

# Autoencoder-Based Anomaly Detection in Collider Experiments

A thesis submitted in partial fulfillment of the requirements  
for the degree of  
**Bachelor of Science (Research) in Physics**

*by*

**Gaurav Niraj Rachh**

*Thesis Supervisors:*

Prof. Biplob Bhattacherjee  
Dr. Swagata Mukherjee



Centre for High Energy Physics  
Indian Institute of Science, Bangalore  
Academic Year 2024–2025

# Certificate

This is to certify that the Bachelor's thesis titled **Autoencoder-Based Anomaly Detection in Collider Experiments** submitted by **Gaurav Rachh** (Sr No. 11-01-00-10-91-21-1-20632) to Indian Institute of Science, Bangalore towards partial fulfilment of requirements for the award of degree of Bachelor of Science (Research) in Physics is a record of bonafide work carried out by him under our supervision and guidance during Academic Year, 2024-25.

*Biplob Bhattacharya*

---

Professor Biplob Bhattacherjee  
Centre for High Energy Physics  
Indian Institute of Science  
Bangalore-560012, India

# **Declaration**

I certify that

1. The work contained in this report has been done by me under the guidance of my supervisors.
2. The work has not been submitted to any other Institute for any degree or diploma.
3. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.
4. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources or people, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.

**April 11, 2025**

**Gaurav Rachh  
IISc Bangalore, India**

A handwritten signature in black ink, appearing to read "Gaurav Rachh".

## Acknowledgments

I would like to express my sincere gratitude to Prof. Biplob Bhattacherjee for his constant support and invaluable guidance throughout the course of this project. I am also deeply thankful to Swagata Ma'am for her insightful guidance in the field of machine learning, and to Camelia Ma'am for her support and helpful discussions related to jet image processing and ML applications.

I would also like to thank my project partner, Tejas, for his collaboration, discussions, and help at various stages of this work.

Finally, I am grateful to my family for their unwavering support and encouragement, and to all my friends whose companionship and motivation made this journey all the more enjoyable.

*Dedicated with love and gratitude  
to my elder sister, Harita.*

# Abstract

Colliders such as the LHC are robust experimental tools for examining Standard Model particle interactions and for exploring phenomena beyond the Standard Model (BSM). To mimic and simulate these processes, a set of computational frameworks, including MadGraph, Pythia, Delphes, and ROOT, are used. This thesis utilises these tools to simulate jet images produced in high-energy proton-proton collisions across various energy scales. The second half of the work explores the application of machine learning, specifically autoencoders, for anomaly detection. The machine learning model learns to reconstruct jet images; the autoencoder's reconstruction error can be interpreted as an anomaly score, used to flag jets that deviate from expected patterns. Our analysis centres on top quark decays, using convolutional autoencoders to distinguish top jets from the QCD background. This thesis mainly employs a semi-supervised learning method to flag top jets as anomalies and lays the groundwork for unsupervised learning. Overall, this work merges high-energy physics simulations with modern machine learning techniques, offering insights into both collider phenomenology and unsupervised anomaly detection strategies.

# Contents

List of Figures . . . . .	xii
List of Tables . . . . .	xiii
Abbreviations . . . . .	xiv
<b>1 Particle Physics and Fundamental Forces</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Standard Model . . . . .	1
1.3 Interactions . . . . .	3
1.4 Jets . . . . .	6
1.4.1 Generalized $k_t$ algorithm . . . . .	7
1.4.2 Prong Finders . . . . .	8
1.5 BSM physics . . . . .	8
1.5.1 Beyond the Standard Model: Heavy Gauge Boson $Z'$ . . . . .	9
<b>2 Collider Physics and Experiments</b>	<b>10</b>
2.1 Introduction . . . . .	10
2.2 Exploring Colliders . . . . .	10
2.2.1 Important Features . . . . .	11
2.3 Large Hadron Collider . . . . .	11
2.3.1 Collision event . . . . .	12
2.3.2 Detectors . . . . .	13
2.4 Collider Physics . . . . .	15
2.4.1 Madlestram Variables . . . . .	15
2.4.2 Pseudorapidity . . . . .	16
<b>3 Computational Framework and Simulations</b>	<b>18</b>
3.1 Introduction . . . . .	18
3.2 Madgraph . . . . .	18
3.3 PYTHIA . . . . .	18
3.4 Delphes . . . . .	19
3.5 ROOT . . . . .	19
3.6 Jet Image Generation Pipeline . . . . .	19
3.6.1 Event Generation Using MadGraph . . . . .	19

