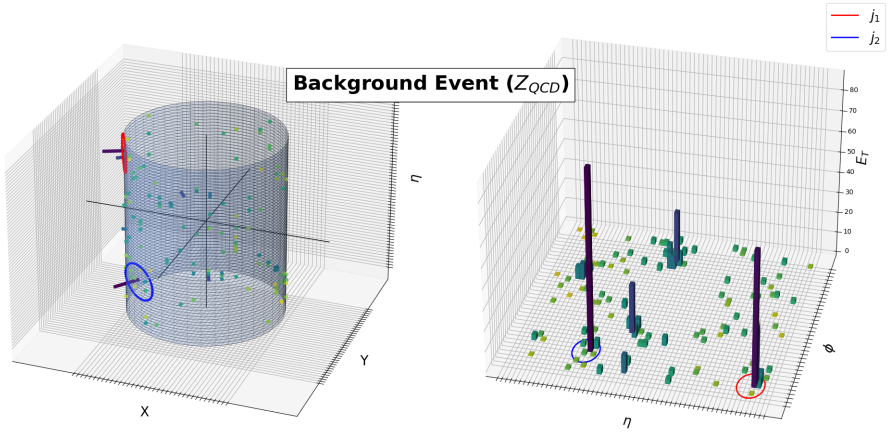


Figure 3.2: Same as figure 3.1, but for a prototype background event originated from a $Z(\nu\bar{\nu}) + jets$ production, where the jets originate from QCD vertices.

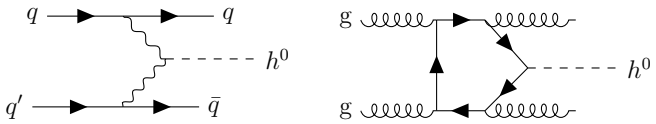


Figure 3.3: Representative diagrams for production of Higgs signal through (left) electroweak VBF channel and (right) a higher-order QCD process in gluon fusion where two QCD jets can be detected along with a sizable missing transverse-energy from invisible Higgs decay.

decay width. Such constraint is essential in many new physics scenarios, such as Higgs portal dark matter [27–31], where new particles do not modify their couplings with SM particles.

The electroweak production of Higgs is dominated by the fusion of two massive vector bosons, which are radiated off two initial (anti-)quarks, as represented in figure 3.3 (left plot). This exchange of color singlet state between two quarks ensures no color connection between two final jets, typically produced in a forward (backward) region of the opposite hemisphere. The central region - between these two jets remain color quiet, lacking any jet activity even after radiation and fragmentation of the two scattered quarks while looking at the hadronic final states. As we have already discussed, an agnostic viewpoint requires a serious re-examination after the inclusion of all other processes, such as non-VBF Higgs signal from gluon fusion. One such sample diagram is shown in figure 3.3 (right plot). Additional radiation from gluons can provide a typical VBF type signal, once again, in the absence of the key attributes like color-quiet central region, etc.

Another interesting feature of VBF Higgs production is that the corresponding cross-section has very modest correction under higher-order QCD, which has been known for a long time [251,252]. Integrated and differential cross sections for VBF

Higgs production have now been calculated up to very high levels of accuracy. QCD corrections are known up to N3LO [253], reducing the scale-uncertainty up to 2%, while Electroweak corrections are known up to NLO [254]. Moreover, non-factorizable contributions have also been calculated for the first time [255], and show up to percent level corrections compared to the leading order (LO) distributions.

At hadron colliders, traditional searches [256–259] of non-hadronically decaying color-singlet particles in the VBF production channel focus on rejecting the large QCD backgrounds from $Z+jets$, and $W+jets$ background via a *central jet-veto*, after a hard cut on the separation of the two forward jets in pseudorapidity $|\Delta\eta_{jj}|$, and the dijet invariant mass m_{jj}. This opens up the possibility of using inclusive event-shape variables like N-jettiness [106], to improve the selection efficiency [260]. In this study, we explore the feasibility of using deep-learning techniques instead of event-shape variables. We study the invisible decay of the Higgs boson as a prototype channel for gauging the power of deep-learning methods in VBF since there is no contamination on the radiation patterns between the two forward jets from the decay products. We closely follow the shape-based analysis performed by the CMS experiment at LHC [261].* As already commented, the central jet veto played a critical role in the usual searches of VBF to control the vast QCD background. The role of additional information from QCD radiation between the tagging jets and within the jet itself was explored in references [263, 264]. It was found that relaxation of the minimum p_T requirement of the central jet improved the sensitivity, and the inclusion of subjet level information resulted in further suppression of backgrounds. However, the present analysis does not rely on a central jet veto, as the main aim is to study the VBF topology with the low-level data, made possible with modern deep-learning algorithms. Therefore, with the relaxed selection requirements on $|\Delta\eta_{jj}|$ and m_{jj}, the selected signal gets a significant contribution from the gluon-fusion production of Higgs on top of VBF processes. Due to the relaxed selection criteria, we also get a substantial contribution from QCD backgrounds.

3.1.1 Signal topology

The present study relies on all dominant contributions to Higgs coming both from electroweak VBF processes and also higher-order in QCD gluon fusion processes. Here at least two jets should be reconstructed along with sizable missing transverse-energy from invisible decay of Higgs. Hence, we classify the full signal contribution in two channels:

- S_{QCD}: Gluon-fusion production of Higgs with two hard jets, where the Higgs decays invisibly.
- S_{EW}: Vector-Boson fusion production of Higgs decaying invisibly.

*For the ATLAS results with similar data, see reference [262].

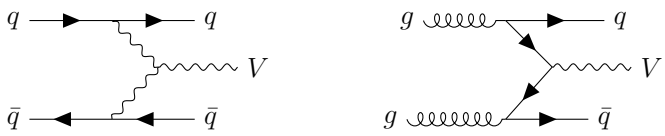


Figure 3.4: Representative diagrams for dominant background processes through (left) VBF type weak production and (right) QCD production of massive vector-bosons V, such as W or Z which decay invisibly by producing undetected lepton or neutrinos.

The subscript $EW(QCD)$ denotes the absence (presence) of strong coupling α_S, at leading order(LO) for the interested topology. This also segregates the channels with absence or presence of color exchange between the two incoming partons at LO. Figure 3.3 shows a representative Feynman diagram of the signal channels in each class.

3.1.2 Backgrounds

The major backgrounds contributing to the invisible Higgs VBF signature can come from the different standard model processes. Among them, VBF type electroweak, and QCD production of massive vector-bosons (W or Z) contribute copiously. All these processes ensure a pair of reconstructed jets along with considerable missing transverse energy from invisible decay of these gauge bosons. A substantial fraction of W and Z can produce neutrinos or a lepton which remain undetected at the detector. We consider the following backgrounds in all our analyses:

- Z_{QCD}: $Z(\nu\bar{\nu}) + jets$ process contributes as the major SM background due to high cross section.
- W_{QCD}: $W^{\pm}(l^{\pm}\nu) + jets$ process also contribute to the SM background when the lepton is not identified.
- Z_{EW} : Electroweak production of Z decaying invisibly along with two hard jets is topologically identical with the electroweak signal and contributes significantly to the background.
- W_{EW} : Electroweak production of $W^{\pm}$ with two hard jets can also produce an identical signal when the lepton does not satisfy the identification criteria.

Similar to the signal processes, the subscript $EW(QCD)$ denotes the absence (presence) of strong coupling α_S, at LO for the interested topology having at least two reconstructed jets in the final state. Figure 3.4 shows representative Feynman diagrams of the background channels divided into four different classes.

There are also other background processes like top-quark production, diboson processes, and QCD multijet backgrounds whose contribution would be highly suppressed compared to these four backgrounds. The top and diboson backgrounds would contribute to leptonic decay channels where charged leptons, if present, are not identified, while the QCD multijet background would contribute when there is severe mismeasurement of the jet energies.

3.1.3 Simulation details

We used `MadGraph5_aMC@NLO` (v2.6.5) [61] to generate parton-level events for all processes at 13 TeV LHC. These events are then showered and hadronised with `Pythia8` (v8.243) [70]. `Delphes3` (v3.4.1) [72] is used for fast-detector simulation of the CMS working conditions. Jets are clustered using the `FastJet` (v3.2.1) [94] package. The signal processes are generated using a modified version of the Higgs Effective Field Theory (HEFT) model [69, 265, 266], where the Higgs boson can decay to a pair of scalar dark matter particle at tree level. We are interested in probing high transverse momentum of Higgs, where the finite mass of top quark in gluon fusion becomes essential. Hence, we have taken into account such effect by reweighting the missing transverse energy (MET) distribution of the events with recommendations from reference [267]. The parton level cross-sections of Z_{QCD} and W_{QCD} were also matched up to four and two jets, respectively, via the MLM procedure [268]. Since the $W^{\pm}$ backgrounds contribute when the leptons are missed within the range of tracker or when they are not reconstructed at the detector, the parton level cuts on the generated leptons are removed to cover the whole range in pseudorapidity (η).

For a consistent comparison with current experimental results, we repeat the shape-analysis of reference [261] with our simulated dataset. The MET cut for the deep-learning study is relaxed from 250 GeV to 200 GeV.

Baseline selection criteria: We apply the following pre-selections:

- **Jet** p_T: At least two jets with leading (sub-leading) jet having minimum transverse momentum $p_T > 80$ (40) GeV.
- **Lepton-veto:** We veto events with the reconstructed electron (muon) with minimum transverse momentum $p_T > 10$ GeV in the central region, *i.e.* $|\eta| < 2.5\,(2.4)$. This rejects leptonic decay of single $W^{\pm}$, and semi-leptonic $t\bar{t}$ backgrounds.
- **Photon-veto:** Events having any photon with $p_T > 15$ GeV in the central region, $|\eta| < 2.5$ are discarded.
- τ **and b-veto**: No tau-tagged jets in $|\eta| < 2.3$ with $p_T > 18$ GeV, and no b-tagged jets in $|\eta| < 2.5$ with $p_T > 20$ GeV are allowed. This rejects leptonic decay of single $W^{\pm}$, semi-leptonic $t\bar{t}$ and single top backgrounds.

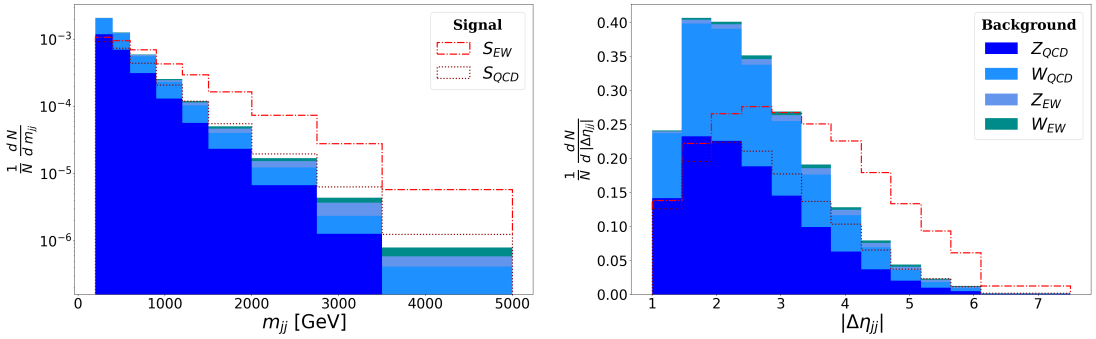


Figure 3.5: Distribution of (left) m_{jj} and (right) $\Delta\eta_{jj}$ of events passed after the passing the tighter selection requirement (MET $>$ 250 GeV). The contribution of each channel to its parent class has been weighted by their cross-sections and the baseline efficiency at 13 TeV. The signal and backgrounds are then individually normalised, and the lines/color show the contribution of each channel to its parent class.

- **MET**: Total missing transverse momentum for the event must satisfy MET $>$ 200 GeV for all our deep-learning study, whereas we compared CMS shape-analysis consistent with MET $>$ 250 GEV.
- **Alignment of MET with respect to jet directions**: Azimuthal angle separation between the reconstructed jet with the missing transverse momentum to satisfy $\min(\Delta\phi(\vec{p}_T^{\,\text{MET}}, \vec{p}_T^{\,j})) > 0.5$ for up to four leading jets with $p_T > 30$ GeV and $|\eta| < 4.7$. QCD multi-jet background that arises due to severe mismeasurement is reduced significantly via this requirement.
- **Jet rapidity**: We require both jets to have produced with $|\eta_j| < 4.7$, and at least one of the jets to have $|\eta_{j_i}| < 3$, since the L1 triggers at CMS do not use the information from the forward regions.
- **Jets in opposite hemisphere**: Those events which have the two leading jets reside in the opposite hemisphere in η are selected. This is done by imposing the condition $\eta_{j_1} \times \eta_{j_2} < 0$.
- **Azimuthal angle separation between jets**: Events with $|\Delta\phi_{jj}| < 1.5$ are selected. This helps in reducing all non-VBF backgrounds.
- **Jet rapidity gap**: Events having minimum rapidity gap between two leading jets $|\Delta\eta_{jj}| > 1$ are selected.
- **Di-jet invariant mass**: We required a minimum invariant mass of two leading jets, $m_{jj} > 200$ GeV. Note that, this along with the previous selection requirements are relatively loose compared to traditional selection criteria of VBF topologies, which result in significant enhancement of the signal from S_{QCD}, although at the cost of increased QCD backgrounds (Z_{QCD} and W_{QCD}).

Interestingly, one can notice that a relaxed selection requirement may give rise to additional contamination from Higgs-strahlung type topologies to the S_{EW} channel, which is included in our EW generation of events. However, these events are not expected to survive a selection of di-jet invariant masses of more than 200 GeV. After extracting the events passing the above selection requirements and the respective selection efficiency (calculated from the weights) for S_{QCD}, the pre-selected events are unweighted again so that we get equal weights for individual events.[†] The background and signal classes are formed by mixing the channels with the expected proportions using appropriate k-factors, cross-sections, and the baseline selection efficiencies. We use cross-sections quoted in reference [267] for both signal processes. For instance, the S_{QCD} is calculated up to NNLL +NNLO accuracy [269], while for S_{EW} it is calculated up to NNLO [270] in QCD and NLO in electroweak. We use the LO distributions with their overall normalisations increased to accommodate the total cross-section at higher perturbative accuracies without accounting for the possible change in shape. Similarly, all background cross sections are calculated by scaling the LO result with global NLO k-factors [271, 272]. We generated 200,000 training and 50,000 validation balanced dataset of events for the deep-learning classifier. The signal class consists of 44.8% S_{EW} and the 55.2% S_{QCD} channels; while the background class consists of 51.221% Z_{QCD}, 44.896% W_{QCD}, 2.295% Z_{EW} and 1.587% W_{EW} channels.

We also extract event sample for all channels with the harder selection requirement on missing transverse momentum (MET $>$ 250 GeV), the value used in reference [261], from the same set of generated events used for the deep-learning analysis. The extracted dataset contains: 39% S_{EW} and the 61% S_{QCD} channels for the signal class; and 54.43% Z_{QCD}, 40.92% W_{QCD}, 3.05% Z_{EW} and 1.58% W_{EW} channels for the background class. The bin-wise stacked histogram of all channels for m_{jj} and $|\Delta\eta_{jj}|$ are shown in figure 3.5. The properties of the EW and the QCD subsets are evident from these distributions: EW contribute more at higher m_{jj} and $|\Delta\eta_{jj}|$, while the opposite is true for QCD.

3.2 Data Representation for the Network

Neural network architectures for deep-learning are mostly designed with two blocks. The first stage generally consists of locally-connected layers (with or without weight sharing) with some particular domain level specifications which extract the features. The second stage consists typically of densely connected layers, whose function is to find a direction in the learned feature-space, which optimally satisfies the particular target of the network locally by learning its projections in different representations at each subsequent layer. For instance, in

[†]See Appendix A, for the distribution of important kinematic variables and details of the re-weighting and unweighting of events.

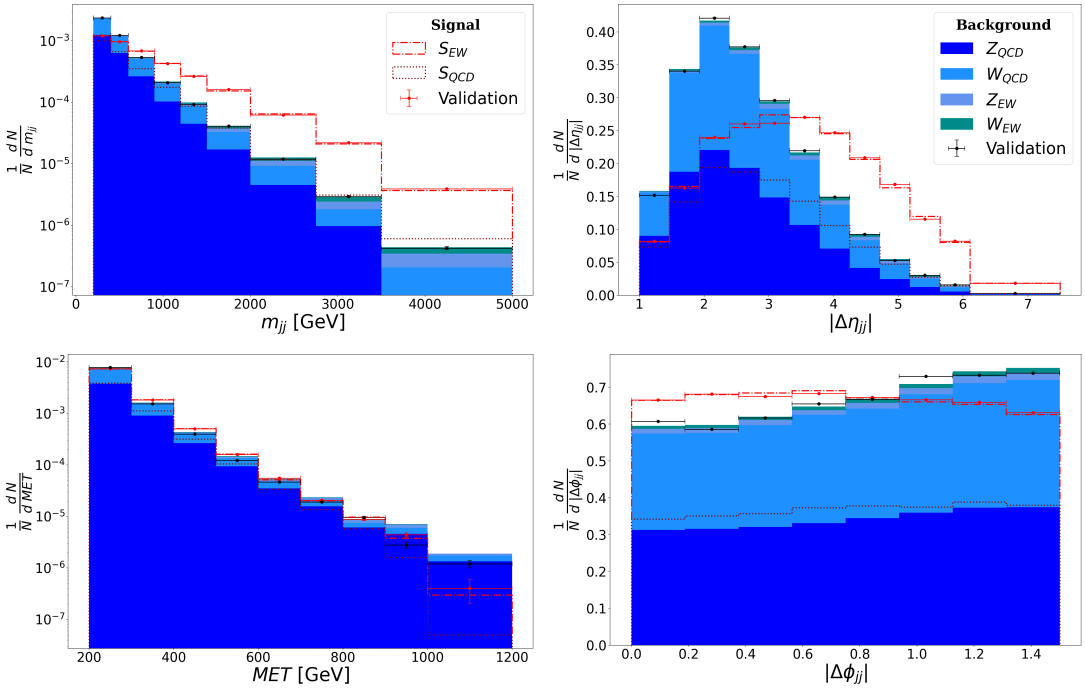


Figure 3.6: Similar to figure 3.5, some of the basic input high-level kinematic variables used for our analysis (MET $>$ 200 GeV) are shown for signal and background.

classification problems, it finds the decision boundary between different classes. At the same time, in an unsupervised clustering, it compresses the feature-space so that the modes become localised in a smaller volume. A synergy between the representation of data and the network architecture is a must for efficient feature extraction. This is evident from the fact that convolutional neural networks perform best with data structures that have an underlying Euclidean structure, while recurrent networks work best with sequential data structures. In the context of classifying boosted heavy particles like W, Higgs, top quark or heavy scalars decaying to large-radius jets from QCD background, a lot of efforts [115–119, 122] went into representing the data like an image in the (η, ϕ) plane to use convolutional layers for feature extraction, while some others [173, 178], use physics-motivated architectures. Convolutional architectures work in these cases because the differences between the signal jet and the background (QCD) follows a Euclidean structure.[‡] The Minkowski structure of space-time prohibits a direct use of convolutional architectures. Although geometric approaches [197] exist to counter the non-Euclidean nature, the number of dimensions makes it computationally expensive. Graph neural networks [200, 216, 273, 274] provide a possible workaround which is computationally less intensive, for feature learning in non-Euclidean domains.

[‡]Most high-level variables designed from QCD knowledge are functions of $\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2}$.

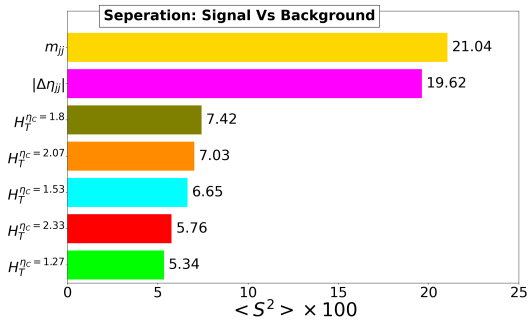


Figure 3.7: The separation of the 7 highest performing variables (given in percentage).

We want to study the difference in radiation patterns between the two forward jets for signal and background events; hence, we primarily choose a convolutional architecture for automatic feature extraction. Therefore, the low-level feature space we prefer is the *tower-image*, in the (η, ϕ)-plane, with the transverse energy E_T, as the pixel values. One can take into account the different resolutions in the central and forward regions of calorimeter towers in LHC detectors. For simplicity, and also to demonstrate the resolution dependence, we construct two images - a high-resolution image with bin size 0.08×0.08, and a low-resolution image with bin size 0.17×0.17, in the full range of the tower, $[-5, 5]$ for η and while $[-\pi, \pi)$ for ϕ. Convolutional neural-networks, in general, look at global differences, and increasing the resolution does not play as important a role. We examine CNNs in these two different resolutions to inspect this for our particular case. The procedure of forming a tower-image does not naturally take the periodicity of the ϕ axis into account. In order to let the network know this inherently, we expand the image obtained after binning, in the ϕ axis such that the connectivity between the two edges is not broken. This is done by taking a predetermined number of ϕ-rows from each edge of the original image and forming a new image where these rows are padded [192, 195] in their corresponding opposite sides, thereby mimicking the periodicity. This is similar to cutting the cylinder at two different points in ϕ for each edge, such that there is an overlapping region in the final image. Taking the jet radius $R = 0.5$, which have a regular geometry since they are clustered with anti-k_t algorithm [91], we choose the number of rows to be 4 (8) for the low (high)-resolution images, with one bin as a buffer. This gives a low-resolution (LR) image of 59×45 and a high-resolution (HR) one of 125×95.

A significant difference between low-level and high-level feature spaces is that the modes of the data in low-level representations are not distinct. Although this is marginally enhanced by preprocessing, high-level features derived from the said low-level features have distinctly localised modes in their distribution. An exemplary ability of deep-learning algorithms is to by-pass this step and learn their own representations which perform better than the high-level variables developed by domain-specific methods. To analyse the relative performance of

physics-motivated variables derived from the calorimeter deposits, we consider two classes of high-level variables. The first one consists of the following kinematic variables:

$$\mathcal{K} \equiv (\ |\Delta\eta_{jj}|,\ |\Delta\phi_{jj}|\ ,\ m_{jj}\ ,\ \text{MET}\ ,\ \phi_{\text{MET}}\ ,\ \Delta\phi^{j_1}_{\text{MET}}\ ,\quad \Delta\phi^{j_2}_{\text{MET}}\ ,\ \Delta\phi^{j_1+j_2}_{\text{MET}}\ )$$

ϕ_{MET} is the azimuthal direction of MET in the lab-frame. $\Delta\phi^{j_1}_{\text{MET}}$, $\Delta\phi^{j_2}_{\text{MET}}$ and $\Delta\phi^{j_1+j_2}_{\text{MET}}$ are the azimuthal separation of MET with the direction of the leading, sub-leading and the vector sum of these two jets, respectively. Clearly, these do not contain any information about the radiation pattern between the tagging jets. The second class of variables: the sum of E_T of the tower constituents in the interval $[-\eta_C, \eta_C]$, incorporates this information:

$$\mathcal{R} \equiv (H_T^{\eta_C} | \eta_C \in \mathcal{E}) \quad , \quad H_T^{\eta_C} = \sum_{\eta<|\eta_C|} E_T \quad . \tag{3.1}$$

$\mathcal{E}$ denotes the set of chosen η_C's. We vary η_C uniformly in the interval [1,5]:

$$\mathcal{E} = \{1, 1.27, 1.53, 1.8, 2.07, 2.33, 2.6, 2.87, 3.13, 3.4, 3.67, 3.93, 4.2, 4.47, 4.73, 5\},$$

to get 16 such variables. Their inclusion helps us to provide a thorough comparison of the high-level and low-level feature spaces. Figure 3.6 shows the signal vs background distribution of some important kinematic-variables. The channel-wise contributions to the parent class are also stacked with different colors/lines. We see that the characteristics of the m_{jj} and $|\Delta\eta_{jj}|$ are the same with figure 3.5, with the electroweak processes contributing more at higher values. A feature seen for $|\Delta\phi_{jj}|$ is the shape of the signal and background distributions. Clearly, the difference is due to the S_{EW} contribution since S_{QCD} has a very similar shape as that of the background. This is another characteristic of VBF processes that the leading jets, originating from electroweak vertices, have lower separation in ϕ compared to those originating from QCD. Similar plots for the remaining four kinematic variables and the $\mathcal{R}$ set of variables are given in Appendix B. A brief discussion of the two feature spaces (mainly $\mathcal{R}$) is also presented. We denote the combined high-level feature-space as $\mathcal{H}$, which is 24-dimensional.

In order to gauge the discriminating power of each feature x, we determine the separation [275] defined as,

$$< S^2 >= \frac{1}{2} \int \frac{(p_S(x) - p_B(x))^2}{p_S(x) + p_B(x)}\, dx \quad . \tag{3.2}$$

$p_S(x)$ and $p_B(x)$ denote the normalised probability distribution of the signal and background classes. It gives a classifier-independent discrimination power of the feature x. A value of zero (one) denotes identical (non-overlapping) distributions.

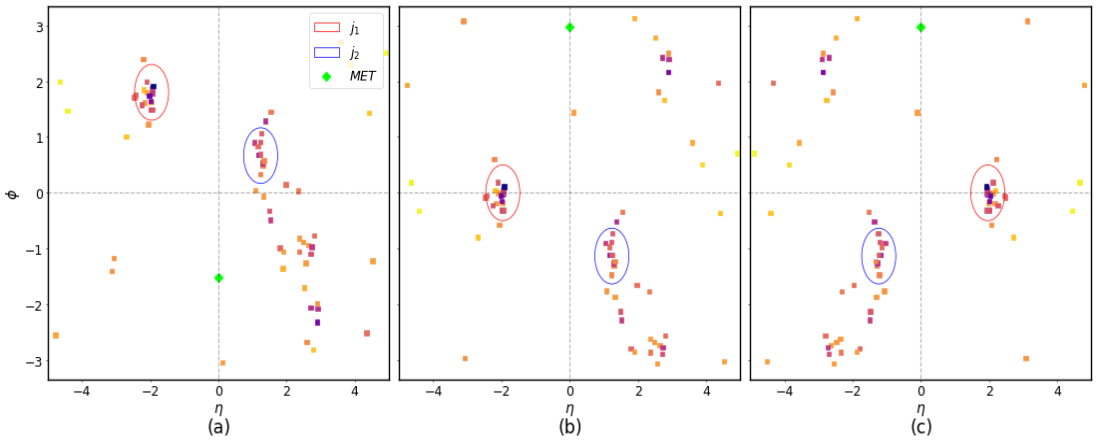


Figure 3.8: Scatter plot of tower constituents of an event in the (η, ϕ)-plane showing: (a) the raw event; and the effects of (b) rotation ($\phi_{j_1} = 0$), and (c) reflection ($\eta_{j_1} > 0$) operations. The pseudorapidity of MET has been set to zero for illustration. It is important to note that the points here are not binned into pixels and the values are the ones extracted from the Delphes Tower constituents.

We plot the separation (in percentage) of the seven highest important variables out of the 24 features in figure 3.7. It is interesting to note that out of these, there are five variables from $\mathcal{R}$, even though the first and the second are from $\mathcal{K}$, and they are much greater in magnitude.

3.3 Preprocessing of feature space

Preprocessing of features is indispensable for shallow machine learning as it helps maximise the statistical output from smaller data sizes. In deep-learning applications, it helps in faster convergence of the training and in approaching optimal accuracy with a lesser amount of data using simpler architectures. Even though the primary aim of our model is to learn the differing QCD radiation patterns, we can only devise preprocessing operations that preserve the Lorentz symmetries of the event. The spatial orientation of the events, in general, can be regularised by the following procedure:

1. **Identify principal directions:** Choose three final-state directions $\{\hat{n}_1, \hat{n}_2, \hat{n}_3\}$. These can be any three final state objects, which are the interest of our studies like photons, leptons, and jets, or they can be chosen to be generic directions in the lab frame.

2. **First Rotation:** Rotate the event such that:

$$\hat{n}_1 \to \hat{n}_1' = (0, 0, 1) \equiv \hat{n}^a \quad , \quad \hat{n}_2 \to \hat{n}_2' \quad , \quad \hat{n}_3 \to \hat{n}_3' \quad .$$

   After this operation, the orientation of $\hat{n}_1$ is the same for all events.

3. **Second Rotation:** Rotate the event along $\hat{n}^a$ such that:

$$\hat{n}'_2 \to \hat{n}''_2 = (0, n^b_y, n^b_z) \equiv \hat{n}^b \quad , \quad \hat{n}'_3 \to \hat{n}''_3 \quad .$$

   The plane formed by $\hat{n}_1$ and $\hat{n}_2$ has the same orientation for all events after this operation.

4. **Reflection:** Reflect along yz-plane such that:

$$n''_3 \to (|n^c_x|, n^c_y, n^c_z) \equiv \hat{n}^c \quad .$$

   The half-space containing $\hat{n}_3$ becomes the same for all events after this step.

These are passive operations which affect the orientation of the reference frame without changing the physics. For most event topologies, we can see that there will be better feature regularisation when $\hat{n}_2$ and $\hat{n}_3$ are equal. In hadron colliders, due to the unknown partonic center-of-mass energy $\sqrt{\hat{s}}$, we set the z-axis as $\hat{n}_1$, preserving the transverse momentum of all final state particles. We choose two different instances of $\hat{n}_2 \in \{\hat{n}_{\text{MET}}, \hat{n}_{j_1}\}$. For our choice of $\hat{n}_1$, the z-direction of $\hat{n}_2$ does not matter and we can take its value for $\hat{n}_{\text{MET}}$ to be zero. However, the z-direction becomes important for the third operation and we choose $\hat{n}_3 = \hat{n}_{j_1}$. This translates to applying the following operations to the four-momenta of each events:

1. Rotate along z-axis such that $\phi_0 = 0$. We choose two instances of $\phi_0 \in \{\phi_{\text{MET}}, \phi_{j_1}\}$.

2. Reflect along the xy-plane, such that the leading jet's η is always positive.

After these two steps, the tower-constituents are binned in the resolutions as mentioned earlier and then padded on the ϕ-axis. We denote the feature-spaces obtained after preprocessing with the two instances of ϕ_0 as $\mathcal{P}_{\text{MET}}$ and $\mathcal{P}_J$. Figure 3.8 shows the different steps of preprocessing steps for an event taking $\phi_0 = \phi_{j_1}$. Averaged low-resolution image of the validation dataset of each class without preprocessing, and for both instances of ϕ_0 are shown in figure 3.9. As emphasised earlier, it is seen that there is a better regularisation when $\hat{n}_2 = \hat{n}_3$ ($\phi_{j_1} = 0,\ \eta_{j_1} > 0$). Clearly, the dominant features are the jets, and while for $\mathcal{P}_J$, these lie in the center; for $\mathcal{P}_{\text{MET}}$ they lie at the ϕ-boundary. Thus, the effect of padding is much more pronounced in $\mathcal{P}_{\text{MET}}$. In analogy, it becomes crucial when the Higgs boson decays in a hadronic channel (say $h^0 \to b\bar{b}$ or even $h^0 \to \tau^+\tau^-$), where we would desire the jets arising from Higgs – be it normal or large-radius, to be at the center of the image. Combining the instances of preprocessing and resolutions, there are four low-level feature spaces, namely: $\mathcal{P}^{LR}_{\text{MET}}$, $\mathcal{P}^{HR}_{\text{MET}}$, $\mathcal{P}^{LR}_J$ and $\mathcal{P}^{HR}_J$. The superscripts LR and HR denote the low and high-resolutions. We notice that all the high-level variables except ϕ_{MET}, are invariant under the two preprocessing operations, although, for our purpose, we extract

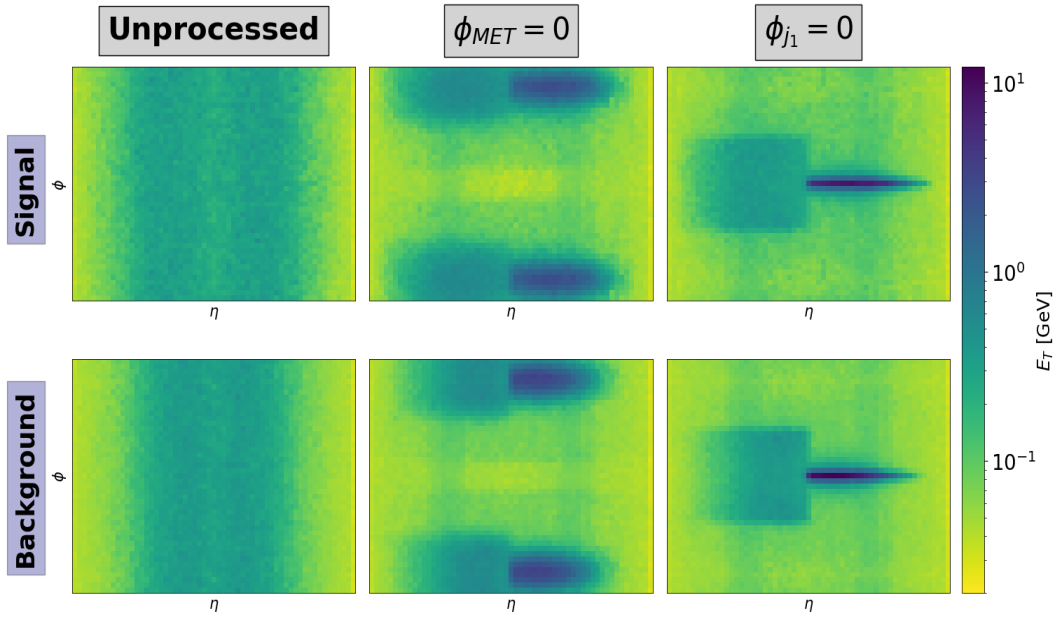


Figure 3.9: Average of 25,000 low-resolution tower-images of (left) unprocessed, (center) processed image with $\phi_{\text{MET}} = 0$ and (right) $\phi_J = 0$ for (top panel) signal and (bottom panel) background classes. The images are binned in the full range of the tower: $\eta \in [-5, 5]$ and $\phi \in [-\pi, \pi)$. We can see that as we go from left to right, there is a discernible improvement in regularisation of the features. There are no distinctly localised hard regions for the unprocessed case, while there are some for the $\phi_{\text{MET}} = 0$ instance, which becomes harder for $\phi_{j_1} = 0$ case with the hardest region around the leading jet.

them prior to their application. This follows from the usual physical intuition that absolute positions in the lab-frame are of no particular importance, and the useful information comes from the relative position of the different final-states.

We regularise the high-level features by mapping the distribution of each variable to their z-scores. Calculating the mean $\bar{x}^j$, and the standard deviation σ^j for each feature of the whole dataset (training and validation data of both classes together), we perform the following operation on each variable of all events,

$$z_i^j = \frac{x_i^j - \bar{x}^j}{\sigma^j} \quad . \tag{3.3}$$

The superscript j denotes the feature index, and the subscript i denotes the per-event index. It is particularly useful since the features have very different ranges (for instance, m_{jj} and $|\Delta\eta_{jj}|$), and the operation minimises this disparity. Furthermore, the features of z^j are now dimensionless. A caveat here is that the values of mean and standard deviations used are calculated from a balanced dataset. In experimental data, the presence of both classes, if at all, there is a positive signal, is never balanced. We justify our choice by their class independence, by virtue of which the relative differences in the shape of the signal and

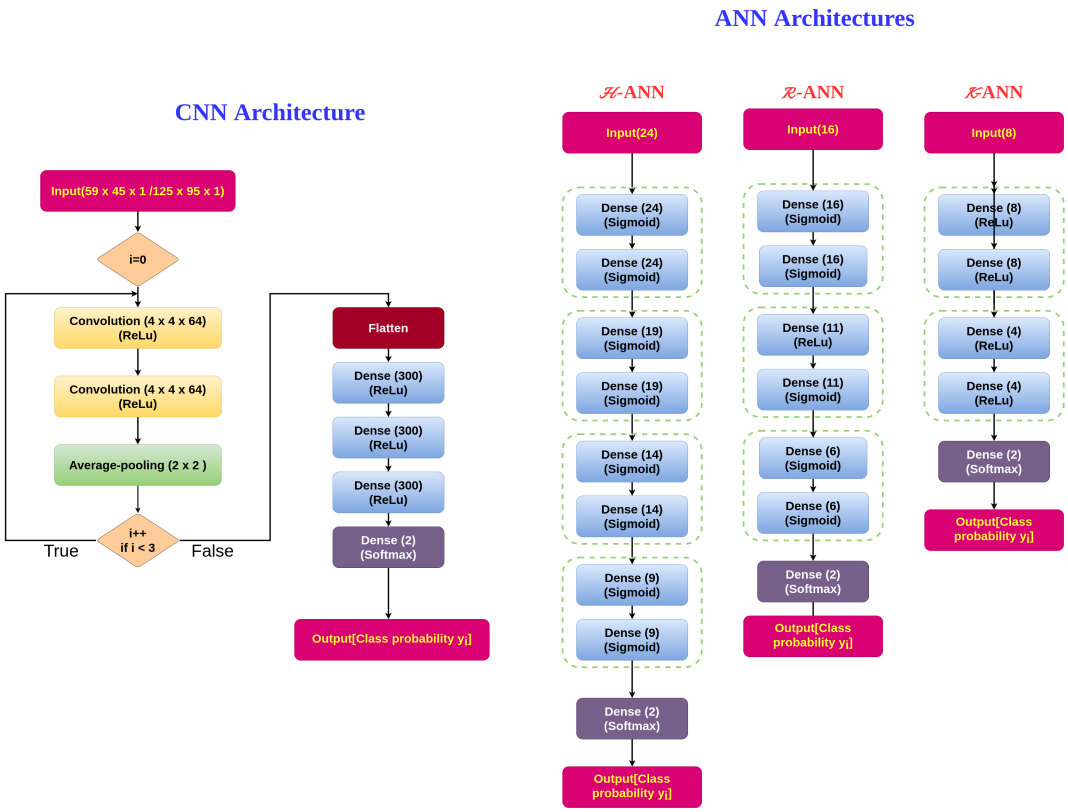


Figure 3.10: Simplified architecture of (left) CNNs and (right) ANNs.

background distributions are conserved, and the same set of values can be used when applying to unknown data with no labels.

3.4 Neural Network architecture and performance

In the previous sections, we have defined seven feature spaces, which are broadly grouped into high-level classes comprising of $\mathcal{K}$ (kinematic), $\mathcal{R}$ (QCD-radiative) and $\mathcal{H}$ (a combination of the two previous spaces); while the low-level spaces are: $\mathcal{P}^{LR}_{\text{MET}}$, $\mathcal{P}^{HR}_{\text{MET}}$, $\mathcal{P}^{LR}_{J}$ and $\mathcal{P}^{HR}_{J}$. With these as inputs, we train neural networks for classification. The generic architecture chosen for the high-level feature spaces are dense Artificial Neural Networks (ANNs) while for low-level ones are Convolutional Neural Networks. Hence, we name the 7 networks as: $\mathcal{K}$-ANN, $\mathcal{R}$-ANN, $\mathcal{H}$-ANN, $\mathcal{P}^{LR}_{\text{MET}}$-CNN, $\mathcal{P}^{HR}_{\text{MET}}$-CNN, $\mathcal{P}^{LR}_{J}$-CNN and $\mathcal{P}^{HR}_{J}$-CNN. All networks were executed in `Keras` (v2.2.4) [232] with the `TensorFlow`(v1.14.1) [230] backend.

3.4.1 Choice of hyperparameters

The CNN is composed of three modules with each module formed by two convolutional layers followed by an average-pooling layer. Each convolutional layer

consists of sixty-four filters with a size 4×4, with a single stride in each dimension. We pad all inputs to maintain the size of the outputs after each convolution. The pool-size is set to be 2×2 for all three modules with 2×2 stride size. The third module's output is flattened and fed into a dense network of three layers having three hundred nodes each, which we pass into the final layer with the two nodes and softmax activation. The convolutional layers and the dense layers before the final layer have ReLu activations. In total, the CNNs for the high-resolution (low-resolution) images have approximately 3.7 (1.2) million trainable parameters. The information bottleneck principle [276] inspires the ANN architectures. It has close connections to coarse-graining of the renormalisation-group evolution and was, in fact, priorly pointed out in reference [277]. We choose the number of nodes in the first layer to be equal to the number of input-nodes, which is then successively reduced after two layers of the same dimension.[§] These reductions in successive nodes are chosen to be five for the $\mathcal{R}$-ANN and $\mathcal{H}$-ANN, while for $\mathcal{K}$-ANN, we consider four due to the low-dimensionality of the input. We stop this process when there is no further reduction possible, or after four such reductions. We checked two activation functions: sigmoid and ReLu for the ANNs. We found that sigmoid activation gave the best validation accuracy for $\mathcal{R}$-ANN and $\mathcal{H}$-ANN, while it decreased over ReLu activations for $\mathcal{K}$-ANN. In total, the $\mathcal{K}$-ANN, $\mathcal{R}$-ANN, and the $\mathcal{H}$-ANN have 210, 991, and 2790 trainable parameters, respectively. Since this is a first exploratory study, we do not optimise the hyperparameters and use the values specified here for extracting the results. Simplified architecture flowcharts for each of the different networks are given in figure 3.10.

We chose categorical cross entropy as the loss function. We used Nadam [172] optimiser with a learning rate of 0.001 to minimise the loss function for all neural-networks. The optimiser's adaptive nature: smaller updates for frequently occurring features while larger updates for rare features, helps in better convergence for the sparse image-data that we have, with the added benefits of Nesterov accelerated gradient descent [278]. Moreover, the learning rate is no longer a hyperparameter. For the CNNs, training does not require more than ten epochs to reach optimal validation accuracy. Nevertheless, we train them five times from random initialisation for twenty epochs. The ANNs are trained for more epochs since the relatively fewer parameters make the convergence slower. For the ANNs, ReLu activation networks are trained for two hundred epochs. In comparison, sigmoid activation networks are trained for one thousand epochs due to their relative difference in convergence compounded with fewer parameters. A batch size of three hundred was chosen for training all networks. Each model, including all of its parameters, is stored after every epoch in the Keras-provided "hdf5" format during training. Out of these, we use the best performing model with the highest validation accuracy for further analysis.

[§]This provides stability of the representations learned at each dimension

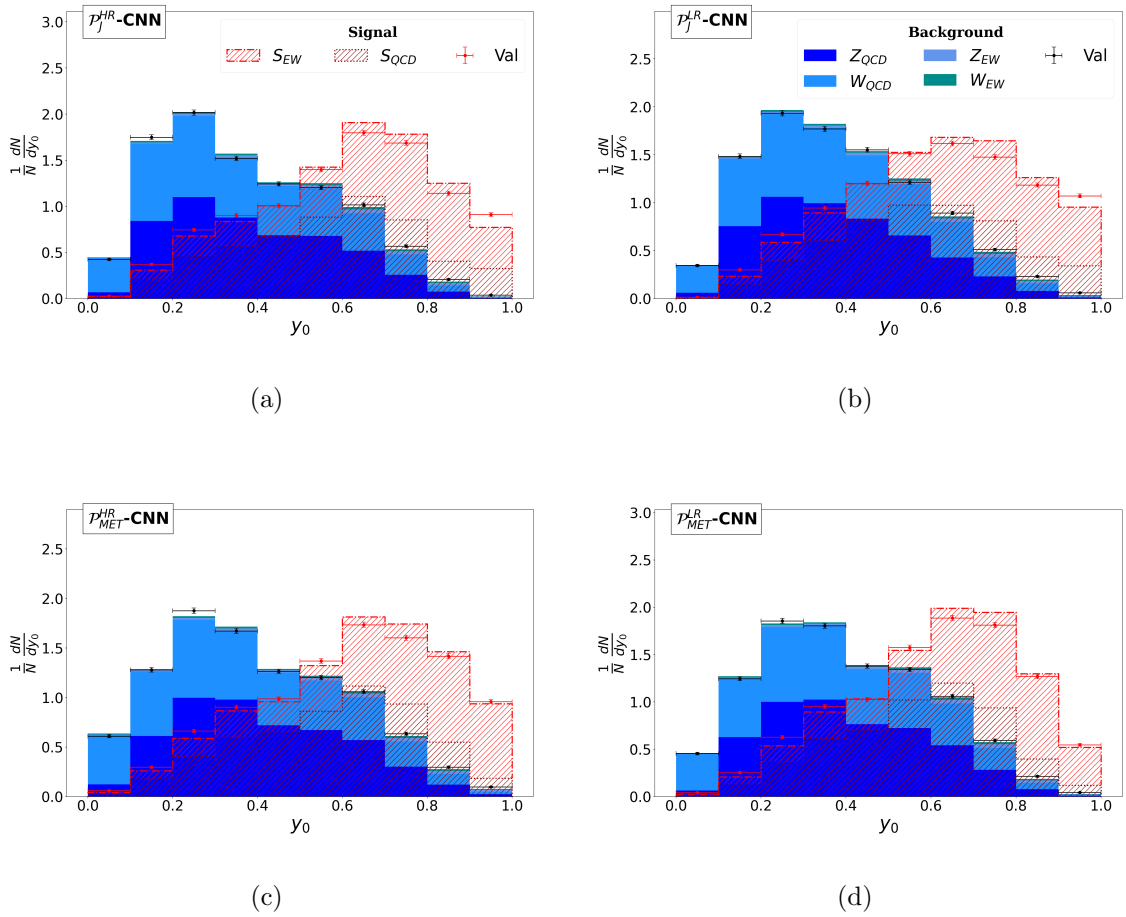


Figure 3.11: Binned distribution of the network output for (a) $\mathcal{P}_J^{HR}$-CNN (top-left), (b) $\mathcal{P}_J^{LR}$-CNN (top-right), (c) $\mathcal{P}_{\text{MET}}^{HR}$-CNN (bottom-left) and (d) $\mathcal{P}_{\text{MET}}^{LR}$-CNN (bottom-right).

3.5 Results

3.5.1 Network Performance

We extract the network output y_0, which is the probability of the event being a signal, from the best performing model from each network class. The class-wise binned distribution of y_0, for training and validation datasets of the low-level and high-level feature spaces, are shown in figure 3.11 and 3.12, respectively. These also show the channel wise contribution to their parent class. The choice of binning is set to the same ones used in extracting the bounds on the invisible branching ratio of the Higgs in Sect. 7.4. It has been set such that the minimum number of entries of each class for the validation data in the edge bins have enough numbers to reduce the statistical fluctuations to less than 15%. Contributions of the S_{EW} and S_{QCD} components to the signal class follow a definite pattern. As expected, all networks find it difficult to distinguish the S_{QCD} signal from the QCD dominated background. Hence, S_{QCD} contributes more in the bins closer

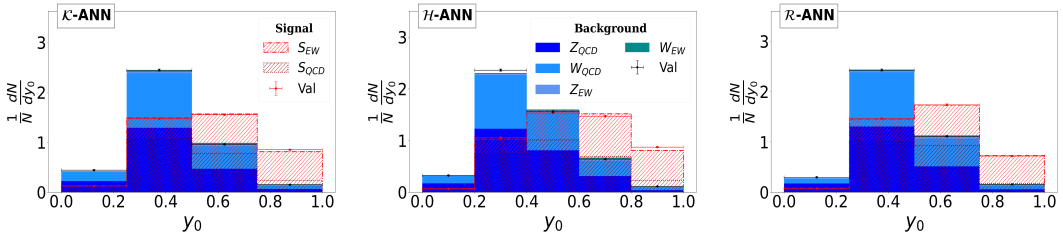


Figure 3.12: Binned distribution of the network output for (left) $\mathcal{K}$-ANN, (center) $\mathcal{H}$-ANN, and (right) $\mathcal{R}$-ANN.

to zero, which is governed by the background class. S_{EW} shows the opposite behavior dominating near one. This same feature, although a little inconspicuous, is present for the background class's EW subset as well. It may be pointed out that even for traditional analysis methods, there is significant contamination from S_{QCD}. A relevant machine-learning paradigm [279] where mixed samples are used in place of pure ones, could have an interesting application in reducing this S_{QCD} contamination of the signal for precision studies. Another notable feature prominent in the CNN outputs is the relative contribution of the Z_{QCD} and W_{QCD} channels to the background in the first bin, which is dominated by W_{QCD}. This can be apprehended from the fact that some of the leptons from $W^{\pm}$ decay, although not reconstructed, can still make calorimeter deposits on top of the QCD radiation to make itself visible to CNNs.

Receiver operating characteristic (ROC) curves between the signal acceptance ϵ_S, and the background rejection $1/\epsilon_B$; and also the area under the curve (AUC) for all networks are shown in figure 3.13. The AUCs were calculated using y_0 and the true class labels y_t with the scikit-learn(v0.22) [280] package. It is interesting to see that the so-called QCD-radiative variables ($\mathcal{R}$) perform almost as good as the kinematic-variables ($\mathcal{K}$) with only less than a percent difference in the validation AUCs. It can be understood by recalling that the radiative variables' definition includes the radiation pattern of the event, including the radiation inside the jet in cumulative η bins. This, in principle, has similar information to $|\Delta\eta_{jj}|$, which is one of the kinematic-variables with high separation. We confirm this by observing the correlations (shown in figure 3.14) between the variables $H_T^{\eta_C=2.07}$ and $H_T^{\eta_C=1.8}$ with $|\Delta\eta_{jj}|$ and m_{jj}. They are relatively more correlated with $|\Delta\eta_{jj}|$ than with m_{jj}. The AUC for our combined variable $\mathcal{H}$-ANN shows that the $\mathcal{R}$ variables may contain some extra information on top of what is extracted from the kinematic variables. As emphasised earlier, we get less than 0.1 percent difference in the validation AUCs of the low and high-resolution networks. The difference in AUC between $\mathcal{P}_J$ and $\mathcal{P}_{\text{MET}}$, although small, is still significant. It can be understood by looking at figure 3.9: there is better feature regularisation in $\mathcal{P}_J$ due to the choice of ϕ_0 than in $\mathcal{P}_{\text{MET}}$. CNNs, in general, are supposed to be robust to these kinds of differences owing to their properties of translational invariance [197]. In our case, the presence of fully-connected layers

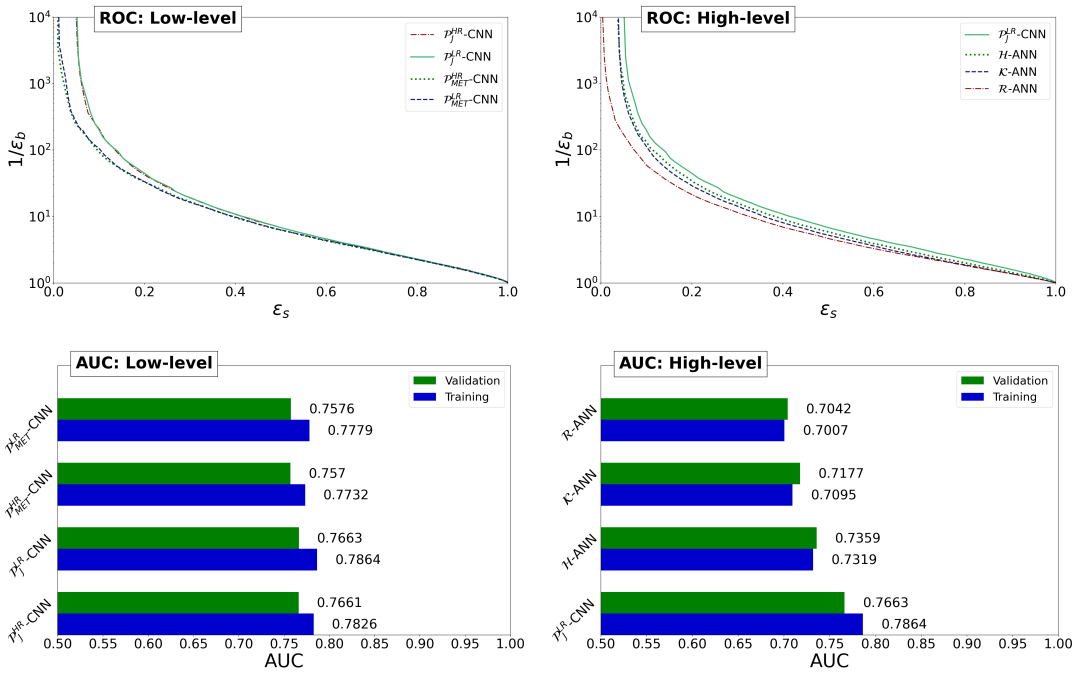


Figure 3.13: The validation (top panel) ROC-curves and (bottom panel) training/validation AUC for (left) low-level and (right) high-level feature spaces. In order to compare the feature spaces, the highest performing CNN is added to the plots on the right. The x-axis of the ROC-curve is the signal acceptance ϵ_S, while the y-axis is the inverse of background acceptance ϵ_B.

and the relatively small training sample hamper the generalisation power of the CNNs. Application of global-pooling instead of using fully-connected layers and an increase in data size coupled with proper hyper-parameter optimisation should reduce this difference in AUCs. These can be explored in future studies.

The class-wise linear correlation matrix between the network-outputs, along with the four high-level variables possessing the highest separations, are shown in figure 3.14. As expected, the outputs within the respective subset of networks are highly correlated. The outputs of the ANNs and the CNNs are also correlated significantly. A closer look reflects the addition of information in the high-level feature spaces: the correlations increase as we go from $\mathcal{R}/\mathcal{K}$-ANN to $\mathcal{H}$-ANN. In fact, if we extrapolate this argument in conjunction with the relative increase in AUC, we find that the CNNs have extracted the most information from the low-level data, which is not present in any of the high-level variables. A detailed description of the correlation of high-level variables and the ANN outputs are given in C.

3.5.2 Bounds on Higgs invisible Branching Ratio

In order to quantify our network performance in terms of expected improvements in the invisible Higgs search results at LHC, we obtain expected upper limits

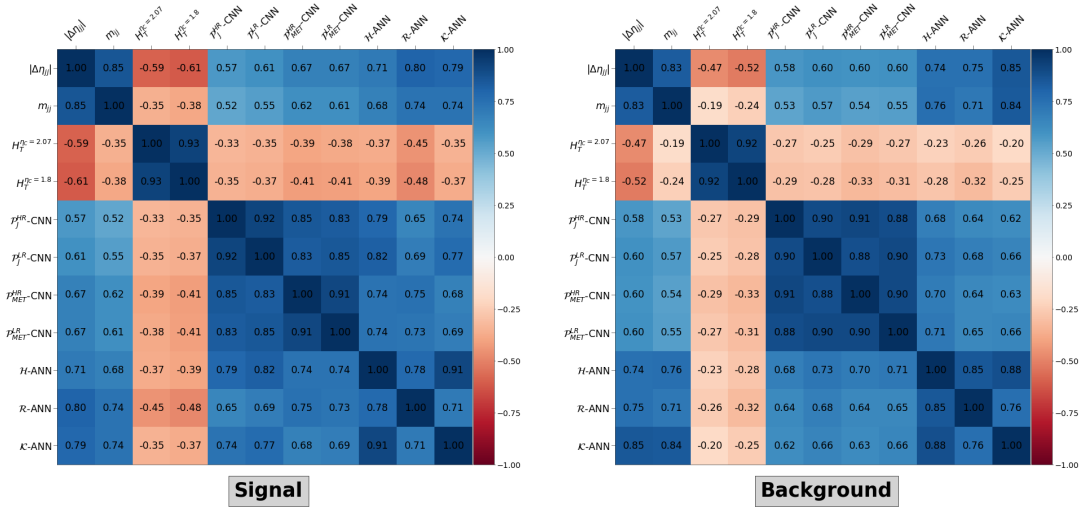


Figure 3.14: Pearson's correlation coefficients amongst the first four high-level variables with highest separation and the network-outputs for (left) signal and (right) background. These have been calculated using the validation dataset.

on the Higgs to invisible BRs from the distribution of the network output. We use CL_s method [281, 282] in the asymptotic approximation [283], to calculate the upper limit on the invisible BR at 95% CL. The method is briefly discussed as follows. In a binned Poisson counting experiment of expected signal s_i and background b_i (which are functions of nuisance parameters jointly denoted by $\boldsymbol{\theta}$) in a bin with observed number n_i of some observable, we can write the likelihood function as:

$$\mathcal{L}(\mu, \boldsymbol{\theta}) = \prod_{i=1}^{N_b} \frac{(\mu\, s_i(\boldsymbol{\theta}) + b_i(\boldsymbol{\theta}))^{n_i}}{n_i!}\, e^{-(\mu\, s_i(\boldsymbol{\theta}) + b_i(\boldsymbol{\theta}))} \quad , \tag{3.4}$$

where N_b is the total number of bins. N_b and the bin-edges for the different variables are chosen as shown in their respective distribution plots (figures 3.5, 3.6, 3.11 and 3.12). The profile-likelihood ratio:

$$\lambda(\mu) = \frac{\mathcal{L}(\mu, \hat{\hat{\boldsymbol{\theta}}})}{\mathcal{L}(\hat{\mu}, \hat{\boldsymbol{\theta}})} \quad , \tag{3.5}$$

where the arguments of the denominator maximises $\mathcal{L}$, and $\hat{\hat{\boldsymbol{\theta}}}$ conditionally maximises $\mathcal{L}$ for the particular μ, is used as a test-statistic in the form of log-likelihood,

$$t_\mu = -2\ln(\lambda(\mu)) \quad . \tag{3.6}$$

The distribution of the test statistic for different values of μ, is required to extract frequentist confidence intervals/limits. Since, we have fixed the total weight of the signal events with respect to the background to correspond to the ones expected

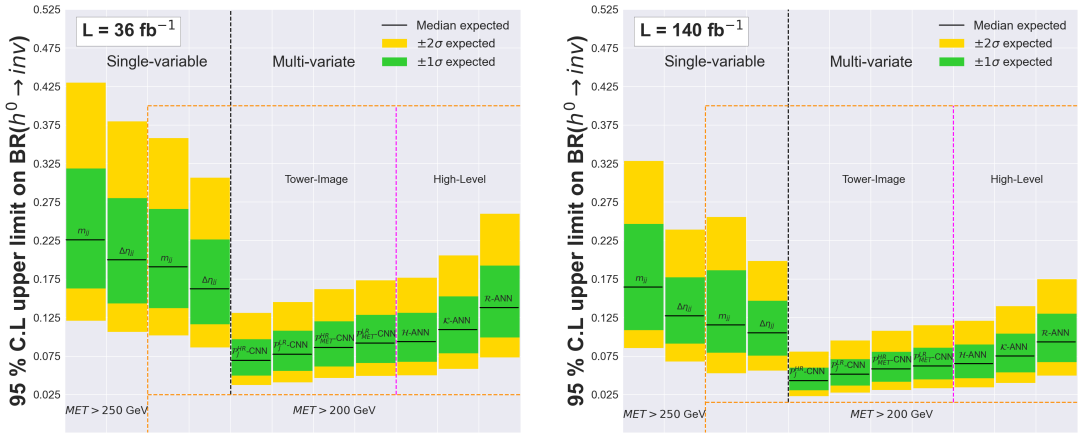


Figure 3.15: Expected 95% C.L median upper limit on the invisible branching ratio of SM Higgs with one and two sigma sidebands for (left) 36 fb^{-1} and (right) 140 fb^{-1} integrated luminosities.

with the total expected production cross-section from SM for each channel(S_{EW} and S_{QCD}), μ corresponds to the invisible branching ratio of the Higgs. In the asymptotic method, for one parameter of interest, approximate analytical expressions for the distribution are derived using a result from Wald [284], in the form of a non-central Chi-square distribution. Monte-Carlo simulations required to extract the unknown parameters are by-passed by choosing the best representative data called the Asimov data, by the authors of reference [283]; which is defined as the data when used to estimate the parameters, produces their true values.

We used `HistFactory` [285] to create the statistical model, and the `RooStats` [286] package to obtain the expected limits. This provides us with greater ease of handling systematic uncertainties. As stated before, we also redo the shape-based analysis of reference [261] with our dataset only considering a few simpler systematics, to consistently gauge the increased sensitivity of the deep-learning approach. We incorporate three overall-systematics: uncertainty of the total cross-section, statistical uncertainty of Monte Carlo simulated events, and approximate luminosity uncertainties. We do not take into account the possible change in the shape of the distributions due to Monte Carlo simulation effects. The per-bin statistical error is taken into consideration by activating each sample's statistical-error while creating the statistical model in HistFactory. This is essentially a shape-systematics that considers the bin-wise change in shape due to the statistical uncertainties. Its inclusion increases the median expected upper-limit by around three percent in the reproduced shape-analysis. The number of events for the analysis with the higher MET cut is set to the expected number at 36 fb^{-1} for all background channels. This result is also scaled for the other luminosities. For the ones with the lower MET cut, we use the validation data scaled by appropriate weights for the respective luminosities.

The median expected upper limit on the invisible branching ratio of SM Higgs

Sl.No	Name	Description	Expected median upper-limit on BR($h^0 \to$ inv)		
			L = 36 fb^{-1}	L = 140 fb^{-1}	L = 300 fb^{-1}
1.	m_{jj}(MET > 250 GeV)	reproduced shape analysis of reference [261]	$0.226^{+0.093}_{-0.063}$	$0.165^{+0.082}_{-0.056}$	$0.130^{+0.089}_{-0.027}$
2.	$\lvert\Delta\eta_{jj}\rvert$(MET > 250 GeV)	$\lvert\Delta\eta_{jj}\rvert$ analysis with shape-cuts of reference [261]	$0.200^{+0.080}_{-0.056}$	$0.128^{+0.050}_{-0.036}$	$0.106^{+0.041}_{-0.025}$
3.	m_{jj}(MET > 200 GeV)	m_{jj} shape analysis with weaker cut	$0.191^{+0.075}_{-0.053}$	$0.116^{+0.071}_{-0.036}$	$0.101^{+0.037}_{-0.045}$
4.	$\lvert\Delta\eta_{jj}\rvert$(MET > 200 GeV)	$\lvert\Delta\eta_{jj}\rvert$ analysis with weaker cut	$0.162^{+0.065}_{-0.045}$	$0.105^{+0.042}_{-0.029}$	$0.087^{+0.034}_{-0.025}$
5.	$\mathcal{P}_J^{LR}$-CNN	Low-Resolution, $\phi_0 = \phi_{j_1}$	$0.078^{+0.030}_{-0.022}$	$0.051^{+0.020}_{-0.014}$	$0.045^{+0.017}_{-0.013}$
6.	$\mathcal{P}_J^{HR}$-CNN	High-Resolution, $\phi_0 = \phi_{j_1}$	$0.070^{+0.027}_{-0.020}$	$0.043^{+0.017}_{-0.012}$	$0.035^{+0.013}_{-0.010}$
7.	$\mathcal{P}_{\text{MET}}^{LR}$-CNN	Low-Resolution, $\phi_0 = \phi_{\text{MET}}$	$0.092^{+0.037}_{-0.025}$	$0.062^{+0.024}_{-0.017}$	$0.053^{+0.023}_{-0.014}$
8.	$\mathcal{P}_{\text{MET}}^{HR}$-CNN	High-Resolution, $\phi_0 = \phi_{\text{MET}}$	$0.086^{+0.035}_{-0.024}$	$0.058^{+0.023}_{-0.016}$	$0.051^{+0.020}_{-0.014}$
9.	$\mathcal{K}$-ANN	8 kinematic-variables	$0.101^{+0.052}_{-0.022}$	$0.075^{+0.029}_{-0.021}$	$0.063^{+0.027}_{-0.017}$
10.	$\mathcal{R}$-ANN	16 radiative $H_T^{\eta_C}$ variables	$0.138^{+0.055}_{-0.039}$	$0.094^{+0.036}_{-0.027}$	$0.079^{+0.032}_{-0.022}$
11.	$\mathcal{H}$-ANN	Combination of $\mathcal{K}$ and $\mathcal{R}$ variables	$0.094^{+0.038}_{-0.026}$	$0.065^{+0.026}_{-0.018}$	$0.057^{+0.022}_{-0.015}$

Table 3.1: Short description of the different analyses shown in figure 3.15 and the expected median upper-limit on BR($h^0 \to$ inv) at 95% CL for each integrated luminosities which also include projections for L = 300fb^{-1}.

at 95% CL along with the one and two sigma error bands are shown in figure 3.15 for integrated luminosities of 36 fb^{-1} and 140 fb^{-1}. A short description of the datasets used, and the corresponding median-expected upper limits with 95 % CL is tabulated in Table 3.1. This also contains the projected limits for 300 fb^{-1}, the integrated luminosity expected at the end of LHC Run III. We emphasise that even though we scale to 300 fb^{-1} luminosity, we use the same dataset, and hence, the statistical uncertainties are not reduced. Consequently, our estimation for 300 fb^{-1} is a conservative one. First and foremost, one can notice that the reproduced result of the shape-analysis of reference [261] for an integrated luminosity of 36 fb^{-1} is quite consistent, and the difference can be accounted to the excluded background channels and experimental systematics. We repeat this analysis with the weaker selection criteria and see a modest improvement in the median-expected upper-limit. We also perform similar analyses with $|\Delta\eta_{jj}|$ distributions, and get an improvement of 2.9 % for MET > 200 GeV, and 2.6 % for MET > 250 GeV cuts. The worst (best) performing neural-network $\mathcal{R}$-ANN ($\mathcal{P}_J^{HR}$-CNN) has an improvement of 8.8% (14.6%) from the repeated experimental analysis. This, although, is with different cuts, and for the same cut in MET, we have an improvement of 5.3% (12.1%) for $\mathcal{R}$-ANN ($\mathcal{P}_J^{HR}$-CNN). For an integrated luminosity of 140 fb^{-1}, we get an improvement of 2.2 % and 7.3 % for $\mathcal{R}$-ANN and $\mathcal{P}_J^{HR}$-CNN, respectively. The reduced difference for higher luminosities is, of course, expected since the significance does not scale linearly with an increase in data size. An expected median upper-limit of about 3.5% can be achieved with 300 fb^{-1} of data using the highest performing network, $\mathcal{P}_J^{HR}$-CNN.

The results of the different feature spaces follow the expected trend. For this discussion, we quote the numbers for an integrated luminosity of 36 fb^{-1}. Comparing the performance of high-level feature spaces, we see that $\mathcal{R}$ performs

the worst while the combined space $\mathcal{H}$ puts the most stringent bounds. The difference is minimal (0.7 %) with $\mathcal{K}$-ANN, and appreciable (4.4%) with $\mathcal{R}$-ANN. Amongst the image-networks, the difference between the low and high-resolution networks is less than a percent (0.8 % for $\mathcal{P}_J$, and 0.6% for $\mathcal{P}_{\text{MET}}$). Differences in performances of the different preprocessing instances are reflected in this analysis: $\mathcal{P}_J$ puts nominally stricter bounds on the branching ratio (1.4 % for LR, and 1.6 % for HR).

Up to now, we demonstrated the capability of our CNN based low-level networks and also ANN-based networks considering particle level data, including detector effects as well as underlying events during our simulations as discussed in Section 3.1. However, we neglected the effect of simultaneous occurrences of multiple proton-proton interactions (pileup) in our analysis. The amount of pileup is relatively moderate in low luminosity data, but increasingly significant once we move towards high luminosity. We believe that its presence would not alter our primary results substantially from the calorimeter image data. CNN architectures look into the global features of an input image. Calorimeter deposits due to pileup are expected to be similar for different classes since they are independent of the hard scattering processes. The same can be identified as redundant information, as a consequence of the optimisation algorithm effectively searching for dissimilarities between the two classes. Optimal pdfs acquired by CNNs remain very similar, whether it is with or without pileup. This issue was analysed before, where it was shown that unlike high-level methods, deep-learning from calorimeter deposits shows robustness to pileup effects in the classification of jet-image [119]. Although, in these studies, the jets have large transverse boosts and mostly reside in the central regions where its effect is reduced. However, various other studies [193, 194] have also shown that deep-learning on the full calorimeter information is less prone to pileup effects. These existing results further elucidate our presumption that CNNs would be less affected by higher pileup expected at future runs of LHC. In contrast, the other analyses, including the ANNs trained on high-level feature spaces, can be relatively more affected.

To present our arguments in perspective, we combined each event (tower-image) with an additional N randomly chosen minimum bias event with CMS switch through Pythia8 and Delphes without any pileup subtraction. At the same time, N follows a Poisson distribution with $< N >= 20, 50, 50$ for integrated luminosity 36, 140 and 300 fb^{-1}, respectively. Merged tower-image with pileup is then trained and tested for our high-resolution CNN scenario ($\mathcal{P}_J^{HR}$-CNN, which can be noted from Sl.No (6) in Table 3.1). We found a very mild depreciation over our estimated median upper-limit at 0.076, 0.059, and 0.045, which all lie within the 1σ error band in the branching ratio constraints. Note that no effort was made to mitigate the effects of the pileup during these estimates, which will not be the case in experimental analyses. In fact, there are extensive studies [153, 287] of using powerful machine-learning algorithms specially designed to reduce pileup contamination of events. A new interpretation of collider events in terms of

optimal transport [288, 289] have also provided promising new techniques for pileup mitigation on top of reinterpretation of existing ones [290, 291]. These developments offer further optimism for better mitigation of pileup effects in the future.

To test the robustness of our proposal, we also consider the effect of an important experimental systematic uncertainty. One of the significant experimental systematic uncertainties affecting the result of this analysis can be the uncertainty on the jet energy scale. Therefore, we estimate the effect of uncertainties on the jet energy scale for our main results with calorimeter input data in CNN architecture. We vary the pixel-wise input values (which has already gone through the smearing in Delphes) by 10% in upward and downward directions,[¶] and record the variation in the shape of the network output without considering any pileup. This is added as a coherent shape systematics, and we obtain an increased expected median upper-limit of $0.071^{+0.028}_{-0.019}$ for $\mathcal{P}_J^{HR}$-CNN at 140 fb^{-1} integrated luminosity, which is still better by a factor of almost two when compared to the latest result from ATLAS [262].

3.6 Summary

In this chapter, we have studied the capability of CNNs in identifying VBF production of Higgs from the dominant QCD backgrounds. We choose VBF production of the Higgs boson decaying to invisible particles as a case study for neural networks to learn the entire event topology without any reconstructed objects. We use the compelling capability of Convolutional Neural Networks (CNN) to examine the potential of deep-learning algorithms using low-level variables. Instead of identifying any particular objects, we utilise the entire calorimeter image to study the event topology, which aims to learn the difference in radiation patterns between the two forward jets of the VBF signal. We specifically develop preprocessing steps that preserve the Lorentz symmetry of the events and are essential for maximising the statistical output of the data.

Apart from low-level variables as calorimeter images for CNN, we also consider two sets of high-level features. One such set is based on the kinematics of the VBF, whereas the other set of variables are designed to portray the radiation pattern H_T calculated in different η ranges of the calorimeter. For a comprehensive analysis, we constructed several neural network architectures and demonstrated the comparative performance of CNN and ANN using different feature spaces. All these networks achieved excellent separation between signal and background. However, we found that CNN based low-level $\mathcal{P}_J^{HR}$-CNN performs the best among all the networks, which is based on the high-resolution images, although the

[¶]Reference [292] reports jet energy scale uncertainty for various observables, which lie well within 5%. However, since such uncertainties are significantly controlled in jets reconstructed with the particle-flow (PF) method, we take a relatively conservative measure for the pixel-wise uncertainty of the measured energies.

dependence on image resolution is relatively insignificant. We also note that deep-learning on the full calorimeter information is less prone to pileup effects as well. Without relying on any exclusive event reconstruction, this novel technique can provide the most stringent bounds on the invisible branching ratio of the SM-like Higgs boson, which can be expected to be constrained up to 4.3% (3.5%) using a dataset corresponding to an integrated luminosity of 140 fb^{-1} (300 fb^{-1}). These limits can severely constrain many BSM scenarios, especially in the context of (Higgs-portal) dark matter models. The techniques presented here can easily be extended to a more complex event topology.

Chapter 4

Sensitivity of CNNs to simulation aspects of VBF Higgs

In the last chapter, we have seen that CNNs can efficiently identify vector boson fusion produced Higgs events from those originating from the QCD background, concentrating on the invisible decay. In this chapter, we scrutinise the training and performance of the CNN against important factors in the data generation process. We note that it is essential to scrutinise the differences in leading-order (LO) and next-to-leading-order (NLO) simulations; the presence of a third jet needs the proper introduction of real and virtual corrections to the tree level process. Another issue of central importance in the simulation of VBF events is the inability of a global-recoil scheme in initial state radiations (ISR) of the parton showering algorithm to describe the central-jet activity correctly [293]. We will systematically investigate these issues for the VBF signal search with CNNs.

Although the preceding arguments apply to the VBF production of weak bosons, we presently ignore its effects as they are much less in proportion ($\sim 5\%$ of the total background for the cuts used here). We also neglect the contribution of the gluon-fusion events in the signal. The global recoil scheme correctly produces the leading logarithmic behaviour, already incorporated in our previous analysis. A precise determination of its various effects demands a very high level of sophistication, requiring much higher perturbative and logarithmic accuracy. Although the cuts used in the analysis have a sizable amount of gluon-fusion contribution, the large amount of data from high-luminosity LHC runs will provide ways to do precision analysis with more stringent cuts, with negligible contribution from gluon-fusion events. These do not impede our final goal, as our intention is not to project experimental sensitivities but to usher pragmatism and careful examination while using inclusive event information as inputs to DNNs.

Although DNNs generally perform better than ML algorithms utilising high-level variables, their usability in phenomelogical analyses is determined by our ability to simulate subtle aspects of the data accurately. To this end, we show the

possibility of CNNs learning inaccurate representations of inclusive events due to a global recoil used in the simulation of VBF events.

The rest of the chapter is organised as follows. In Section 4.1 we discuss the parton shower scheme and NLO effects in the simulation of VBF Higgs signal. In Section 4.2, we examine the impact of the different signal simulations on the trained network output and its performance. We conclude in Section 5.4.

4.1 Impact of NLO corrections and recoil schemes

Although VBF processes have relatively lower higher-order corrections, utilising the hadronic activity between the two tagging jets would use information not captured by a leading-order simulation. This inadequacy is due to the inherent assumption in parton shower generators, primarily focusing on the soft and collinear regions. A next-to-leading-order hard partonic simulation merged with a parton shower algorithm would accurately describe the kinematics of the third leading jet (if present) over the full range of transverse momentum. Additionally, for event topologies with no colour flow between the two incoming partons from the colliding protons, a parton shower algorithm with a global-recoil scheme for the initial-state radiation (ISR) is known to have a further inefficient simulation of the wide-angle soft radiation patterns. The cause for this inaccuracy is due to the incorrect assumption of an `II` dipole in the global-recoil scheme [293], while VBF processes have a double DIS scattering topology with an `IF/FI` dipole structure. Existing phenomenological studies [294, 295] are consistent with this known limitation of the global-recoil scheme, and recent experimental results [296–298] have employed the dipole recoil scheme [299–302] for the relevant VBF topologies. The effects of both higher-order virtual corrections and the recoil scheme are even more important when using powerful deep-learning algorithms with low-level inputs.

4.1.1 Signal generation

Since VBF Higgs processes are our primary interest, we do not include the gluon-fusion processes in the present analysis. We, therefore, study the four different possible combinations of the perturbative accuracy and the parton shower's recoil scheme for the VBF channel. These are described as follows:

1. **Global LO**: Parton level events simulated at leading-order perturbative accuracy showered with a global-recoil scheme for the ISR parton shower. This recoil scheme is the default implementation in `Pythia8` and was used in chapter 3 for the VBF processes.

2. **Dipole LO**: Parton level events simulated at leading-order perturbative accuracy showered with a dipole-recoil scheme.

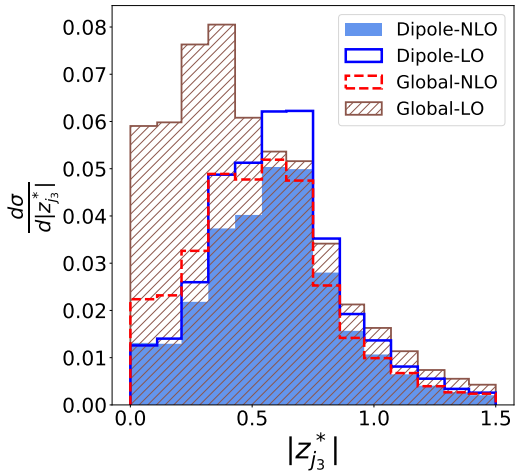


Figure 4.1: Distribution of the absolute Zeppenfeld variable $|z^*_{j_3}|$ for the four signal simulations. To capture the relative occurrence of the third jet, we set each event weight so that the total sum with or without an additional jet in each signal simulation sum to unity.

3. **Global NLO**: Parton level events simulated at next-to-leading order accuracy merged with parton shower employing the global-recoil scheme for ISR.

4. **Dipole NLO**: Parton level events simulated at next-to-leading order accuracy merged with parton shower using the dipole-recoil scheme.

We use the same set of parton-level events for the LO and NLO simulations to shower with the two recoil schemes. The parton-level events at LO were generated with `MadGraph5_aMC@NLO`, while the NLO events were generated with the `POWHEG-BOX` [62–65]. The renormalisation and factorisation scales for both orders are set for each event as,

$$\mu_0^2 = \frac{m_h}{2}\sqrt{\left(\frac{m_h}{2}\right)^2 + p_{T,h}^2} \quad , \tag{4.1}$$

where m_h =125 GeV is the mass of the Higgs and $p_{T,h}$ is the transverse momentum of the Higgs boson in the event. For the parton level generation, we use the `PDF4LHC15_nlo_100_pdfas` [303] parton distribution function (PDF) set implemented with `LHAPDF6` [304] (v6.1.6) package. This PDF set is a combination [305] of `CT14` [306], `MMHT14` [307], and `NNPDF3.0` [308] PDF sets using the Hessian reduction method proposed in reference [309]. We use `MadSpin` [310] to decay the Higgs boson at parton-level to two scalar dark matter particles for the NLO events, while we simulate the full decay chain for the LO events. All parton showers are performed in `Pythia8.235`. For the NLO events, we perform the POWHEG-merging with recommended values from reference [311]. The switch to a dipole-recoil scheme is done by setting `"SpaceShower:dipoleRecoil=on"`

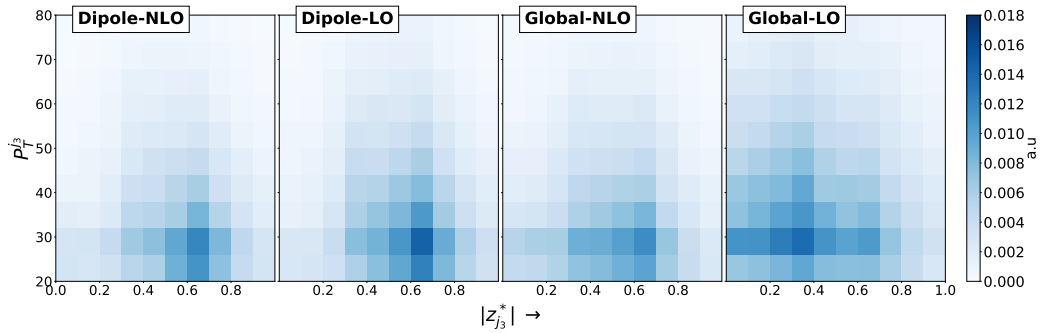


Figure 4.2: Two dimensional histogram of events with the transverse momentum($P_T^{j_3}$) of the third jet and $|z^*_{j_3}|$ for four different cases of signal simulations, such as, dipole NLO, dipole LO , global NLO and global LO of the VBF Higgs signal.

for the parton shower. We note that the events generated at NLO and showered with the dipole-recoil scheme should be the most physically accurate simulation of the VBF Higgs process. These four sets of showered events are then passed through the same detector simulation and selection criteria described in the previous chapter for the deep-learning analysis with MET $>$ 200 GeV. We divide the dataset of each of these simulations into 100k training and 25k validation samples for the neural network analysis.

4.1.2 Characteristics of the third jet

To compare the different signal simulations, we plot distributions of the Zeppenfeld variable z^*_{j3} in Figure 4.1 for events passing the selection criteria and having a third jet with $p_T > 20$ GeV. It is defined as,

$$z^*_{j_3} = \frac{\eta_{j_3} - (\eta_{j_1} + \eta_{j_2})/2}{|\Delta\eta_{j_1 j_2}|} \quad , \tag{4.2}$$

where η_{j_i} is the pseudorapidity of the i^{th} hardest jet, and $\Delta\eta_{j_1 j_2}$ is the rapidity gap between the two tagging jets. This variable looks at the position of the third jet relative to the tagging jets and is important when considering the additional information available beyond the two-jet system. We set the normalisation such that the cumulative sum of the bins corresponds to the fraction of events that satisfy the requirement on the third jet. The dipole-NLO signal has the least proportion of events passing the additional criteria at 30%, while the global NLO has 35%. The fraction for LO events with dipole and global recoil schemes are 37% and 55% respectively. From these values and the shape of the distribution in Figure 4.1, we can infer that out of the four, global LO should be most similar to the QCD-dominated background, and dipole-NLO should be the least identical. Consequently, we expect these to be reflected on the performance of any statistical model utilising radiative information beyond the two jets. Although the

proportion of events with a third jet is very close for global NLO and dipole-LO, note that the former has more jets in the central regions from the shape of $|z^*_{j_3}|$ distribution. Hence, we would expect better discrimination for Dipole-LO.

Even though $z^*_{j_3}$ is a good variable, a model like a CNN that uses the inclusive event information will use the third jet's position as well as its transverse momentum implicitly to find the decision boundary. To this end, in Figure 4.2 we plot the 2-D histogram plot of the transverse momentum $P_T^{j_3}$ of the third jet and $|z^*_{j_3}|$. Due to the artificial enhancement from the `II`-assumed global showering scheme in the central regions, we can see that the third jet is relatively harder than their dipole counterparts for both orders. Moreover, since the third jet results from the parton shower for LO, there is a drastic difference between global LO and dipole LO relative to the same comparison at NLO. From this, we can infer that events that do not have a third reconstructed jet would still follow the same pattern and expect the same effect on the performance of the CNN.

4.2 Results

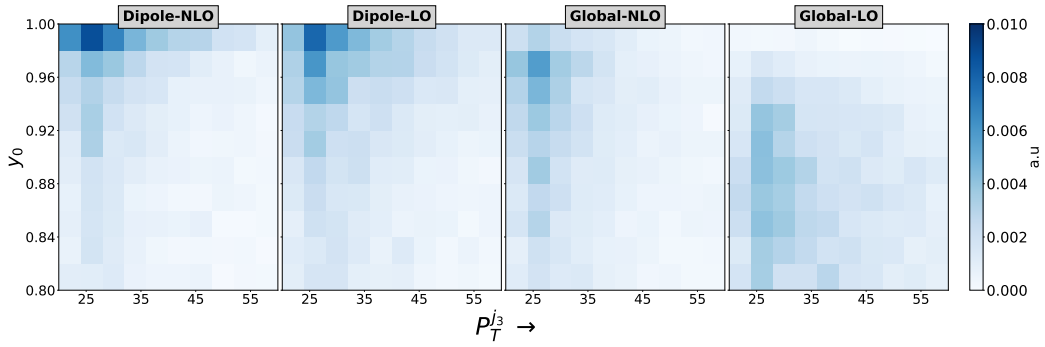
In this section, we examine the performance of CNNs in identifying the different simulations of the same signal from the same background dataset used in the previous chapter. When trained with the same architecture, the relative discrimination power should reflect the physical intuition we presented in the preceding section. The four sets of signal events are preprocessed so that $\phi_{j_1} = 0$ and $\eta_{j_1} > 0$ and binned with ther the lower resolution. Therefore, the network $\mathcal{P}_J^{LR}$-CNN is trained with the procedure used in the previous chapter. The performance on the higher resolution should follow the same trend, and hence unwarranted for the aim pf this study.

4.2.1 Effects of central radiation on the network output

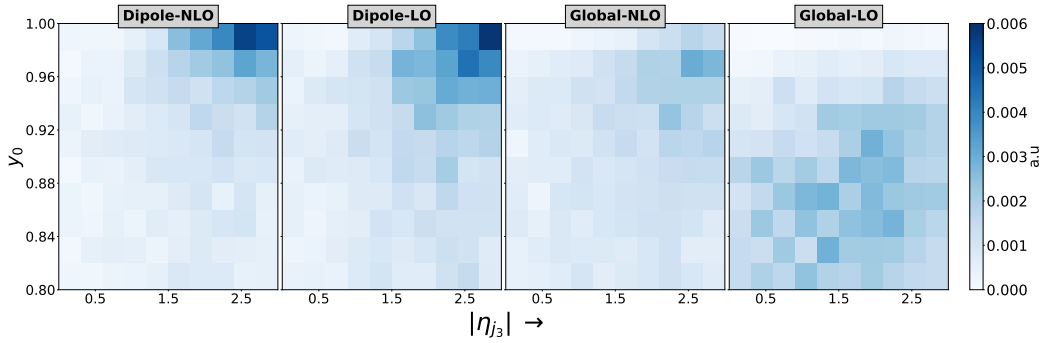
The two-dimensional histogram of the network output y_0 (the probability of an event being a signal) of the signal validation datasets with various variables quantifying the additional information beyond the two-jet system are shown in Figure 4.3. The weight of each event is set such that the total sum of all events with or without the third jet corresponds to one. Therefore, the total sum of the histogram with the physical quantities of the third jet corresponds to the fraction with at least one additional jet.* The comparatively lower concentration of events for the global LO simulation is due to the lower performance of the network (presented in Section 4.2.2) compared to the other three simulations.

In Figure 4.3(a), in which the histogram is with the transverse momentum of the third jet, we see that for the dipole recoil, both orders have the maxi-

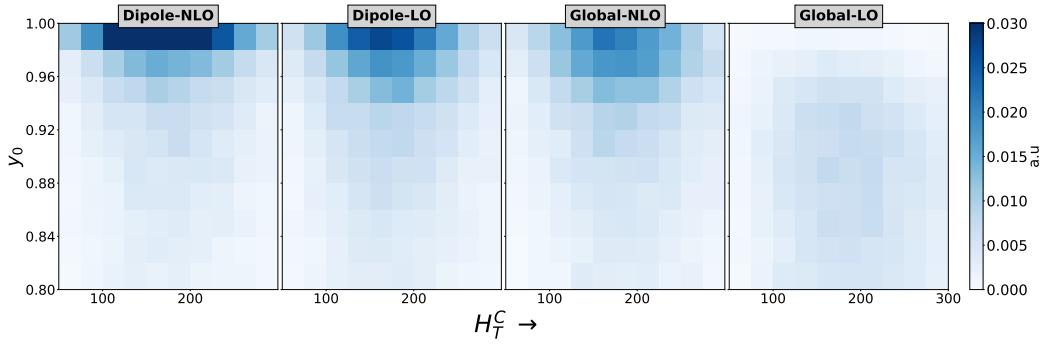
*Due to the range of the variables, the total sum is not equal to the fraction presented in Section 4.1.2



(a)



(b)



(c)

Figure 4.3: Two dimensional histogram of events of the network output y_0 for each signal simulation with the (a) $P_T^{j_3}$ and (b) $|\eta_{j_3}|$ of the third jet (when present), and (c) the H_T^C between the two tagging jets.

mum concentration of events in the top-left corner. The third jet has the least transverse momentum in this region, and the network identifies the event as most signal-like. For the case of the global recoil, we see that the NLO simulation has a higher concentration near the top-left corner. In contrast, the LO simulation has significantly reduced events near the top left, with the shift toward the bottom in the y-axis more prominent. The greater change in the network output can be understood by recalling from Figure 4.2 that the relative position of the third jet is much more central for the global LO simulation event if its transverse momentum is in a similar range. This property is further confirmed in Figure 4.3(b) where the histogram is on the $|\eta_{j_3}|$ and y_0 plane. The events for the global LO simulation is closer to the left side: implying that the third jets are much more central; and lower in the y_0 axis: indicating that the network identifies the signal less efficiently. Similarly, the same histogram for the dipole-recoil scheme and different orders shows a concentration of events in the top right corner, where the third jets are more forward, and the network identifies the signal with greater confidence.

To look collectively into the events with or without a third jet, we define the scalar sum of p_T between the two tagging jets as,

$$H_T^C = \sum_{\eta_i \in [\eta_l, \eta_u]} p_T^i \,, \tag{4.3}$$

where the range is determined by the pseudorapidity of the two jets: η_{j_1} and η_{j_2} mapped such that $\eta_l < \eta_u$. We do not remove the particles within the jets when calculating H_T^C, thus giving a non-zero value for all events. As expected, we see in Figure 4.3(c), that the dipole-NLO simulation has the highest proportion of events near the top left corner, followed by dipole LO and global NLO, with global LO having a larger concentration in the central regions of the (H_T^C, y_0)-plane. Therefore, we see that events without the third jet also follow a pattern similar to those with the additional jet.