3.6.2	Parton Showering and Detector Simulation . . . . .	20
3.6.3	Conversion to ROOT Format . . . . .	20
3.6.4	Data Extraction Using C++ . . . . .	20
3.6.5	Jet Image Construction . . . . .	20
3.7	Problem definition . . . . .	26
3.7.1	Feynman Diagrams for Signal and Background Processes . . . . .	27
<b>4</b>	<b>Machine learning</b>	<b>29</b>
4.1	Introduction . . . . .	29
4.2	Deep learning . . . . .	29
4.2.1	Logistic Regression . . . . .	30
4.2.2	Convolutional Neural Networks . . . . .	32
4.3	Supervised and Unsupervised learning . . . . .	34
4.3.1	Supervised Learning . . . . .	34
4.3.2	Unsupervised Learning . . . . .	34
4.4	AutoEncoders . . . . .	35
4.4.1	Encoders . . . . .	35
4.4.2	Decoders . . . . .	36
4.5	Activation Function of the Output Layer . . . . .	36
4.5.1	ReLU (Rectified Linear Unit) . . . . .	36
4.5.2	Sigmoid . . . . .	37
4.5.3	Linear . . . . .	37
4.5.4	Tanh (Hyperbolic Tangent) . . . . .	38
4.6	Loss Function and Training Objective . . . . .	39
4.6.1	Mean Squared Error (MSE) . . . . .	39
4.6.2	Mean Absolute Error (MAE) . . . . .	40
4.6.3	Structural Similarity Index (SSIM) . . . . .	41
4.6.4	Final Objective . . . . .	41
4.7	Model Evaluation Metrics . . . . .	41
4.7.1	Receiver Operating Characteristic (ROC) Curve . . . . .	41
4.7.2	Signal Efficiency vs. Background Rejection . . . . .	42
4.7.3	Confusion Matrix . . . . .	42
4.8	Reconstructing MNIST Data . . . . .	43
4.9	Visualization . . . . .	43
4.10	Anomaly Detection . . . . .	44
<b>5</b>	<b>Anomaly Detection in Jet Images Using Autoencoders</b>	<b>45</b>
5.1	Introduction . . . . .	45
5.1.1	Jet Images and Physics Motivation . . . . .	46
5.1.2	Autoencoder-Based Anomaly Detection . . . . .	46
5.1.3	Chapter Overview . . . . .	46
5.2	Convolutional Autoencoder Architecture . . . . .	47

5.2.1	Training Setup . . . . .	48
5.2.2	Training Setup and Model Configuration . . . . .	48
5.2.3	Reconstructions of the Jet Images . . . . .	48
5.3	Model Evaluation and Anomaly Scoring using MSE . . . . .	50
5.4	Model Evaluation and Anomaly Scoring using MAE . . . . .	56
5.5	Model Evaluation and Anomaly Scoring using Structural Similarity (SSIM) . . . . .	60
5.6	Conclusion . . . . .	64
<b>A</b>	<b>Youden's J Threshold</b>	<b>66</b>
<b>B</b>	<b>Model Evaluation and Anomaly Scoring for different <math>p_T</math> values</b>	<b>67</b>
<b>C</b>	<b>Unsupervised Learning</b>	<b>72</b>
C.0.1	Autoencoder architecture . . . . .	73
<b>D</b>	<b>Model Evaluation and Anomaly Scoring for Higgs process</b>	<b>74</b>
	References . . . . .	78