4.2.2 Dependence of performance on the signal simulation

The normalised distribution of the network output y_0 of each class are shown in Figure 4.4 for the four different signal simulation approaches. One can see that the CNN trained and validated with the dipole-NLO simulation has the highest separation from the background. To better quantify the power, we look at the receiver operating characteristic (ROC) curves between the signal acceptance ϵ_S and the background rejection $1/\epsilon_B$, and the area under the ROC curve (AUC). These are shown in Figure 5.4. As expected, the highest discrimination is obtained for dipole NLO with a validation AUC of 0.9355, followed by dipole LO with 0.9243 validation AUC. Inadvertently, the dipole-NLO signal happens to be the most physically accurate simulation. The hierarchy suggests that the recoil

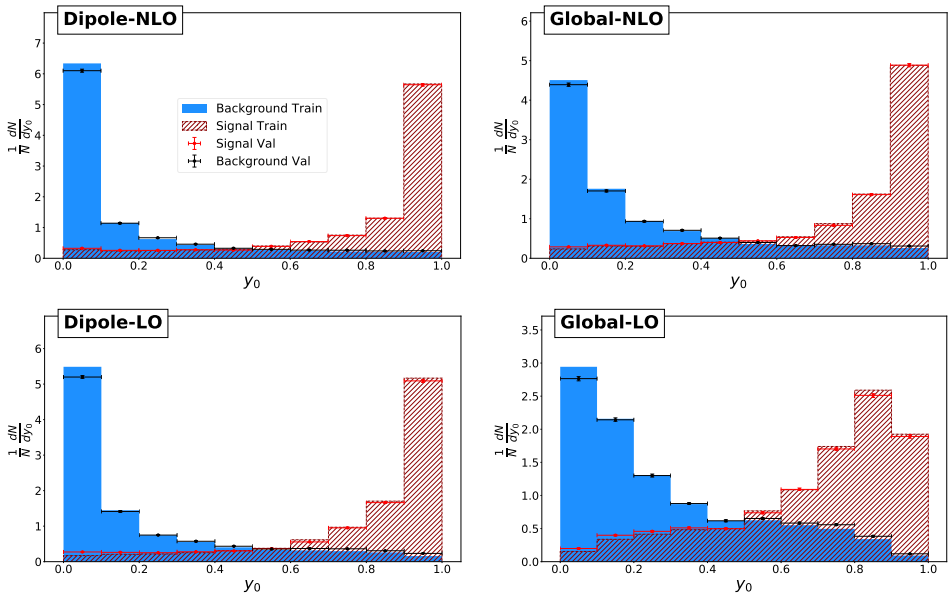


Figure 4.4: Normalised binned distributions of the network output discriminating background from signal class for four different instances of signal simulations.

Sl.No	Train Signal Dataset	Test Signal Dataset			
		Globalo-LO	Global-NLO	Dipole-LO	Dipole-NLO
1.	Global-LO	0.8599	0.8956	0.9027	0.9201
2.	Global-NLO	0.8486	0.9036	0.9112	0.9288
3.	Dipole-LO	0.8036	0.8878	0.9243	0.9335
4.	Dipole-NLO	0.8234	0.8922	0.9200	0.9355

Table 4.1: The table shows the test AUC evaluated for all signal simulation for each CNN trained on the different signal simulations.

scheme is of greater importance than the perturbative accuracy for the CNN analysis with tower images. Looking at the global recoil for each order, we see that global NLO has better performance, with the CNN trained and validated with global LO having the least discriminatory power. To understand this relative power, we note that the third jet in an NLO simulation has a leading-order accuracy. Whereas, for the LO case, the third jet, if present, is a consequence of the parton shower. The global-recoil scheme enhances the radiation in the central regions for both orders; however, it is partially controlled by the NLO simulation of the first real emission, while there is no such control for the LO case.

Although we have seen that the network trained and tested with different signal simulations shows notable differences, it is worth investigating how a CNN trained on a specific simulation fares when tested on other signal simulations. The validation AUC for all signal simulations evaluated on each of the networks trained on the different signal simulations is shown in table 4.1. For each signal type, the network it was trained on has the maximum discrimination, which is

unsurprising given that the purpose of the training is to encode its behaviour into the network. Moreover, the trend of increasing performance is the same regardless of the signal dataset used in training, pointing towards all networks learning the underlying difference between the signal and the background. Another feature of interest is the relatively higher range of AUCs for the LO datasets than NLO ones, pointing towards their relatively high uncertainties. Interestingly, regardless of the nature of the simulation used during the training, the most accurate simulation among the four, dipole-NLO events, has a very stable validation AUC with only a 1.6% deviation. This stability shows that CNNs can learn the underlying differences between VBF events and non-VBF events even when the VBF simulation is suboptimal.

To gauge the possible improvement in using a dipole scheme over the global scheme used in our previous work, we train the CNN with the combined gluon-fusion signal and the instance of dipole-NLO simulation of the VBF process in the same proportion as described in Section 3.1 and extract the bounds on the branching ratio. We find the median upper limit on the invisible branching ratio for an integrated luminosity $L = 300$ fb^{-1} to be 2.22%.

In all preceding analyses, we have used LO samples without any matching, and the third jet originates exclusively from the parton-shower, which is inaccurate in describing harder emissions. It is worth examining how a matching procedure between the hard matrix element and the parton shower, which improves the description of the third jet in the harder regions, influences the network performance. To inspect the possible improvement of such matching procedures, we generate VBF events matched with an additional jet via the MLM procedure [312,313] for both parton-recoil schemes. We found a continuous differential jet rate and transverse momentum distribution of the different jet samples for an `xqcut` value of 100. As recommended for VBF processes, the `auto_ptj_mjj` flag was set to false. All other aspects of the simulation including the renormalisation and factorisation scale, PDFs, and baseline selection criteria are the same as described in Section 4.1.1. We generated about 25k events after baseline selection for both recoil schemes. Testing with these samples for the networks trained with the leading order unmatched samples with the same parton-shower recoil against the validation background dataset, we find an AUC of 0.8651 and 0.9261 for the global and dipole matched LO samples, respectively. Compared to the full NLO simulation values tested on these networks (Table 4.1), these values lie closer to the LO simulation, indicating that the matching procedure does not help alleviate the issues of the global parton shower. In contrast, the matched dipole value is still relatively stable, although closer to the LO value than the NLO value, signifying the relative importance of the virtual corrections of the NLO simulation.

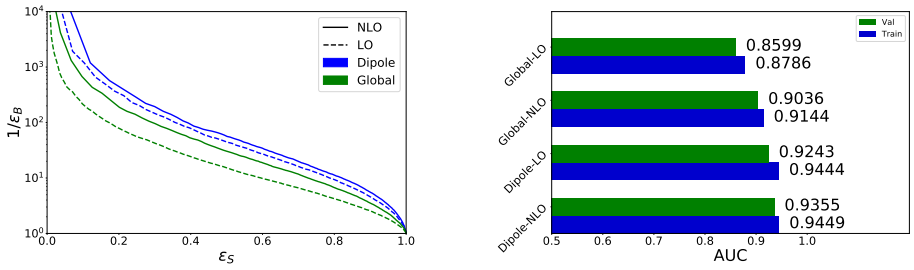


Figure 4.5: Comparison of the performances in terms of Receiver operating characteristic (ROC) curves (left plot) on the validation dataset between the signal acceptance ϵ_S and the background rejection $1/\epsilon_B$, and the corresponding areas under these ROC curves (AUC) (right plot) for the training and validation data are shown for the four different cases of signal simulations.

4.3 Summary

In this chapter, we carried out a quantitative analysis to investigate the dependence of a CNN's performance on the recoil scheme of the parton shower and the perturbative accuracy of the matrix element simulation for a VBF Higgs signal decaying to invisible particles. The difference between the leading order and next-to-leading order, although present, is not very pronounced for the physically correct dipole-recoil scheme. We found that the training is highly dependent on the recoil scheme, with a better performance coming for the physically accurate dipole recoil. With this fortunate coincidence, a complete analysis with all VBF processes showered with a dipole recoil scheme will possibly reduce the upper limits on the invisible branching ratio even further than the projection which used a global recoil scheme. Furthermore, we find that

- the training performance is greatly reduced when we use signal simulated with a global-recoil scheme on parton level events generated with leading-order or next-to-leading accuracy and improves for a dipole recoil, with events generated at next-to-leading-order accuracy showered with a dipole recoil having the highest training accuracy.

- for each set of signal simulations, the highest validation accuracy is achieved for the network that used the same type during the training process with the same trend as the training accuracies. However, the validation performance of the NLO events showered with dipole recoil (which is the most accurate description of the actual events amongst the four signals used) is affected mildly by the kind of data used during the training.

Our findings show that CNNs can learn the underlying differences between VBF type events and the dominant QCD backgrounds, even when trained on sub-optimal simulated data.

Chapter 5

An infra-red and collinear safe message-passing algorithm

Although the findings of the previous chapter suggest that CNNs can learn the underlying differences in the QCD radiation pattern even with suboptimal data, one would like deep-learning algorithms to have a better property within the theoretical biases involved in perturbative QCD calculations. One such theoretical requirement is the infra-red and collinear safety of observables. It ensures the appropriate handling of real and virtual corrections order by order in perturbative calculations via the KLN theorem [52, 53]. Thus, an IRC safe deep-learning algorithm would learn features that are, in principle, calculable in perturbative QCD. In this chapter, we devise such an IRC safe Graph Neural Network and study its performance on the problem of jet-tagging and its resilience to nearly soft and collinear emissions.

A closely connected algorithm to GNNs: the deep-sets framework for feature learning on point clouds, has been explored for jet physics. *Energy Flow Networks* (EFNs) [314] are IRC safe deep learning models for point clouds, where the feature extraction component learns a per-particle-map to a latent space. The process of constructing graphs out of the point cloud imposes additional structures into the data, which can be efficiently extracted with the help of MPNNs. Concretely, an MPNN based feature extraction phase improves the per-particle-map in the following ways:

- It can extract inter particle information courtesy of the trainable message-passing function $\Phi(p_i, p_j)$, acting on each pair of nodes p_i and p_j connected by each of the edges in the graph.

- The node readout updates the node feature as a permutation invariant function of all incoming messages. The readout, along with the message-passing step, forms one *message-passing* operation. It controls the extent of information passed from one layer to another. Therefore, the graph construction algorithm directly controls the nature of the information that goes into learning the parameters of the message function of the first layer.

- Since the updated node features are functions of all the neighbouring node features, the range of information in the node features gradually increases with the repetitive application of the message-passing operation. This is not the case for EFNs, as the function is dependent on single-node features. Thus, applying a subsequent learnable function to the updated node features becomes a functional composition, which does not add additional complexity to the process of feature extraction.

- On top of the graph construction itself, the number of applications of the message-passing operation also controls the amount of local information encoded into the final node features. For EFNs, this is always limited to single particles.

Forgoing the permutation invariance of EFNs, for permutation equivariance [315] has better feature extraction by partially taking care of the last two points, at the additional cost of having to abandon the variable-length inputs. On the contrary, an IRC safe MPNN would improve upon the EFNs and still be permutation invariant. Intrinsically, this is because they are very similar, which is also self-evident within the discussed reasons. Once the graph construction algorithm is taken care of, we find that implementing an IRC safe MPNN can be done via an energy-weighted message (feature) with summed aggregation at the node (graph) level. We find that the network, which we refer to as *Energy-weighted Message-Passing Network* (EMPN), improves upon EFNs with a single message-passing operation. Moreover, EMPNs can, in principle, improve upon EFNs in all the four points discussed above as the iterative application does not spoil the IRC safety.

The rest of the chapter is organised as follows. We present the main results of this work in Section 5.1, where we devise the graph construction algorithm and the MPNN architecture, which guarantees the IRC safety of the network output. We describe in detail the application of EMPNs to three jet-tagging scenarios: gluon vs quark, QCD vs W, and QCD vs top, in Section 5.2. The results of these three scenarios are presented in Section 7.4. We conclude in Section 5.4.

5.1 IRC safe message-passing

In this section, we examine the subtleties of building an IRC safe message-passing neural network. We can divide this into three steps: graph construction, message-passing and node readout, and graph readout. In the following, we analyse the graph construction algorithm in Section 5.1.1, and the message-passing, node readout and graph readout together in Section 5.1.2.

5.1.1 Constructing the neighbourhood of a particle

An infra-red and collinear safe observable has to be equal in the presence or absence of soft or collinear particles. Specifically, given a set $\mathcal{S}$ of n massless particles with their four momenta $p_i = (z_i, \hat{p}_i)$, with $z_i = p_T^i / \sum_{j\in\mathcal{S}} p_T^j$ denoting the relative hardness of the particle, and $\hat{p}_i$ being the directional (angular) coordinates. If a particle q undergoes a splitting $q \to r + s$, with $p_q = p_r + p_s$, an IRC safe observable $\mathcal{O}_n$ must satisfy

$$\begin{aligned} &\mathcal{O}_{n+1}(p_a, ..., p_b, p_r, p_s, p_c, ...) \to \mathcal{O}_n(p_a, ..., p_b, p_q, p_c, ...) \quad \text{as} \quad z_r \to 0 \quad , \\ &\mathcal{O}_{n+1}(p_a, ..., p_b, p_r, p_s, p_c, ...) \to \mathcal{O}_n(p_a, ..., p_b, p_q, p_c, ...) \quad \text{as} \quad \Delta_{rs} \to 0 \quad , \end{aligned} \tag{5.1}$$

where z_r is the relative hardness of p_r, and Δ_{rs} is the angle between $\vec{p}_r$ and $\vec{p}_s$. Consequently, the algorithm for constructing graphs should allow for the addition of soft or collinear particles without changing the whole structure of the graph. The graph constructed by a vertex deletion of a soft or collinear particle should be equal to the one formed in its absence, with proper substitution of the four-momenta in the case of collinear particles. For instance, a k-nearest neighbour (k-NN) graph would not allow for an IRC safe message-passing since adding a particle in the vicinity of a node i could change the neighbourhood set $\mathcal{N}(i)$ to $\mathcal{N}'(i)$ with a fixed cardinality. The fixed cardinality would induce a *domino effect* in the neighbourhood sets of the subsequent neighbours and change the graph's structure to a large degree. As a concrete example, for a k-NN graph in the (η, ϕ) plane, the addition of a particle closer to the node could, in principle, omit the hardest particle out of the neighbourhood in $\mathcal{N}'(i)$. This is diagrammatically shown in Figure 5.1, where a particle q splitting to two particles r and s excludes another particle b from the neighbourhod of particle i. Thus, in the node readout for particle i, a message-passing algorithm based on a k-NN graph cannot smoothly extrapolate between the two scenarios, when taking the IRC limits of the daughter particles r and s. This warrants a careful examination of the graph construction algorithm.

Since our final aim is to have a message-passing neural network whose output is IRC safe, the correctness of the graph construction algorithm is intimately connected with the subsequent operations the network will perform on the graph's nodes. From the perspective of QCD, the node readout and the graph readout functions are on the same footing, with the only difference being the scale. We look into the jet substructure with the help of the nodes and the edge connections, which gives us a representation of the whole jet.* In graph theory, self-loops are often ignored, and most of the efforts concentrate on analysing simple graphs. However, from the perspective of QCD, the destination node itself can also emit

*This could be extrapolated to the event shape, where an IRC safe graph neural network would look into the subsequent scales present in the event and construct an event level representation which would have the desirable property of being less affected by soft and collinear radiations.

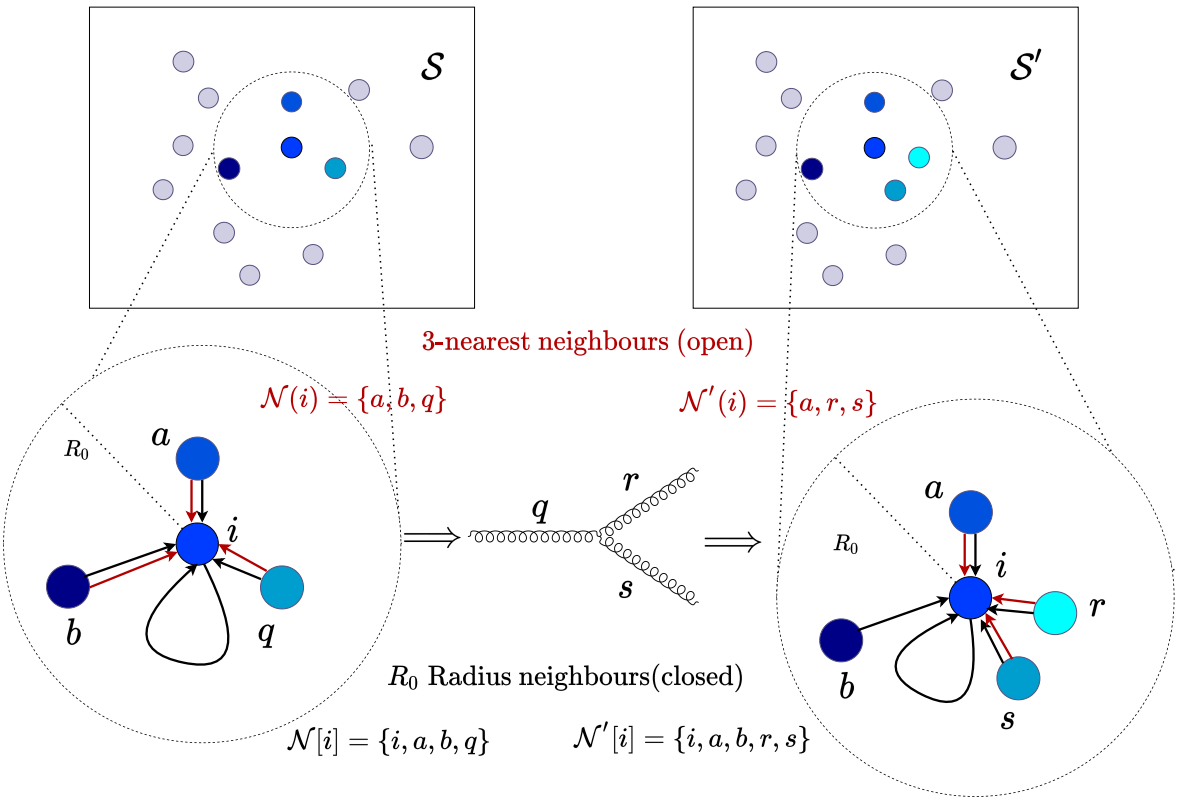


Figure 5.1: A k-nearest neighbour graph in the (η, ϕ)-plane will have a different structure when any particle q splits to r and s. The set $\mathcal{S}$ denote the particles in the jet when there is no splitting, while $\mathcal{S}'$ denotes the particles with q splitting. We show the directed edge connection to i from its three nearest neighbours with red on either side. The neighbourhood set $\mathcal{N}(i)$ has b in it, however when q splits, $\mathcal{N}'(i)$ does not contain b. Therefore, the graph's structure prevents a smooth extrapolation between the two scenarios in the infra-red and collinear limit. This is not the case for a radius graph with radius R_0 in the (η, ϕ) plane, which is shown with black connections. We also include the self-loop of i, by using the closed neighbourhood sets $\mathcal{N}[i]$ and $\mathcal{N}'[i]$, since the node i could also split into two particles.

soft or collinear particles. Therefore, an IRC safe aggregation must act on the closed neighbourhood $\mathcal{N}[i]$, which includes the destination node i.

Let us take a set $\mathcal{S}$ of the four momenta of n massless particles. Out of these, any particle q could undergo a splitting to r and s, which enlarges the set $\mathcal{S}$ to $\mathcal{S}'$ with $\mathcal{S}' = \mathcal{S} \setminus \{q\} \cup \{r, s\}$. The three four momenta can be written in general as

$$p_q = (z_q, \hat{p}_q) \quad , \quad p_r = (z_r, \hat{p}_r) \;\; z_r = \lambda z_q \quad , \quad p_s = (z_s, \hat{p}_s), \;\; z_s = (1 - \lambda)\, z_s \quad , \tag{5.2}$$

with $\lambda \in [0, 1]$, and $p_q = p_r + p_s$. Following are the limits that are of interest:

- **IR limit:** $\lambda \to 0 (\lambda \to 1)$, for r(or s) in the soft limit,

- **C limit:** $\hat{p}_r \to \hat{p}_s \to \hat{p}_q$ or equivalently $\Delta_{rs} \to 0$, for any λ.

For the IR limit, the two cases are for either of the daughter particles becoming soft, and it suffices to take one of them, say $\lambda \to 0 \implies z_r \to 0$ in the following presentation. A graph construction method on $\mathcal{S}$ would allocate to each particle i a neighbourhood set $\mathcal{N}[i] \subseteq \mathcal{S}$. We would have to apply the same method to $\mathcal{S}'$, which would give neighbourhood sets $\mathcal{N}'[i] \subseteq \mathcal{S}'$. To devise an IRC safe message passing operation, a simple procedure is to assume that the neighbourhood sets, $\mathcal{N}[i]$, behave the same way as the total set $\mathcal{S}$. By keeping the behaviour of the sets the same, the graph structure essentially works as a control over the scale of the message-passing operation. In the IR limit, the emitter q and the daughter r need not fall in the same neighbourhood since the other daughter s will have the same four momenta of q in the limit $z_r \to 0$. However, in the C limit, if the emitter q is in the neighbourhood of $\mathcal{N}[i]$, we need both the daughters to be in $\mathcal{N}'[i]$. Mathematically, we can write this condition as:

- **IR limit:** If $r \notin \mathcal{N}'[i] \implies \mathcal{N}'[i] = \mathcal{N}[i]$

$$\text{else } \mathcal{N}'[i] \setminus \{r\} = \mathcal{N}[i] \quad \text{, when } \; z_r = 0 \; ; \tag{5.3}$$

- **C limit:** If $\{r, s\} \cap \mathcal{N}'[i] = \emptyset \implies \mathcal{N}'[i] = \mathcal{N}[i]$

$$\text{else } \mathcal{N}[i] = \mathcal{N}'[i] \setminus \{r, s\} \cup \{q\} \quad \text{, when } \; \Delta_{rs} = 0 \; . \tag{5.4}$$

The aim now is to devise a graph construction algorithm that will give us neighbourhood sets satisfying these conditions. Constructing graphs from sets sampled from a point cloud uses functions defined on the features $\vec{\beta}_i$. The algorithm can be surmised by comparing two functions, which, in general, depend on features $\vec{\beta}_i$ (which need not be the same as the node features $\mathbf{h}_i$) of elements i belonging to subsets of the whole sample set $\mathcal{S}$, which itself can change as the edge set $\mathcal{E}$ grows. Calling these two functions as the *decision* function $\mathbf{D}$, and the *threshold* function $\mathbf{T}$, we say that a particle j will be placed into the neighbourhood of i, if $\mathbf{D}$ is less than or equal to $\mathbf{T}$,

$$\mathbf{D}(\vec{\beta}_i, \vec{\beta}_j | \vec{\beta}_k, \vec{\beta}_l, ...) \leq \mathbf{T}(\vec{\beta}_i, \vec{\beta}_j | \vec{\beta}_k, \vec{\beta}_l, ...) \implies j \in \mathcal{N}[i] \quad . \tag{5.5}$$

The features $\vec{\beta}_i$ can generally contain any quantity of i like charge, four-momenta, or the identity of the sub-detector component of i. Graphs are versatile data structures that can encode the detector components together into a compact, unified representation. However, as our current aim is to incorporate IRC safety, it restricts us to calorimeter or particle flow constituents with no charge information and the four vectors of the particles. In the following, we systematically reduce the possible four-vectors which could come into the arguments of the decision and the threshold functions.

As was previously discussed, $\mathbf{D}$ or $\mathbf{T}$ cannot be dependent on the cardinality of the neighbourhood set $\mathcal{N}[i]$. Consider the functions depending on another particle p_q to decide whether p_j should be in $\mathcal{N}[i]$. A splitting on p_q can create situations where p_j can be in $\mathcal{N}[i]$ and not in $\mathcal{N}'[i]$ or vice versa. Thus, the functions can not depend on any other four vectors than the two particles in question. Looking at eq. 5.4, we see that the emitter and the daughter particles of a collinear splitting need to be in both in the neighbourhood $\mathcal{N}[i]$ and $\mathcal{N}'[i]$, respectively, or not at all. We have the following condition on the decision and threshold functions,

$$\begin{aligned}\mathbf{D}(p_i, p_r + p_s) \leq \mathbf{T}(p_i, p_r + p_s) \Leftrightarrow \mathbf{D}(p_i, p_r) \leq \mathbf{T}(p_i, p_r) \text{ and } \mathbf{D}(p_i, p_s) \leq \mathbf{T}(p_i, p_s) \quad , \\ \mathbf{D}(p_r + p_s, p_i) \leq \mathbf{T}(p_r + p_s, p_i) \Leftrightarrow \mathbf{D}(p_r, p_i) \leq \mathbf{T}(p_r, p_i) \text{ and } \mathbf{D}(p_s, p_i) \leq \mathbf{T}(p_s, p_i) \quad ,\end{aligned} \tag{5.6}$$

in the exact collinear limit of $\Delta_{rs} = 0$. The second line arises when considering the emitter or daughters as the destination node, with p_i denoting any particle in their respective sets. A simple way to satisfy these inequalities is by using the condition of collinearity and making the functions dependent only on the directional coordinates,

$$\mathbf{D} = \mathbf{D}(\hat{p}_i, \hat{p}_j) \quad , \quad \mathbf{T} = \mathbf{T}(\hat{p}_i, \hat{p}_j) \quad . \tag{5.7}$$

The functions can also have additional dependence on any IRC safe quantity defined on the set $\mathcal{S}$.

For our network analysis, we explore the simplest possible graphs to gauge the power of this method by constructing graphs with constant radius R_0,

$$\mathbf{D} = \Delta R_{ij} \quad , \quad \mathbf{T} = R_0 \quad , \tag{5.8}$$

in the (η, ϕ)-plane. Complicated dependencies on the directional variables and on IRC safe quantities like the jet's p_T can be explored in future work. The black connections to particle i in Figure 5.1 show a case where a split in particle q preserves the other particles in the neighbourhood sets, except for the emitter and the daughters.

5.1.2 Energy-weighted Message-Passing

Now that we have the graph construction algorithm, we look into building an IRC safe message-passing function. The message-function at the first layer $\Phi^{(0)}$ would take two four-vectors p_i and p_j for a *directed edge* from j to i, to give the message ${}^i\mathbf{m}_j^{(0)}$. The node features are then updated to $\mathbf{h}_i^{(1)}$, by applying a permutation invariant function on the messages ${}^i\mathbf{m}_j^{(0)}$ for all possible $j \in \mathcal{N}[i]$. Commonly used permutation invariant functions can be classified in the sense of QCD into *exclusive* or *inclusive* functions: the function output depends on a specific subset

of the neighbourhood, or it depends equally on all the neighbourhood particles. Max/min falls within the first class, while mean/sum falls under the second class. As one can presume, it is inherently problematic to build IRC safety into exclusive functions. Building IRC safety into a mean readout is not straightforward since it depends explicitly on the number of particles in $\mathcal{N}[i]$. In the following, we examine the conditions which give IRC safety of the updated node features on the message-passing function for the exclusive and summed node readout operations.

Max/Min readout: Since, the only difference between max and min readout is the comparison, we look at max readout. The same for min readout follows by replacing the greater-than with the less-than symbol in the message comparisons. We have the messages $\Phi^{(0)}(p_i, p_j) = {}^i\mathbf{m}_j^{(0)}$ with the max update as,

$$\mathbf{h}_i^{(1)} = \max_{j \in \mathcal{N}[i]} \Phi^{(0)}(p_i, p_j) \quad .$$

For a splitting q to r and s with $p_q = p_r + p_s$ and assuming that the neighbourhood sets follow eq. 5.3 and 5.4. In the soft limit when $z_r \to 0$, we have

$$z_j > z_r \implies \Phi^{(0)}(p_i, p_j) > \Phi^{(0)}(p_i, p_r) \quad \text{for IR safety.}$$

For the collinear limit $\Delta_{rs} \to 0$, we have

$$\Phi^{(0)}(p_i, p_j) \geq \Phi^{(0)}(p_i, p_r) \quad \text{and} \quad \Phi^{(0)}(p_i, p_j) \geq \Phi^{(0)}(p_i, p_s) \; \forall \; j \; \in \mathcal{N}[i] \quad \text{for C safety.}$$

Implementing C safety in a max/min node readout is *not possible* since the angle Δ_{rs} needs to control the ordering of the messages ${}^i\mathbf{m}_r^{(0)}$ and ${}^i\mathbf{m}_s^{(0)}$ with all other messages ${}^i\mathbf{m}_j^{(0)}$. The max function chooses the maximum value out of all ${}^i\mathbf{m}_j^{(0)}$, with the ordering essentially determined by the second argument in $\Phi^{(0)}$. Consider an exactly collinear splitting of the particle contributing to the highest message vector in $\mathcal{N}[i]$, say $p_M \to \lambda\, p_M + (1-\lambda)\, p_M$, with $\lambda \in [0, 1]$. The max value in both the scenarios can be equal only at the endpoints $\lambda \in \{0, 1\}$, which is essentially the soft *and* collinear limit. The same is true for min readout when considering the particle determining the minimum value of $\Phi^{(0)}$ in the neighbourhood.

Sum readout: The updated node features are given by,

$$\mathbf{h}_i^{(1)} = \sum_{j \in \mathcal{N}[i]} \Phi^{(0)}(p_i, p_j) \quad . \tag{5.9}$$

For a splitting $q \in \mathcal{N}[i]$ to $r, s \in \mathcal{N}'[i]$ changing the neighbourhood set $\mathcal{N}[i]$ to $\mathcal{N}'[i]$. The requirements on the message function $\Phi^{(0)}$ are

$$\textbf{IR safety:} \quad \Phi^{(0)}(p_i, p_r) \to 0 \quad \text{as} \quad z_r \to 0 \tag{5.9a}$$

$$\textbf{C safety:} \quad \Phi^{(0)}(p_i, p_r + p_s) = \Phi^{(0)}(p_i, p_r) + \Phi^{(0)}(p_i, p_s) \quad \text{as} \quad \Delta_{rs} \to 0 \quad . \tag{5.9b}$$

Satisfying these conditions gives IRC safe updated node features

$$\mathbf{h}_i^{(1)} = \mathbf{h}_i'^{(1)} = \sum_{j \in \mathcal{N}'[i]} \Phi^{(0)}(p_i, p_j) \quad . \tag{5.10}$$

We have written these conditions for the second argument only, even though the splitting can occur in the destination node, since it has a special status in the message passing operation. Applying similar conditions for the first argument is highly restrictive with no practical gain. Nodes corresponding to the daughters are present in the graph even in the IRC limit, which is precisely the objective of the present study – to get fixed-length representations of two graphs, with one having an additional node, which is the same when the additional node or particle is soft or collinear. Including the destination node in the neighbourhood set makes it possible for the emitter and the two daughters to have the same updated node features in the exact collinear limit, with the two collinear copies propagating forward simultaneously. In either of the limits (soft or collinear), these copies are then taken care of separately by an IRC safe graph readout. These are explained in more detail in the following paragraphs.

We now present an implementation of message-passing operation which satisfies the IRC safe conditions for a summed node readout. The message function $\Phi^{(0)}$ has a dependence on two four-vectors, which allows an MPNN to extract richer features than the ones employed in EFNs [314] with a single particle map. However, the per-particle map can be functionally regarded as a special message function constant for the second argument. The point cloud could then be regarded as a graph of N nodes, with N disconnected components with only self-loops entering the edge set. Therefore, generalising the per-particle map, we define the message function as

$${}^{i}\mathbf{m}_j^{(0)} = \Phi^{(0)}(p_i, p_j) = \omega_j^{(\mathcal{N}[i])} \;\; \hat{\Phi}^{(0)}(\hat{p}_i, \hat{p}_j) \quad , \tag{5.11}$$

where $\hat{\Phi}$ takes only the directional information of the four vectors and we define scalar weights $\omega_j^{(\mathcal{K})}$, dependent on the scope $\mathcal{K}$ of the readout operation,

$$\omega_j^{(\mathcal{K})} = \frac{p_T^j}{\sum_{k \in \mathcal{K}} p_T^k} \quad . \tag{5.12}$$

Clearly, for the full set $\mathcal{S}$, $\omega_j^{(\mathcal{S})} = z_j$, and $z_j \to 0 \implies \omega_j^{(\mathcal{K})} \to 0$ regardless of $\mathcal{K}$, thereby satisfying[†] eq. 5.9a. Moreover, as long as the neighbourhood sets $\mathcal{N}[i]$ and $\mathcal{N}'[i]$ satisfy eq. 5.4 which is true even when i undergoes a splitting we have,

$$\omega_q^{(\mathcal{N}[i])} = \omega_r^{(\mathcal{N}'[i])} + \omega_s^{(\mathcal{N}'[i])} \quad ,$$

[†]Note, when a soft particles has no other neighbour except itself, the node readout might change to a finite value. However, the graph readout, and therefore the network output, will remain unchanged, as $\mathcal{K} = \mathcal{S}$.

where q is the emitter and r and s are the daughter particles. Since $\hat{p}_q = \hat{p}_r = \hat{p}_s$ in the collinear limit, we have

$$\begin{aligned} \hat{\Phi}^{(0)}(\hat{p}_i, \hat{p}_q) &= \hat{\Phi}^{(0)}(\hat{p}_i, \hat{p}_r) = \hat{\Phi}^{(0)}(\hat{p}_i, \hat{p}_s) \quad , \\ \hat{\Phi}^{(0)}(\hat{p}_q, \hat{p}_i) &= \hat{\Phi}^{(0)}(\hat{p}_r, \hat{p}_i) = \hat{\Phi}^{(0)}(\hat{p}_s, \hat{p}_i,) \quad . \end{aligned} \tag{5.13}$$

Hence, the updated node features

$$\mathbf{h}_i^{(1)} = \sum_{j \in \mathcal{N}[i]} \omega_j^{(\mathcal{N}[i])} \; \hat{\Phi}^{(0)}(\hat{p}_i, \hat{p}_j) \quad , \tag{5.14}$$

satisfies the IRC safety condition eq. 5.10. Note that the expression does not limit the form of the function $\hat{\Phi}^{(0)}$ other than differentiability which is required for back propagation. Thus, we can modify any existing message-passing algorithm into the IRC safe version by implementing the appropriate message weights and restricting the input to the directional coordinates. We therefore implement the IRC safe version of edge-convolutions as a proof-of-principle analysis.

Looking at the structure of the updated node features after the first message-passing operation, we can see that it is a function of all the four-momenta of its neighbourhood particles. If n is the number of nodes in the set $\mathcal{N}[i]$, we have the updated IRC safe node feature as $\mathbf{h}_i^{(1)}(p_1, p_2, ...p_n)$. We want to investigate the IRC safety of another message passing on the updated quantities $\mathbf{h}_i^{(1)}$. If true, the architecture could accommodate multiple iterations of message-passing operations, thereby increasing the model's expressive power. For simplicity, one can consider static graphs with the same neighbourhood sets. A weighted message-passing with of the same form as eq. 5.14

$$\mathbf{h}_i^{(2)} = \sum_{j \in \mathcal{N}[i]} \omega_j^{(\mathcal{N}[i])} \; \hat{\Phi}^{(1)}(\mathbf{h}_i^{(1)}, \mathbf{h}_j^{(1)}) \quad ,$$

with the same weights $\omega_j^{(\mathcal{N}[i])}$, but with the updated node features $\mathbf{h}_i^{(1)}$ satisfies IR safety. For it to be C safe, the features $\mathbf{h}_i^{(1)}$ should behave just like the directional coordinates $\hat{p}_i$. Note that the neighbourhood sets for the two collinear daughters are the same. The emitter also has the same neighbourhood after replacing the daughters with their summed four-vector (cf. eq. 5.4). Their aggregated node vectors become equal to that of the emitter in $\mathcal{S}$ via the cancellation of the λ factors in the weights. Thus, the updated node vectors after the first message-passing of the daughters and the emitter are exactly equal in the collinear limit $\mathbf{h}_q^{(1)} = \mathbf{h}_r^{(1)} = \mathbf{h}_s^{(1)}$. Hence, they have essentially the same characteristics as the directional coordinates. This ensures that $\hat{\Phi}^{(1)}(\mathbf{h}_i^{(1)}, \mathbf{h}_j^{(1)})$ follow analogous equations to eq. 5.13, thereby making the weighted message $\omega_j^{(\mathcal{N}[i])} \; \hat{\Phi}^{(1)}(\mathbf{h}_i^{(1)}, \mathbf{h}_j^{(1)})$ follow similar equations to eq. 5.9. Moreover, the new features $\mathbf{h}_i^{(2)}$, would have this same property. Hence, *repeating the energy weighted message passing operation any number of times satisfies IRC safety at the level of each updated node*

feature. Denoting the node features for the l^{th} iteration as $\mathbf{h}_i^{(l)}$ with $\mathbf{h}_i^{(0)} = \hat{p}_i$, we have the iterative application of the energy-weighted message passing as

$$\mathbf{h}_i^{(l+1)} = \sum_{i \in \mathcal{N}[i]} \omega_j^{(\mathcal{N}[i])} \ \hat{\Phi}^{(l)}(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}) \quad . \tag{5.15}$$

As seen above, there will be copies of emitted particles even in the IRC limit, propagating forward in the graph formed after a soft or collinear splitting. Thus, any generically defined graph readout operation acting on the node features of the full graph will not be IRC safe. The graph readout should guarantee the equality of the obtained representation of the two graphs in the IRC limit, with one having an additional node. The node features at the final message-passing layer, say $\mathbf{h}_i^{(L)}$, will behave the same way as the directional variables, regardless of L, the number of message-passing iterations. Thus, a graph readout of the form

$$\mathbf{g} = \sum_{i \in G} \omega_i^{(\mathcal{S})} \ \mathbf{h}_i^{(L)} \quad , \tag{5.16}$$

with $z_i = \omega_i^{(\mathcal{S})}$, is IRC safe. This is an analogue of the sum over the per-particle representation employed in EFNs. The graph convolution operation now replaces the per-particle maps. The scale of the representation which undergoes the sum, which contains local structural information, is determined by the number of message-passing operations and the graph construction algorithm. A schematic representation of such a network for $L = 1$ is shown in Figure 5.2.

5.2 Details of network implementation

In this section, we present the numerical results of the IRC safe message passing neural network. The details of the datasets are given first, followed by the network hyperparameters and training aspects.

5.2.1 Analysis setup

For assessing the power of *Energy-weighted Message Passing Networks* (EMPN), we consider three scenarios: quark/gluon discrimination as a benchmark for IRC safe, supervised identification of normal radius quark jets from gluon jets, boosted W vs QCD jet tagging as an example of two-prong tagging, and boosted top vs QCD jet tagging as an example of three-prong tagging. We use publicly available datasets for the quark vs gluon tagging [314, 316], and the top tagging scenarios [178, 317]. These datasets were generated at 14 TeV center-of-mass energy proton-proton collisions in `Pythia8` [70]. The parton level events were generated in `Pythia 8.226` using the processes `WeakBosonAndParton:qqbar2gmZg` and `WeakBosonAndParton:qg2gmZq` for the gluon and quark samples respectively. These events were showered with the default tunings of the shower parameters

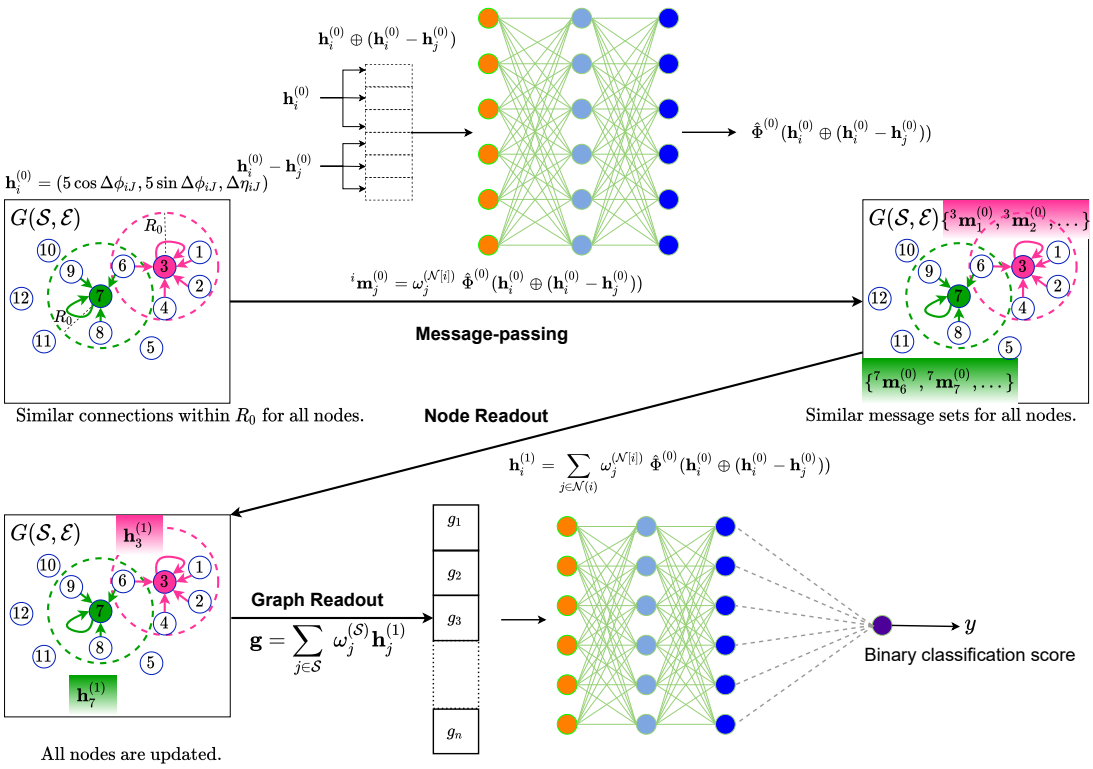


Figure 5.2: The specific architecture used for the three jet tagging scenarios of an *Energy-weighted Message-Passing network*(EMPN), with a single energy-weighted message passing operation. It takes graphs of constant radius R_0 in the (η, ϕ)-plane. The message-passing network $\Phi^{(0)}$, takes the directional inputs of the four-vectors in the form of $\mathbf{h}_i^{(0)}$, and calculates a weighted message ${}^i\mathbf{m}_j^{(0)}$ with $\omega_j^{(\mathcal{N}[i])}$ as the weights. It then undergoes a summed node readout operation to update their features to $\mathbf{h}_i^{(1)}$. The graph representation $\mathbf{g}$ obtained after a summed graph readout operation on the node features $\mathbf{h}_j^{(1)}$ weighted with $\omega_j^{(\mathcal{S})}$, is fed into a DNN which outputs a binary classification score.

with multi-parton interactions (MPI) and hadronisation. All final state particles except neutrinos were clustered with `FastJet 3.3.0` [94] into anti-k_T [91] jets of radius $R = 0.4$, with no detector simulation. Jets are required to have $p_T \in [500, 550]$ GeV and rapidity $|y| < 2$. Parton level events for QCD jets and top jets in the top tagging dataset were generated with `Pythia 8.2.15`. These were showered without MPI effects and passed through `Delphes3` [72], with the default ATLAS detector card. The particle-flow objects are clustered into anti-k_T jets with $R = 0.8$. The jets are required to have $p_T \in [550, 650]$ GeV, with pseudorapidity $|\eta| < 2$. For the top-jets, the parton-level top quark and its decay products were required to fall within $\Delta R = 0.8$ of the reconstructed jet axis. QCD jets from this dataset was used for the W-tagging scenario. For the W jets, we generated the parton level process `p p > w`$^{\pm}$ `z` in `MadGraph5_aMC@NLO`(v2.6.5) [61], at 14 TeV proton-proton collisions, forcing the W boson to decay hadronically, and

Sl.No	Jet Class	Parton-level	MPI	Detector Sim.	Jet-radius
1.	Gluon	`Pythia8`	Yes	No	0.4
2.	Quark	`Pythia8`	Yes	No	0.4
3.	QCD	`Pythia8`	No	Yes	0.8
4.	Top	`Pythia8`	No	Yes	0.8
5.	W	`MadGraph5_aMC@NLO`	No	Yes	0.8

Table 5.1: A summary of the different classes of data used in the three classification scenarios. The W data was generated for this study, while for the first four classes, we use publicly availaible datasets [316, 317]. All datasets were showered and hadronised with `Pythia8`, while the detector simulation was done with `Delphes3`, with the default ATLAS card.

the Z boson to decay to neutrinos. Parton level cuts on the missing-transverse energy with $\not{E}_T > 500$ GeV, and the pseudorapidity of the W bosons, $|\eta_w| < 3$, were applied during the generation. Further downstream simulation of these partonic events was done by implementing the same configuration details of the top-dataset, including the jet-reconstruction and baseline selection criteria. We also matched the parton level W and its decay products to be within $\Delta R = 0.8$ of the reconstructed W jet axis. Up to two-hundred hardest constituents within the jet were used to construct the graphs for the three large-radius jet tagging datasets. For all three scenarios, we have 1.2 million training, 400k validation, and 400k test jets.

5.2.2 Constructing the jet graphs

The jet graphs of each jet are constructed by taking their constituents. We calculate the interparticle distance $\Delta R_{ij} = \sqrt{\Delta\eta^2 + \Delta\phi^2}$, in the (η, ϕ) plane. For each node i, we define the neighbourhood set $\mathcal{N}[i]$ as the set of all the particles i with $\Delta R_{ij} \leq R_0$. After the neighbourhood sets, or equivalently the edge set $\mathcal{E}$ of the graph are obtained, we shift the coordinates of each constituents (η_i, ϕ_i) to $(\Delta\eta_{iJ}, \Delta\phi_{iJ})$, their distance between the jet axis (η_J, ϕ_J). The node features that the network takes has the ϕ coordinates mapped to two-dimensional coordinates $(a\cos\phi, a\sin\phi)$. Keeping in mind the total allowed range of $\eta \in [-5, 5]$, we choose $a = 5$. Thus, for each jet constituent, we have the node features of the input graph as

$$\mathbf{h}_i^{(0)} = (5\cos\Delta\phi_{iJ}, 5\sin\Delta\phi_{iJ}, \Delta\eta_{iJ}) \quad .$$

This choice of representation makes the edge-convolution (which we will be using) look at the ϕ information through an *embedding* in a two-dimensional Euclidean space. This is essential since multilayer perceptrons (MLPs), the building blocks of neural networks, are essentially sequential *affine* maps interspersed by non-linear activation functions, and the periodicity of ϕ may not be evident to it directly even if the graph has the periodicity. The range of ϕ for jets consid-

ered here are not wide enough for the periodicity to become a major bottleneck. However, it is crucial when considering the inclusive event information. We also calculate the weights $\omega_j^{(\mathcal{K})}$ defined in eq. 5.12, for all neighbourhood sets $\mathcal{N}[i]$ and the full set $\mathcal{S}$. The jets are not preprocessed with steps like rotation and reflection before extracting the node features. Doing so should improve the network performance as these symmetries are not built into the architecture. Incorporating these symmetries into the architecture could also improve the performance in the absence of preprocessing.

5.2.3 Network hyperparameters and training

As this is a proof-of-principle study, we examine the simplest of architectures to showcase the ability of EMPNs at the different classification scenarios. We implement an `EnergyWeighted` message-passsing module in `PyTorch-Geometric-1.7.2` [234], for the analysis of the EMPN network. The message-passing function corresponds to an energy-weighted edge convolution [203],

$$ {}^{i}\mathbf{m}_j^{(0)} = \omega_j^{(\mathcal{N}[i])}\ \hat{\Phi}^{(0)}\left(\mathbf{h}_i^{(0)} \oplus (\mathbf{h}_j^{(0)} - \mathbf{h}_i^{(0)})\right) \quad . \tag{5.17} $$

The learnable function $\hat{\Phi}^{(0)}$ is an MLP having two hidden layers. The input layer takes the six-dimensional concatenated vector $\mathbf{h}_i^{(0)} \oplus (\mathbf{h}_j^{(0)} - \mathbf{h}_i^{(0)})$, and maps it to a 128-dimensional representation. Both hidden layers are also fixed to have 128 nodes each with `ReLU` activations, while the output layer has `Linear` activation. The graph representation obtained after applying the IRC safe-readout (cf. eq. 5.16) is fed into a downstream MLP, which outputs the binary classification score. This MLP has three hidden layers, with all of them having sixty-four nodes and `ReLU` activations. The structure of the EMPN network is summarised in Figure 5.2. We use Adam [171] optimiser with an initial learning rate of 0.001, which reduces with a decay-on-plateau condition by a factor of 0.5, with the patience of two epochs without any cooldown. We scan over a set of R_0 values for each classification scenario. For the W and top tagging with large-radius jets ($R = 0.8$), we choose $R_0 \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$, while for the quark-gluon classification with normal-radius jets ($R = 0.4$), we choose $R_0 \in \{0.1, 0.2, 0.3, 0.4\}$. For all three scenarios and each R_0, we train the same network from random initialisation five times. All networks were trained for seventy epochs. The epoch with minimum validation loss is used for evaluating the model with their respective test datasets for each instance of the training. Note that we do not perform any hyperparameter optimisation, and doing so should further improve the performance.

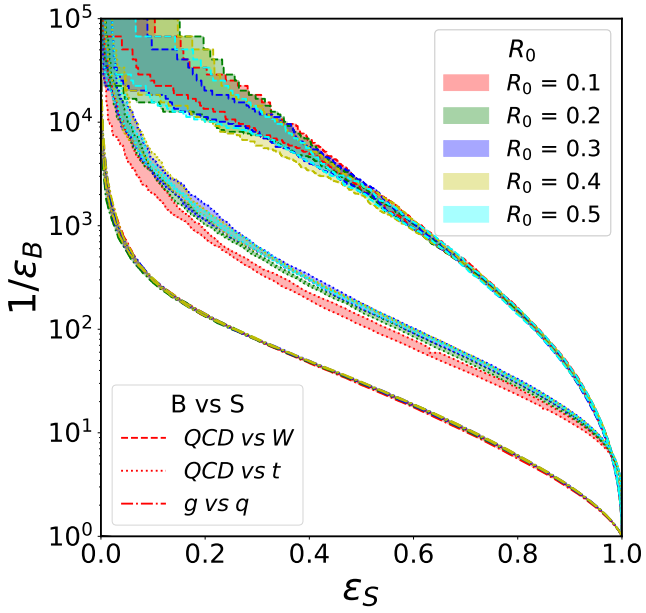


Figure 5.3: ROC curve for the three tagging scenarioes and different values of R_0. The three sets of curves correspond to QCD vs W, QCD vs top, and gluons vs quarks from respectively from top to bottom. The band shows the maximum and the minimum values of the inverse of background acceptance $1/\epsilon_B$, for fixed values of signal efficiency ϵ_S from seperate runs.

5.3 Results

5.3.1 Tagging performance

The ROC curve for the three jet-tagging scenarios for the various values of R_0 are shown in Figure 7.6. We evaluate the background acceptance ϵ_B, at the same set of signal efficiencies ϵ_S for all instances of the trained networks. For a specific tagging case and fixed R_0, the boundary indicates the maximum and the minimum values of $1/\epsilon_B$ from the five training instances. The variation of the mean AUC and their error for the five training instances for each R_0 and the three cases are shown in Figure 5.4. These values, along with the background rejection $1/\epsilon_B$ at 50% signal efficiency, are shown in Tables 5.2, 5.3 and 5.4 for gluon/quark, top and W tagging respectively. For comparison, we also include relevant numbers for gluon vs quark and top tagging scenarios from reference [314] for Energy Flow Networks (EFNs). Since we have not preprocessed our data, the values for top discrimination is for the unprocessed case. The quark-gluon tagging networks already show improvement at $R_0 = 0.1$ with an AUC of 0.8888 over EFNs with 0.8824. However, for the top tagging case, the AUC (0.9734) for $R_0 = 0.1$ is less then that of EFNs (0.9760). This decrease indicates that the local structural information at that scale does not help distinguish QCD jets from top jets with a single message-passing operation. The local information learned by the

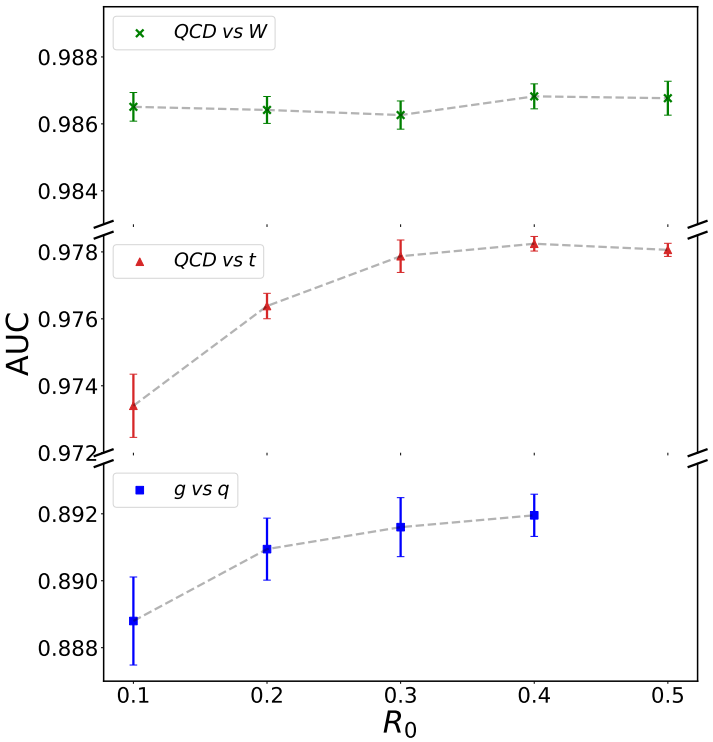


Figure 5.4: Variation of mean AUC with R_0 for the three tagging scenarios. For the W tagging scenario (green cross), the AUC has saturated at $R_0 = 0.1$ and does not increase when compared to the other two. AUCs for the top vs QCD (red triangle), and the gluon vs quark (blue square) classification increases as we increase R_0. The error bands are the standard deviation from five training instances.

message-passing phase confuses the downstream MLP, decreasing its performance compared to EFNs. Although, the message function or the downstream MLP we used is not exactly the same as the analogous per-particle map and the downstream MLP used in reference [314], and hence the comparison is not exactly like-for-like. The difference reaches parity at $R_0 = 0.2$, which further increases and reaches a stable value for higher R_0. Thus, for both scenarios, the energy-weighted message-passing help in better feature extraction of the local features. For the W tagging results, we see very stable values of AUC (see Figure 5.4), which do not vary appreciably with R_0 compared to the other two cases. The EMPN can already extract very rich features for the graphs at $R_0 = 0.1$, giving an AUC of 0.9865. Increasing the complexity of the graph by enlarging R_0 does not add new information which the current architecture can extract. The stability of the AUC shown in Figure 5.4 is likely due to the high kinematic range of the jets compared to the W mass, giving the separation between the two decay products as $\Delta R \sim 2m_W/p_T \sim 0.25$. To check whether the performance decreases for smaller R_0, we repeat the training process for $R_0 = 0.02$ and find that the mean AUC indeed falls mildly to 0.9845 for five training instances.

Sl.No	R_0	AUC	$1/\epsilon_B$ at $\epsilon_S = 50\%$
$L = 1$			
1.	(EFN [314])	0.8824±0.0005	28.6±0.3
2.	0.1	0.8888±0.0013	30.1±0.3
3.	0.2	0.8909±0.0009	30.1±0.2
4.	0.3	0.8916±0.0008	30.7±0.2
5.	0.4	0.8919±0.0006	31.0±0.1
$L = 2$ (Discussed Later)			
1.	0.1	0.8932±0.0006	30.8±0.2

Table 5.2: AUC values and the background rejection for different values of R_0 for *gluons vs quark* tagging dataset. Uncertainties for AUC are the standard deviation from five training instances, while for the background rejection $1/\epsilon_B$ are half of the inter-quartile range. The first entry is quoted from the cited reference.

Sl.No	R_0	AUC	$1/\epsilon_B$ at $\epsilon_S = 50\%$
1.	(EFN [314])	0.9760±0.0001	143± 2
2.	0.1	0.9734±0.0009	115±2
3.	0.2	0.9764±0.0004	151±2
4.	0.3	0.9779±0.0005	167±4
5.	0.4	0.9782±0.0002	174±2
6.	0.5	0.9781±0.0002	168±3

Table 5.3: AUC values and the background rejection for different values of R_0 for *top tagging* dataset. Uncertainties for AUC are the standard deviation from five training instances, while for the background rejection $1/\epsilon_B$ are half of the inter-quartile range. The first entry is quoted from the cited reference.

Other than the apparent variation of the mean AUCs and the ROC curves, we also see interesting features in the error bars of the AUC and the band of the ROC curves. If the AUC increases, its errors also gradually decrease as one increases R_0. On the other hand, across the different scenarios, the errors do not follow the same relation. The variation of AUC for each R_0 is due to the random initialisation of weights from the same underlying weight space with the same distribution[‡] for all networks. The optimisation proceeds via a gradient descent algorithm that goes to a local minimum of the loss function accessible from the initialised point in the space of weights for each instance. We can infer the relative quality of the local minima accessible from the initialised point. Lower the error, the easier it is to get to approximately similar values of the stable loss function. Comparing the three scenarios with stable AUC for the same R_0, we see that the top tagging case has a minor variance, followed by W tagging. Thus, even though the performance of W tagging is relatively higher, the distinguishing features for

[‡]We are using the same initialiser for all networks.

Sl.No	R_0	AUC	$1/\epsilon_B$ at $\epsilon_S = 50\%$
1.	0.1	0.9865±0.0004	2415±104
2.	0.2	0.9864±0.0004	2332±95
3.	0.3	0.9863±0.0004	2381±71
4.	0.4	0.9868±0.0004	2254±80
5.	0.5	0.9868±0.0005	2300±226

Table 5.4: AUC values and the background rejection for different values of R_0 for *W tagging* dataset. Uncertainties for AUC are the standard deviation from five training instances, while for the background rejection $1/\epsilon_B$ are half of the inter-quartile range.

QCD vs top jets have a higher number of equally good local minima. The ROC band also enlarges with increased performance due to the decreasing statistics of the finite test sample.

5.3.2 Examining IRC safety

We now check the numerical stability of the network output for additional emissions. Since the network respects IRC safety, a jet with an additional splitting in the exact collinear or soft limit would have the same output without any splitting. We explicitly verified that the difference between the network output of jets and their respective copies with one additional splitting in the exact collinear or soft limit are zero within numerical precision. In order to check the network output's stability, we create copies of an original top jet belonging to the test dataset by splitting the hardest constituent. We choose the hardest constituent since numerically, it should have the maximum effect on the probabilistic output due to the $\omega_i^{(\mathcal{K})}$ weighted node and graph readouts. The splitting is done as follows. We create a scaled copy $z_r\ p_q$ of the hardest four-momentum p_q. Taking the plane formed by the hardest particle and the softest particle in the jet, we rotate it by an angle θ giving us the four-momentum of one daughter p_r. The second daughter's four-momentum p_s is determined by the enforcing conservation of energy and momentum. We vary the two quantities z_r and θ independently to get the network output of the jet with an additional split $y_{\mathcal{S}'}(z_r, \Delta R_{rs})$ as a function of z_r and ΔR_{rs}.

The contour of the absolute difference $|y_{\mathcal{S}} - y_{\mathcal{S}'}(z_r, \Delta R_{rs})|$ between the network output of the initial jet $y_{\mathcal{S}}$ and those with an additional splitting $y_{\mathcal{S}'}(z_r, \Delta R_{rs})$ for different values of R_0 is shown in Figure 5.5. We evaluate the difference of the best network from each of the five instances of training. For each R_0, we have plotted the contour having the maximum variance. The value of $y_{\mathcal{S}}$, which is the probability of the jet being a top, is also displayed. It can be seen that the difference goes to zero independently in the soft or collinear limits for all networks. This difference is low in considerable portions of the domain, indicating that the

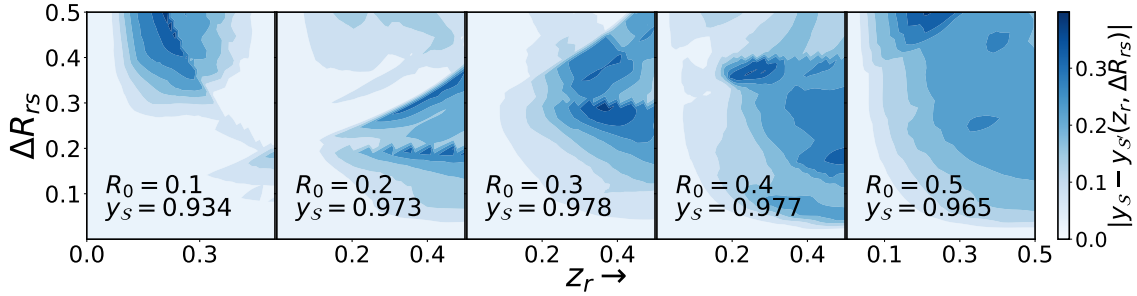


Figure 5.5: Variation of the network output with one additional particle, emitted from the hardest constituent in a top jet, on trained networks for various values of R_0. The contour figures show the contour of $|y_S - y_{S'}(z_r, \Delta R_{rs})|$ in the two-dimensional place $(z_r, \Delta R_{rs})$. Although, the differences are finite for non zero z_r or ΔR_{rs}, it goes to zero independently at the infra-red or collinear limits.

network output is relatively stable (at least for the particular jet).[§]

We see an increase of the area with non-zero differences as one increases R_0. To understand this behaviour, we examine how the neighbourhood sets of each particle in the jet with an additional splitting evolve as one increases R_0. For a fixed ΔR_{rs}, the two daughter's neighbourhood set would grow as one increases R_0. In contrast, for the remaining particles, the number of particles that have either of the daughter particles in them would increase with increasing R_0. Since they are greater in number, we expect this second aspect to influence the network output to a greater degree. Thus, even though the network performances are generally lower for smaller values of R_0, the network output is more stable for additional emissions that are not too soft or collinear. *Increasing R_0, therefore, increases network performance at the cost of increased computational load (due to the addition of edges) but decreases the network's stability to QCD emissions.*

Since the increase in performance for increasing R_0 comes at a price of a growing sensitivity to additional emissions, it is worth investigating how a deeper EMPN with more message-passing operations (which should increase the discrimination for a fixed R_0) fare against the same QCD radiations. We, therefore, train EMPN with two different message passing operations for $R_0 = 0.1$. For demonstration, we chose the gluon vs quark scenario because both classes' one-prong nature elevates the importance of the differing soft radiation patterns. We keep the structure of the downstream MLP and the message function at the first layer identical to the previously presented network. The second one is chosen to correspond to an edge-convolution operation given in eq. 6.4, with $l = 1$ in the superscript instead of $l = 0$. The MLP, therefore, takes a 256-dimensional input and outputs a 128-dimensional vector. It contains two hidden layers of 128 nodes

[§]We tested with multiple jets from the different classes and found similar features, for brevity we have only included a single plot.

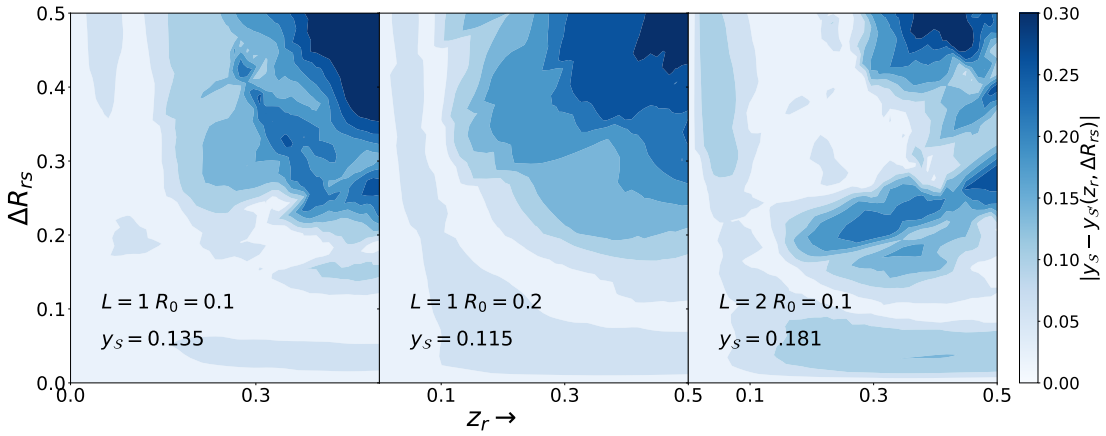


Figure 5.6: Comparing the variation of the network output $|y_{\mathcal{S}} - y_{\mathcal{S}'}(z_r, \Delta R_{rs})|$ of a gluon jet for a deeper network (right) with $L = 2$ and smaller radius $R_0 = 0.1$ against a shallower network (middle) with $L = 1$ and different $R_0 \in \{0.1, 0.2, 0.4\}$. For comparison, the variation with $L = 1$ and $R_0 = 0.1$ is also shown on the left.

each, with `ReLu` activation. The training is done five times with the same set of hyperparameters. We find a mean AUC of 0.8932 ± 0.0006 over the five training instances, confirming our presumption implying increasing performance with deeper models. Moreover, from Table 5.2, one finds that the value is even better than $R_0 = 0.4$ at $L = 1$ with AUC=0.8919±0.0006, which indicates that the performance scales much faster with the number of message-passing operations L than with R_0.

We now turn to investigate the phenomenologically important resilience to additional emissions. Following the procedure explained in the preceding paragraphs, the contours of $|y_{\mathcal{S}} - y_{\mathcal{S}'}(z_r, \Delta R_{rs})|$ for a gluon jet with

$$(L, R_0) \in \{(1, 0.1), (1, 0.2), (2, 0.1)\}$$

are shown in Figure 5.6. Along with the increasing discrimination, the model with $L = 2$, $R_0 = 0.1$ also provides better stability to additional emissions. The variation reduces with increasing depth for a constant R_0. Naturally, compared to $R_0 = 0.2$, which is less stable than with $R_0 = 0.1$ for constant L, we find that increasing L has an overall better phenomenological suitability than increasing R_0. Thus, deeper networks increase the performance and enhance the stability of additional emissions. This stability might be due to the larger number of functional compositions that a deeper model applies to the input, thereby reducing the sensitivity of the weight space (fixed after the training) to perturbations in the data. However, it needs a more detailed study since our present analysis is not extensive and does not reflect a truly realistic QCD picture.

5.4 Summary

In this chapter, we have devised an IRC-safe EMPN algorithm, and applied this approach to the discrimination of hadronically decaying top quarks and W bosons from QCD jets and the classification of jets into quark or gluon-induced jets. We find this algorithm to be highly performant, at par with other state-of-the-art neural network classification methods quoted in the literature. Thus, our definition of an IRC-safe Energy-weighted Message-Passing Network paves the way to highly performant jet classification algorithms that are at the same time insensitive to often poorly modelled parts of the event simulation, i.e. phase-space regions in the training event samples that are plagued by large theoretical uncertainties.

Chapter 6

Detecting anomalous jets with a Graph Autoencoder

In the preceding chapters, we have primarily considered supervised applications of deep-learning algorithms where we have some particular signal models. Such studies concentrate on a specific region of phase space to look for the hypothesised signal. However, the LHC, to date, has not found any tell-tale signs of new physics, which motivates model-unspecific searches. In this chapter, we explore the capability of graph neural networks in the form of graph autoencoders to the problem of anomaly detection of any non-QCD jet.

Convolutional autoencoders have been proposed and studied in [167, 169, 170, 191, 318, 319] for distinguishing QCD jets from non-QCD jets using "jet-images" [115, 117] as the input space. However, as discussed in chapter 2, convolutions on these images are expensive due to their extreme sparsity and are limited to the Euclidean domain. GNNs mitigate these two inadequacies, so studying their performance as anomaly finders is motivated. A study of particle graph autoencoders for anomaly detection has been carried out in the LHC Olympics community challenge [320]. A typical obstacle of GNN-based autoencoders is achieving an appropriate reflection of all network features on the decoding side. Existing graph-autoencoders in the literature [321–325] are designed mostly for node-classification or link prediction, while we desire a network capable of classifying graphs. Moreover, jets provide us with multidimensional edge information, along with node features; classifying the entire graph thereby exploits the full kinematic information of the event. To solve this known difficulty of graph-autoencoders, we design a decoder network capable of simultaneously reconstructing multidimensional edge, and node features with the help of *Inner Product Layers*.

This chapter is organised as follows: Section 6.1 introduces our analysis setup. The graph neural network methodology that we use in this work is described in Section 6.2, where we provide details on the network's architecture and its performance. Results are presented in Section 7.4, and we summarise in Section 7.5.

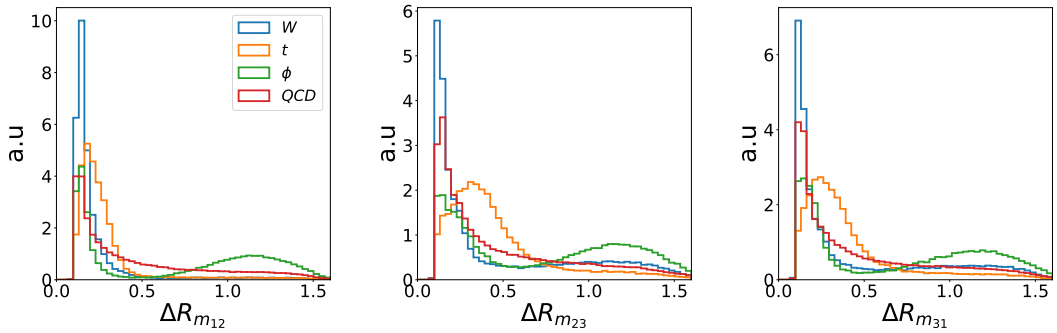


Figure 6.1: Normalised angular separation distribution between three leading microjets in the fat jet for the physics scenarios discussed in this work.

6.1 Elements of the Simulation

For our proof-of-principle analysis*, we generate events using `MadGraph5` [61] at leading order (LO), followed by `Pythia8` [326] for showering and hadronization. The hadronic final states are then clustered into jets using the anti-k_t algorithm [91] with parameter $R = 1.5$ using `FastJet` [94]. Along with a requirement that the rapidity of jets is $|y| < 2.5$, the minimum transverse momentum of a jet is required to be $p_T > 1$ TeV for this "fat jet" cluster. Only the leading jet from each multi-jet event is used as an input to the graph network and we do not include detector effects to our analysis. The sample used for training of the autoencoder (for details see below) is a QCD multi-jet background sample, consisting of 200k generated $pp \to jj$ events.

To test the autoencoder's anomaly detection performance we use three different signal samples, each consisting of 100k events generated with `MadGraph5`, using the same procedure described above. These samples consist of

(i) boosted hadronically-decaying W bosons as a benchmark for two-prong jet structure,

(ii) boosted hadronically-decaying top quarks, as a benchmark for a three-prong structure, and

(iii) a boosted scalar ϕ decaying as $\phi \to W^+W^- \to 4j$ to give a four-prong structure. The interaction is based on a simplified Lagrangian

$$\mathcal{L} \supset -\frac{c_1}{v}\phi W^{\mu\nu}W_{\mu\nu} - c_2(u\bar{u} + d\bar{d})\phi, \tag{6.1}$$

where c_1 and c_2 are dimensionless constants and v is the Higgs field's vacuum expectation value (vev). We choose $m_\phi = 700$ GeV for demonstration purposes, but note that our results are not too sensitive to the ϕ mass scale.

*Throughout this work, we will focus on 13 TeV LHC collisions.

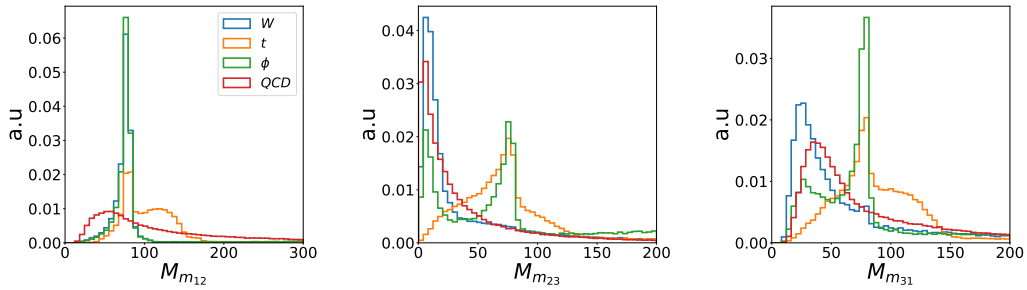


Figure 6.2: Similar to figure 6.1, but showing the normalised invariant mass distribution between three leading microjets in the fat jet.

To map out an infrared safe input to the graph network, we first use the anti-k_T jet algorithm to re-cluster the fat jet constituents into microjets[†] with a finer resolution of $R = 0.1$ and minimum $p_T = 5$ GeV. We consider fat jets with at least three microjets for our neural network analysis.

Identifying each microjet as a node in the network, we construct a graph associated with each jet as follows:

- **Node feature vectors**: We associate five microjet observables as the node's feature vector $\vec{x}$. These are $\log p_t$, $\Delta\eta$, $\Delta\phi$, ΔR, and $\bar{m}$. Here, p_t is the transverse momentum of the microjet, $\Delta\eta, \Delta\phi$, and ΔR are differences in pseudorapidity, azimuthal angle, and angular distance between the microjet and the jet axis respectively. $\bar{m}$ is the mass of the microjet divided by 100 GeV, which, along with the log on p_t, reduces the disparity in the range with the other three angular variables.

- **Edge feature vectors**: After the nodes are defined, we define the graph as the complete graph with all possible edge connections. For each edge, we construct an associated edge-feature vector of three dimensions. Its components are the two distance parameters between the nodes as defined below, and one invariant mass parameter: $\vec{e}_{ij} \equiv (d_{ij}^{\text{CA}}, \log d_{ij}^{k_t}, \log m_{ij})$. The metric d_{ij} is given by

$$d_{ij} = \min(p_{ti}^{2p}, p_{tj}^{2p}) \frac{R_{ij}^2}{R^2},$$

  where $p = 0$ for Cambridge-Aachen (CA) jets, $p = 1$ for k_t jets and R is radius parameter for the fat jet. The CA measure provides information about the geometric distance between two microjets, whereas the k_t measure is motivated from QCD splittings [89, 328]. m_{ij} is the invariant mass of the two microjets. These three variables capture the essential physics between two nodes.

- **Adjacency Matrix**: We also construct the adjacency matrix for each edge feature to facilitate their reconstruction at the decoder side. It is defined

[†]As shown in reference [327], such objects are under good experimental control.

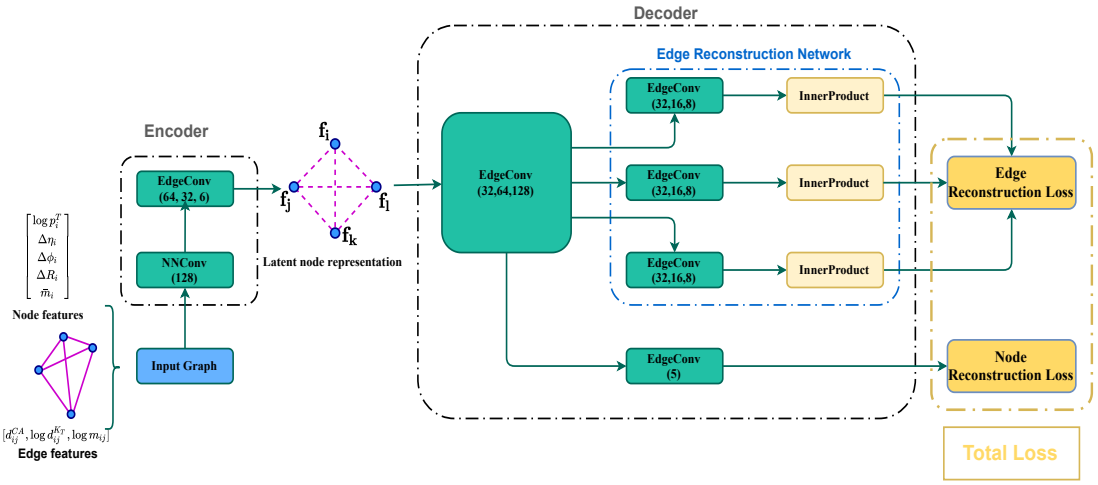


Figure 6.3: A schematic representation of a graph-autoencoder network. The network contains the (a) Encoder and the (b) Decoder. We employ an edge reconstruction network in the decoder to reconstruct the multidimensional edge information.

as

$$A^a_{ij} = A^a_{ji} = \begin{cases} e^a_{ij} & \text{if } i \neq j \\ 1 & \text{otherwise} \end{cases} ,$$

where a is the vectorial index. Thus, for a jet-graph of N nodes, we have three $N \times N$ matrices. The network outputs the edge-features in this representation, and hence the edge-loss is defined as a function of these adjacency matrices.

The distribution of ΔR_{ij} and m_{ij} for the 3-leading microjets of each jet are shown in figures 6.1 and 6.2. The construction of the graphs and the network analysis are performed using the `Deep Graph Library` [233] with the `PyTorch` [231] backend.

6.2 Graph Autoencoders

In this section, we describe the various components of our neural network analysis. We briefly detail the conceptual structure of the graph autoencoder before moving on to describe the ones we utilise in our analysis, along with the explicit form of the autoencoder's loss function. The network architecture and the process of training are described thereafter.

Graph-autoencoders are typically designed for classifying nodes or edges, focusing on learning local features of a huge graph. However, as our goal is to classify small graphs, the network needs to learn global graph structures *and* local features. To overcome this, we design an edge-reconstruction network within the decoder, making our network capable of learning graph structures by reconstructing the graph in its entirety. The complete structure of our network is

shown in figure 6.3, where the black boxes encase the encoder and the decoder. The edge-reconstruction network is shown bounded by the blue box. These are described in greater detail in the following passages.

6.2.1 Designing a graph autoencoder

Autoencoders are neural networks that map an input space to a bottleneck dimension (the latent dimension) and then back again to a space identical to the input. We use the graph-convolutions proposed in reference [199] to incorporate the multi-dimensional edge information along with the input node features. Our network, therefore, learns the physics information that is encoded into our 3-dimensional edge feature. The timesteps until we reach the latent space employ edge-convolution [203], which has proved excellent performance in supervised learning scenarios [210, 219]. We refer to these two layers as *NNConv*, and *EdgeConv* respectively, according to the python class name implemented in the `Deep Graph Library`. The encoder block outputs a graph with the same edge connections as that of the input with updated latent features $\vec{f}_i$ for each node. The decoder reconstructs the node and edge features from this latent node representation. As shown in figure 6.3, the decoder has a shared block of edge convolutions, after which the output feeds into four different blocks of edge convolutions: a single layer for the node reconstruction, and three edge reconstruction blocks. These three blocks are identical in structure and reconstruct each edge feature independently from the propagated information from the shared block. We use an *Inner Product Layer* [321] to reconstruct the edge information in the form of three adjacency matrices. These three components and the composition of the loss function are explained in the subsequent paragraphs.

NNConv: The first layer takes the node and edge features as input and performs a weighted graph convolution by making use of an MLP, referred to as edge function F_w. This takes the edge features as input and maps it to a dimension of $m \times n$, where m is the input node's dimensions (5 in the present case), and n is the dimension of the updated node features. The message passing function performs a broadcasted element-wise multiplication of the form

$$ {}^{ab}m_{ij}^{(1)} = {}^{ab}F_e(\vec{e}_{ij}) \times {}^{ab}\tilde{h}_j^{(0)} , \tag{6.2} $$

where a and b are the indices of the matrix, and ${}^{ab}\tilde{h}_j^{(0)}$ is formed by repeating $\vec{h}_j^{(0)}$, the input node features, n times. The aggregation step takes the mean of ${}^{ab}m_{ij}^{(1)}$ over all neighbouring nodes j, and then sums over the a index of the matrix:

$$ {}^{b}h_i^{(1)} = \sum_a \text{mean}_{j\in\mathcal{N}(i)} \left(\left\{ {}^{ab}m_{ij}^{(1)} \right\} \right) , \tag{6.3} $$

to give updated n dimensional node features $\vec{h}_i^{(1)}$.

EdgeConv: The backbone of our architecture is the edge convolution operation [203]. This involves two linear layers: Θ_w and Φ_w, with identical input and output dimensions, which determine the dimensions of original and updated node features respectively. The message passing function is defined as

$$\vec{m}_{ij}^{(l)} = \Theta_w(\vec{h}_j^{(l)} - \vec{h}_i^{(l)}) + \Phi_w(\vec{h}_i^{(l)}) \,, \tag{6.4}$$

while the aggregation step involves taking the maximum value

$${}^a h_i^{(l+1)} = \max_{j \in \mathcal{N}(i)} \{ {}^a m_{ij}^{(l)} \} \,, \tag{6.5}$$

in each component a of the incoming message vectors to give the updated node features $\vec{h}_i^{(l+1)}$.

Inner Product Layer: The edge-reconstruction network uses an Inner Product Layer to reconstruct the edge features from the node features of the final edge convolution output. The inner product makes the correspondence to the two-node indices for each edge. Since our graphs are undirected, the layer constructs a symmetric $N \times N$ matrix, N being the number of nodes in the graph. Its components are therefore

$$\hat{A}_{ij} = \vec{h}_i \,.\, \vec{h}_j \,, \tag{6.6}$$

where $\vec{h}_i$ and $\vec{h}_j$ are node-feature vectors.

6.2.2 Loss Function

We use root-mean squared error (RMSE) for the node as well as the edge reconstruction losses. For the node feature this is

$$L_{node} = \sqrt{\sum_{ia} \frac{(\hat{x}_i^a - x_i^a)^2}{N \times 5}} \,, \tag{6.7}$$

where a is the node-feature index, i is the node index, $\hat{x}_i^a$ and x_i^a are the reconstructed and input node features, respectively. We define the edge reconstruction loss as the sum of three individual RMSEs for each edge feature

$$L_{edge} = \sum_a \sqrt{\sum_{ij} \frac{(\hat{A}_{ij}^a - A_{ij}^a)^2}{N \times N}} \quad , \tag{6.8}$$

where a is the edge-feature index, i and j are node indices. $\hat{A}_{ij}^a$ and A_{ij}^a are the reconstructed and input adjacency matrices respectively. The total loss is the weighted sum of the individual losses,

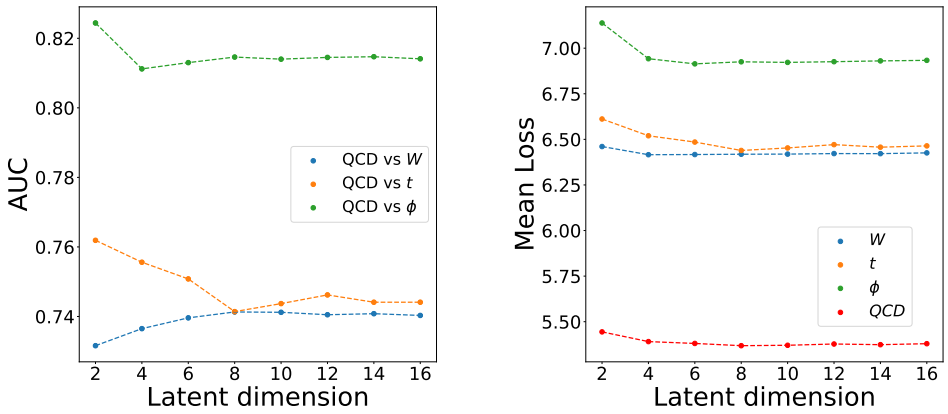
$$L_{auto} = \lambda_{node} \, L_{node} + \lambda_{edge} \, L_{edge} \tag{6.9}$$

We choose $\lambda_{node} = 0.3$ and $\lambda_{edge} = 1$, so that the combined node features get the same weight as each individual edge feature, which carry more relevant physics information. Note that the loss function is invariant to node permutations of the input graph since, mean is a permutation invariant function, and the architecture respects permutation invariance: any change in the node ordering changes the output of each layer (via the node readout) in conjunction with the adjacency matrix. Our network however, does not reconstruct an arbitrarily permuted graph for a given input, which is not strictly necessary since we concentrate on the reconstruction error of a single graph and not of an equivalence class of graphs.

6.2.3 Network Architecture and training

Neural networks require a careful optimal choice of hyperparameters. As this is a proof-of-principle analysis, we do not perform an extensive hyperparameter scan. However, we scan over the latent dimension, which is critical for any autoencoder. For the first layer of the graph-encoder (NNConv), we use an MLP of hidden dimensions: 256, 128, 64, and 32 as the edge function to map the 3-dimensional edge features to a 5×128 dimensional output. The hidden layers have ReLU activations, whereas the final layer has a sigmoid activation. The limited range of the sigmoid activation helps in giving the addition operation in aggregation (as defined in eq. 6.3) an interpretation of a weighted sum over messages in an additional dimension without the dynamics being entirely dominated by the outputs of the edge function. Each hidden layer has a dropout layer with fraction 0.2 of disconnected nodes between layers to avoid overfitting and achieve better generalisation. After the aggregation, we get a 128-dimensional output that feeds into a series of edge-convolution layers with linear layers as Θ_w and Φ_w. The output dimensions of the linear layers are 64 and 32 and outputs a 6 dimensional latent node encoding. This value is chosen after a scan over different latent dimensions which we elaborate on in the next section. The shared block of the decoder uses the encoder's reversed dimensions: 32, 64, and 128. With the 128-dimensional vector as input, the node reconstruction layer performs an edge-convolution to give the reconstructed node vectors $\hat{\vec{x}}$. Similarly, each edge reconstruction network has three successive edge convolutions of output dimensions 32, 16, and 8. We calculate the inner products on the 8-dimensional vector space to give the reconstructed adjacency matrices $\hat{A}^a_{ij}$.

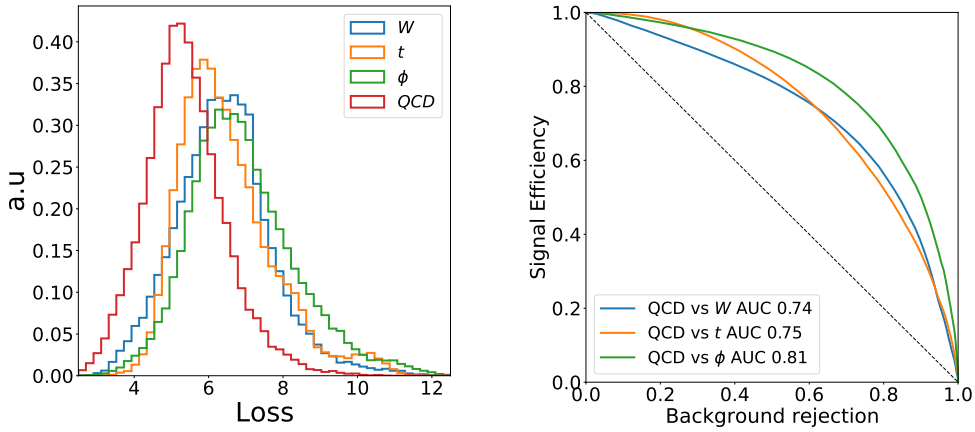
We train the network with the Adam optimiser [171] initialised with a 0.001 learning rate on mini-batches of 64 samples. The learning rate is decayed with a reduce-on-plateau condition with decay factor 0.5, and a patience of five epochs with an additional five epochs of cool-down. We use 85k jets to train the network. After each epoch, we calculate the loss of an independent validation dataset containing 28k QCD jets. We stop the training once the learning rate goes below 10^{-8}. The epoch with minimum validation loss is used for further inference.



(a) (b)

Figure 6.4: The AUC and mean loss for the three signal classes as a function of latent dimension from 2 to 12 for the given architecture

6.3 Results and Discussion



(a) (b)

Figure 6.5: The loss of the graph-autoencoder (a) and ROC curves (b) for a network trained only on QCD jets.

6.3.1 Performance for benchmark signals

In order to test the performance of the graph-autoencoder for the different non-QCD signals described in section 6.1, we evaluate the discrimination power of the total loss function as defined in eq. 6.9. We use an independent testing data set of 28k for QCD jets and a similar number for the signal samples. We first

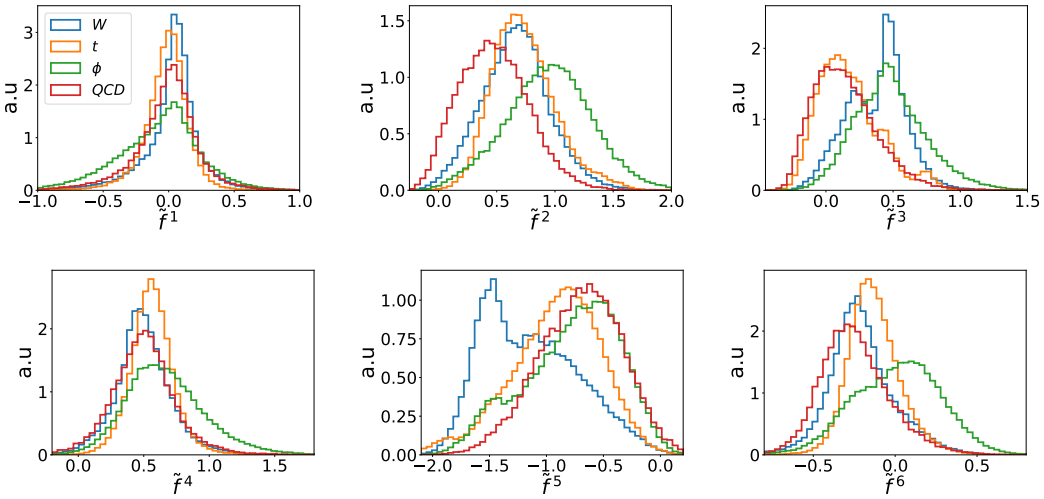


Figure 6.6: The distribution of six dimensional latent space after the training is performed only on QCD jets.