# List of Figures

1.1	Standard Model; [1]	2
1.2	As we can observe, at sufficiently high energies, the coupling constants of these forces approach a common value, suggesting the existence of a hypothetical unified theory of forces.[2]	4
1.3	Feynman diagram for QED interactions	4
1.4	Feynman diagram for QCD interactions	5
1.5	Feynman diagram for Weak interactions	6
1.6	Feynman diagram for Higgs interactions	6
1.7	Pictorial representation of proton-proton collision forming a BSM particle which decays to a SM/BSM particles[5]	8
2.1	Visualization of a Hadronic Collision at the LHC[9]	12
2.2	CMS detector cross-section[11]	14
2.3	Two particles with momenta $p_1$ and $p_2$ interact, producing two outgoing particles with momenta $p_3$ and $p_4$ .[12]	15
2.4	The letters s, t, and u are also used in the terms s-channel (timelike channel), t-channel, and u-channel (both spacelike channels).	15
2.5	Here is the pictorial representation of the Madlestram variables.	15
2.6	Pictorial representation of pseudorapidity.[1]	16
3.1	Raw jet images, where each jet is represented as a heatmap in the $\eta$ - $\phi$ plane, with pixel intensity proportional to the transverse momentum ( $p_T$ ).	21
3.2	Jet image representations after preprocessing. The constituent with the highest transverse momentum ( $p_T$ ) in each jet has been translated to $\phi = 0$ and its $p_T$ normalized to 1. Each 2D histogram shows the spatial distribution of jet constituents in the $(\eta, \phi)$ plane, weighted by normalized $p_T$ values. This standardization centers and scales jets for consistent input to machine learning model	22
3.3	Side-by-side visualization of two preprocessed jet images. The $(\eta, \phi)$ coordinates represent constituent positions, colored by their normalized transverse momentum ( $p_T$ ). Each image has been centered around the constituent with maximum $p_T$ at $\phi = 0$ , and all $p_T$ values have been normalized, enabling consistent comparison across jets.	22

3.4	Jet images after rotational alignment using the principal axis of energy flow. The jets are rotated such that their maximum variance direction (principal component) lies along a fixed orientation, improving consistency across samples. This step enhances the performance of machine learning models by removing rotational ambiguity. . . . .	23
3.5	Jet images after applying mirror flipping to enforce a consistent orientation. Jets are flipped such that the side with greater transverse momentum ( $p_T$ ) appears on the left in $\eta$ and the top in $\phi$ . This normalization reduces symmetry-induced variation in training. . . . .	23
3.6	Visualization of 20 jet images represented in the $\eta$ - $\phi$ plane. Each image is constructed using a 2D histogram where the color intensity reflects the transverse momentum ( $p_T$ ) of constituents. These jet images serve as the raw input to preprocessing and machine learning models. . . . .	24
3.7	Construction of jet images from calorimeter data. Constituent particles' transverse momenta are projected onto a 2D $\eta$ - $\phi$ grid, forming a pixelated image. The image is then cropped to a fixed central region and normalized such that the total intensity sums to 1. . . . .	24
3.8	Average jet image computed from the training dataset. The image highlights the typical spatial distribution of transverse momentum ( $p_T$ ) across the $\eta$ - $\phi$ plane. High-intensity regions correspond to more frequently occupied calorimeter cells by jet constituents. . . . .	25
3.9	Feynman diagram for the QCD process, where two high energy protons collide to give to quark jets . . . . .	27
3.10	Representative Feynman diagrams showing (a) top quark pair production via gluon fusion, (b) top quark decay into a W boson and a b quark, and (c) subsequent W boson decay into a quark-antiquark pair. . . . .	28
3.11	Representative Feynman diagrams showing (a) top quark pair production via gluon fusion, (b) top quark decay into a W boson and a b quark, and (c) subsequent W boson decay into a quark-antiquark pair. . . . .	28
4.1	X is a set of tennis racquets with the features described on the neurons, and we also have a hidden layer in the neural network that represents the player's ability; Y will give the best racquet for a player based on the requirement and the level of their game. ( <b>It is a fully connected neural structure; however, I have not connected each neuron in the diagram to clarify and avoid visual clutter.</b> ) . . . . .	30
4.2	CNN architecture[24] . . . . .	32
4.3	Example of convolution operation after pooling. As the input vector undergoes the pooling layer, its spatial dimensions get reduced, resulting in a pooled vector. A kernel is then convolved with the pooled vector to compute the destination pixel. The kernel underlines directional features, and the final value (-8) reflects the weighted sum of overlapping elements.[23] . . . . .	34
4.4	Autoencoder architecture . . . . .	35