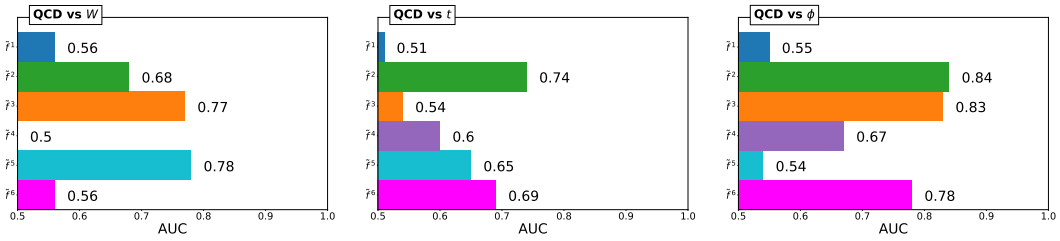


Figure 6.7: The AUC for all three signal classes corresponding to each latent dimension in the network.

scan the latent dimension from 2 to 12 in multiples of two, keeping all other hyperparameters fixed. The Area-Under-(the)Curve (AUC) between the signal acceptance and the background rejection for each latent dimension is shown in figure 6.4(a). In figure 6.4(b), we show the mean loss for each class as a function of the latent dimension. We can see that although the mean loss is relatively stable for QCD jets after 4-dimensions, the AUC of the different signal vs. QCD scenarios varies significantly. The variation is due to the unsupervised nature of the algorithm; the network has no information about the signal classes.

On the other hand, from the different nature of the AUC curves, we can understand the information passed for differing latent dimensions. The increasing AUCs for the W classification hints that the network sees them as similar to QCD jets when the information passed in the bottleneck is smaller, but the features of a typical QCD jet are not fully modelled for low dimensions, thus making this discrimination not reliable. Increasing the bottleneck dimension makes the network learn QCD features, which then leads to robust anomaly detection for top quarks and W bosons. The ϕ jets, which have the most noticeably different structure from QCD jets, reach a stable AUC much faster than tops and W bosons. We

infer that latent dimension of ~ 6 shows a stable performance for all three classes (in particular for QCD jets) and has reached the plateau in the mean loss. Since we cannot optimise the network to each class in anomaly detection, we fix six as the latent dimension parameter. The normalised distribution of the loss function for all classes is shown in figure 6.5(a). As the network is trained using QCD jets, the autoencoder reconstructs them with lower loss, while all signal classes have a relatively higher loss. By vetoing the QCD jets with lower losses, we tag (new physics) signal jets (anomalous class); the Receiver-Operator-Characteristic (ROC) curve between the signal acceptance and background rejection is shown in figure 6.5(b)[‡]. The performance increases as the prong structure becomes richer for the signal classes.

6.3.2 Looking at the latent graph representation

We also investigate the latent representation learned by the graph-autoencoder to explore compressed representations for QCD jets. Latent representations have also been investigated in similar, and indeed different, physical scenarios recently in references [329–331]. Even though we do not perform graph readouts during the training, the graph-autoencoder learns the graph structure via the edge reconstruction network. We use a graph readout that takes the mean in each dimension of the latent node features to obtain a fixed-dimensional latent graph representation. More precisely, we consider

$$\tilde{f}^a = \frac{1}{N} \sum_{i \in G} f_i^a \,,$$

where a is the vector-index, i is the node index and G is the set of all nodes of the graph. The normalised distribution of the four classes for each latent dimension is shown in figure 6.6, while the corresponding AUC for each signal vs. QCD discrimination is shown in figure 6.7. We find that $\tilde{f}^2$ performs best for top and ϕ jets, while $\tilde{f}^5$ gives the maximum AUC for W jets. The AUC for top quarks and scalar ϕ from $\tilde{f}^2$ is 0.74 and 0.84 respectively. Thus, we find a significant improvement for ϕ from the value obtained with the loss function, which is also the case for the W jets whose AUC is 0.78 from $\tilde{f}^5$. The latent distributions are prone to training uncertainties since they do not have any regularising terms in the loss function.

More precisely, the shapes and location of these distributions will vary significantly for different training instances even when they give very similar distributions of the loss function. There are available remedies [332–334] for the training class, but controlling the signal distributions during unsupervised training is not possible by design. However, it may be possible to control them using physically

[‡]We compare the results obtained for our dataset with particle graph autoencoders used in reference [320] in Appendix D.

motivated priors, which is beyond the scope of our present work. Nevertheless, once we have a single training instance, latent dimension-based anomaly finders can be used by trimming the encoder network after training to contain only these two outputs. Control samples can be used to quantify the latent space distributions and could therefore find applications in trigger optimisation.

6.3.3 Correlation of the loss function with jet observables

The correlation of the loss function with different jet variables is essential in determining the trained network's biases. Although perfectly decorrelated discriminants to the jet's physical variables like transverse momentum (p_T), mass (M), or the number of constituents are highly coveted, it is not possible in practice – known methods to decorrelate them, like adversarial training, diminish the power of the discriminant. We discuss the correlation of our network's loss function with the quantities mentioned earlier in this section. The class-wise correlation of the four quantities is shown in figure 6.8. We see that the loss function and the p_T are uncorrelated with small positive values (the highest being 0.27 for ϕ), indicating that the loss function tends to increase with an increase in transverse momentum of the jet slightly, although the increase is minimal for the background QCD jets (~ 0.10). Jet mass is an important variable that helps in discriminating different classes of jets. However, a discriminant (the loss function) needs to be decorrelated entirely with jet mass as putting a cut on a correlated variable will lead to artificial bumps in the jet-mass distribution of the selected events. As can be seen, from figure 6.8, the loss function is reasonably correlated with the jet mass even for the QCD jets. Decorrelating the jet mass from the loss can be done via an adversarial network [169].

The reconstruction efficiency of convolutional autoencoders has been shown to decrease with an increase in the number of non-zero pixels [191], which leads to the possibility of missing out on potential signals with lower complexities than QCD jets. We find that our network behaves in the opposite way: the reconstruction error reduces with an increase in the number of microjets. More importantly, this reduction is minimal for the QCD jets, suggesting that the network learns a uniform feature of the jet graph regardless of the number of microjets. We can understand this independence via the structure of a graph neural network. A graph convolution layer essentially learns a set of weights shared for all the nodes and edges and hence learns the underlying feature regardless of the number of nodes/edges in the graphs. However, there is a strong negative correlation of the loss of the different signal classes with the number of microjets which can be understood via the fact that any extra radiation other than the said multiplicities essentially arise from QCD splittings. To further understand this behavior, we plot the median loss of the events with a fixed number of microjets for the four classes in figure 6.9. The initial increase in the median loss from three to five for the four-pronged ϕ jets further solidifies the preceding argument regarding the

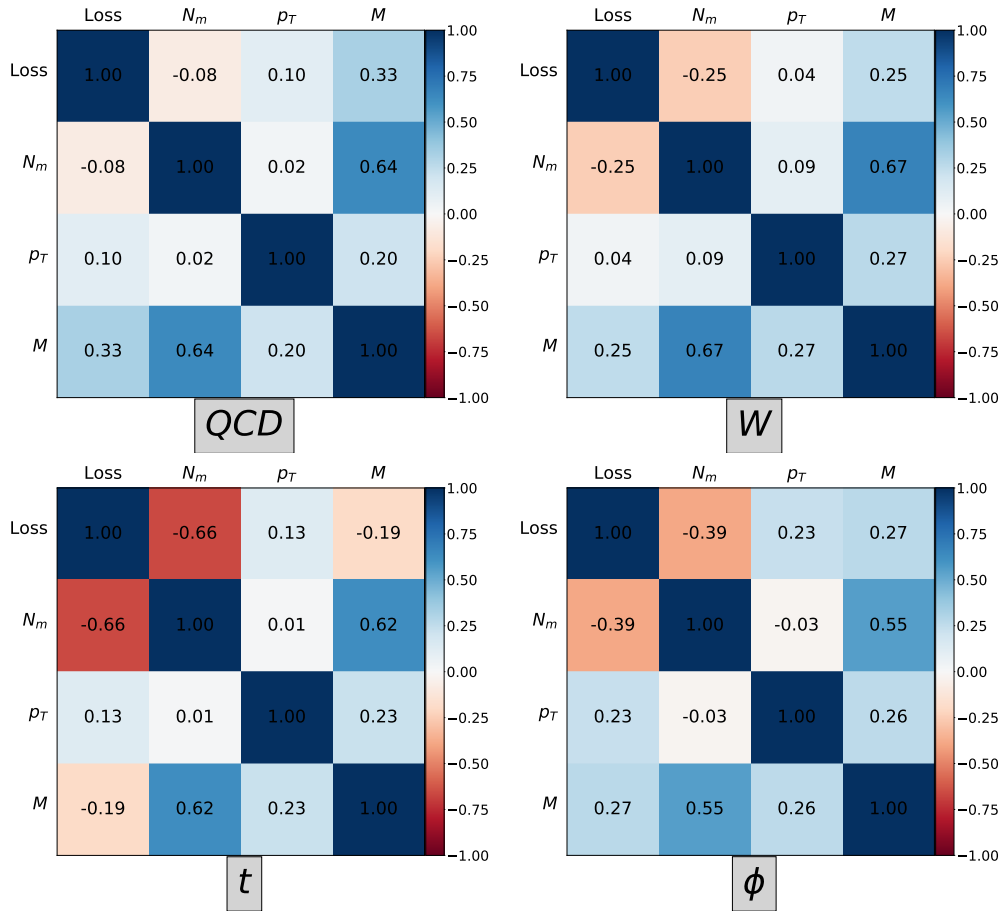


Figure 6.8: Linear correlation coeeficient between the loss, number of microjets(N_m), transverse momentum (p_T), mass(M) of the jet for the four jet classes: QCD(top left), W-boson (top right), top-quark (bottom left) and ϕ(bottom right).

decrease of the loss function with an increase in the microjet multiplicity. Such a peak is absent for the lower multiplicity signal classes.

6.4 Summary

In this chapter, we have introduced a GNN-based autoencoder for unsupervised anomaly detection in QCD boosted jet data. We design a novel edge-reconstruction network for the graph-decoder, which allows us to reconstruct multidimensional edge information. This gives the graph-autoencoder the capacity to classify entire graphs, unlike previously existing graph-autoencoders. We use NNConv to incorporate the multidimensional edge and node features as inputs to a graph-autoencoder while utilising edge convolutions to learn inductive latent space representations of QCD jets' graph-structured data.

The anomaly finder based on the reconstruction loss shows good performance for the non-QCD scenarios that we consider. We further explore the possibility of

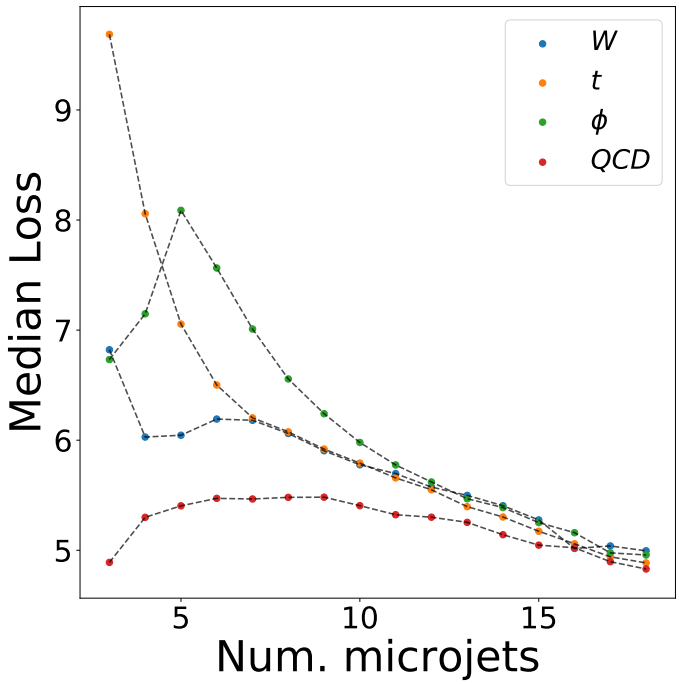


Figure 6.9: The median loss of events (from the test dataset) with fixed number of microjets for the various types of jets.

exploiting latent space variables as discriminants for anomalous jets and find that latent variables can indeed lead to improved anomaly detection by accessing the compressed information of the QCD data. While GNNs are known to be good candidates for trigger-level implementations, we study latent dimension-based anomaly finders with graph-autoencoders. Using latent dimensions instead of the loss has the additional appeal of halving the number of layers, thus resulting in a shallower network. Studying the latent dimension representation of QCD jets therefore provides a compressed arena for new physics discovery by using these observables directly.

Chapter 7

Anomaly detection with variational quantum circuits

With the advent of widely available noisy intermediate-scale quantum computers (NISQ) [335] the interest in quantum algorithms applied to high-energy physics problems has spurred. Today's quantum computers have a respectable quantum volume and can perform highly non-trivial computations. This technical development has resulted in a community-wide effort [336, 337] exploring the applications of quantum computers for studying quantum physics in general and in particular, the application to challenges in the theoretical description of particle physics. Some recent studies in the direction of LHC physics include evaluating Feynman loop integrals [338], simulating parton showers [339] and structure [340], quantum algorithm for evaluating helicity amplitudes [341], and simulating quantum field theories [342–347]. An interesting application of quantum computers is the nascent field of quantum machine learning–leveraging the power of quantum devices for machine learning tasks, with the capability of classical* machine learning algorithms for various applications at the LHC already recognised, it is only natural to explore whether quantum machine learning (QML) can improve the classical algorithms [348–355].

In this chapter, we explore the feasibility and potential advantages of using quantum autoencoders (QAE) for anomaly detection. Most quantum algorithms consist of a quantum state, encoded through qubits, which evolves through the application of a unitary operator. The necessary compression and expansion of data in the encoding and decoding steps are manifestly non-unitary, which has to be addressed by the QAE using entanglement operations and reference states which disallow information to flow from the encoder to the decoder. To this end, a QAE should, in principle, be able to perform tasks ordinarily accomplished by a classical autoencoder (CAE) based on deep neural networks (DNN). The ability of DNNs are known to scale with data [356], and large datasets are necessary

*By classical, we mean any machine learning algorithm that leverages only discrete bit computations, while by quantum, we imply a computation that uses the properties of quantum mechanics and qubits, even if they are simulated on classical hardware.

to bring out their better performance over other machine-learning algorithms. Interestingly, we find that a quantum autoencoder, augmented using quantum gradient descent [351, 357] for its training, is much less dependent on the number of training samples and reaches optimal reconstruction performance with minuscule training datasets. Since the use of quantum gradient descent is a relatively new way of improving the convergence speed and reliability of the quantum network training, we provide a detailed introduction in Section 7.1.1. Moreover, compared to CAEs, which use the same input variables as the QAE, QAEs have better anomaly detection capabilities for the two benchmark processes we use in our study. This better performance is particularly interesting as the CAE has $\mathcal{O}(1000)$ parameters compared to just $\mathcal{O}(10)$ for the QAE. The study indicates the possibility to study quantum latent representations of high-energy collisions, in analogy to classical autoencoders [358–360]. With the current state-of-the-art quantum computers capable of processing more than a hundred qubits, quantum anomaly detection could already rival classical state-of-the-art anomaly detection techniques.

The rest of the chapter is organised as follows. We describe the basic ideas of quantum machine learning and a quantum autoencoder in section 7.1. The details of the data simulation, network architecture, and training are described in section 7.3. We present the performance of a quantum autoencoder compared to a classical autoencoder in section 7.4. We conclude in section 7.5.

7.1 Quantum machine learning

Quantum machine learning broadly deals with extending classical machine learning problems to the quantum domain with variational quantum circuits [361]. We can divide these circuits into three blocks: a state preparation that encodes classical inputs into quantum states, a unitary evolution circuit that evolves the input states, and a measurement and post-processing part that measures the evolved state and processing the obtained observables further. For this discussion, we will always work in the computational basis with the basis vectors $\{|0\rangle, |1\rangle\}$ denoting the eigen states of the Pauli Z operator $\hat{\sigma}_z$ for each qubit.

There are many examples of state preparation in literature [362], which has their own merits in various applications. We prepare the states using *angle encoding*, which encodes real-valued observables ϕ_j as rotation angles along the x-axis of the Bloch sphere

$$|\Phi\rangle = \bigotimes_{i=1}^{n} R_x(\phi_j)\,|0\rangle = \bigotimes_{j=1}^{n} \left(\cos\frac{\phi_j}{2}\,|0\rangle - i\sin\frac{\phi_j}{2}|1\rangle\right) , \tag{7.1}$$

where $R_x = e^{-i\frac{\phi_j}{2}\hat{\sigma}_x}$ denote the rotation matrix. The number of qubits required n, is same as the dimensions of the input vector. A parametrised unitary circuit

$\mathbf{U}(\Theta)$, with Θ denoting the set of parameters, evolves the prepared state $|\Phi\rangle$ to a final state $|\Psi\rangle$,

$$|\Psi\rangle = \mathbf{U}(\Theta)\,|\Phi\rangle\ . \tag{7.2}$$

The final measurement step involves the measurement of an observable on the final state $|\Psi\rangle$. Since measurements in quantum mechanics are inherently probabilistic, we measure multiple times (called shots) to get an accurate result. In order to do that, we need quantum hardware that can prepare a large number of pure identical input states $|\Phi\rangle$ for each data point.

After defining a cost function, the parameters Θ can be trained and updated using an optimisation method. To better capture the geometry of the underlying Hilbert space and to achieve a faster training of the quantum network,[†] we will use quantum gradient descent [357], where the direction of steepest descent is evaluated according to the Fubini-Study metric [363, 364]. The general idea is to make the optimisation procedure aware of the weight space's underlying quantum geometry, which improves the speed and reliability of finding the global minimum of the loss function. A brief outline of quantum gradient descent is given in Section 7.1.1.

While we have not discussed the specific form of the parametrised unitary operation $\mathbf{U}(\Theta)$, it is important to note that one of the major advantages of quantum computation is due to its ability to produce *entangled states*, a phenomenon absent in devices based on classical bits. The prepared input state is separable into the component qubits, and a product of unitaries acting on single-qubit states will not entangle the subsystems. The CNOT gate is a standard two-qubit gate, which will be used in our circuit to entangle the subsystems.

7.1.1 Quantun Gradient Descent

We discuss the basic idea behind quantum gradient descent [357] in this section. The general idea is to make the optimisation procedure aware of the underlying quantum geometry of the weight space. Denoting any generic weight vector by Θ, we have the vanilla gradient descent update as,

$$\Theta_{i+1} = \Theta_i - \gamma\ \nabla_\Theta\ L(\Theta) \quad , \tag{7.3}$$

where L is a well-behaved loss function. This expression implicitly assumes that l_2 distances correctly describe the underlying geometry of the weight space, placing all directions in the weight space on an equal footing. In reality, however, the geometry of the weight space can be much more complicated, and such a straightforward update rule may not converge to the optimal point. Therefore,

[†]See [351] for a brief presentation of the Fubini-Study metric and a comparison of natural and quantum gradient descent for the training of classical and quantum networks. It was shown that quantum gradient descent improves the training of a variational quantum circuit significantly.

to have an idea of the underlying geometry, we modify eq. 7.3 with the metric tensor $\mathbf{G}$,

$$\Theta_{i+1} = \Theta_i - \gamma\ \mathbf{G}^{-1}(\Theta_i)\ (\nabla_\Theta L(\Theta))_{\Theta=\Theta_i} \quad , \tag{7.4}$$

to get the Natural Gradient descent [365]. Note that Natural Gradient descent gives the usual gradient descent (eq. 7.3) for a Euclidean metric $\mathbf{G} = \mathbf{I}$. Due to the extremely large parameter space, it is computationally prohibitive to put metric-restrained optimisation in deep neural networks, which is not the case for currently used variational quantum circuits. The natural metric on complex projective Hilbert Spaces (the space containing physical quantum states) is the Fubiny-Study metric [363, 364],

$$g_{ij} = \mathrm{Re}[\ \langle \partial_i \phi_0 | \partial_j \phi_0 \rangle - \langle \partial_i \phi_0 | \phi_0 \rangle\ \langle \phi_0 | \partial_j \phi_0 \rangle\ ] \quad . \tag{7.5}$$

Here, $|\partial_i \phi_0\rangle = \frac{\partial |\phi_0\rangle}{\partial \theta_i}$, with θ_i, a component of the weight vector Θ and $|\phi_0\rangle$, a state in the Hilbert space. The inverse of the metric is evaluated in `Pennylane` using the Moore-Penrose pseudo inverse [366, 367]

$$g^+ = (g^T g)^{-1} g^T \quad ,$$

which is well-behaved even when $\det g = 0$ and is numerically equal to the inverse when it exists.

7.2 Quantum autoencoders on variational circuits

Quantum autoencoders based on variational circuit models have been proposed for quantum data compression [368]. In our work, we want to learn the parameters of such a network to compress the background data efficiently. Along the same principles as anomaly detection on classical autoencoders, we expect that the compression and subsequent reconstruction will work poorly on data with different characteristics to the background.

A quantum autoencoder, in analogy to the classical autoencoders has an encoder circuit which evolves the input state $|\Phi\rangle$ to a latent state $|\chi\rangle$ via a unitary transformation $\mathbf{U}(\Theta)$, and then reconstructs the input state, via its hermitian conjugate $|\Phi\rangle = \mathbf{U}^\dagger(\Theta)|\chi\rangle$. However, note that since unitary transformations are probability conserving and act on spaces having identical dimensions, there is no data compression in such a set-up. In order to have data compression, some qubits at the initial encoding $|\chi\rangle$ are discarded and replaced by freshly prepared reference states. Such a setup for a four feature input and two dimensional latent space is shown in figure 7.1. The unitary operators output identical number of qubits, however at the encoder step, two of its outputs (shown by green lines) are replaced by freshly prepared reference states (shown in orange lines), devoid of

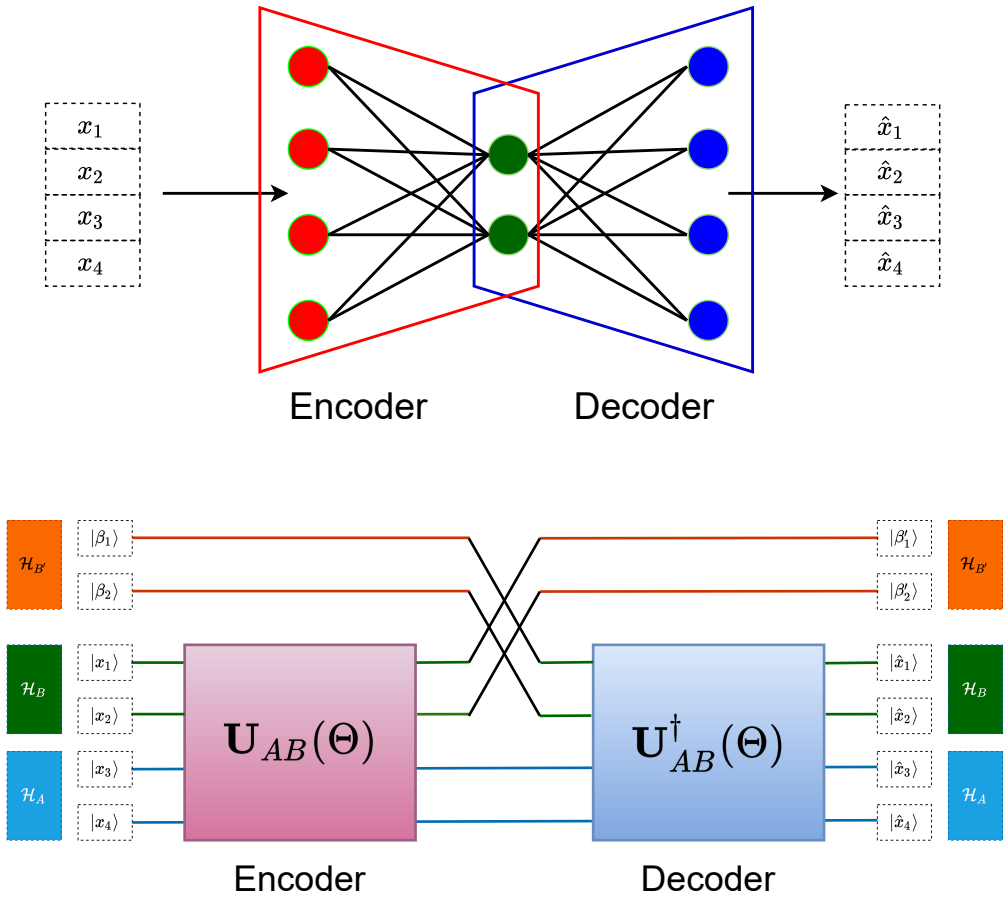


Figure 7.1: Schematic representation of a simple dense classical autoencoder (left) and a quantum autoencoder (right) for a four dimensional input space and a two dimensional latent space. To induce an information bottleneck in quantum unitary evolutions, we throw away states $|\beta'_i\rangle$ (trash states) at the encoder output (green lines), which are replaced by reference states $|\beta_i\rangle$ (shown in orange lines), containing no information of the input $|x_j\rangle$. The mechanism can be better understood by dividing the Hilbert space of the complete system into three parts: $\mathcal{H}_A$ the subspace formed by the qubits that are fed to the decoder, $\mathcal{H}_B$ the subspace of the qubits that are discarded after encoding, and $\mathcal{H}_{B'}$ the subspace where a fixed reference state (initialised as $|0\rangle^{\otimes \dim \mathcal{H}_B}$) unacted by the encoder is fed to the decoder. SWAP gates can achieve the exchange of states denoted by black lines.

any information of the input states. We describe the basics of quantum autoencoding in the following, mainly based on the discussion of quantum autoencoders for data compression from reference [368]. Quantum anomaly detection of simulated quantum states has been investigated in reference [369]. To the best of our knowledge, our study is the first to explore anomaly detection of classical inputs via a quantum autoencoder. The main difference between existing studies and ours is that the input states for the former are inherently quantum mechanical. In contrast, the choice of input embedding of the classical numbers in our case

determines the nature of the quantum state. We will use angular encoding, where the quantum states are separable into the constituent qubits. We will, however, be extensively using CNOT gates in the unitary evolution which will entangle the different qubits.

Let us denote the Hilbert space containing the input states by $\mathcal{H}$. For describing a quantum autoencoder, it is convenient to expand $\mathcal{H}$ as the product of three subspaces,

$$\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B \otimes \mathcal{H}_{B'} \ , \tag{7.6}$$

with subspace $\mathcal{H}_A$ denoting the space of qubits fed into the decoder from the encoder, and $\mathcal{H}_B$ denoting the space corresponding to the ones that are re-initialised, and $\mathcal{H}_{B'}$ denoting the Hilbert space containing the reference state. In the following, we will denote states belonging to any subspace with suffixes while the full set will have no suffix. For example, $|a\rangle_{AB} \in \mathcal{H}_A \otimes \mathcal{H}_B$, $|\kappa\rangle \in \mathcal{H}$, $|b\rangle_{B'} \in \mathcal{H}_{B'}$ etc. We will use the same convention for operators acting on the various subspaces.

Since we entangle the separable input qubits in the subspaces $\mathcal{H}_A \otimes \mathcal{H}_B$ via $\mathbf{U}_{AB}(\Theta)$, the latent state $|\chi\rangle_{AB} \in \mathcal{H}_A \otimes \mathcal{H}_B$, in general, is not seperable. The input of the larger composite system including the reference state is $|\Phi\rangle_{AB} \otimes |\beta\rangle_{B'}$, with $|\beta\rangle_{B'}$ denoting a freshly prepared *reference state* (initialised as $|0\rangle^{\otimes \dim \mathcal{H}'_B}$) not acted on by the unitary $\mathbf{U}_{AB}$. The process of encoding can be therefore written as,

$$|\chi\rangle_{AB} \otimes |\beta\rangle_{B'} = (\mathbf{U}_{AB}(\Theta) \otimes \mathbf{I}_{B'})\ |\Phi\rangle_{AB} \otimes |\beta\rangle_{B'} \quad , \tag{7.7}$$

where $\mathbf{I}_{B'}$ denotes the identity operator on $\mathcal{H}_{B'}$. Explicitly, the dimensions of the subspaces $\mathcal{H}_A$, $\mathcal{H}_B$, and $\mathcal{H}_{B'}$ are $2^{N_{lat}}$, $2^{N_{trash}}$, and $2^{N_{trash}}$, respectively, where N_{lat} is the number of qubits passed to the decoder directly from the encoder, while N_{trash} are the ones that are discarded. Swapping the B and B', gives the input to the decoder as

$$|\chi'\rangle = \mathbf{I}_A \otimes \mathcal{V}_{BB'}\ \ |\chi\rangle_{AB} \otimes |\beta\rangle_{B'} \ , \tag{7.8}$$

where $\mathcal{V}_{BB'}$ indicates a unitary that performs the swap operation,[‡] and $\mathbf{I}_A$ is the identity operator on $\mathcal{H}_A$. The output of the decoder can now be written as

$$|\Psi\rangle = \mathbf{U}^{\dagger}_{AB}(\Theta) \otimes \mathbf{I}_{B'}\ |\chi'\rangle \ , \tag{7.9}$$

with $\mathbf{I}_{B'}$ being the identity operator on $\mathcal{H}_{B'}$. The decoding, therefore, takes the swapped latent state $|\chi'\rangle$, and the unitary $\mathbf{U}^{\dagger}_{AB}$ evolves it with no information from the encoder in the subspace $\mathcal{H}_B$. The reconstruction efficiency of the au-

[‡]For instance swapping the state of two qubits in the basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$, can be implemented via the unitary matrix

$$\mathcal{V}_{BB'} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad .$$

toencoder can be quantified in terms of the *fidelity* between the input and output states in the subspace $\mathcal{H}_A \otimes \mathcal{H}_B$, which quantifies their similarity. For two quantum states $|\psi\rangle$ and $|\phi\rangle$, it is defined as

$$F(|\phi\rangle, |\psi\rangle) = F(|\psi\rangle, |\phi\rangle) = |\langle\phi|\psi\rangle|^2 \ .$$

For normalised states, we have $0 \leq F \leq 1$, with $F = 1$ only when $|\phi\rangle$ and $|\psi\rangle$ are exactly identical. We can write the fidelity of the complete system as

$$\begin{aligned} &F(|\Phi\rangle_{AB} \otimes |\beta\rangle_{B'}, |\Psi\rangle) \\ &\quad = F(|\Phi\rangle_{AB} \otimes |\beta\rangle_{B'}, \mathbf{U}_{AB}^{\dagger}\ \mathcal{V}_{BB'}\ \mathbf{U}_{AB}\ |\Phi\rangle_{AB} \otimes |\beta\rangle_{B'}) \ , \end{aligned}$$

where we have implicitly assumed that the unitary operators are extended to the whole space via a direct product with the identity operator on the subspace it does not act on, for notational compactness. Noting that $\mathbf{U}_{AB}|\Phi\rangle_{AB} = |\chi\rangle_{AB}$, we can write this as,

$$\begin{aligned} &F(|\Phi\rangle_{AB} \otimes |\beta\rangle_{B'}, |\Psi\rangle) \\ &\quad = F(|\chi\rangle_{AB} \otimes |\beta\rangle_{B'}, \mathcal{V}_{BB'}\ |\chi\rangle_{AB} \otimes |\beta\rangle_{B'}) \ . \end{aligned}$$

Writing the swaped state as $\mathcal{V}_{BB'}\ |\chi\rangle_{AB} \otimes |\beta\rangle_{B'} = |\chi\rangle_{AB'} \otimes |\beta\rangle_{B}$, we have

$$F(|\Phi\rangle_{AB} \otimes |\beta\rangle_{B'}, |\Psi\rangle) = F(|\chi\rangle_{AB} \otimes |\beta\rangle_{B'}, |\chi\rangle_{AB'} \otimes |\beta\rangle_{B}) \ . \tag{7.10}$$

Since we are interested in the wave functions belonging to the subspace $\mathcal{H}_A \otimes \mathcal{H}_B$, we trace over B' to get the required fidelity. However, a perfect fidelity between the input and outputs of the AB system can be achieved when the complete information of the input state passes to the decoder, i.e.

$$\mathbf{U}_{AB}|\Phi\rangle_{AB} = |\Phi^c\rangle_A \otimes |\beta\rangle_B \quad . \tag{7.11}$$

The state $|\Phi^c\rangle_A$ denotes a compressed form of $|\Phi\rangle_{AB}$, i.e it should contain the information of the AB system in the input, while $|\beta\rangle_B$ is equivalent to the reference state, with no information of the input. If the B and B' systems are identical during the swap operation, the entire circuit reduces to the identity map. The output of the B' system, hereby referred to as the *trash state*, is itself the determining factor of the output state fidelity. The output of the B' system can be obtained after tracing over the A system as: $\hat{\rho}_{B'} = \mathrm{Tr}_A\ \{|\chi\rangle\langle\chi|_{AB'}\}$ and the required fidelity of the B' system is $F(|\beta\rangle_{B'}, \hat{\rho}_{B'})$.

A perfect reconstruction of the input is possible only when the trash state fidelity $F(|\beta\rangle_{B'}, \hat{\rho}_{B'}) = 1$. Thus a quantum autoencoder can be trained by maximising the trash state fidelity instead of the output fidelity, which has the advantage of reducing the resource requirements during training. Although, the output fidelity obtained by tracing over the B' system is numerically not equal to

the trash state fidelity, we can use the latter in anomaly detection as well, since it is a faithful measurement of the output fidelity. Thus, unlike vanilla classical autoencoders, we can reduce the execution and training of QAEs into the encoder circuit for anomaly detection.

The above discussions have focused on the underlying principles behind a quantum autoencoding process on single input states. As stated before, we need to prepare identical input-states for each data point and repeat the unitary evolution and measurement to get a useful estimate of the fidelity, evident also from the use of density operators to express the output state. Referring to the ensemble of the input states as $\{p_i, |\Phi_i\rangle_{AB}\}$, we obtain for the cost function

$$C(\Theta) = -\sum_i p_i \, F(|\beta\rangle_{B'}, \hat{\rho}_{B'}) \quad , \tag{7.12}$$

where the negative sign converts the optimisation process into minimising the cost function. It is important to note that the ensemble should not be taken as being analogous to the batch training in classical neural networks, as it is required for the accurate prediction of the network output even when testing the autoencoder network.

7.3 Analysis Setup

7.3.1 Data simulation

To show the prowess of the quantum autoencoders, we study two processes with distinctive features: a QCD continuum background of top pair production taking possible signal signatures of resonant heavy Higgs decaying to a pair of top quarks, and invisible Z decays into neutrinos with a likely signal of the 125 GeV Higgs decaying to two dark matter particles. As we shall see in the following sections, the relative performance of QAEs over CAEs show parallels in these two different signatures, pointing towards an advantage of QAEs over CAEs not governed by the specific details of the final state.

Resonant Higgs signal over continuum top pair background

The first background and signal samples used in our analysis consist of the QCD $t\bar{t}$ continuum production, $pp \to t\bar{t}$, and the scalar resonance production $pp \to H \to t\bar{t}$, respectively. The background and the signal events are generated with a centre-of-mass energy of 14 TeV, as expected during future LHC runs. Each top decays to a bottom quark and a W boson, and we focus on the decay of the W's into muons exclusively. We consider four different masses of the scalar resonance, $m_H = 1.0, 1.5, 2.0$, and 2.5 TeV. All events are generated with `MadGraph5_aMC@NLO` [61], and showered and hadronisation is performed by `Pythia8` [70]. `Delphes3` [72] is utilised for the detector simulation, where the

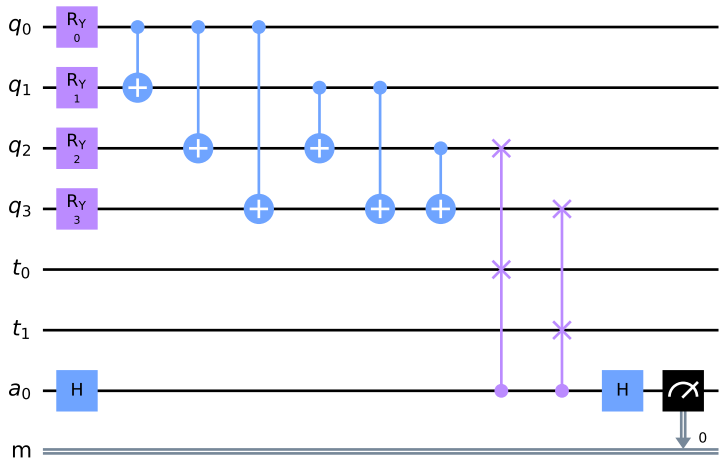


Figure 7.2: The figure shows a Quantum autoencoder circuit for a four qubit input and two latent qubits. The inputs are already embedded in q_i (by the input embedding circuit), which are then rotated by tunable angles θ_i in the y-direction of the Bloch sphere by $R_y(\theta_i)$ gates (shown in purple boxes). Each pair of these qubits are entangled via CNOT gates(shown with blue lines). For the trash training, we need a two-dimensional reference state denoted by t_i qubits and an ancillary qubit a_0. The fidelity between two qubits at the encoder output and the reference states is measured via a SWAP test.

jets are clustered using `FastJet` [94]. We generate about 30k events for the background samples, while for each signal sample, we generate about 15k. The background events are divided into 10k training, 5k validation and 15k testing samples.

For the object reconstruction, a standard jet definition using the anti-k_t algorithm [91] with the jet radius $R = 0.5$ is used. For the signal bottom jets, the output from Delphes 3 is used and require $p_T^b > 30$ GeV. For isolated leptons, we require $p_T^l > 30$ GeV and its isolation criteria with $R = 0.5$. We extracted four variables $\{p_T^{b_1}, p_T^{l_1}, p_T^{l_2}, \not{E}_T\}$ for our analysis, keeping in mind the limitations of current devices. To conserve the aperiodic topology of these variables in the angle embedding (given in eq. 7.7) we fix the range of each variable to $[0, 1000]$ by adding two points[§] and map the whole dataset to a range $[0, \pi]$ via the `MinMaxScaler` implemented in `scikit-learn` [280]. The two added points are then removed from the dataset. This maps each feature's minimum and maximum to two distinct angles separated by a finite distance due to the selection criteria.

Invisible Higgs signal over invisible Z background

To test the anomaly detection capabilities of QAEs in a different scenario, we study invisible decays of a Z boson produced with two jets originating from QCD vertices. As a possible signal, we take the production of the 125 GeV Higgs boson and two jets originating from Electroweak vertices, decaying to two scalar dark matter particles. The generation is carried out in the same manner as in the previous case, including the definition of jets. We demand that we have at least two reconstructed jets with $p_T > 30$ GeV, and the events have a missing transverse momentum $\not{E}_T > 30$ GeV. For the background, we have 30k events divided into 10k training, 5k validation, and 15k test events, while for the signal, we have 15k test events. We extract six variables to train the QAE and the CAE. They are the absolute separation in pseudorapidity between the two jets $|\Delta\eta_{jj}|$, the invariant mass of the dijet system m_{jj} and the sum of transverse energies

$$H_T^{\eta_C} = \sum_{|\eta_i|<\eta_C} E_T^i \quad ,$$

within four ranges of pseudorapidity $\eta_C \in \{1.0, 1.5, 2.0, 2.5\}$. The mapping to conserve the aperiodic topology of these variables in the angular embedding is done by increasing their range on the higher side.

7.3.2 Network architecture and training

The QAE was implemented and trained using `Pennylane` [370]. As stated before, we train and test the QAE model with only the encoder circuit. After the input

[§]Events with the variables lying above 1000 GeV are very rare and excluded in our case. In a realistic analysis, the upper bound can be determined from the data.

features are embedded as the rotation angle of the x-axis in the Bloch sphere, the unitary evolution $\mathbf{U}(\Theta)$ consists of two stages. In the first step, each qubit is rotated by an angle θ_i in the y-axis of the Bloch sphere. The values of these angles are to be optimised via gradient descent. After this, we apply the CNOT gate to all the possible pairs of qubits, with the ordering determined by the explicit number of the qubit. This circuit is shown in figure 7.2 for a four qubit input QAE with two-qubit latent representation. It is given by,

$$\mathbf{U}_{AB} = C_{23}\otimes C_{13}\otimes C_{12}\otimes C_{23}\otimes C_{03}\otimes C_{02}\otimes C_{01}\otimes R_y^0(\theta_0)\otimes R_y^1(\theta_1)\otimes R_y^2(\theta_2)\otimes R_y^3(\theta_3) \quad ,$$

where C_{ij} is the CNOT operation acting on the composite space of two qubits i and j, and $R_y^i(\theta_i)$ is the rotation of a single qubit i about the y-axis of the Bloch sphere. Note that the expression does not contain the operations of the SWAP test, which will be explained in the following paragraphs. The training proceeds to find the optimal values for θ_i.

The number of qubits discarded at the encoder, the size of the trash-state, fixes the latent dimension[¶] via $N_{lat} = N_{in} - N_{trash}$, with N_{lat} the latent dimension, N_{in} the size of the input state, and N_{trash} the number of discarded qubits. The reference state $|\beta\rangle_{B'}$, has the same number of qubits N_{trash}, and it is initialised to be

$$|\beta\rangle_{B'} = |0\rangle^{\otimes N_{trash}} \quad .$$

We measure the fidelity between the trash-state $\hat{\rho}_{B'}$ and the reference state $|\beta\rangle_{B'}$ via a SWAP test [371]. It is a way to measure the fidelity between two multi-qubit states. For any two states $|\phi\rangle$ and $|\psi\rangle$ with the same dimensions, the fidelity $F(|\phi\rangle, |\psi\rangle)$ can be measured as the output of an ancilliary qubit $|a\rangle_{anc}$ after the following operation,

$$\mathbf{H}_{anc}\otimes\mathbf{I}\ (\text{c-SWAP})\ \mathbf{H}_{anc}\otimes\mathbf{I}\ \ |0\rangle_{anc}\otimes|\phi\rangle\otimes|\psi\rangle \quad , \tag{7.13}$$

where $\mathbf{H}_{anc}$ is the Hadamard gate acting on the ancilliary qubit, and c-SWAP is the controlled swap operation between the states $|\phi\rangle$ and $|\psi\rangle$ controlled by the ancilliary qubit. Thus the total number of qubits required for a fixed N_{in} and N_{trash} is $N_{in} + N_{trash} + 1$. Due to the limitation of current quantum devices we limit the input feature to four, and scan over the possible latent dimensions.

The quantum network is trained by minimising the cost function (c.f eq. 7.12) with quantum gradient descent for the one, two and three-dimensional latent spaces. We train these instances for different training sizes of 1, 10, 100, 1000 and 10000 events to study the dependence of the QAE's performance on the size of the training data. We update the weights for each data sample, with 5000 shots in all training scenarios. For training sizes greater than or equal to 100, we train the networks for 50 epochs. In comparison, for sample sizes 1 and 10, we train

[¶]In our discussions, we will use the number of latent qubits as the latent dimension, although the Hilbert space would have $2^{N_{lat}}$ dimensions.

the QAE for 500 and 200 epochs, respectively. To benchmark the performance of a QAE on a quantum computer, we train a QAE with the two inputs $p_T^{l_1}$ and $p_T^{b_1}$ with quantum-gradient descent on `Pennylane`, and compare the test performance with the simulation and the IBM-Q. For running on the IBM-Q, we build and implement the test circuits in `Qiskit` [372].

We also train classical autoencoders using `Keras` (v2.4.0) [232] with `Tensorflow` (v2.4.1) [230] for the same input features, for comparison. The encoder is a dense network mapping the input space to a latent dimension of $N_{lat} \in \{1, 2, 3\}$, and has three hidden layers with 20, 15, and 10 nodes. The hidden layers have `ReLU` activations while the latent output has `Linear` activation. The decoder has a symmetric configuration to the encoder. The networks are trained with `Adam` [373] optimiser with a learning-rate of 10^{-3} to minimise the root-mean-squared error between the input vector $\mathbf{x}$ and the reconstructed vector $\hat{\mathbf{x}}$. For the CAEs, we found that training with single data per update (technically batch size=1) has a volatile validation loss per epoch, with slow convergence. Therefore, we choose a batch size of 64 to train the CAEs.[‖]

We train the QAE with analogous architecture for a six-dimensional input for the second scenario for a two-dimensional latent space in a similar fashion for all training sizes. For the CAE keeping the number of nodes and layers identical to the previous case for six-dimensional input and output vectors, we perform a hyperparameter scan, the details of which is given in Appendix E. All results shown in the next section for this scenario is for the best performing hyperparameters.

7.4 Results

Results of the various training scenarios are presented in this section. We present a detailed investigation of the QAE and CAE's properties for the $t\bar{t}$ background scenario in Sections 7.4.1 to 7.4.4. The lessons learnt from these analyses, particularly the training size independence and the relative performance, are then tested for the invisible Z background in Section 7.4.5.

7.4.1 Dependence of test reconstruction efficiency on the number of training samples

The distribution of the loss function of the independent background test samples for different training sizes of the CAE is shown in figure 7.3. Although training with a single data point is inherently inaccurate, we perform such an exercise as a sanity check of the CAE's comparison to a QAE. The test distribution shifts

[‖] The network performs one update per epoch for training with a number of samples less than 64. These training sizes are too small for a CAE to have any good learning capability. Hence, we do not try to modify the batch sizes or interpret the test distributions.

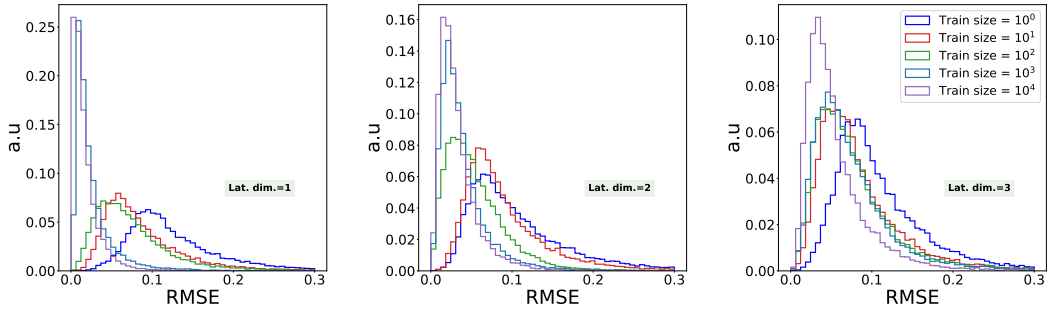


Figure 7.3: Loss distribution of the test background samples (15k) for different sizes of training dataset. We can see that the distribution shifts significantly towards the left (direction with lower loss) as one increases the training data size, which reflects that there is noticeable incraase in learning with larger data samples.

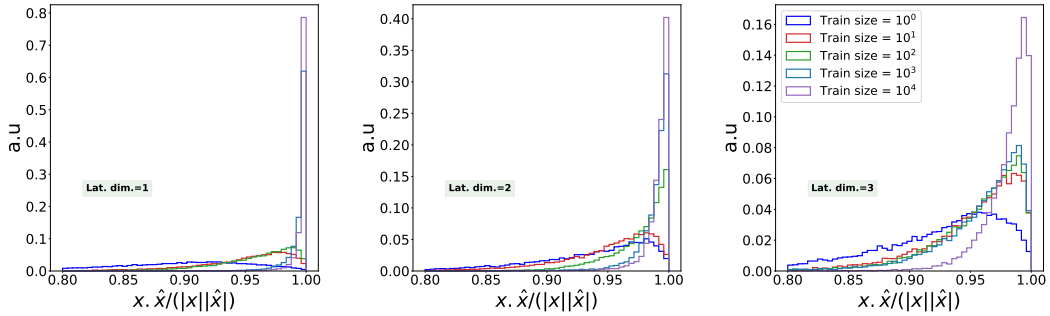


Figure 7.4: Cosine similarity (analogous to quantum fidelity) distribution of the test background samples (15k) for different sizes of training dataset of the CAE.

towards the left as one increases the training size, thereby signifying increased reconstruction efficiency. For training sizes of up to 10^2, the limited statistics will produce a very high statistical uncertainty. Since it is not the main emphasis of our present work, we do not comment any further. Looking at the distribution across different latent dimensions for 10^3 and 10^4 training samples, one can see the impact of the information bottleneck. For a singular latent dimension, the passed information is already available from 10^3 samples, and hence the loss distribution is very close to the one trained on 10^4. This relative separation increases as we go to higher latent dimensions, denoting the higher information passed to the decoder to reconstruct the input, which is exploited with higher training samples. For an analogous comparison with the quantum fidelity, we define the cosine similarity between the input vector $\mathbf{x}$ and the reconstructed vector $\hat{\mathbf{x}}$ as,

$$\cos\alpha = \frac{\mathbf{x}.\hat{\mathbf{x}}}{|\mathbf{x}|\ |\hat{\mathbf{x}}|}\ , \tag{7.14}$$

where the dot product is done with a Euclidean signature. The distribution of the cosine similarity shown in figure 7.4, shows similar features to the loss function's

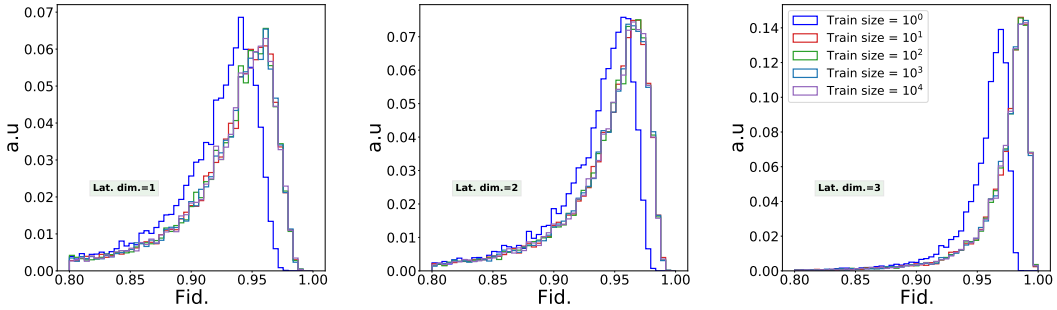


Figure 7.5: Fidelity distribution of the test background samples (15k) for different sizes of training dataset. The peak shifts towards the right in analogy to the CAE, however the shift is not as pronounced. With a single training sample, the network is not able to converge completely while for anything greater than 10, the increase in training size has practically no effect.

distribution, with efficient reconstruction possible only when the train size is at least 10^3.

We have seen that CAEs cannot be trained with limited statistics to reconstruct the statistically independent test dataset. From the distribution of the test sample's fidelity in figure 7.5, we see that QAEs are much more effective in learning from small data samples. Although training with a single data point has not reached the optimal reconstruction efficiency, it is obtained with ten sample events. Unlike CAEs, see figures 7.3 and 7.4, the test fidelity distribution for all latent spaces are identical for training sizes greater than or equal to ten. The independence of the sample size is particularly important in LHC searches where the background cross section is small. This particularly interesting feature may be due to the interplay of an enhancement of statistics via the uncertainty of quantum measurements and the relatively simple circuits employed in our QAE circuit. For a single input point and assuming that we have hardware capable of building exact copies, a finite number of measurement processes always introduces a non-zero uncertainty in the network output. This uncertainty can act as additional information in the quantum gradient minimisation, which is performed after the measurement process, increasing the convergence for smaller data samples. Moreover, existing studies [374, 375] show the advantage of quantum machine learning over classical approaches. Additionally, the use of quantum gradient descent [351] makes the loss landscape more convex, thereby speeding up convergence.

7.4.2 Classification Performance

We compare the QAE and the CAE's performance for the four-dimensional input feature space. The metrics used in this presentation bear similarity to those used in a supervised framework. It also assumes that a randomly chosen event

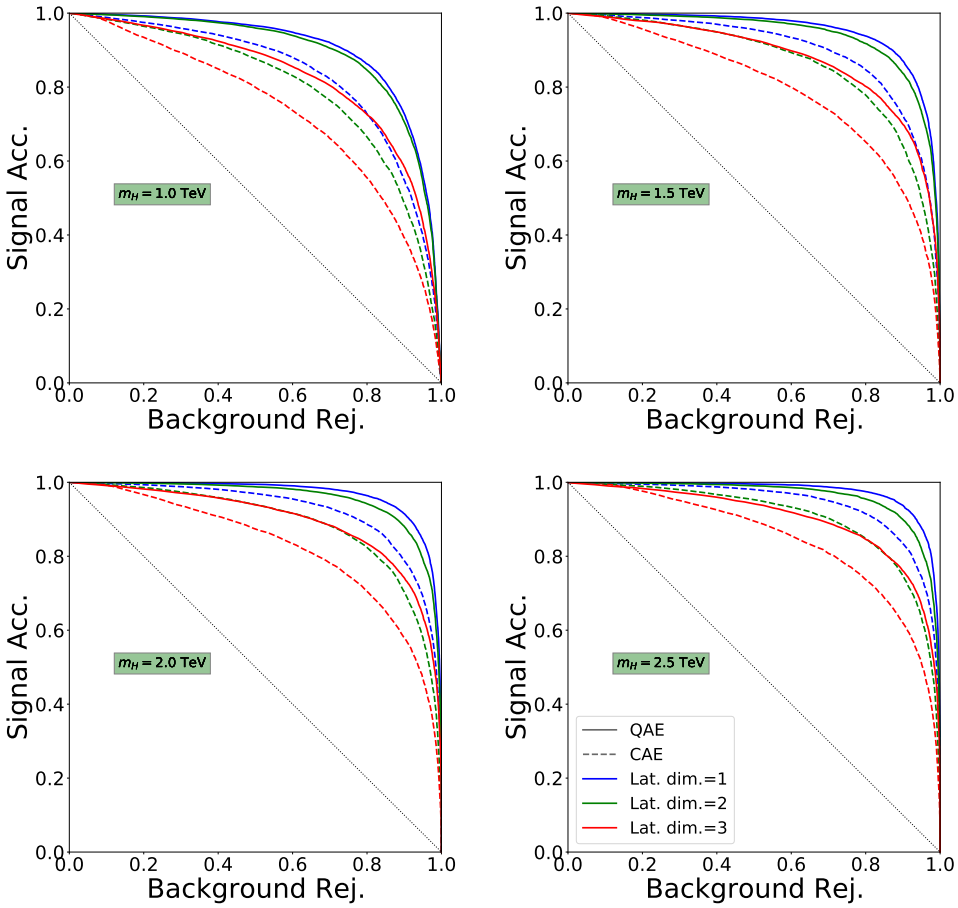


Figure 7.6: ROC curve between signal acceptance vs background rejection for Quantum Autoencoder (QAE) and Classical Autoencoder (CAE) for various values of m_H and different latent dimensions for a training datasize of 10k samples. The trend across latent dimensions is same for both QAE and CAE with QAEs performing better in all cases.

is equally likely to be either the background or the signal. This assumption is not sound in the context of LHC searches or in an anomaly detection technique since the background's cross-section is orders of magnitude larger than that of the signal. Nevertheless, they are handy when comparing different classifiers.

For each value of m_H, we plot the Receiver-Operator-Characteristics (ROC) curve between the signal acceptance and the background background rejection in figure 7.6, for the networks trained with 10k samples. The black dotted lines denote the performance of a random classifier with no knowledge of either the signal or the background, and the lines further away from it indicate better performance than those in its vicinity. The performance reduces with increasing latent dimensions for CAEs and QAEs, with the highest background rejection coming for a singular latent dimension. Comparing the QAEs and the CAEs (dotted vs solid lines for each colour), we find that QAEs perform better than CAEs consistently in all latent dimensions and the different values of m_H. This

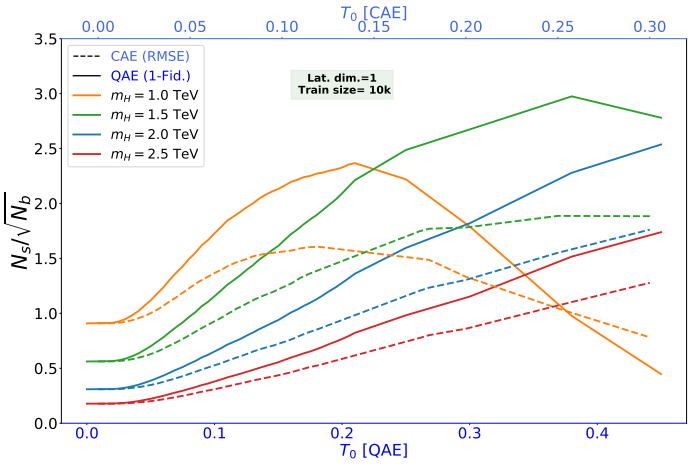


Figure 7.7: Significance as a function of the threshold T_0 on the fidelity and root-mean-squared-error (RMSE) of the QAE and the CAE, respectively, for each of the signal scenarios and singular latent dimension trained on 10k samples. To keep the signal rich region on the right side for both, we have used $(1 - \text{Fidelity})$ for the QAE. We fix the cross-section of all signals to 10 fb, and evaluate the yields at an integrated luminosity of 3000 fb^{-1}.

better performance may be a universal property of QAEs. However, as our analysis is a proof-of-concept, an in-depth exploration of the properties of QAEs in general and anomaly detection at colliders, in particular, is needed to affirm this observation.

7.4.3 Anomaly detection

We now explore the performance of the autoencoders in a semi-realistic search scenario. When we scale the normalisation of the signal and the background by their respective probability of occurrence, i.e. their respective cross-sections, we are essentially in an anomaly detection scenario since the background is orders of magnitude larger than the signal. The performance of the autoencoders can then be quantified in terms of statistical significance as a function of the threshold applied on the loss. For the background, we scale the cross-section obtained from Madgraph by a global k-factor of 1.8 [376], while for all the signal masses, we fix a reference value of 10 fb. The yield is then calculated as

$$N_p = \epsilon_p \, \sigma_p \, L \, E_p(T_0) \quad ,$$

where ϵ_p is the baseline selection efficiency, σ_p the cross-section, and $E_p(T_0)$ the efficiency at a threshold T_0 of the loss distribution, for a process p, while L is the integrated luminosity which we take to be 3000 fb^{-1}.

Since it is natural to use the best classifier in a search, we evaluate the significance of the autoencoders with one latent dimension, trained on 10k samples. We

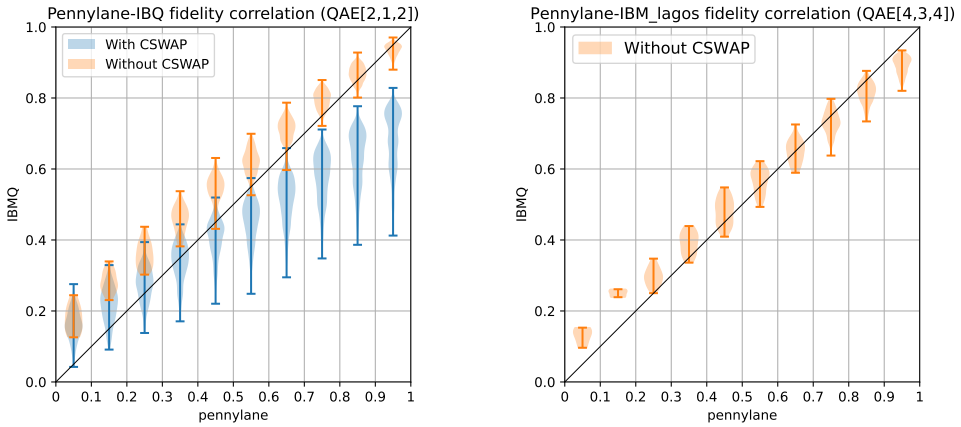


Figure 7.8: The correlation between the fidelity values obtained by `Pennylane` and by the IBM-Q backends. On the left we show the comparison of a 2-1-2 QAE, where we directly measure the trash state (orange) and with a SWAP test employing a CSWAP gate. We find that the shallower implementation without the CSWAP gate has lesser decoherence effects, and hence better agreement with the simulation. The correlation with the direct measurement for the 4-3-4 case is shown on the right.

apply the threshold for the QAE and the CAE on the quantum trash state fidelity and the RMSE loss, respectively. We use $(1 - \text{Fidelity})$ for the QAE to make the signal-rich regions same in both scenarios. RMSE loss is chosen over the cosine similarity since the former was found to have a higher performance. The significance $N_S/\sqrt{N_B}$ for each of the signal masses as a function of the threshold T_0 is shown in figure 7.7. We fix the threshold range so that there are enough background test statistics in the least background like bin. Looking at the peak of the significance, we note that QAEs outperform CAEs, which is only natural from the preceding discussions. However, an interesting development is the relative performance for the different masses. Even though the ROCs indicated higher discrimination with increasing mass, the significance increases for $m_H = 1.0$ TeV to 1.5 TeV and decreases for higher masses. Since we have fixed a fiducial cross-section for each signal mass, it plays no role in this irregularity. The trend arises via an interplay between the higher discrimination by the autoencoder output and the decrease in baseline efficiency with increasing mass m_H. The decreasing selection efficiency is due to the isolation criteria of the jets and the leptons, which would be naturally boosted when we go to higher resonant masses m_H, thereby becoming more collimated.

7.4.4 Benchmarking on a quantum device

We now compare the performance of the quantum simulator and the actual quantum hardware. Since there is a limitation on the available number of qubits, we limit the feature space in two dimension, which consists of $\{p_T^{b_1}, p_T^{l_1}\}$. For our QAE setup, in addition to the two qubits for embedding the input features, one qubit

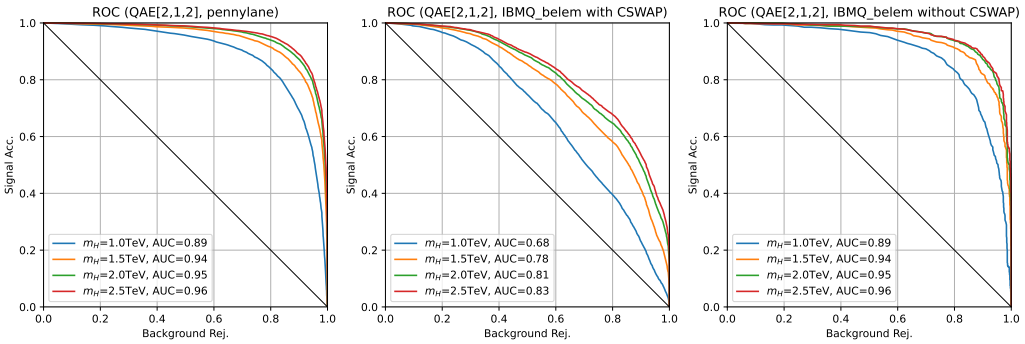


Figure 7.9: ROC curves based on the fidelity distributions. Those evaluated by the `Pennylane` simulator (left panel), by the quantum device IBM-Q belem backend with the SWAP test (central panel), and with the second qubit measurements (right panel) are shown.

for the reference state and another ancilliary qubit for the SWAP test are needed. We use the simpler version of the quantum circuit shown in figure 7.2, which is implemented and trained using `PennyLane`. To compare the performance, we use the same circuit with the same optimised parameters both for `PennyLane` and for the IBM-Q belem backend. Accessing the IBM hardware was done through `Qiskit`.

In figure 7.8, we show the fidelity distributions for the background and the signal samples for our QAE circuit with the optimised circuit parameters computed by the simulator in `Pennylane` and in the actual quantum device of IBM-Q belem backend. The plot shows the shape of the distribution (denoted by the width of the shaded region) in the y-axis for each bins of size 0.1 in the x-axis (plotted at each bin center). The lines at each ending denote the range of the data of the y-axis. Since IBM-Q does not have a shallow implementation of the CSWAP operation, the fidelity distributions are smeared toward 0.5, and especially it is worse around 1. One of the advantages of using the SWAP test is to reduce the number of qubits for the evaluations of the fidelity during the optimisation process. For example, to check the performance of the current circuit, directly measuring the fidelity between the reference state and the output for the second qubit would be enough. It can be achieved by the simple Pauli z measurements. The correlation of the fidelities obtained by `Pennylane` and by IBM-Q belem, based on the SWAP test and on the Pauli z measurement are shown in the right panel as the violin plots, in blue and in orange, respectively. The correlation is better for the Pauli z measurements for the same circuit part with the identical input parameters. It suggests that the decoherence effects from a deeper circuit obscure the performance.

In figure 7.9, we show the ROC curves based on the fidelity distributions for the background and the signal samples evaluated by `Pennylane` simulator in the left panel. The central panel shows the ROC curves based on the fidelities evaluated by the SWAP test, while the right panel shows those by the second

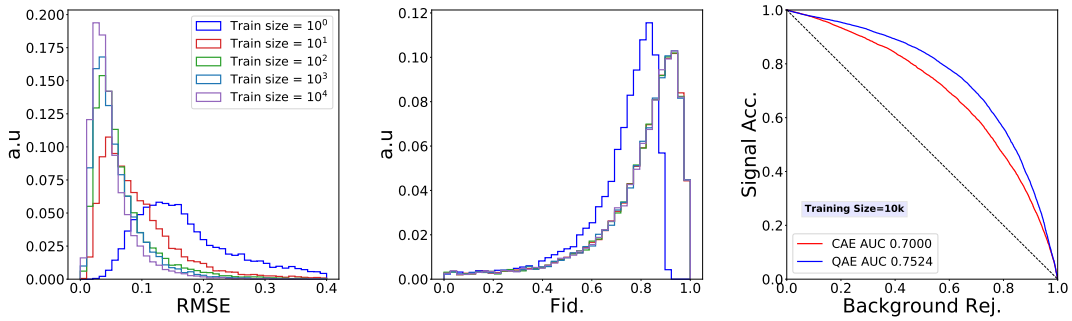


Figure 7.10: The test distribution of the invisible Z background scenario for different training sizes of a CAE (left) and a QAE (center) for a two-dimensional latent representation, and their respective ROC curve (right) for the training done with 10k events. Similar to the previous case, the QAE has converged with much smaller datasets than the CAE. Moreover, the QAE performs relatively better than the CAE for the particular signal.

qubit Pauli z measurements, for the same IBM-Q device of belem backend. As one can see, the performances based on the Pauli z measurements on the IBM-Q device follow those obtained by the `Pennylane` simulator. The AUCs for them are also essentially the same. Thus, the deficit in the performance with the SWAP test is due to the too deep circuit realisation for the CSWAP operation in the IBM-Q device. Therefore, the realisation of a CSWAP operation with a shallow circuit is necessary.

To check the efficacy of quantum hardware for the four input QAE, we evaluate the trash state fidelity of a QAE with four-dimensional input features. Due to hardware limitations discussed above, we estimate it without the SWAP test for a single trash qubit giving us a three-dimensional latent representation. The correlation between the `Pennylane` evaluated fidelity and the output from IBM-Q lagos, shown in figure 7.8, displays a good agreement between the simulation and the hardware.

7.4.5 Comparative training efficiency and performance for invisible Z background

We have seen that a QAE trains efficiently and performs better than a CAE in a hypothetical resonant signal scenario. To gauge how these important behaviours carry over to a different process, we study the training size dependence and performance of a QAE and CAE for an invisible background (and signal), detailed in the last paragraph of Section 7.3.1 for a two-dimensional latent space. Note that all the results for the CAE are for the best model chosen after a hyperparameter scan described in Appendix E.

The loss distribution of the test dataset for the background for different sizes of training data and their ROC curve for the case of 10k training samples are

shown in figure 7.10. The characteristics are similar to the previous scenario, giving further evidence that the training efficiency of the QAE is not limited to a specific kind of process. Moreover, from the ROC and the AUC value, we see that the QAE also performs better than the CAE. This superior performance is particularly noteworthy given that the CAE's hyperparameters has been chosen after a hyperspace scan restricted to a fixed width and depth.

7.5 Summary

The lack of evidence for new interactions and particles at the Large Hadron Collider has motivated the high-energy physics community to explore model-agnostic data-driven approaches to search for new physics. Machine-learning anomaly detection methods, such as autoencoders, have shown to be a powerful and flexible tool to search for outliers in data. Autoencoders learn the kinematic features of the background data by training the network to minimise the reconstruction error between input features and neural network output. As the kinematic characteristics of the signal are different to the background, the reconstruction error for the signal is expected to be larger, allowing signal events to be identified as anomalous.

Although quantum architecture capable of processing huge volumes of data is not yet feasible, noisy-intermediate scale devices could have very real applications at the Large Hadron Collider in the near future. With the origin of the collisions being quantum-mechanical, a quantum autoencoder could, in principle, learn quantum correlations in the data that a bit based autoencoder fails to see. We have shown that quantum-autoencoders based on variational quantum circuits have potential applications as anomaly detectors at the Large Hadron Collider. Our analysis shows that for the scenario we consider, i.e. the same set of input variables, quantum autoencoders outperform dense classical autoencoders based on artificial neural networks, asserting that quantum autoencoders can indeed go beyond their classical counterparts. They are very judicious with data and converge with very small training samples. This independence opens up the possibility of training quantum autoencoders on small control samples, thereby opening up data-driven approaches to inherently rare processes.

Chapter 8

Summary and future directions

The Large Hadron Collider (LHC), to date, is the most sophisticated machine built to probe physics at the sub-nuclear length scales. It generates huge amounts of data that are stored all over the world in the worldwide LHC computing grid. Therefore, the physics goals of the LHC present in itself a humongous data analysis task. The particle physics community has been using machine learning techniques in data analysis for decades. However, with the advent of modern deep-learning algorithms propelled by the wide availability of high-end GPU acceleration and a rich research ecosystem unearthing state-of-the-art algorithms and architectures, there is an unprecedented increase in applying such algorithms at various stages of the analysis pipeline.

The nature of investigations in fundamental physics presents a unique situation where one cannot simply use these deep-learning algorithms as a black box without understanding their behaviour in various working conditions. Although their universal approximation capabilities provide innovative ways of solving various problems of interest, their flexibility, high dimensionality, and the numeric nature of finding effective approximators impede a complete understanding of their workings. Understanding their behaviour is paramount for phenomenological applications at the Large Hadron Collider, which heavily depends on the accurate, but ultimately approximate simulation of Quantum Chromodynamics at different energy scales extensively validated on experimental data. They take in the four-vector information of the measured particles, bypassing and (most of the time) improving over the use of distinct variables created by extensive domain knowledge and physics intuition. This thesis studies some aspects of utilising deep-learning algorithms for phenomenological searches at the Large Hadron Collider.

In the first study (chapter 3), we investigated the improvement in the searches of invisible Higgs decays produced via Vector Boson Fusion (VBF) with Convolutional Neural Networks (CNNs). Such processes have a different radiation pattern from most QCD backgrounds due to the absence of colour exchange between the two protons in the underlying hard partonic process. The upper bounds on the invisible branching ratio of the Higgs boson, which is very important for various

Higgs-portal dark matter scenarios, is still much higher than the one expected in the Standard Model. We find that CNNs can put much stricter constraints on this branching ratio, outperforming both univariate shape analysis of single variables and multivariate analysis of high-level observables.

Deep-learning algorithms can utilise minute differences in the data; therefore, it is imperative to scrutinise the phenomenological implications of various aspects of the simulation. For the VBF production, we study the dependence of the CNN's performance on the recoil scheme used in the parton shower and the perturbative accuracy of the matrix-element simulation of the hard process in chapter 4. We find that the testing performance is dependent on the signal used during the training. However, this dependence is mild for the next-to-leading-order simulated signal suggesting that the CNNs can partially understand the better stability of higher-order simulations from the tree level simulations.

The result above indicates that neural networks, although highly expressive, are not well understood from a physical perspective. As a step toward resolving these issues, we build an infrared and collinear safe graph neural network algorithm in chapter 5. This network performs comparably to state-of-the-art IRC unsafe algorithms on standard top tagging datasets.

In the remainder of the thesis, we take a different approach to signal-dependent classification scenarios and study unsupervised anomaly detection models trained only on the background for wider model-independent search strategies relying on the background-only hypothesis. Such model-independent search strategies are important given that direct searches for various well-motivated new physics scenarios have been inconclusive. In chapter 6, we devise a graph autoencoder capable of learning inductive jet graph representations by utilising edge reconstruction networks. We find that they can recognise higher-multiplicity decays captured within large radius jets when trained to only reconstruct QCD jets. In chapter 7, we compare the capabilities of a quantum autoencoder based on variational quantum circuits against a bit-based autoencoder with the same set of features. The quantum autoencoder, augmented with quantum gradient descent, converges with very small datasets and considerably outperforms classical autoencoders for two benchmark signal scenarios.

The influx of modern deep-learning methods into high energy phenomenology is not restricted to LHC applications but to other allied fields like neutrino physics [377–379], cosmology [380–383], gravitational waves detection [384–386], and dark matter [387, 388] searches. Due to their superior performance in many regards, some of which we have touched upon in this thesis, and with community-wide interest in their inner workings, their reach will expand further in the coming years. However, applications in fundamental physics demand a level of rigour and understanding beyond their formal universal approximation capabilities.

For instance, a practical aspect of importance in experimental analyses with wide phenomenological implications is the propagation of systematic uncertainties [163, 389–392] when using deep-learning algorithms. These uncertainties can

be of theoretical origin like the factorisation and renormalisation scale uncertainty in the simulation of the events. Rather than relying only on data-driven methods to quantify such systematics, it is important to look into the inner workings of deep-learning algorithms to understand how exactly these theoretical considerations affect the nature of the converged minima in the weight space. Doing so could not only unearth ways of reducing the uncertainties but also deepen our understanding of deep-learning algorithms in a more general setting.

A related and highly promising avenue for the near future is using quantum computing technology to bolster machine learning models. Existing noisy-intermediate-scale-quantum devices already show favourable performance against their classical counterparts, one of which we found in our study of quantum autoencoders. The technological advancements in quantum noise corrections would increase near-term devices' overall stability and usability. There are various unexplored avenues in using such quantum devices for machine learning tasks and the efficient simulation of the inherently quantum mechanical collisions at the Large Hadron Collider.

Although the studies in the thesis have focussed on the backdrop of signal searches based on data simulated using traditional techniques, these conventional techniques can be augmented using deep-learning methods. With the future high-luminosity phase placing huge computational requirements [393–395] overall, there is already a considerable effort in inspecting machine learning-based alternatives for the more computationally intensive part of the simulation pipeline. One such stage is the generation of parton-level events traditionally done using Monte-Carlo methods. Along with reducing the computational requirements [396–398], there is also considerable focus on improving the precision and robustness [399–401] of the generated data using novel techniques.

The development of powerful and data-hungry deep-learning algorithms has propelled the nature of mainstream scientific inquiry into a data-driven era. This situation has percolated into fundamental physics research, with a multifaceted effort within various fields to utilise big data to deepen our understanding of the universe. Nevertheless, applying such techniques as a black box could (and most likely would) yield spurious results. Due to the extreme mathematical complexity of such algorithms, it is important to ascertain their behaviour under various extreme conditions.* Therefore, a pragmatic optimism toward deep-learning algorithms in the community should lead to fertile progress of fundamental science in the information age.

*In the sense of statistical outliers as well as physics scenarios one could consider

Appendix A

Finite mass effect of top quark in gluon-fusion events

We generate the gluon-fusion production of the Higgs boson used in the study of chapter 3 by using the Higgs Effective Field Theory (HEFT) model, where the interaction of the Higgs boson with gluons is approximated by an effective vertex calculated by taking the top-quark mass to infinity. This is a reasonable approximation only when all relevant scales in the physical process are less than $2\ m_t$. The distribution of p_T of the Higgs boson (equivalently MET with detector effects introduced via Delphes) has a significant portion of events in regions where the approximation is not valid. We remove this inconsistency by reweighting the MET distribution of the events obtained after Delphes. We extract weights (ratio of the full SM results to HEFT) and bins in p_T of the Higgs for the present final state topology from figure 30 on reference [267]. Each event is then assigned the corresponding weight of the bin of its MET. After reweighting the events, we apply the preselection-cuts and extract the cut efficiency using the weights.

Since we need unweighted events for the neural network training, the passed events are again unweighted. This is done in the following steps. We divide all events into sets with unique weights. This is nothing but grouping the events into the extracted bins in MET. We get mutually exclusive subsets of events $\mathcal{S}_i$, with i being the bin-index. The per-bin weights are divided by their maximum value. We get a weight $w_i \in (0, 1]$ for each $\mathcal{S}_i$. From each set $\mathcal{S}_i$, we randomly choose w_i proportion of events rounded to the closest integer. We show in figure A.1, the distribution of some kinematic-variables of the three datasets: unweighted events generated with HEFT model, weighted events with finite-top mass effects, and unweighted events used in neural network training. The effect of rounding to the nearest integer is seen in the later bins in MET, where the statistics are weaker due to fewer events.

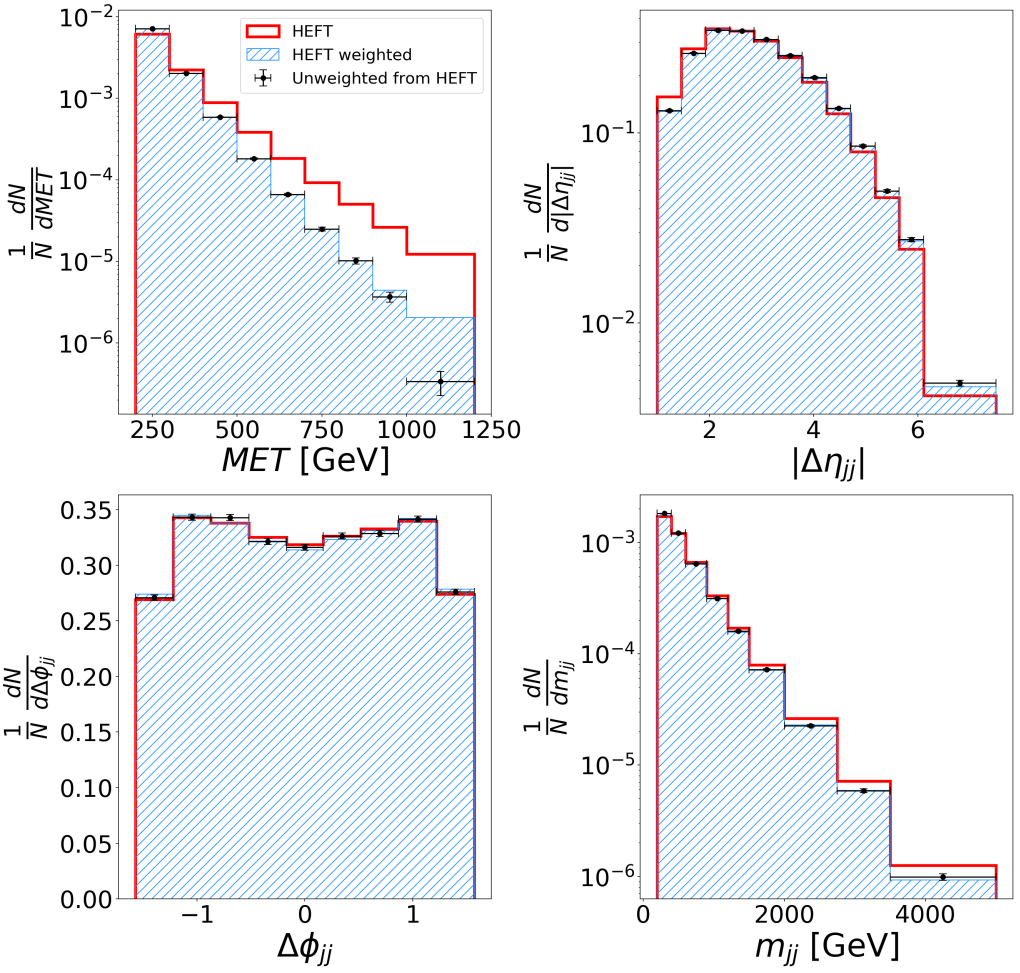


Figure A.1: Comparative distribution of kinematic variables for HEFT, weighted with finite-top mass effects and unweighted distributions for passed events used in deep-learning training and validation.

Appendix B

Characteristics of High-level variables

In this appendix, we take a closer look at the high-level variables, especially the $\mathcal{R}$ variables defined in eq. 3.1 in chapter 3. A key element in the extraction of variables belonging to the two spaces $\mathcal{K}$ and $\mathcal{R}$ is that the $\mathcal{K}$ variables are functions of four-momenta of reconstructed objects while the $\mathcal{R}$ variables are functions of four-momenta of tower-constituents (in our case from the Tower class of Delphes). The $\mathcal{R}$ variables do not take into account the tower-resolutions in the strict sense. This may point to a further reduction in the performance of ANNs compared to CNNs, where the tower-resolutions are better modeled.

We show the signal vs background distribution of all $\mathcal{R}$ variables in figure B.1. The contribution of S_{EW} and S_{QCD} to the total signal is stacked. The separation, as defined in eq. 3.2, are shown for these variables for the total signal (also, S_{EW}) and background in figure B.2. We can see that the trends in the distribution are in accordance with their respective values of separation. The shape of S_{QCD} and the background distributions are similar for all values of η_C, and the overall differences, if any, comes from the contribution of S_{EW}. The separation is minimal and remains constant for $\eta_C > 4$. This can be attributed to the fact that above these values, almost all of the calorimeter hits contribute to $H_T^{\eta_C}$. It increases continuously up to $\eta_C = 1.8$ and then decreases till $\eta_C = 1.0$. The increase is expected from the VBF topology, while the decrease can be attributed to the smallness of the region $[-\eta_C, \eta_C]$.

In figure B.3, we show the remaining kinematic variables not shown in figure 3.6. As can be seen, there is not much discriminatory information in any of these variables: ϕ_{MET} is uniform for all channels since the beams are unpolarised, while $\Delta\phi^J_{\text{MET}}$ ($J \in \{j_1, j_2, j_1 + j_2\}$) has most contributions around $\pm\pi$, due to the imposed separation of two jets $\Delta\phi_{jj}$ and momentum conservation in the recoil of quarks/gluons against heavy bosons ($W^\pm$, Z^0 and h^0).

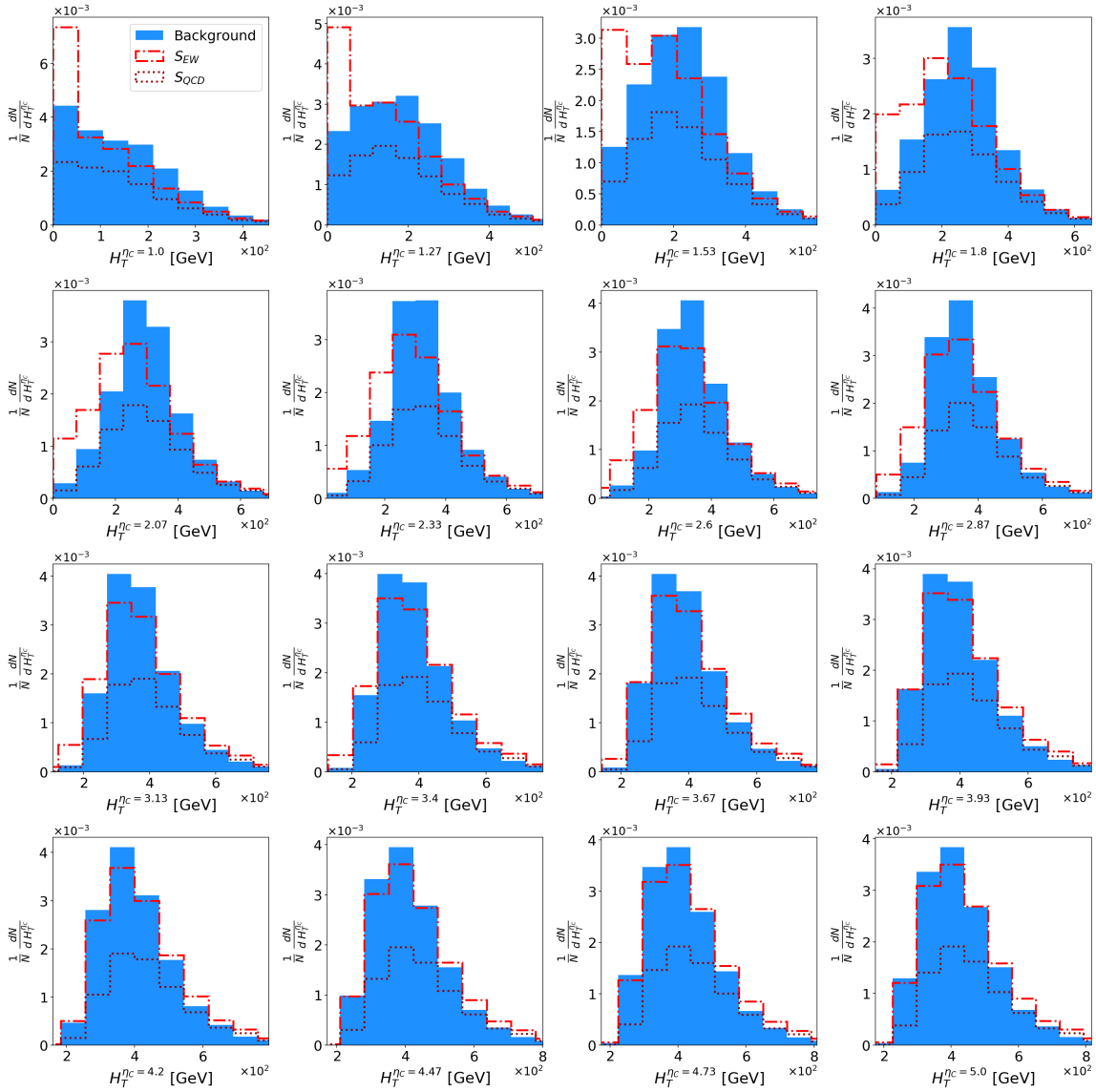


Figure B.1: Signal vs Background distribution for all $H_T^{\eta_C}$ variables. We can see that for higher values of η_C the signal and background are not that different and the difference grows as we approach the cut value of η cut.

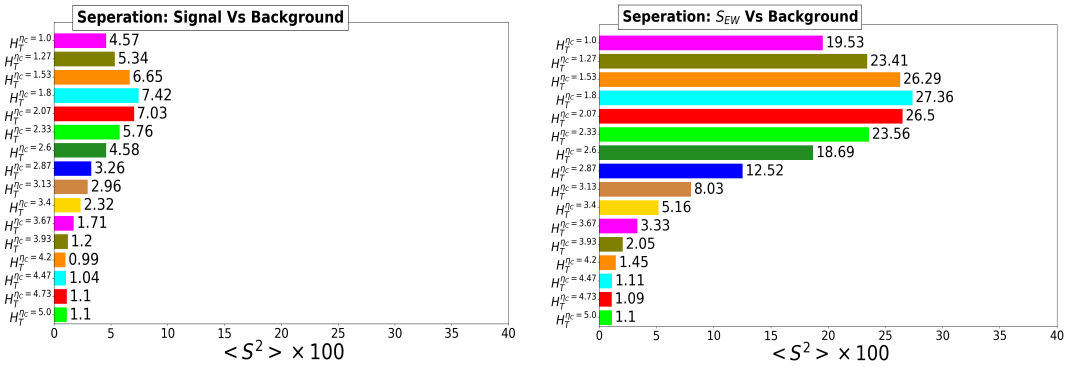


Figure B.2: Seperation of all $H_T^{\eta C}$ variables for (left) signal vs background and (right) S_{EW} vs background. These have been calculated with 25000 events for each of the three datasets with the same binning. We can see that the presence of S_{QCD} significantly reduces the discriminating power of $H_T^{\eta C}$ variables on the left.

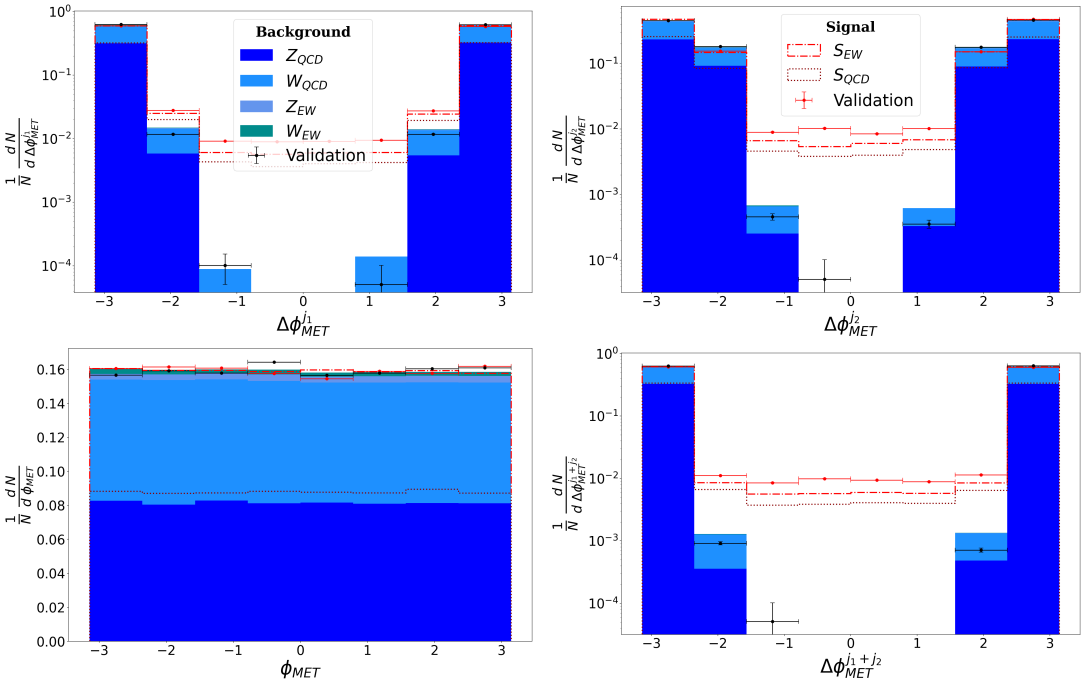


Figure B.3: Signal vs Background distribution of the high-level kinematic variables excluded in figure 3.6

Appendix C

Correlation between High-level variables and network-outputs

Salient features of the correlation of important variables with all neural network outputs have been given in the chapter 3 (figure 3.14). We examine the correlation of the ANNs with their inputs in this section. All correlations have been calculated using the inbuilt function in NumPy(v1.17.2) [402].