4.5	Graph for the ReLU activation function[27] . . . . .	36
4.6	Graph for the ReLU activation function[27] . . . . .	37
4.7	Graph for the linear activation function[27] . . . . .	37
4.8	Graph for the tanh activation function[27] . . . . .	38
4.9	Reconstruction of MNIST dataset . . . . .	43
5.1	Architecture of the convolutional autoencoder used for anomaly detection in jet images. The input is a grayscale image of size $40 \times 40 \times 1$ that passes through the encoder composed of successive Conv2D and MaxPooling2D layers, progressively reducing the spatial dimensions while increasing feature depth, leading to a compressed latent representation. The decoder then reconstructs the original image using Conv2D and UpSampling2D layers. This reconstruction is used to differentiate between QCD and anomalous (Top) jets based on the reconstruction error. . . . .	47
5.2	Sample QCD jet images used as input to the autoencoder. Grayscale, resolution: $40 \times 40 \times 1$ . . . . .	48
5.3	QCD jet images at $p_T = 450$ GeV. These images represent typical two-prong structures. . . . .	49
5.4	Top jet images at $p_T = 450$ GeV. These images represent typical three-prong structures. . . . .	49
5.5	Reconstructed Top jet images from the trained autoencoder. . . . .	49
5.6	Training and validation loss curve for the convolutional autoencoder over 50 epochs. The loss is measured using Mean Squared Error (MSE) between the input and reconstructed QCD jet images. The smooth convergence of both training and validation loss indicates that the model has learned to effectively reconstruct QCD-like jets without significant overfitting. . . . .	50
5.7	Left: original QCD image; Middle: reconstruction; Right: pixel-wise squared error. . . . .	51
5.8	Left: original Top image; Middle: reconstruction; Right: pixel-wise squared error. . . . .	51
5.9	Scatter plot of Mean Absolute Error (MAE) scores for QCD and Top jets. The red line indicates the anomaly threshold. . . . .	52
5.10	Scatter plot of Mean Absolute Error (MAE) scores for QCD and Top jets. The red line indicates the anomaly threshold. . . . .	52
5.11	Confusion matrix for anomaly detection using reconstruction error on 450 GeV jets. The autoencoder was trained on QCD (normal) jets and tested on both QCD and top (anomalous) jets. . . . .	54
5.12	ROC curve obtained using MSE-based anomaly score. Area Under Curve (AUC) = 0.8454. . . . .	55
5.13	Signal efficiency ( $\epsilon_S$ ) vs. background rejection ( $1/\epsilon_B$ ). The red marker indicates the chosen anomaly threshold. . . . .	56
5.14	Normal QCD jet: original (left), reconstructed (middle), and MAE heatmap (right). . . . .	57

5.15	Anomalous Top jet: original (left), reconstructed (middle), and MAE heatmap (right). . . . .	57
5.16	Scatter plot of Mean Absolute Error (MAE) scores for QCD and Top jets. The red line indicates the anomaly threshold. . . . .	57
5.17	Confusion matrix showing the classification of QCD (normal) and Top (anomalous) jets at the chosen threshold. . . . .	58
5.18	ROC plot for MAE loss function . . . . .	59
5.19	SSIM visualization for a QCD jet: original (left), reconstruction (middle), and SSIM error map (right). . . . .	60
5.20	SSIM visualization for a Top jet: original (left), reconstruction (middle), and SSIM error map (right). . . . .	60
5.21	Distribution of SSIM scores for QCD and Top jets. Lower SSIM indicates higher anomaly. . . . .	61
5.22	SSIM anomaly score with chosen threshold (red line). . . . .	62
5.23	RROC curve for anomaly detection using SSIM scores . . . . .	62
5.24	SSIM Confusion Matrix . . . . .	63
B.1	Scatter plot of Mean Absolute Error (MSE) scores for QCD and Top jets at 750 GeV. . . . .	67
B.2	Histogram of Mean Squared Error (MSE) scores for QCD and Top jets at 750 GeV. . . . .	68
B.3	Signal Efficiency vs Background rejection for 750 GeV $p_T$ jets. . . . .	68
B.4	Scatter of Mean Squared Error (MSE) scores for QCD and Top jets. The red line indicates the anomaly threshold. . . . .	69
B.5	Histogram of Mean Squared Error (MSE) scores for QCD and Top jets. The red line indicates the anomaly threshold. . . . .	69
B.6	Signal Efficiency vs Background rejection for 1500 GeV $p_T$ jets. . . . .	70
B.7	Scatter plot of Mean Squared Error (MSE) scores for QCD and Top jets. The red line indicates the anomaly threshold. . . . .	70
B.8	Histogram of Mean Squared Error (MSE) scores for QCD and Top jets. The red line indicates the anomaly threshold. . . . .	71
B.9	Signal Efficiency vs Background rejection for 5000 GeV $p_T$ jets . . . . .	71
C.1	Model evaluation for various bottle dimentions as described above in the text. . . . .	72
D.1	Scatter plot of Mean Absolute Error (MSE) scores for QCD and Higgs decay jets at 450 GeV. . . . .	74
D.2	Histogram of Mean Squared Error (MSE) scores for QCD and Higgs decay jets at 450 GeV. . . . .	75
D.3	Signal Efficiency vs Background rejection QCD and Higgs decay jets at 450 GeV. . . . .	75