In figure C.1 we show the correlations amongst the $\mathcal{K}$ variables including the $\mathcal{K}$-ANN network output for each class. As expected, the $\mathcal{K}$-ANN output is highly correlated with the two most discriminating variables $|\Delta\eta_{jj}|$ and m_{jj}. The next highest correlation with $\mathcal{K}$-ANN is found to be with MET for background and $|\Delta\phi_{jj}|$ for signal. Except for $|\Delta\phi_{jj}|$, all other ϕ variables are almost uncorrelated with $\mathcal{K}$-ANN for both classes. The uniformity of ϕ_{MET} results in its negligible correlation with all other variables. In the correlation among $\mathcal{K}$ variables, we can see two distinct sets of variables with comparatively moderate to high correlations formed amongst $\{|\Delta\eta_{jj}|, m_{jj}, \text{MET}\}$ and $\{\Delta\phi^{j_1}_{\text{MET}}, \Delta\phi^{j_2}_{\text{MET}}, \Delta\phi^{j_1+j_2}_{\text{MET}}\}$. In the first set, $|\Delta\eta_{jj}|$ and m_{jj} are almost completely correlated since, the angular opening between two four vectors $p^{\mu}_{j_1}$ and $p^{\mu}_{j_2}$, determine the invariant mass $m_{jj} = (p^{\mu}_{j_1} + p^{\mu}_{j_2})^2$. The MET shows a moderate correlation with both $|\Delta\eta_{jj}|$ and m_{jj} as momentum conservation forces $|\vec{p}_{j_1} + \vec{p}_{j_2}|$ to be higher for higher MET. The correlation amongst the second subset can also be explained by transverse momentum conservation in the collision, with contamination from subsidiary QCD radiation and detector effects.

The class-wise correlations amongst the outputs of $\mathcal{R}$-ANN and $\mathcal{H}$-ANN along with six variables from $\mathcal{R}$ with high separation, and the two kinematic variables $|\Delta\eta_{jj}|$ and m_{jj} are shown in figure C.2. As expected, we see that the $\mathcal{R}$ variables are highly correlated with one another, which decreases with increasing distance in η_C. Another highlight is the negative correlation between them and the kinematic variables. It can be understood if we recall that the dominant radiation in the tower comes from the two leading jets, and an increase in $|\Delta\eta_{jj}|$ will decrease the calorimeter hits in the central regions. In the case of correlations between

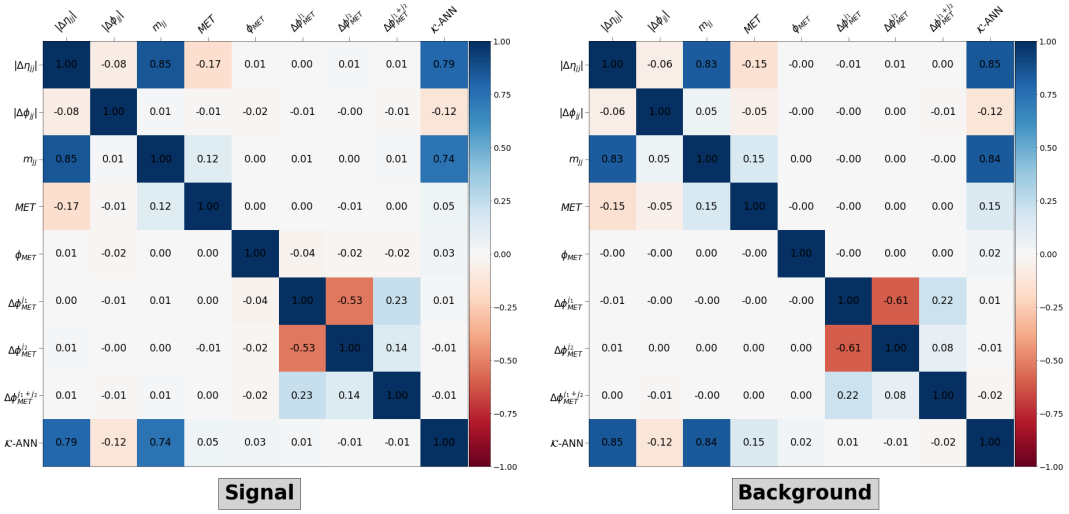


Figure C.1: Correlation between the high-level kinematic variables $\mathcal{K}$ and the network-output of $\mathcal{K}$-ANN for (left) signal and (right) background.

neural-network outputs and their respective inputs, the sign of the correlation is not much relevant for binary classification due to the probabilistic interpretation of the outputs y_i: $y_0 + y_1 = 1$ and $y_i > 0$. On the contrary, the relative difference in sign and magnitude in correlations between the different input features and the output is relevant. In the case of $\mathcal{H}$-ANN, we can see that in terms of both magnitude (importance as plotted in figure 3.7) and sign (as discussed here), the relations amongst $\mathcal{K}$ and $\mathcal{R}$ variables are carried over to their corresponding correlations with the network-output.

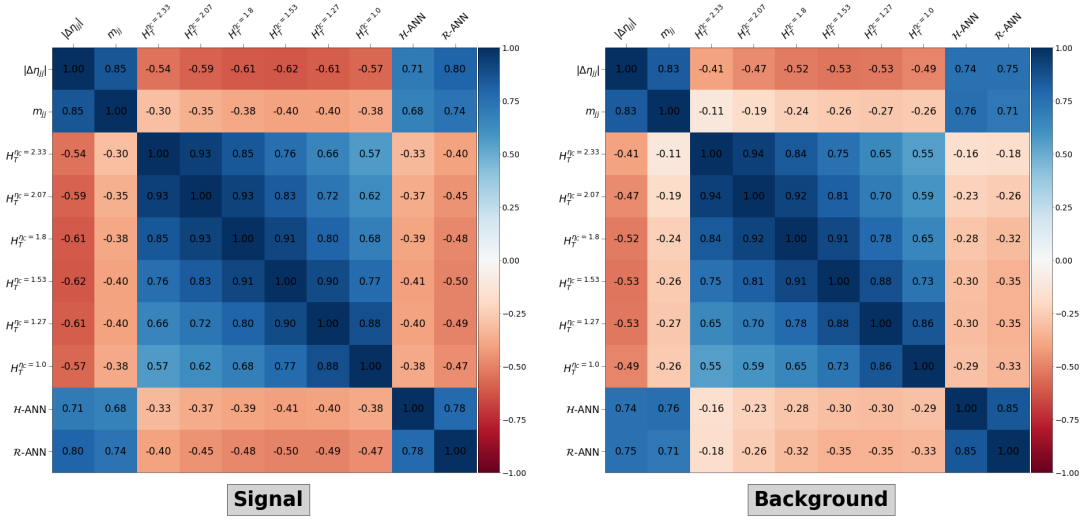
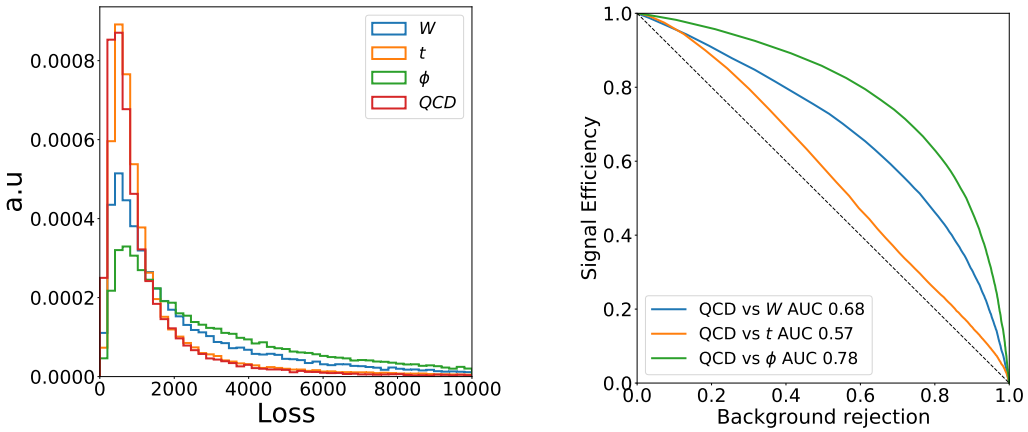


Figure C.2: Correlation between the high-level variables $\mathcal{H}$ and the network-outputs of $\mathcal{R}$-ANN and $\mathcal{H}$-ANN for (left) signal and (right) background. For better representation we have chosen variables with non-negligible correlations with the network outputs.

Appendix D

Comparison with Particle Graph Autoencoder



(a) (b)

Figure D.1: Distribution of the loss function of the PGAE (a) and the corresponding ROC curves (b) for the different signal classes for a network trained only on QCD jets.

We compare the performance of the network proposed in chapter 6 with the particle graph autoencoders (PGAE) proposed in reference [320]* with our dataset. This study focussed on identifying anomalous events with dijet signatures (large-radius jets and high p_T) and used the two leading jets in the event to learn latent event representations. In contrast, our present focus is jet-level classification. For the input, we consider the four-vector of each microjet as the node feature and use a complete graph with all possible connections. The network is a graph-autoencoder that takes the vectors as input with a single edge convolu-

*We use the code available in https://github.com/stsan9/AnomalyDetection4Jets/tree/rnd_cuts

tion to map it to a two-dimensional latent node representation and maps it back with another single edge convolution. We use the mean squared error as the loss function and train with a batch size of 32. For more details of the architecture, we refer the reader to Section 3.7 of reference [320]. We show the distribution of the loss function and the corresponding ROC curve in figure D.1. The first thing that we notice is that the location of the peaks is identical for all four classes, with the only difference coming in the tail of the distribution. The value of the AUCs is significantly reduced for W and top jets compared to our work, while for ϕ-jets, the reduction is not that drastic. Out of the three signal classes, ϕ jets are the least QCD-like, and hence, the networks find it easier to distinguish them with less information. At the same time, the edge-reconstruction employed in our architecture helps identify the W and top jets more efficiently. Hence, we infer that the edge-reconstruction and the multidimensional edge feature representation is crucial for a graph-autoencoder as these complex and physically relevant features are not learned by the network even though they are, in principle, constructed from the node features. Moreover, using only the node features, the graph autoencoder is insensitive to the n-prong structure of the signals as the AUCs do not follow the usual QCD intuition. The addition of the edge features and their reconstruction enables the graph-autoencoder to learn the signal jets' n-prong topology in an unsupervised manner.

Appendix E

Details of hyperparameter scan for Classical autoencoder

Sl. no.	Hyper Parameter	Value Space	Best value
1.	Activation function	tanh, ReLu, Sigmoid, Linear	ReLu
2.	L1 Regularisation	0,0.1,0.01,0.001,0.0001	0
3.	L2 Regularisation	0,0.1,0.01,0.001,0.0001	0
4.	Dropout	0,0.1,0.2,0.3	0
5.	Learning Rate	0.01,0.001,0.0003	0.0003
6.	Batch Size	32,64,128,256,512,1024	64

Table E.1: The table shows the different values of the hyperparameters and their best values after the scan.

The details of the hyperparameter scan of classical autoencoder with six-dimensional inputs and outputs are given in this appendix. We use the `RandomSearch` algorithm implemented in `KerasTuner` [403] for the scan. The number of nodes in the hidden layers of the encoder is kept fixed to 20, 15, and 10. With a (fixed) two-dimensional latent space, we use a symmetric decoder setup. Once the skeleton of the architecture is fixed, we scan over the activation function of the layers, L1 regularisation and L2 regularisation of the weights, the dropout value between two successive layers, and the training's learning rate and batch size. Their respective values along with the best one chosen for the final training are given in table E.1. The best value of the hyperparameters are from thousand trials trained for hundred epochs, and the training is terminated if the validation loss does not improve for ten epochs (implemented as the `EarlyStopping` callback during training).

We do not vary the width or the depth to compare the capabilities of CAEs with at least some degree of comparability to the simple QAE used in the study. Increasing the width and depth will undoubtedly increase the expressive power of a CAE, which is not the objective of the current study. Networks like Convolutional or graph autoencoders acting on low-level high dimensional data will

undoubtedly perform better than currently executable QAEs. However, existing quantum resources cannot process such high dimensional data.

Bibliography

[1] S. L. Glashow, *Partial Symmetries of Weak Interactions*, *Nucl. Phys.* **22** (1961) 579–588.

[2] N. Cabibbo, *Unitary Symmetry and Leptonic Decays*, *Phys. Rev. Lett.* **10** (1963) 531–533.

[3] F. Englert and R. Brout, *Broken Symmetry and the Mass of Gauge Vector Mesons*, *Phys. Rev. Lett.* **13** (1964) 321–323.

[4] P. W. Higgs, *Broken Symmetries and the Masses of Gauge Bosons*, *Phys. Rev. Lett.* **13** (1964) 508–509.

[5] G. S. Guralnik, C. R. Hagen and T. W. B. Kibble, *Global Conservation Laws and Massless Particles*, *Phys. Rev. Lett.* **13** (1964) 585–587.

[6] A. Salam and J. C. Ward, *Gauge theory of elementary interactions*, *Phys. Rev.* **136** (1964) B763–B768.

[7] S. Weinberg, *A Model of Leptons*, *Phys. Rev. Lett.* **19** (1967) 1264–1266.

[8] G. 't Hooft, *Renormalizable Lagrangians for Massive Yang-Mills Fields*, *Nucl. Phys. B* **35** (1971) 167–188.

[9] M. Kobayashi and T. Maskawa, *CP Violation in the Renormalizable Theory of Weak Interaction*, *Prog. Theor. Phys.* **49** (1973) 652–657.

[10] D. J. Gross and F. Wilczek, *Asymptotically Free Gauge Theories - I*, *Phys. Rev. D* **8** (1973) 3633–3652.

[11] D. J. Gross and F. Wilczek, *Asymptotically free gauge theories. II*, *Phys. Rev. D* **9** (1974) 980–993.

[12] Planck collaboration, N. Aghanim et al., *Planck 2018 results. VI. Cosmological parameters*, *Astron. Astrophys.* **641** (2020) A6, [1807.06209].

[13] G. Hinshaw, D. Larson, E. Komatsu, D. N. Spergel, C. L. Bennett, J. Dunkley et al., *Nine-year wilkinson microwave anisotropy probe (WMAP) observations: cosmological parameter results*, *The Astrophysical Journal Supplement Series* **208** (Sep, 2013) 19.

[14] Super-Kamiokande collaboration, Y. Fukuda et al., *Evidence for oscillation of atmospheric neutrinos*, *Phys. Rev. Lett.* **81** (1998) 1562–1567, [hep-ex/9807003].

[15] SNO collaboration, Q. R. Ahmad et al., *Direct evidence for neutrino flavor transformation from neutral current interactions in the Sudbury Neutrino Observatory*, *Phys. Rev. Lett.* **89** (2002) 011301, [nucl-ex/0204008].

[16] CMS collaboration, S. Chatrchyan et al., *Observation of a New Boson at a Mass of 125 GeV with the CMS Experiment at the LHC*, *Phys. Lett. B* **716** (2012) 30–61, [1207.7235].

[17] ATLAS collaboration, G. Aad et al., *Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC*, *Phys. Lett. B* **716** (2012) 1–29, [1207.7214].

[18] B. P. Roe, H.-J. Yang, J. Zhu, Y. Liu, I. Stancu and G. McGregor, *Boosted decision trees, an alternative to artificial neural networks*, *Nucl. Instrum. Meth. A* **543** (2005) 577–584, [physics/0408124].

[19] B. H. Denby, *Neural Networks and Cellular Automata in Experimental High-energy Physics*, *Comput. Phys. Commun.* **49** (1988) 429–448.

[20] L. Lonnblad, C. Peterson and T. Rognvaldsson, *Finding Gluon Jets With a Neural Trigger*, *Phys. Rev. Lett.* **65** (1990) 1321–1324.

[21] Muon g-2 collaboration, B. Abi et al., *Measurement of the Positive Muon Anomalous Magnetic Moment to 0.46 ppm*, *Phys. Rev. Lett.* **126** (2021) 141801, [2104.03281].

[22] H. Harari, *How Many Quarks Are There?*, in *11th Rencontres de Moriond: International Meeting on Storage Ring Physics*, pp. 461–482, 1976.

[23] D. J. Gross and R. Jackiw, *Effect of anomalies on quasirenormalizable theories*, *Phys. Rev. D* **6** (1972) 477–493.

[24] C. Bouchiat, J. Iliopoulos and P. Meyer, *An Anomaly Free Version of Weinberg's Model*, *Phys. Lett. B* **38** (1972) 519–523.

[25] M. L. Perl et al., *Evidence for Anomalous Lepton Production in e+ - e- Annihilation*, *Phys. Rev. Lett.* **35** (1975) 1489–1492.

[26] P. Langacker and M.-x. Luo, *Implications of precision electroweak experiments for M_t, ρ_0, $\sin^2\theta_W$ and grand unification*, *Phys. Rev. D* **44** (1991) 817–822.

[27] G. Arcadi, A. Djouadi and M. Raidal, *Dark Matter through the Higgs portal*, *Phys. Rept.* **842** (2020) 1–180, [1903.03616].

[28] A. Djouadi, O. Lebedev, Y. Mambrini and J. Quevillon, *Implications of LHC searches for Higgs–portal dark matter*, *Phys. Lett. B* **709** (2012) 65–69, [1112.3299].

[29] A. Djouadi, A. Falkowski, Y. Mambrini and J. Quevillon, *Direct Detection of Higgs-Portal Dark Matter at the LHC*, *Eur. Phys. J. C* **73** (2013) 2455, [1205.3169].

[30] H. Han, J. M. Yang, Y. Zhang and S. Zheng, *Collider Signatures of Higgs-portal Scalar Dark Matter*, *Phys. Lett. B* **756** (2016) 109–112, [1601.06232].

[31] S. Baek, P. Ko, W.-I. Park and E. Senaha, *Higgs Portal Vector Dark Matter : Revisited*, *JHEP* **05** (2013) 036, [1212.2131].

[32] T. D. Lee, *A Theory of Spontaneous T Violation*, *Phys. Rev. D* **8** (1973) 1226–1239.

[33] G. C. Branco, P. M. Ferreira, L. Lavoura, M. N. Rebelo, M. Sher and J. P. Silva, *Theory and phenomenology of two-Higgs-doublet models*, *Phys. Rept.* **516** (2012) 1–102, [1106.0034].

[34] R. Barbieri, L. J. Hall and V. S. Rychkov, *Improved naturalness with a heavy Higgs: An Alternative road to LHC physics*, *Phys. Rev. D* **74** (2006) 015007, [hep-ph/0603188].

[35] S. Kashiwase and D. Suematsu, *Baryon number asymmetry and dark matter in the neutrino mass model with an inert doublet*, *Phys. Rev. D* **86** (2012) 053001, [1207.2594].

[36] M. E. Peskin and T. Takeuchi, *A New constraint on a strongly interacting Higgs sector*, *Phys. Rev. Lett.* **65** (1990) 964–967.

[37] M. E. Peskin and T. Takeuchi, *Estimation of oblique electroweak corrections*, *Phys. Rev. D* **46** (1992) 381–409.

[38] P. Sikivie, L. Susskind, M. B. Voloshin and V. I. Zakharov, *Isospin Breaking in Technicolor Models*, *Nucl. Phys. B* **173** (1980) 189–207.

[39] CDF collaboration, T. Aaltonen et al., *High-precision measurement of the W boson mass with the CDF II detector*, *Science* **376** (2022) 170–176.

[40] P. Asadi, C. Cesarotti, K. Fraser, S. Homiller and A. Parikh, *Oblique Lessons from the W Mass Measurement at CDF II*, 2204.05283.

[41] J. Fan, L. Li, T. Liu and K.-F. Lyu, *W-Boson Mass, Electroweak Precision Tests and SMEFT*, 2204.04805.

[42] E. Bagnaschi, J. Ellis, M. Madigan, K. Mimasu, V. Sanz and T. You, *SMEFT Analysis of m_W*, 2204.05260.

[43] D. J. Gross and F. Wilczek, *Ultraviolet behavior of non-abelian gauge theories*, *Phys. Rev. Lett.* **30** (Jun, 1973) 1343–1346.

[44] H. D. Politzer, *Reliable perturbative results for strong interactions?*, *Phys. Rev. Lett.* **30** (Jun, 1973) 1346–1349.

[45] R. K. Ellis, W. J. Stirling and B. R. Webber, *QCD and Collider Physics*. Cambridge Monographs on Particle Physics, Nuclear Physics and Cosmology. Cambridge University Press, 1996, 10.1017/CBO9780511628788.

[46] F. Herzog, B. Ruijl, T. Ueda, J. A. M. Vermaseren and A. Vogt, *The five-loop beta function of Yang-Mills theory with fermions*, *JHEP* **02** (2017) 090, [1701.01404].

[47] V. N. Gribov and L. N. Lipatov, *Deep inelastic e p scattering in perturbation theory*, *Sov. J. Nucl. Phys.* **15** (1972) 438–450.

[48] L. N. Lipatov, *The parton model and perturbation theory*, *Yad. Fiz.* **20** (1974) 181–198.

[49] Y. L. Dokshitzer, *Calculation of the Structure Functions for Deep Inelastic Scattering and e+ e- Annihilation by Perturbation Theory in Quantum Chromodynamics.*, *Sov. Phys. JETP* **46** (1977) 641–653.

[50] G. Altarelli and G. Parisi, *Asymptotic Freedom in Parton Language*, *Nucl. Phys. B* **126** (1977) 298–318.

[51] J. C. Collins, D. E. Soper and G. F. Sterman, *Factorization of Hard Processes in QCD*, *Adv. Ser. Direct. High Energy Phys.* **5** (1989) 1–91, [hep-ph/0409313].

[52] T. Kinoshita, *Mass singularities of feynman amplitudes*, *Journal of Mathematical Physics* **3** (1962) 650–677, [https://doi.org/10.1063/1.1724268].

[53] T. D. Lee and M. Nauenberg, *Degenerate systems and mass singularities*, *Phys. Rev.* **133** (Mar, 1964) B1549–B1562.

[54] M. Dasgupta and G. P. Salam, *Resummation of nonglobal QCD observables*, *Phys. Lett. B* **512** (2001) 323–330, [hep-ph/0104277].

[55] A. Banfi, G. P. Salam and G. Zanderighi, *Principles of general final-state resummation and automated implementation*, *JHEP* **03** (2005) 073, [hep-ph/0407286].

[56] A. Banfi, G. P. Salam and G. Zanderighi, *Resummed event shapes at hadron - hadron colliders*, *JHEP* **08** (2004) 062, [hep-ph/0407287].

[57] S. Alioli, A. Broggio, A. Gavardi, S. Kallweit, M. A. Lim, R. Nagar et al., *Resummed predictions for hadronic Higgs boson decays*, *JHEP* **04** (2021) 254, [2009.13533].

[58] H.-M. Chang, M. Procura, J. Thaler and W. J. Waalewijn, *Calculating Track-Based Observables for the LHC*, *Phys. Rev. Lett.* **111** (2013) 102002, [1303.6637].

[59] A. Chakraborty, S. Chakraborty and T. S. Roy, *Chasing New Physics in Stacks of Soft Tracks*, *Phys. Rev. D* **94** (2016) 111703, [1606.07826].

[60] A. J. Larkoski and J. Thaler, *Unsafe but Calculable: Ratios of Angularities in Perturbative QCD*, *JHEP* **09** (2013) 137, [1307.1699].

[61] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer et al., *The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations*, *JHEP* **07** (2014) 079, [1405.0301].

[62] P. Nason, *A New method for combining NLO QCD with shower Monte Carlo algorithms*, *JHEP* **11** (2004) 040, [hep-ph/0409146].

[63] S. Frixione, P. Nason and C. Oleari, *Matching NLO QCD computations with Parton Shower simulations: the POWHEG method*, *JHEP* **11** (2007) 070, [0709.2092].

[64] P. Nason and C. Oleari, *NLO Higgs boson production via vector-boson fusion matched with shower in POWHEG*, *JHEP* **02** (2010) 037, [0911.5299].

[65] S. Alioli, P. Nason, C. Oleari and E. Re, *A general framework for implementing NLO calculations in shower Monte Carlo programs: the POWHEG BOX*, *JHEP* **06** (2010) 043, [1002.2581].

[66] H. Murayama, I. Watanabe and K. Hagiwara, *HELAS: HELicity amplitude subroutines for Feynman diagram evaluations*, .

[67] C. Degrande, C. Duhr, B. Fuks, D. Grellscheid, O. Mattelaer and T. Reiter, *UFO - The Universal FeynRules Output*, *Comput. Phys. Commun.* **183** (2012) 1201–1214, [1108.2040].

[68] N. D. Christensen and C. Duhr, *FeynRules - Feynman rules made easy*, *Comput. Phys. Commun.* **180** (2009) 1614–1641, [0806.4194].

[69] A. Alloul, N. D. Christensen, C. Degrande, C. Duhr and B. Fuks, *FeynRules 2.0 - A complete toolbox for tree-level phenomenology*, *Comput. Phys. Commun.* **185** (2014) 2250–2300, [1310.1921].

[70] T. Sjöstrand, S. Ask, J. R. Christiansen, R. Corke, N. Desai, P. Ilten et al., *An Introduction to PYTHIA 8.2*, *Comput. Phys. Commun.* **191** (2015) 159–177, [1410.3012].

[71] J. Schwinger, *On the classical radiation of accelerated electrons*, *Phys. Rev.* **75** (Jun, 1949) 1912–1925.

[72] DELPHES 3 collaboration, J. de Favereau, C. Delaere, P. Demin, A. Giammanco, V. Lemaître, A. Mertens et al., *DELPHES 3, A modular framework for fast simulation of a generic collider experiment*, *JHEP* **02** (2014) 057, [1307.6346].

[73] CMS collaboration, V. Khachatryan et al., *The CMS trigger system*, *JINST* **12** (2017) P01020, [1609.02366].

[74] ATLAS collaboration, M. Aaboud et al., *Performance of the ATLAS Trigger System in 2015*, *Eur. Phys. J. C* **77** (2017) 317, [1611.09661].

[75] ATLAS collaboration, G. Aad et al., *Performance of the ATLAS muon triggers in Run 2*, *JINST* **15** (2020) P09015, [2004.13447].

[76] CMS collaboration, C. Ramon Alvarez, *Level 1 muon triggers algorithms for the CMS upgrade at the HL-LHC*, *PoS* **EPS-HEP2021** (2022) 764.

[77] D. Bertolini, P. Harris, M. Low and N. Tran, *Pileup Per Particle Identification*, *JHEP* **10** (2014) 059, [1407.6013].

[78] M. Cacciari, G. P. Salam and G. Soyez, *SoftKiller, a particle-level pileup removal method*, *Eur. Phys. J. C* **75** (2015) 59, [1407.0408].

[79] ATLAS collaboration, G. Aad et al., *Performance of pile-up mitigation techniques for jets in pp collisions at* $\sqrt{s} = 8$ *TeV using the ATLAS detector*, *Eur. Phys. J. C* **76** (2016) 581, [1510.03823].

[80] CMS collaboration, V. Khachatryan et al., *Jet energy scale and resolution in the CMS experiment in pp collisions at 8 TeV*, *JINST* **12** (2017) P02014, [1607.03663].

[81] C. Frye, A. J. Larkoski, M. D. Schwartz and K. Yan, *Precision physics with pile-up insensitive observables*, 1603.06375.

[82] CMS collaboration, A. M. Sirunyan et al., *Pileup mitigation at CMS in 13 TeV data*, *JINST* **15** (2020) P09018, [2003.00503].

[83] G. Hanson, G. S. Abrams, A. M. Boyarski, M. Breidenbach, F. Bulos, W. Chinowsky et al., *Evidence for jet structure in hadron production by e^+e^- annihilation*, *Phys. Rev. Lett.* **35** (Dec, 1975) 1609–1612.

[84] G. Sterman and S. Weinberg, *Jets from quantum chromodynamics*, *Phys. Rev. Lett.* **39** (Dec, 1977) 1436–1439.

[85] JADE collaboration, W. Bartel et al., *Experimental Studies on Multi-Jet Production in e+ e- Annihilation at PETRA Energies*, *Z. Phys. C* **33** (1986) 23.

[86] JADE collaboration, S. Bethke et al., *Experimental Investigation of the Energy Dependence of the Strong Coupling Strength*, *Phys. Lett. B* **213** (1988) 235–241.

[87] S. Catani, Y. L. Dokshitzer, M. H. Seymour and B. R. Webber, *Longitudinally invariant K_t clustering algorithms for hadron hadron collisions*, *Nucl. Phys. B* **406** (1993) 187–224.

[88] G. C. Blazey et al., *Run II jet physics*, in *Physics at Run II: QCD and Weak Boson Physics Workshop: Final General Meeting*, pp. 47–77, 5, 2000. hep-ex/0005012.

[89] S. D. Ellis and D. E. Soper, *Successive combination jet algorithm for hadron collisions*, *Phys. Rev. D* **48** (1993) 3160–3166, [hep-ph/9305266].

[90] Y. L. Dokshitzer, G. D. Leder, S. Moretti and B. R. Webber, *Better jet clustering algorithms*, *JHEP* **08** (1997) 001, [hep-ph/9707323].

[91] M. Cacciari, G. P. Salam and G. Soyez, *The anti-k_t jet clustering algorithm*, *JHEP* **04** (2008) 063, [0802.1189].

[92] J. E. Huth et al., *Toward a standardization of jet definitions*, in *1990 DPF Summer Study on High-energy Physics: Research Directions for the Decade (Snowmass 90)*, pp. 0134–136, 12, 1990.

[93] G. P. Salam and G. Soyez, *A Practical Seedless Infrared-Safe Cone jet algorithm*, *JHEP* **05** (2007) 086, [0704.0292].

[94] M. Cacciari, G. P. Salam and G. Soyez, *FastJet User Manual*, *Eur. Phys. J. C* **72** (2012) 1896, [1111.6097].

[95] M. Wobisch and T. Wengler, *Hadronization corrections to jet cross-sections in deep inelastic scattering*, in *Workshop on Monte Carlo Generators for HERA Physics (Plenary Starting Meeting)*, pp. 270–279, 4, 1998. hep-ph/9907280.

[96] A. T. Pierce and B. R. Webber, *Comparisons of new jet clustering algorithms for hadron hadron collisions*, *Phys. Rev. D* **59** (1999) 034014, [hep-ph/9807532].

[97] J. M. Butterworth, A. R. Davison, M. Rubin and G. P. Salam, *Jet substructure as a new Higgs search channel at the LHC*, *Phys. Rev. Lett.* **100** (2008) 242001, [0802.2470].

[98] CMS collaboration, A. M. Sirunyan et al., *Observation of Higgs boson decay to bottom quarks*, *Phys. Rev. Lett.* **121** (2018) 121801, [1808.08242].

[99] ATLAS collaboration, M. Aaboud et al., *Observation of $H \to b\bar{b}$ decays and VH production with the ATLAS detector*, *Phys. Lett. B* **786** (2018) 59–86, [1808.08238].

[100] M. Dasgupta, L. Magnea and G. P. Salam, *Non-perturbative QCD effects in jets at hadron colliders*, *JHEP* **02** (2008) 055, [0712.3014].

[101] M. Dasgupta, A. Fregoso, S. Marzani and G. P. Salam, *Towards an understanding of jet substructure*, *JHEP* **09** (2013) 029, [1307.0007].

[102] D. E. Kaplan, K. Rehermann, M. D. Schwartz and B. Tweedie, *Top Tagging: A Method for Identifying Boosted Hadronically Decaying Top Quarks*, *Phys. Rev. Lett.* **101** (2008) 142001, [0806.0848].

[103] CMS collaboration, *Boosted Top Jet Tagging at CMS*, .

[104] L. G. Almeida, S. J. Lee, G. Perez, G. F. Sterman, I. Sung and J. Virzi, *Substructure of high-p_T Jets at the LHC*, *Phys. Rev. D* **79** (2009) 074017, [0807.0234].

[105] J. Thaler and K. Van Tilburg, *Identifying Boosted Objects with N-subjettiness*, *JHEP* **03** (2011) 015, [1011.2268].

[106] I. W. Stewart, F. J. Tackmann and W. J. Waalewijn, *N-Jettiness: An Inclusive Event Shape to Veto Jets*, *Phys. Rev. Lett.* **105** (2010) 092002, [1004.2489].

[107] A. J. Larkoski, D. Neill and J. Thaler, *Jet Shapes with the Broadening Axis*, *JHEP* **04** (2014) 017, [1401.2158].

[108] A. J. Larkoski, G. P. Salam and J. Thaler, *Energy Correlation Functions for Jet Substructure*, *JHEP* **06** (2013) 108, [1305.0007].

[109] S. D. Ellis, C. K. Vermilion and J. R. Walsh, *Techniques for improved heavy particle searches with jet substructure*, *Phys. Rev. D* **80** (2009) 051501, [0903.5081].

[110] S. D. Ellis, C. K. Vermilion and J. R. Walsh, *Recombination Algorithms and Jet Substructure: Pruning as a Tool for Heavy Particle Searches*, *Phys. Rev. D* **81** (2010) 094023, [0912.0033].

[111] D. Krohn, J. Thaler and L.-T. Wang, *Jet Trimming*, *JHEP* **02** (2010) 084, [0912.1342].

[112] A. J. Larkoski, I. Moult and B. Nachman, *Jet Substructure at the Large Hadron Collider: A Review of Recent Advances in Theory and Machine Learning*, *Phys. Rept.* **841** (2020) 1–63, [1709.04464].

[113] R. Kogler et al., *Jet Substructure at the Large Hadron Collider: Experimental Review*, *Rev. Mod. Phys.* **91** (2019) 045003, [1803.06991].

[114] S. Marzani, G. Soyez and M. Spannowsky, *Looking inside jets: an introduction to jet substructure and boosted-object phenomenology*, vol. 958. Springer, 2019, 10.1007/978-3-030-15709-8.

[115] J. Cogan, M. Kagan, E. Strauss and A. Schwartzman, *Jet-Images: Computer Vision Inspired Techniques for Jet Tagging*, *JHEP* **02** (2015) 118, [1407.5675].

[116] L. G. Almeida, M. Backović, M. Cliche, S. J. Lee and M. Perelstein, *Playing Tag with ANN: Boosted Top Identification with Pattern Recognition*, *JHEP* **07** (2015) 086, [1501.05968].

[117] L. de Oliveira, M. Kagan, L. Mackey, B. Nachman and A. Schwartzman, *Jet-images — deep learning edition*, *JHEP* **07** (2016) 069, [1511.05190].

[118] G. Kasieczka, T. Plehn, M. Russell and T. Schell, *Deep-learning Top Taggers or The End of QCD?*, *JHEP* **05** (2017) 006, [1701.08784].

[119] P. Baldi, K. Bauer, C. Eng, P. Sadowski and D. Whiteson, *Jet Substructure Classification in High-Energy Physics with Deep Neural Networks*, *Phys. Rev.* **D93** (2016) 094034, [1603.09349].

[120] J. Barnard, E. N. Dawe, M. J. Dolan and N. Rajcic, *Parton Shower Uncertainties in Jet Substructure Analyses with Deep Neural Networks*, *Phys. Rev. D* **95** (2017) 014018, [1609.00607].

[121] G. Kasieczka, S. Marzani, G. Soyez and G. Stagnitto, *Towards Machine Learning Analytics for Jet Substructure*, *JHEP* **09** (2020) 195, [2007.04319].

[122] P. T. Komiske, E. M. Metodiev and M. D. Schwartz, *Deep learning in color: towards automated quark/gluon jet discrimination*, *JHEP* **01** (2017) 110, [1612.01551].

[123] S. Choi, S. J. Lee and M. Perelstein, *Infrared Safety of a Neural-Net Top Tagging Algorithm*, *JHEP* **02** (2019) 132, [1806.01263].

[124] W. S. McCulloch and W. Pitts, *A logical calculus of the ideas immanent in nervous activity*, *The bulletin of mathematical biophysics* **5** (Dec, 1943) 115–133.

[125] F. Rosenblatt, *The perceptron: A probabilistic model for information storage and organization in the brain.*, *Psychological Review* **65** (1958) 386–408.

[126] G. Cybenko, *Approximation by superpositions of a sigmoidal function*, *Mathematics of Control, Signals and Systems* **2** (Dec, 1989) 303–314.

[127] M. Leshno, V. Y. Lin, A. Pinkus and S. Schocken, *Multilayer feedforward networks with a nonpolynomial activation function can approximate any function*, *Neural Networks* **6** (1993) 861–867.

[128] B. Hanin, *Universal function approximation by deep neural nets with bounded width and relu activations*, *Mathematics* **7** (2019) .

[129] A. Kratsios and E. Bilokopytov, *Non-euclidean universal approximation*, .

[130] P. Werbos, *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Harvard University, 1975.

[131] D. J. C. MacKay, *A practical bayesian framework for backpropagation networks*, *Neural Comput.* **4** (may, 1992) 448–472.

[132] A. Kendall and Y. Gal, *What uncertainties do we need in bayesian deep learning for computer vision?*, in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan et al., eds.), vol. 30, Curran Associates, Inc., 2017.

[133] S. Bollweg, M. Haußmann, G. Kasieczka, M. Luchmann, T. Plehn and J. Thompson, *Deep-Learning Jets with Uncertainties and More*, *SciPost Phys.* **8** (2020) 006, [1904.10004].

[134] G. Kasieczka, M. Luchmann, F. Otterpohl and T. Plehn, *Per-Object Systematics using Deep-Learned Calibration*, *SciPost Phys.* **9** (2020) 089, [2003.11099].

[135] ATLAS Collaboration collaboration, *Evaluating statistical uncertainties and correlations using the bootstrap method*, tech. rep., CERN, Geneva, Apr, 2021.

[136] CMS collaboration, *Combination of standard model Higgs boson searches and measurements of the properties of the new boson with a mass near 125 GeV*, .

[137] B. Efron, *Bootstrap Methods: Another Look at the Jackknife*, *Annals Statist.* **7** (1979) 1–26.

[138] J. Bendavid, *Efficient Monte Carlo Integration Using Boosted Decision Trees and Generative Deep Neural Networks*, 1707.00028.

[139] S. Otten, S. Caron, W. de Swart, M. van Beekveld, L. Hendriks, C. van Leeuwen et al., *Event Generation and Statistical Sampling for Physics with Deep Generative Models and a Density Information Buffer*, *Nature Commun.* **12** (2021) 2985, [1901.00875].

[140] M. S. Albergo, G. Kanwar and P. E. Shanahan, *Flow-based generative models for Markov chain Monte Carlo in lattice field theory*, *Phys. Rev. D* **100** (2019) 034515, [1904.12072].

[141] A. Butter, T. Plehn and R. Winterhalder, *How to GAN LHC Events*, *SciPost Phys.* **7** (2019) 075, [1907.03764].

[142] J. Arjona Martínez, T. Q. Nguyen, M. Pierini, M. Spiropulu and J.-R. Vlimant, *Particle Generative Adversarial Networks for full-event simulation at the LHC and their application to pileup description*, *J. Phys. Conf. Ser.* **1525** (2020) 012081, [1912.02748].

[143] S. Carrazza and F. A. Dreyer, *Lund jet images from generative and cycle-consistent adversarial networks*, *Eur. Phys. J. C* **79** (2019) 979, [1909.01359].

[144] C. Gao, S. Höche, J. Isaacson, C. Krause and H. Schulz, *Event Generation with Normalizing Flows*, *Phys. Rev. D* **101** (2020) 076002, [2001.10028].

[145] C. Krause and D. Shih, *CaloFlow II: Even Faster and Still Accurate Generation of Calorimeter Showers with Normalizing Flows*, 2110.11377.

[146] A. Butter, S. Diefenbacher, G. Kasieczka, B. Nachman and T. Plehn, *GANplifying event samples*, *SciPost Phys.* **10** (2021) 139, [2008.06545].

[147] S. Badger et al., *Machine Learning and LHC Event Generation*, 2203.07460.

[148] M. Paganini, L. de Oliveira and B. Nachman, *CaloGAN : Simulating 3D high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks*, *Phys. Rev. D* **97** (2018) 014021, [1712.10321].

[149] M. Paganini, L. de Oliveira and B. Nachman, *Accelerating Science with Generative Adversarial Networks: An Application to 3D Particle Showers in Multilayer Calorimeters*, *Phys. Rev. Lett.* **120** (2018) 042003, [1705.02355].

[150] P. Musella and F. Pandolfi, *Fast and Accurate Simulation of Particle Detectors Using Generative Adversarial Networks*, *Comput. Softw. Big Sci.* **2** (2018) 8, [1805.00850].

[151] LHCb collaboration, L. Anderlini, *Machine Learning for the LHCb Simulation*, 10, 2021. 2110.07925.

[152] G. R. Khattak, S. Vallecorsa, F. Carminati and G. M. Khan, *Fast Simulation of a High Granularity Calorimeter by Generative Adversarial Networks*, 2109.07388.

[153] P. T. Komiske, E. M. Metodiev, B. Nachman and M. D. Schwartz, *Pileup Mitigation with Machine Learning (PUMML)*, *JHEP* **12** (2017) 051, [1707.08600].

[154] J. Arjona Martínez, O. Cerri, M. Pierini, M. Spiropulu and J.-R. Vlimant, *Pileup mitigation at the Large Hadron Collider with graph neural networks*, *Eur. Phys. J. Plus* **134** (2019) 333, [1810.07988].

[155] V. Mikuni and F. Canelli, *ABCNet: An attention-based method for particle tagging*, *Eur. Phys. J. Plus* **135** (2020) 463, [2001.05311].

[156] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair et al., *Generative Adversarial Networks*, 1406.2661.

[157] D. Rezende and S. Mohamed, *Variational inference with normalizing flows*, in *Proceedings of the 32nd International Conference on Machine Learning* (F. Bach and D. Blei, eds.), vol. 37 of *Proceedings of Machine Learning Research*, (Lille, France), pp. 1530–1538, PMLR, 07–09 Jul, 2015. 1505.05770.

[158] S. Kullback and R. A. Leibler, *On information and sufficiency*, *Ann. Math. Statist.* **22** (03, 1951) 79–86.

[159] J. Neyman, E. S. Pearson and K. Pearson, *Ix. on the problem of the most efficient tests of statistical hypotheses*, *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* **231** (1933) 289–337, [https://royalsocietypublishing.org/doi/pdf/10.1098/rsta.1933.0009].

[160] M. A. Kramer, *Nonlinear principal component analysis using autoassociative neural networks*, *AIChE Journal* **37** (1991) 233–243, [https://aiche.onlinelibrary.wiley.com/doi/pdf/10.1002/aic.690370209].

[161] R. T. D'Agnolo and A. Wulzer, *Learning New Physics from a Machine*, *Phys. Rev. D* **99** (2019) 015014, [1806.02350].

[162] J. A. Aguilar-Saavedra, J. H. Collins and R. K. Mishra, *A generic anti-QCD jet tagger*, *JHEP* **11** (2017) 163, [1709.01087].

[163] A. Blance, M. Spannowsky and P. Waite, *Adversarially-trained autoencoders for robust unsupervised new physics searches*, *JHEP* **10** (2019) 047, [1905.10384].

[164] V. Mikuni, B. Nachman and D. Shih, *Online-compatible Unsupervised Non-resonant Anomaly Detection*, 2111.06417.

[165] E. Govorkova et al., *Autoencoders on FPGAs for real-time, unsupervised new physics detection at 40 MHz at the Large Hadron Collider*, 2108.03986.