# List of Tables

3.1	Motivation Behind Jet Image . . . . .	26
4.1	Activation functions for the output layer and their typical use cases. . . . .	39
4.2	Confusion matrix for binary classification. . . . .	42
5.1	from the obtained confusion matrix 5.11 we are calculating accuracy, precision, recall and the F1 score values . . . . .	54
5.2	From the obtained confusion matrix 5.17 we are calculating accuracy, precision, recall and the F1 score values . . . . .	59
5.3	From the obtained confusion matrix 5.24 we are calculating accuracy, precision, recall and the F1 score values . . . . .	64
C.1	Autoencoder architecture with bottleneck size as a variable parameter. . . . .	73
D.1	Performance metrics for anomaly detection based on the confusion matrixD.3. .	76

## Abbreviations

DNN	Deep Neural Network
CNN	Convolutional Neural Network
SM	Standard Model
BSM	Beyond Standard Model
COM	Center Of Mass
QFT	Quantum Field Theory
QED	Quantum Electrodynamics
QCD	Quantum Chromodynamics
ML	Machine Learning
ROC	Receiver Operating Characteristic
AUC	Area Under Curve
MSE	Mean Square Error
MAE	Mean Absolute Error
SSIM	Structural Similarity Index Measure
HCAL	Hadronic Calorimeter
ECAL	Electromagnetic Calorimeter

# Chapter 1

## Particle Physics and Fundamental Forces

### 1.1 Introduction

After the significant breakthrough of discovering the Higgs boson in 2012, the standard model now provides a well-confirmed map of the fundamental particles within the accessible energy scales. However, according to theoretical models at higher energy scales, this may not be the complete picture reason, as we predict the existence of additional particles at higher energy scales. While the current and future experiments will aim to explore physics beyond the standard model or the new physics, it is essential to understand the current framework in detail first. This chapter explains the standard model, fundamental forces, particle physics interactions, particle jets and BSM physics. .

### 1.2 Standard Model

The Standard Model of Particle Physics describes all fundamental particles that can be categorized into fermions, bosons, and hadrons. Fermions, the particles that follow Fermi-Dirac statistics, constitute the building blocks of matter, including quarks and leptons. Bosons are the particles which obey Bose-Einstein statistics and mediate fundamental interactions, such as the photon for electromagnetism and the gluon for the strong force. Conversely, Hadrons are composite particles of quarks and can be either fermionic (such as baryons) or bosonic (such as mesons), depending on their quark composition.

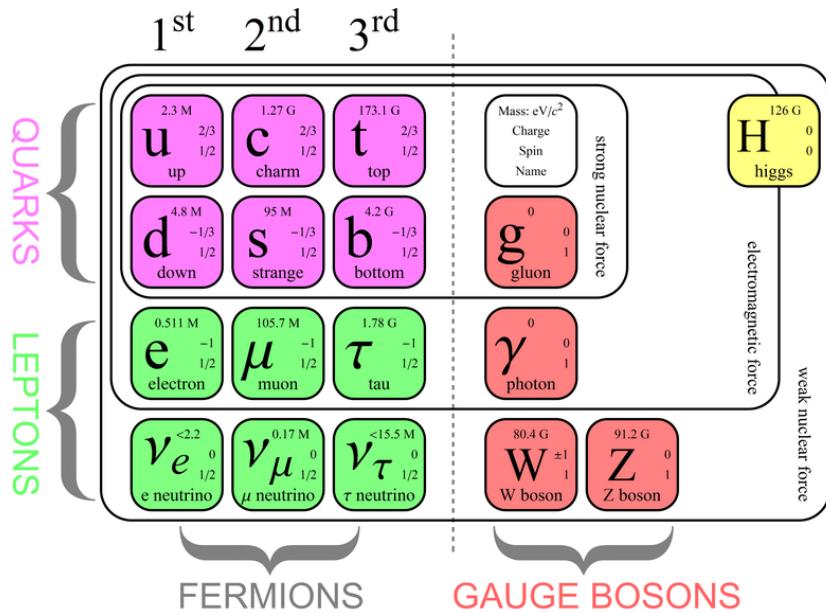


Figure 1.1: Standard Model; [1]

## Fermions

Fermions are spin- $\frac{1}{2}$  particles that obey the Pauli exclusion principle and follow Fermi-Dirac statistics. The Standard Model consists of 12 fermions divided into three generations, each consisting of two quarks and two leptons; these three generations of fermions are categorised in increasing order of mass. Both quarks and leptons carry an electric charge and interact via the electromagnetic force; along with an electrical charge, quarks also possess an additional charge called the *colour charge*, enabling them to participate in the strong interaction mediated via gluons.