[166] J. Hajer, Y.-Y. Li, T. Liu and H. Wang, *Novelty Detection Meets Collider Physics*, *Phys. Rev. D* **101** (2020) 076015, [1807.10261].

[167] T. S. Roy and A. H. Vijay, *A robust anomaly finder based on autoencoders*, 1903.02032.

[168] S. Tsan, R. Kansal, A. Aportela, D. Diaz, J. Duarte, S. Krishna et al., *Particle Graph Autoencoders and Differentiable, Learned Energy Mover's Distance*, in *35th Conference on Neural Information Processing Systems*, 11, 2021. 2111.12849.

[169] T. Heimel, G. Kasieczka, T. Plehn and J. M. Thompson, *QCD or What?*, *SciPost Phys.* **6** (2019) 030, [1808.08979].

[170] M. Farina, Y. Nakai and D. Shih, *Searching for New Physics with Deep Autoencoders*, *Phys. Rev. D* **101** (2020) 075021, [1808.08992].

[171] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 1412.6980.

[172] T. Dozat, *Incorporating nesterov momentum into adam*, in *ICLR 2016 Workshop*, 2016.

[173] M. Erdmann, E. Geiser, Y. Rath and M. Rieger, *Lorentz Boost Networks: Autonomous Physics-Inspired Feature Engineering*, *JINST* **14** (2019) P06006, [1812.09722].

[174] S. H. Lim and M. M. Nojiri, *Spectral Analysis of Jet Substructure with Neural Networks: Boosted Higgs Case*, *JHEP* **10** (2018) 181, [1807.03312].

[175] A. Chakraborty, S. H. Lim and M. M. Nojiri, *Interpretable deep learning for two-prong jet classification with jet spectra*, *JHEP* **07** (2019) 135, [1904.02092].

[176] A. Bogatskiy et al., *Symmetry Group Equivariant Architectures for Physics*, in *2022 Snowmass Summer Study*, 3, 2022. 2203.06153.

[177] S. Villar, D. W. Hogg, K. Storey-Fisher, W. Yao and B. Blum-Smith, *Scalars are universal: Equivariant machine learning, structured like classical physics*, 2106.06610.

[178] A. Butter, G. Kasieczka, T. Plehn and M. Russell, *Deep-learned Top Tagging with a Lorentz Layer*, *SciPost Phys.* **5** (2018) 028, [1707.08966].

[179] S. Gong, Q. Meng, J. Zhang, H. Qu, C. Li, S. Qian et al., *An Efficient Lorentz Equivariant Graph Neural Network for Jet Tagging*, 2201.08187.

[180] A. Bogatskiy, B. Anderson, J. T. Offermann, M. Roussi, D. W. Miller and R. Kondor, *Lorentz Group Equivariant Neural Network for Particle Physics*, 2006.04780.

[181] K. Desai, B. Nachman and J. Thaler, *SymmetryGAN: Symmetry Discovery with Deep Learning*, 2112.05722.

[182] B. M. Dillon, G. Kasieczka, H. Olischlager, T. Plehn, P. Sorrenson and L. Vogel, *Symmetries, Safety, and Self-Supervision*, 2108.04253.

[183] D. Kim, K. Kong, K. T. Matchev, M. Park and P. Shyamsundar, *Deep-Learned Event Variables for Collider Phenomenology*, 2105.10126.

[184] CMS collaboration, A. M. Sirunyan et al., *Identification of heavy, energetic, hadronically decaying particles using machine-learning techniques*, *JINST* **15** (2020) P06005, [2004.08262].

[185] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, *Gradient-based learning applied to document recognition*, *Proceedings of the IEEE* **86** (1998) 2278–2324.

[186] K. Fukushima, *Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position*, *Biological Cybernetics* **36** (Apr, 1980) 193–202.

[187] A. Dhillon and G. K. Verma, *Convolutional neural network: a review of models, methodologies and applications to object detection*, *Progress in Artificial Intelligence* **9** (Jun, 2020) 85–112.

[188] A. Khan, A. Sohail, U. Zahoora and A. S. Qureshi, *A survey of the recent architectures of deep convolutional neural networks*, *Artificial Intelligence Review* **53** (Dec, 2020) 5455–5516.

[189] S. Macaluso and D. Shih, *Pulling Out All the Tops with Computer Vision and Deep Learning*, *JHEP* **10** (2018) 121, [1803.00107].

[190] J. H. Kim, M. Kim, K. Kong, K. T. Matchev and M. Park, *Portraying Double Higgs at the Large Hadron Collider*, *JHEP* **09** (2019) 047, [1904.08549].

[191] T. Finke, M. Krämer, A. Morandini, A. Mück and I. Oleksiyuk, *Autoencoders for unsupervised anomaly detection in high energy physics*, 2104.09051.

[192] S. Diefenbacher, H. Frost, G. Kasieczka, T. Plehn and J. M. Thompson, *CapsNets Continuing the Convolutional Quest*, 1906.11265.

[193] W. Bhimji, S. A. Farrell, T. Kurth, M. Paganini, Prabhat and E. Racah, *Deep Neural Networks for Physics Analysis on low-level whole-detector data at the LHC*, *J. Phys. Conf. Ser.* **1085** (2018) 042034, [1711.03573].

[194] CMS collaboration, M. Andrews, M. Paulini, S. Gleyzer and B. Poczos, *Exploring End-to-end Deep Learning Applications for Event Classification at CMS*, *EPJ Web Conf.* **214** (2019) 06031.

[195] J. Lin, M. Freytsis, I. Moult and B. Nachman, *Boosting $H \to b\bar{b}$ with Machine Learning*, *JHEP* **10** (2018) 101, [1807.10768].

[196] M. Andrews, M. Paulini, S. Gleyzer and B. Poczos, *End-to-End Physics Event Classification with CMS Open Data: Applying Image-Based Deep Learning to Detector Data for the Direct Classification of Collision Events at the LHC*, *Comput. Softw. Big Sci.* **4** (2020) 6, [1807.11916].

[197] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam and P. Vandergheynst, *Geometric deep learning: Going beyond euclidean data*, *IEEE Signal Processing Magazine* **34** (July, 2017) 18–42.

[198] M. Gori, G. Monfardini and F. Scarselli, *A new model for learning in graph domains*, in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 2, pp. 729–734 vol. 2, 2005. DOI.

[199] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals and G. E. Dahl, *Neural message passing for quantum chemistry*, in *International Conference on Machine Learning*, pp. 1263–1272, PMLR, 2017. 1704.01212.

[200] J. Shlomi, P. Battaglia and J.-R. Vlimant, *Graph Neural Networks in Particle Physics*, 2007.13681.

[201] M. L. Mavrovouniotis and S. Chang, *Hierarchical neural networks*, *Computers & Chemical Engineering* **16** (Apr, 1992) 347–369.

[202] T. N. Kipf and M. Welling, *Semi-supervised classification with graph convolutional networks*, 1609.02907.

[203] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein and J. M. Solomon, *Dynamic graph cnn for learning on point clouds*, *Acm Transactions On Graphics (tog)* **38** (2019) 1–12, [1801.07829].

[204] R. Q. Charles, H. Su, M. Kaichun and L. J. Guibas, *Pointnet: Deep learning on point sets for 3d classification and segmentation*, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 77–85, IEEE Computer Society, 2017. 1612.00593.

[205] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov and A. J. Smola, *Deep sets*, *Advances in Neural Information Processing Systems* **30** (2017) , [1703.06114].

[206] C. R. Qi, L. Yi, H. Su and L. J. Guibas, *Pointnet++: Deep hierarchical feature learning on point sets in a metric space*, *Advances in Neural Information Processing Systems* **30** (2017) , [1706.02413].

[207] ATLAS collaboration, G. Aad et al., *Measurements of the W production cross sections in association with jets with the ATLAS detector*, *Eur. Phys. J.* **C75** (2015) 82, [1409.8639].

[208] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang et al., *Graph neural networks: A review of methods and applications*, 1812.08434.

[209] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang and P. S. Yu, *A comprehensive survey on graph neural networks*, *IEEE Transactions on Neural Networks and Learning Systems* **32** (2021) 4–24, [1901.00596].

[210] F. A. Dreyer and H. Qu, *Jet tagging in the Lund plane with graph networks*, *JHEP* **03** (2021) 052, [2012.08526].

[211] B. Andersson, G. Gustafson, L. Lonnblad and U. Pettersson, *Coherence Effects in Deep Inelastic Scattering*, *Z. Phys. C* **43** (1989) 625.

[212] A. Lifson, G. P. Salam and G. Soyez, *Calculating the primary Lund Jet Plane density*, *JHEP* **10** (2020) 170, [2007.06578].

[213] O. Knapp, O. Cerri, G. Dissertori, T. Q. Nguyen, M. Pierini and J.-R. Vlimant, *Adversarially Learned Anomaly Detection on CMS Open Data: re-discovering the top quark*, *Eur. Phys. J. Plus* **136** (2021) 236, [2005.01598].

[214] V. Mikuni and F. Canelli, *Unsupervised clustering for collider physics*, 2010.07106.

[215] G. Dezoort, S. Thais, I. Ojalvo, P. Elmer, V. Razavimaleki, J. Duarte et al., *Charged particle tracking via edge-classifying interaction networks*, 2103.16701.

[216] M. Abdughani, J. Ren, L. Wu and J. M. Yang, *Probing stop pair production at the LHC with graph neural networks*, *JHEP* **08** (2019) 055, [1807.09088].

[217] Y. Iiyama et al., *Distance-Weighted Graph Neural Networks on FPGAs for Real-Time Particle Reconstruction in High Energy Physics*, *Front. Big Data* **3** (2020) 598927, [2008.03601].

[218] I. Henrion, J. Brehmer, J. Bruna, K. Cho, K. Cranmer, G. Louppe et al., *Neural message passing for jet physics*, .

[219] H. Qu and L. Gouskos, *ParticleNet: Jet Tagging via Particle Clouds*, *Phys. Rev. D* **101** (2020) 056019, [1902.08570].

[220] E. Bernreuther, T. Finke, F. Kahlhoefer, M. Krämer and A. Mück, *Casting a graph net to catch dark showers*, *SciPost Phys.* **10** (2021) 046, [2006.08639].

[221] E. A. Moreno, O. Cerri, J. M. Duarte, H. B. Newman, T. Q. Nguyen, A. Periwal et al., *JEDI-net: a jet identification algorithm based on interaction networks*, *Eur. Phys. J. C* **80** (2020) 58, [1908.05318].

[222] E. A. Moreno, T. Q. Nguyen, J.-R. Vlimant, O. Cerri, H. B. Newman, A. Periwal et al., *Interaction networks for the identification of boosted $H \to b\bar{b}$ decays*, *Phys. Rev. D* **102** (2020) 012010, [1909.12285].

[223] A. Blance and M. Spannowsky, *Unsupervised Event Classification with Graphs on Classical and Photonic Quantum Computers*, 2103.03897.

[224] A. Butter et al., *The Machine Learning landscape of top taggers*, *SciPost Phys.* **7** (2019) 014, [1902.09914].

[225] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam and P. Vandergheynst, *Geometric deep learning: Going beyond euclidean data*, *IEEE Signal Processing Magazine* **34** (2017) 18–42.

[226] A. Sperduti and A. Starita, *Supervised neural networks for the classification of structures*, *IEEE Transactions on Neural Networks* **8** (1997) 714–735.

[227] P. Frasconi, M. Gori and A. Sperduti, *A general framework for adaptive processing of data structures*, *IEEE Transactions on Neural Networks* **9** (1998) 768–786.

[228] M. Natali, S. Biasotti, G. Patanè and B. Falcidieno, *Graph-based representations of point clouds*, *Graphical Models* **73** (2011) 151–164.

[229] F. A. Dreyer, G. P. Salam and G. Soyez, *The Lund Jet Plane*, *JHEP* **12** (2018) 064, [1807.04758].

[230] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro et al., *TensorFlow: Large-scale machine learning on heterogeneous systems*, 2015.

[231] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan et al., *Pytorch: An imperative style, high-performance deep learning library*, 1912.01703.

[232] F. Chollet et al., "Keras." https://keras.io, 2015.

[233] M. Wang, D. Zheng, Z. Ye, Q. Gan, M. Li, X. Song et al., *Deep graph library: A graph-centric, highly-performant package for graph neural networks*, 1909.01315.

[234] M. Fey and J. E. Lenssen, *Fast graph representation learning with PyTorch Geometric*, in *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019. 1903.02428.

[235] S. T. Y. Dokshitzer, V. Khoze, *Proceedings of the international conference*, Physics in Collision VI, (Chicago, Illinois), p. 365, World Scientific, Singapore, 1986.

[236] J. D. Bjorken, *Rapidity gaps and jets as a new-physics signature in very-high-energy hadron-hadron collisions*, *Phys. Rev. D* **47** (Jan, 1993) 101–113.

[237] R. S. Fletcher and T. Stelzer, *Rapidity gap signals in higgs-boson production at the ssc*, *Phys. Rev. D* **48** (Dec, 1993) 5162–5167.

[238] CMS collaboration, V. Khachatryan et al., *Search for the standard model Higgs boson produced through vector boson fusion and decaying to $b\bar{b}$*, *Phys. Rev. D* **92** (2015) 032008, [1506.01010].

[239] R. Cahn and S. Dawson, *Production of very massive higgs bosons*, *Physics Letters B* **136** (1984) 196 – 200.

[240] D. L. Rainwater and D. Zeppenfeld, *Searching for $H \to \gamma\gamma$ in weak boson fusion at the LHC*, *JHEP* **12** (1997) 005, [hep-ph/9712271].

[241] D. L. Rainwater and D. Zeppenfeld, *Observing $H \to W^*W^* \to e^{\pm}\mu\mp \not{p}_T$ in weak boson fusion with dual forward jet tagging at the CERN LHC*, *Phys. Rev. D* **60** (1999) 113004, [hep-ph/9906218].

[242] D. L. Rainwater, D. Zeppenfeld and K. Hagiwara, *Searching for $H \to \tau^+\tau^-$ in weak boson fusion at the CERN LHC*, *Phys. Rev. D* **59** (1998) 014037, [hep-ph/9808468].

[243] T. Plehn, D. L. Rainwater and D. Zeppenfeld, *Determining the Structure of Higgs Couplings at the LHC*, *Phys. Rev. Lett.* **88** (2002) 051801, [hep-ph/0105325].

[244] V. Hankele, G. Klamke, D. Zeppenfeld and T. Figy, *Anomalous Higgs boson couplings in vector boson fusion at the CERN LHC*, *Phys. Rev. D* **74** (2006) 095001, [hep-ph/0609075].

[245] T. Han, S. Mukhopadhyay, B. Mukhopadhyaya and Y. Wu, *Measuring the CP property of Higgs coupling to tau leptons in the VBF channel at the LHC*, *JHEP* **05** (2017) 128, [1612.00413].

[246] D. Zanzi, ATLAS and C. Collaborations, *Measurement of the Higgs Boson Couplings and CP Structure Using Tau Leptons at the LHC*, *Nuclear and Particle Physics Proceedings* **287–288** (2017) 115–118.

[247] O. J. Eboli and D. Zeppenfeld, *Observing an invisible Higgs boson*, *Phys. Lett. B* **495** (2000) 147–154, [hep-ph/0009158].

[248] A. Datta, P. Konar and B. Mukhopadhyaya, *Signals of neutralinos and charginos from gauge boson fusion at the Large Hadron Collider*, *Phys. Rev. D* **65** (2002) 055008, [hep-ph/0109071].

[249] D. Choudhury, A. Datta, K. Huitu, P. Konar, S. Moretti and B. Mukhopadhyaya, *Slepton production from gauge boson fusion*, *Phys. Rev. D* **68** (2003) 075007, [hep-ph/0304192].

[250] P. Konar and D. Zeppenfeld, *Next-to-leading order QCD corrections to slepton pair production via vector-boson fusion*, *Phys. Lett. B* **647** (2007) 460–465, [hep-ph/0612119].

[251] T. Han, G. Valencia and S. Willenbrock, *Structure function approach to vector boson scattering in p p collisions*, *Phys. Rev. Lett.* **69** (1992) 3274–3277, [hep-ph/9206246].

[252] T. Figy, C. Oleari and D. Zeppenfeld, *Next-to-leading order jet distributions for Higgs boson production via weak boson fusion*, *Phys. Rev. D* **68** (2003) 073005, [hep-ph/0306109].

[253] F. A. Dreyer and A. Karlberg, *Vector-Boson Fusion Higgs Production at Three Loops in QCD*, *Phys. Rev. Lett.* **117** (2016) 072001, [1606.00840].

[254] M. Ciccolini, A. Denner and S. Dittmaier, *Strong and electroweak corrections to the production of Higgs + 2jets via weak interactions at the LHC*, *Phys. Rev. Lett.* **99** (2007) 161803, [0707.0381].

[255] T. Liu, K. Melnikov and A. A. Penin, *Nonfactorizable QCD Effects in Higgs Boson Production via Vector Boson Fusion*, *Phys. Rev. Lett.* **123** (2019) 122002, [1906.10899].

[256] A. Datta, P. Konar and B. Mukhopadhyaya, *New Higgs signals from vector boson fusion in R-parity violating supersymmetry*, *Phys. Rev. D* **63** (2001) 095009, [hep-ph/0009112].

[257] P. Konar and B. Mukhopadhyaya, *Gauge boson fusion as a probe of inverted hierarchies in supersymmetry*, *Phys. Rev. D* **70** (2004) 115011, [hep-ph/0311347].

[258] A. G. Delannoy et al., *Probing Dark Matter at the LHC using Vector Boson Fusion Processes*, *Phys. Rev. Lett.* **111** (2013) 061801, [1304.7779].

[259] A. Berlin, T. Lin, M. Low and L.-T. Wang, *Neutralinos in Vector Boson Fusion at High Energy Colliders*, *Phys. Rev. D* **91** (2015) 115002, [1502.05044].

[260] F. Braren, *Selection of Vector Boson Fusion Events in the $H \to \gamma\gamma$ Decay Channel Using an Inclusive Event Shape*, Master's thesis, Hamburg U., 4, 2015.

[261] CMS collaboration, A. M. Sirunyan et al., *Search for invisible decays of a Higgs boson produced through vector boson fusion in proton-proton collisions at $\sqrt{s} = 13$ TeV*, *Phys. Lett.* **B793** (2019) 520–551, [1809.05937].

[262] ATLAS collaboration, M. Aaboud et al., *Search for invisible Higgs boson decays in vector boson fusion at $\sqrt{s} = 13$ TeV with the ATLAS detector*, *Phys. Lett. B* **793** (2019) 499–519, [1809.06682].

[263] C. Bernaciak, T. Plehn, P. Schichtel and J. Tattersall, *Spying an invisible Higgs boson*, *Phys. Rev. D* **91** (2015) 035024, [1411.7699].

[264] A. Biekötter, F. Keilbach, R. Moutafis, T. Plehn and J. Thompson, *Tagging Jets in Invisible Higgs Searches*, *SciPost Phys.* **4** (2018) 035, [1712.03973].

[265] T. G. Rizzo, *Gluon final states in higgs-boson decay*, *Phys. Rev. D* **22** (Jul, 1980) 178–183.

[266] R. P. Kauffman and S. V. Desai, *Production of a higgs pseudoscalar plus two jets in hadronic collisions*, *Phys. Rev. D* **59** (Feb, 1999) 057504.

[267] LHC Higgs Cross Section Working Group collaboration, D. de Florian et al., *Handbook of LHC Higgs Cross Sections: 4. Deciphering the Nature of the Higgs Sector*, 1610.07922.

[268] S. Hoeche, F. Krauss, N. Lavesson, L. Lonnblad, M. Mangano, A. Schalicke et al., *Matching parton showers and matrix elements*, in *HERA and the LHC: A Workshop on the Implications of HERA for LHC Physics: CERN - DESY Workshop 2004/2005 (Midterm Meeting, CERN, 11-13 October 2004; Final Meeting, DESY, 17-21 January 2005)*, pp. 288–289, 2005. hep-ph/0602031. DOI.

[269] I. W. Stewart, F. J. Tackmann, J. R. Walsh and S. Zuberi, *Jet p_T resummation in Higgs production at $NNLL' + NNLO$*, *Phys. Rev. D* **89** (2014) 054001, [1307.1808].

[270] M. Cacciari, F. A. Dreyer, A. Karlberg, G. P. Salam and G. Zanderighi, *Fully Differential Vector-Boson-Fusion Higgs Production at Next-to-Next-to-Leading Order*, *Phys. Rev. Lett.* **115** (2015) 082002, [1506.02660].

[271] J. Lindert et al., *Precise predictions for $V+$ jets dark matter backgrounds*, *Eur. Phys. J. C* **77** (2017) 829, [1705.04664].

[272] C. Oleari and D. Zeppenfeld, *QCD corrections to electroweak nu(l) j j and l+ l- j j production*, *Phys. Rev. D* **69** (2004) 093004, [hep-ph/0310156].

[273] J. Ren, L. Wu and J. M. Yang, *Unveiling CP property of top-Higgs coupling with graph neural networks at the LHC*, *Phys. Lett. B* **802** (2020) 135198, [1901.05627].

[274] A. Mullin, H. Pacey, M. Parker, M. White and S. Williams, *Does SUSY have friends? A new approach for LHC event analysis*, 1912.10625.

[275] BaBar collaboration, D. Boutigny et al., *The BABAR physics book: Physics at an asymmetric B factory*. 10, 1998.

[276] N. Tishby and N. Zaslavsky, *Deep learning and the information bottleneck principle*, in *2015 IEEE Information Theory Workshop (ITW)*, pp. 1–5, IEEE, 2015. 1503.02406.

[277] P. Mehta and D. J. Schwab, *An exact mapping between the variational renormalization group and deep learning*, 1410.3831.

[278] Y. E. Nesterov, *A method for solving the convex programming problem with convergence rate $o(1/k^2)$*, *Dokl. Akad. Nauk SSSR* **269** (1983) 543–547.

[279] E. M. Metodiev, B. Nachman and J. Thaler, *Classification without labels: Learning from mixed samples in high energy physics*, *JHEP* **10** (2017) 174, [1708.02949].

[280] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel et al., *Scikit-learn: Machine learning in Python*, *Journal of Machine Learning Research* **12** (2011) 2825–2830.

[281] T. Junk, *Confidence level computation for combining searches with small statistics*, *Nucl. Instrum. Meth. A* **434** (1999) 435–443, [hep-ex/9902006].

[282] A. L. Read, *Presentation of search results: the CLs technique*, *Journal of Physics G: Nuclear and Particle Physics* **28** (sep, 2002) 2693–2704.

[283] G. Cowan, K. Cranmer, E. Gross and O. Vitells, *Asymptotic formulae for likelihood-based tests of new physics*, *Eur. Phys. J. C* **71** (2011) 1554, [1007.1727].

[284] A. Wald, *Tests of statistical hypotheses concerning several parameters when the number of observations is large*, *Transactions of the American Mathematical Society* **54** (1943) 426–482.

[285] ROOT collaboration, K. Cranmer, G. Lewis, L. Moneta, A. Shibata and W. Verkerke, *HistFactory: A tool for creating statistical models for use with RooFit and RooStats*, .

[286] L. Moneta, K. Belasco, K. S. Cranmer, S. Kreiss, A. Lazzaro, D. Piparo et al., *The RooStats Project*, *PoS* **ACAT2010** (2010) 057, [1009.1003].

[287] J. Arjona Martínez, O. Cerri, M. Pierini, M. Spiropulu and J.-R. Vlimant, *Pileup mitigation at the Large Hadron Collider with graph neural networks*, *Eur. Phys. J. Plus* **134** (2019) 333, [1810.07988].

[288] P. T. Komiske, E. M. Metodiev and J. Thaler, *Metric Space of Collider Events*, *Phys. Rev. Lett.* **123** (2019) 041801, [1902.02346].

[289] P. T. Komiske, E. M. Metodiev and J. Thaler, *The Hidden Geometry of Particle Collisions*, *JHEP* **07** (2020) 006, [2004.04159].

[290] M. Cacciari and G. P. Salam, *Pileup subtraction using jet areas*, *Phys. Lett. B* **659** (2008) 119–126, [0707.1378].

[291] P. Berta, M. Spousta, D. W. Miller and R. Leitner, *Particle-level pileup subtraction for jets and jet shapes*, *JHEP* **06** (2014) 092, [1403.3108].

[292] CMS collaboration, V. Khachatryan et al., *Jet energy scale and resolution in the CMS experiment in pp collisions at 8 TeV*, *JINST* **12** (2017) P02014, [1607.03663].

[293] B. Cabouat and T. Sjöstrand, *Some Dipole Shower Studies*, *Eur. Phys. J. C* **78** (2018) 226, [1710.00391].

[294] A. Ballestrero et al., *Precise predictions for same-sign W-boson scattering at the LHC*, *Eur. Phys. J. C* **78** (2018) 671, [1803.07943].

[295] B. Jäger, A. Karlberg, S. Plätzer, J. Scheller and M. Zaro, *Parton-shower effects in Higgs production via Vector-Boson Fusion*, *Eur. Phys. J. C* **80** (2020) 756, [2003.12435].

[296] CMS collaboration, A. Tumasyan et al., *Evidence for WW/WZ vector boson scattering in the decay channel $\ell\nu qq$ produced in association with two jets in proton-proton collisions at $\sqrt{s}$ = 13 TeV*, 2112.05259.

[297] CMS collaboration, *Search for Higgs boson pair production via vector boson fusion with highly Lorentz-boosted Higgs bosons in the four b quark final state at $\sqrt{s} = 13$ TeV*, .

[298] CMS collaboration, A. M. Sirunyan et al., *Search for nonresonant Higgs boson pair production in final states with two bottom quarks and two photons in proton-proton collisions at $\sqrt{s}$ = 13 TeV*, *JHEP* **03** (2021) 257, [2011.12373].

[299] G. Gustafson, *Dual description of a confined colour field*, *Physics Letters B* **175** (1986) 453–456.

[300] G. Gustafson and U. Pettersson, *Dipole formulation of qcd cascades*, *Nuclear Physics B* **306** (1988) 746–758.

[301] S. Schumann and F. Krauss, *A Parton shower algorithm based on Catani-Seymour dipole factorisation*, *JHEP* **03** (2008) 038, [0709.1027].

[302] S. Platzer and S. Gieseke, *Coherent Parton Showers with Local Recoils*, *JHEP* **01** (2011) 024, [0909.5593].

[303] J. Butterworth et al., *PDF4LHC recommendations for LHC Run II*, *J. Phys. G* **43** (2016) 023001, [1510.03865].

[304] A. Buckley, J. Ferrando, S. Lloyd, K. Nordström, B. Page, M. Rüfenacht et al., *LHAPDF6: parton density access in the LHC precision era*, *Eur. Phys. J. C* **75** (2015) 132, [1412.7420].

[305] S. Forte, *Parton distributions at the dawn of the LHC*, *Acta Phys. Polon. B* **41** (2010) 2859–2920, [1011.5247].

[306] S. Dulat, T.-J. Hou, J. Gao, M. Guzzi, J. Huston, P. Nadolsky et al., *New parton distribution functions from a global analysis of quantum chromodynamics*, *Phys. Rev. D* **93** (2016) 033006, [1506.07443].

[307] L. A. Harland-Lang, A. D. Martin, P. Motylinski and R. S. Thorne, *Parton distributions in the LHC era: MMHT 2014 PDFs*, *Eur. Phys. J. C* **75** (2015) 204, [1412.3989].

[308] NNPDF collaboration, R. D. Ball et al., *Parton distributions for the LHC Run II*, *JHEP* **04** (2015) 040, [1410.8849].

[309] S. Carrazza, S. Forte, Z. Kassabov, J. I. Latorre and J. Rojo, *An Unbiased Hessian Representation for Monte Carlo PDFs*, *Eur. Phys. J. C* **75** (2015) 369, [1505.06736].

[310] P. Artoisenet, R. Frederix, O. Mattelaer and R. Rietkerk, *Automatic spin-entangled decays of heavy resonances in Monte Carlo simulations*, *JHEP* **03** (2013) 015, [1212.3460].

[311] https://pythia.org/latest-manual/POWHEGMerging.html.

[312] M. Mangano, *The so-called MLM prescription for ME/PS matching*, *Fermilab ME/MC Tuning Workshop, October 4, 2002*, *http://www-cpd.fnal.gov/personal/mrenna/tuning/nov2002/mlm.pdf.gz* (2002) .

[313] M. L. Mangano, M. Moretti, F. Piccinini and M. Treccani, *Matching matrix elements and shower evolution for top-quark production in hadronic collisions*, *JHEP* **01** (2007) 013, [hep-ph/0611129].

[314] P. T. Komiske, E. M. Metodiev and J. Thaler, *Energy Flow Networks: Deep Sets for Particle Jets*, *JHEP* **01** (2019) 121, [1810.05165].

[315] M. J. Dolan and A. Ore, *Equivariant Energy Flow Networks for Jet Tagging*, *Phys. Rev. D* **103** (2021) 074022, [2012.00964].

[316] P. Komiske, E. Metodiev and J. Thaler, *Pythia8 quark and gluon jets for energy flow*, May, 2019. 10.5281/zenodo.3164691.

[317] G. Kasieczka, T. Plehn, J. Thompson and M. Russel, *Top quark tagging reference dataset*, Mar., 2019. 10.5281/zenodo.2603256.

[318] C. K. Khosa and V. Sanz, *Anomaly Awareness*, 2007.14462.

[319] T. Cheng, J.-F. Arguin, J. Leissner-Martin, J. Pilette and T. Golling, *Variational Autoencoders for Anomalous Jet Tagging*, 2007.01850.

[320] G. Kasieczka, B. Nachman, D. Shih, O. Amram, A. Andreassen, K. Benkendorfer et al., *The lhc olympics 2020: A community challenge for anomaly detection in high energy physics*, 2101.08320.

[321] T. N. Kipf and M. Welling, *Variational graph auto-encoders*, 1611.07308.

[322] P. V. Tran, *Learning to make predictions on graphs with autoencoders*, in *2018 IEEE 5th international conference on data science and advanced analytics (DSAA)*, pp. 237–245, IEEE, 2018. 1802.08352.

[323] G. Salha, R. Hennequin and M. Vazirgiannis, *Simple and effective graph autoencoders with one-hop linear models*, 2001.07614.

[324] S. Pan, R. Hu, G. Long, J. Jiang, L. Yao and C. Zhang, *Adversarially regularized graph autoencoder for graph embedding*, 1802.04407.

[325] J. Park, M. Lee, H. J. Chang, K. Lee and J. Y. Choi, *Symmetric graph convolutional autoencoder for unsupervised graph representation learning*, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6519–6528, 2019. 1908.02441.

[326] T. Sjostrand, S. Mrenna and P. Z. Skands, *PYTHIA 6.4 Physics and Manual*, *JHEP* **05** (2006) 026, [hep-ph/0603175].

[327] ATLAS collaboration, "Performance of shower deconstruction in ATLAS." http://cdsweb.cern.ch/record/1648661, 2, ATLAS-CONF-2014-003.

[328] S. Catani, Y. L. Dokshitzer, M. Olsson, G. Turnock and B. R. Webber, *New clustering algorithm for multi - jet cross-sections in e+ e- annihilation*, *Phys. Lett.* **B269** (1991) 432–438.

[329] B. M. Dillon, D. A. Faroughy, J. F. Kamenik and M. Szewc, *Learning the latent structure of collider events*, *Journal of High Energy Physics* **2020** (Oct, 2020) .

[330] B. Bortolato, B. M. Dillon, J. F. Kamenik and A. Smolkovič, *Bump hunting in latent space*, 2103.06595.

[331] B. M. Dillon, T. Plehn, C. Sauer and P. Sorrenson, *Better latent spaces for better autoencoders*, 2104.08291.

[332] D. P. Kingma and M. Welling, *Auto-encoding variational bayes*, 1312.6114.

[333] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow and B. Frey, *Adversarial autoencoders*, 1511.05644.

[334] G. Patrini, R. van den Berg, P. Forré, M. Carioni, S. Bhargav, M. Welling et al., *Sinkhorn autoencoders*, 1810.01118.

[335] J. Preskill, *Quantum Computing in the NISQ era and beyond*, *Quantum* **2** (Aug., 2018) 79, [1801.00862].

[336] R. P. Feynman, *Simulating physics with computers*, *International Journal of Theoretical Physics* **21** (Jun, 1982) 467–488.

[337] I. M. Georgescu, S. Ashhab and F. Nori, *Quantum Simulation*, *Rev. Mod. Phys.* **86** (2014) 153, [1308.6253].

[338] S. Ramírez-Uribe, A. E. Rentería-Olivo, G. Rodrigo, G. F. R. Sborlini and L. Vale Silva, *Quantum algorithm for Feynman loop integrals*, 2105.08703.

[339] S. Williams, S. Malik, M. Spannowsky and K. Bepari, *A quantum walk approach to simulating parton showers*, 2109.13975.

[340] T. Li, X. Guo, W. K. Lai, X. Liu, E. Wang, H. Xing et al., *Partonic Structure by Quantum Computing*, 2106.03865.

[341] K. Bepari, S. Malik, M. Spannowsky and S. Williams, *Towards a quantum computing algorithm for helicity amplitudes and parton showers*, *Phys. Rev. D* **103** (2021) 076020, [2010.00046].

[342] S. P. Jordan, K. S. M. Lee and J. Preskill, *Quantum Algorithms for Fermionic Quantum Field Theories*, 1404.7115.

[343] J. Preskill, *Simulating quantum field theory with a quantum computer*, *PoS* **LATTICE2018** (2018) 024, [1811.10085].

[344] C. W. Bauer, W. A. de Jong, B. Nachman and D. Provasoli, *Quantum Algorithm for High Energy Physics Simulations*, *Phys. Rev. Lett.* **126** (2021) 062001, [1904.03196].

[345] S. Abel, N. Chancellor and M. Spannowsky, *Quantum computing for quantum tunneling*, *Phys. Rev. D* **103** (2021) 016008, [2003.07374].

[346] S. Abel and M. Spannowsky, *Observing the fate of the false vacuum with a quantum laboratory*, *P. R. X. Quantum.* **2** (2021) 010349, [2006.06003].

[347] Z. Davoudi, N. M. Linke and G. Pagano, *Toward simulating quantum field theories with controlled phonon-ion dynamics: A hybrid analog-digital approach*, *Phys. Rev. Res.* **3** (2021) 043072, [2104.09346].

[348] A. Mott, J. Job, J.-R. Vlimant, D. Lidar and M. Spiropulu, *Solving a higgs optimization problem with quantum annealing for machine learning*, *Nature* **550** (Oct, 2017) 375–379.

[349] A. Blance and M. Spannowsky, *Unsupervised Event Classification with Graphs on Classical and Photonic Quantum Computers*, *JHEP* **21** (2020) 170, [2103.03897].

[350] S. L. Wu et al., *Application of quantum machine learning using the quantum variational classifier method to high energy physics analysis at the LHC on IBM quantum computer simulator and hardware with 10 qubits*, *J. Phys. G* **48** (2021) 125003, [2012.11560].

[351] A. Blance and M. Spannowsky, *Quantum machine learning for particle physics using a variational quantum classifier*, *JHEP* **2021** (Feb, 2021) 212, [2010.07335].

[352] S. Abel, A. Blance and M. Spannowsky, *Quantum Optimisation of Complex Systems with a Quantum Annealer*, 2105.13945.

[353] S. L. Wu et al., *Application of quantum machine learning using the quantum kernel algorithm on high energy physics analysis at the LHC*, *Phys. Rev. Res.* **3** (2021) 033221, [2104.05059].

[354] S. Y.-C. Chen, T.-C. Wei, C. Zhang, H. Yu and S. Yoo, *Hybrid Quantum-Classical Graph Convolutional Network*, 2101.06189.

[355] K. Terashi, M. Kaneda, T. Kishimoto, M. Saito, R. Sawada and J. Tanaka, *Event Classification with Quantum Machine Learning in High-Energy Physics*, *Comput. Softw. Big Sci.* **5** (2021) 2, [2002.09935].

[356] I. H. Sarker, *Machine learning: Algorithms, real-world applications and research directions*, *SN Computer Science* **2** (Mar, 2021) 160.

[357] J. Stokes, J. Izaac, N. Killoran and G. Carleo, *Quantum Natural Gradient*, *Quantum* **4** (May, 2020) 269.

[358] B. M. Dillon, T. Plehn, C. Sauer and P. Sorrenson, *Better Latent Spaces for Better Autoencoders*, 2104.08291.

[359] B. Bortolato, B. M. Dillon, J. F. Kamenik and A. Smolkovič, *Bump Hunting in Latent Space*, 2103.06595.

[360] B. M. Dillon, D. A. Faroughy and J. F. Kamenik, *Uncovering latent jet substructure*, *Phys. Rev. D* **100** (2019) 056002, [1904.04200].

[361] M. Benedetti, E. Lloyd, S. Sack and M. Fiorentini, *Parameterized quantum circuits as machine learning models*, *Quantum Science and Technology* **4** (Nov, 2019) 043001.

[362] R. LaRose and B. Coyle, *Robust data encodings for quantum classifiers*, *Physical Review A* **102** (Sep, 2020) .

[363] G. Fubini, *Sulle metriche definite da una forme hermitiana*, *Atti del Reale Istituto Veneto di Scienze, Lettere ed Arti* **63** (1904) 502–513.

[364] E. Study, *Kürzeste wege im komplexen gebiet*, *Mathematische Annalen* **60** (Sep, 1905) 321–378.

[365] S. Amari, *Natural Gradient Works Efficiently in Learning*, *Neural Computation* **10** (02, 1998) 251–276, [https://direct.mit.edu/neco/article-pdf/10/2/251/813415/089976698300017746.pdf].

[366] A. Dresden, *The fourteenth western meeting of the American Mathematical Society*, *Bulletin of the American Mathematical Society* **26** (1920) 385 – 396.

[367] R. Penrose, *A generalized inverse for matrices*, *Mathematical Proceedings of the Cambridge Philosophical Society* **51** (1955) 406–413.

[368] J. Romero, J. P. Olson and A. Aspuru-Guzik, *Quantum autoencoders for efficient compression of quantum data*, *Quantum Science and Technology* **2** (2017) 045001, [1612.02806].

[369] K. Kottmann, F. Metz, J. Fraxanet and N. Baldelli, *Variational Quantum Anomaly Detection: Unsupervised mapping of phase diagrams on a physical quantum computer*, *Phys. Rev. Res.* **3** (2021) 043184, [2106.07912].

[370] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, M. S. Alam, S. Ahmed et al., *Pennylane: Automatic differentiation of hybrid quantum-classical computations*, 1811.04968.

[371] H. Buhrman, R. Cleve, J. Watrous and R. de Wolf, *Quantum fingerprinting*, *Phys. Rev. Lett.* **87** (Sep, 2001) 167902.

[372] M. S. A. et. al, *Qiskit: An open-source framework for quantum computing*, .

[373] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 1412.6980.

[374] D. Ristè, M. P. da Silva, C. A. Ryan, A. W. Cross, A. D. Córcoles, J. A. Smolin et al., *Demonstration of quantum advantage in machine learning*, *npj Quantum Information* **3** (Apr, 2017) 16.

[375] H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven et al., *Power of data in quantum machine learning*, *Nature Communications* **12** (May, 2021) 2631.

[376] M. Czakon, P. Fiedler and A. Mitov, *Total Top-Quark Pair-Production Cross Section at Hadron Colliders Through $O(\alpha_S^4)$*, *Phys. Rev. Lett.* **110** (2013) 252004, [1303.6254].

[377] F. Psihas, M. Groh, C. Tunnell and K. Warburton, *A Review on Machine Learning for Neutrino Experiments*, *Int. J. Mod. Phys. A* **35** (2020) 2043005, [2008.01242].

[378] R. Abbasi et al., *A Convolutional Neural Network based Cascade Reconstruction for the IceCube Neutrino Observatory*, *JINST* **16** (2021) P07041, [2101.11589].

[379] G. Karagiorgi, G. Kasieczka, S. Kravitz, B. Nachman and D. Shih, *Machine Learning in the Search for New Fundamental Physics*, 2112.03769.

[380] C. Escamilla-Rivera, M. A. C. Quintero and S. Capozziello, *A deep learning approach to cosmological dark energy models*, *JCAP* **03** (2020) 008, [1910.02788].

[381] H. Tilaver, M. Salti, O. Aydogdu and E. E. Kangal, *Deep learning approach to Hubble parameter*, *Comput. Phys. Commun.* **261** (2021) 107809.

[382] G.-J. Wang, S.-Y. Li and J.-Q. Xia, *ECoPANN: A Framework for Estimating Cosmological Parameters using Artificial Neural Networks*, *Astrophys. J. Suppl.* **249** (2020) 25, [2005.07089].

[383] R. Arjona and S. Nesseris, *What can Machine Learning tell us about the background expansion of the Universe?*, *Phys. Rev. D* **101** (2020) 123525, [1910.01529].

[384] R. Biswas et al., *Application of machine learning algorithms to the study of noise artifacts in gravitational-wave data*, *Phys. Rev. D* **88** (2013) 062003, [1303.6984].

[385] M. D. Morales, J. M. Antelis, C. Moreno and A. I. Nesterov, *Deep learning for gravitational-wave data analysis: A resampling white-box approach*, *Sensors* **21** (2021) 3174, [2009.04088].

[386] M. B. Schäfer and A. H. Nitz, *From one to many: A deep learning coincident gravitational-wave search*, *Phys. Rev. D* **105** (2022) 043003, [2108.10715].

[387] C. K. Khosa, L. Mars, J. Richards and V. Sanz, *Convolutional Neural Networks for Direct Detection of Dark Matter*, *J. Phys. G* **47** (2020) 095201, [1911.09210].

[388] J. Herrero-Garcia, R. Patrick and A. Scaffidi, *A semi-supervised approach to dark matter searches in direct detection data with machine learning*, *JCAP* **02** (2022) 039, [2110.12248].

[389] C. Englert, P. Galler, P. Harris and M. Spannowsky, *Machine Learning Uncertainties with Adversarial Neural Networks*, *Eur. Phys. J. C* **79** (2019) 4, [1807.08763].

[390] A. Ghosh, B. Nachman and D. Whiteson, *Uncertainty-aware machine learning for high energy physics*, *Phys. Rev. D* **104** (2021) 056026, [2105.08742].

[391] A. Ghosh and B. Nachman, *A cautionary tale of decorrelating theory uncertainties*, *Eur. Phys. J. C* **82** (2022) 46, [2109.08159].

[392] R. T. d'Agnolo, G. Grosso, M. Pierini, A. Wulzer and M. Zanetti, *Learning new physics from an imperfect machine*, *Eur. Phys. J. C* **82** (2022) 275, [2111.13633].

[393] D. Lange, K. Bloom, T. Boccali, O. Gutsche and E. Vaandering, *CMS Computing Resources: Meeting the Demands of the High-Luminosity LHC Physics Program*, *EPJ Web Conf.* **214** (2019) 03055.

[394] A. Collaboration, *ATLAS Software and Computing HL-LHC Roadmap*, tech. rep., CERN, Geneva, Mar, 2022.

[395] HSF Physics Event Generator WG collaboration, E. Yazgan et al., *HL-LHC Computing Review Stage-2, Common Software Projects: Event Generators*, 2109.14938.

[396] E. Bothmann, T. Janßen, M. Knobbe, T. Schmale and S. Schumann, *Exploring phase space with Neural Importance Sampling*, *SciPost Phys.* **8** (2020) 069, [2001.05478].

[397] K. Danziger, T. Janßen, S. Schumann and F. Siegert, *Accelerating Monte Carlo event generation – rejection sampling using neural network event-weight estimates*, *SciPost Phys.* **12** (2022) 164, [2109.11964].

[398] J. Aylett-Bullock, S. Badger and R. Moodie, *Optimising simulations for diphoton production at hadron colliders using amplitude neural networks*, *JHEP* **08** (2021) 066, [2106.09474].

[399] R. Winterhalder, V. Magerya, E. Villa, S. P. Jones, M. Kerner, A. Butter et al., *Targeting multi-loop integrals with neural networks*, *SciPost Phys.* **12** (2022) 129, [2112.09145].

[400] S. Badger, A. Butter, M. Luchmann, S. Pitz and T. Plehn, *Loop Amplitudes from Precision Networks*, 2206.14831.

[401] A. Butter, T. Heimel, S. Hummerich, T. Krebs, T. Plehn, A. Rousselot et al., *Generative Networks for Precision Enthusiasts*, 2110.13632.

[402] S. van der Walt, S. C. Colbert and G. Varoquaux, *The numpy array: A structure for efficient numerical computation*, *Computing in Science Engineering* **13** (2011) 22–30.

[403] T. O'Malley, E. Bursztein, J. Long, F. Chollet, H. Jin, L. Invernizzi et al., "Keras Tuner." https://github.com/keras-team/keras-tuner, 2019.