So now let us discuss the three generations of the fermions. The first generation consists of up and down quarks, the electron, and the electron neutrino. The second generation consists of charm and strange quarks, the muon, and the muon neutrino, and the third generation consists of the top and bottom quarks, the tau, and the tau neutrino. The first-generation particles are stable under normal conditions and form the most visible matter in the Universe. For example, protons (uud) and neutrons (udd) are colour-neutral combinations of up and down quarks. Higher-generation fermions are significantly more massive than the previous-generation Fermions, and because of that, second and third-generation particles rapidly decay into first-generation particles.

For example, the muon ( $\mu^-$ ) of the second generation decays into an electron ( $e^-$ ), an electron antineutrino ( $\bar{\nu}_e$ ) and a muon neutrino ( $\nu_\mu$ ):

$$\mu^- \rightarrow e^- + \bar{\nu}_e + \nu_\mu$$

This decay process illustrates how unstable particles ultimately transform into the stable first-generation fermions that make up ordinary matter.

## Bosons

Bosons are spin-0 (scalar) and spin-1 (vector or gauge) particles that follow Bose-Einstein statistics. Some of these bosons mediate fundamental forces between the fermions; as we know, there are four fundamental forces: Strong, Weak, Gravitational and Electromagnetic. Gauge bosons are described through gauge theories, whereby the gauge bosons preserve the local gauge symmetries of the specific field.

## Fundamental forces

The **Strong force** is a short-range force that acts upon the colour-changing particles such as quarks and gluons. It is attractive in nature and binds quarks to form hadrons, such as protons and neutrons. It also acts upon the protons and neutrons to form the nuclei. Quantum chromodynamics (QCD) provides the theoretical framework for describing this force. The gluon serves as its mediator, being massless, color-charged and spin-one particle. Interestingly, the name gluon comes from the idea that it holds the quarks like glue to form stable hadrons like protons and neutrons.

The **Weak force** is a short-range interaction that affects all fermions, including neutrinos. It plays a crucial role in various types of radioactive decay, such as alpha and beta decay. In the modern Standard Model, the weak and electromagnetic forces are unified under a single framework known as electroweak (EW) theory. The mediators of the weak force are massive, spin-one particles, with the  $W^\pm$  bosons being charges and Z boson as neutral.

The **Electromagnetic force** is a long-range force that influences electrically charged particles such as quarks and electrons. Depending on the charge signs, it can be attractive or repulsive. Quantum Electrodynamics (QED) provides the theoretical framework for describing this force. The photon ( $\gamma$ ) serves as its mediator, being a massless, chargeless particle with a spin of one. It interacts exclusively with particles that possess an electromagnetic charge.

The **Gravitational force** is a long-range force, attractive and acts upon all the objects with mass and energy. We don't have a unified theory of gravity and QFT, but the theory expects a hypothetical boson with spin 2 to mediate gravitational field at quantum levels.

## 1.3 Interactions

Interactions are one of the most essential parts of particle physics, as they allow us to gather more information about a particular particle. Interactions are described through perturbative QFT, and the calculations are done graphically using Feynman diagrams.

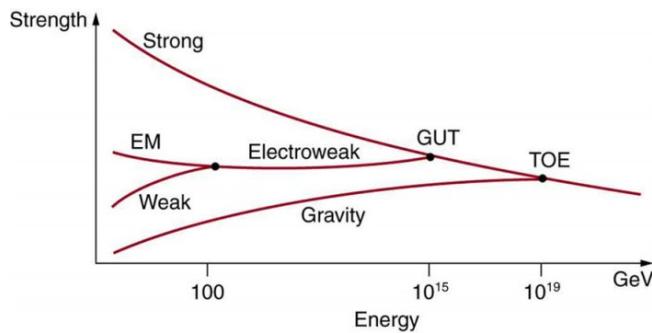


Figure 1.2: As we can observe, at sufficiently high energies, the coupling constants of these forces approach a common value, suggesting the existence of a hypothetical unified theory of forces.[2]

### Quantum Electrodynamics

Quantum Electrodynamics describes the interaction of the charged particle through the **electromagnetic force**. The photon( $\gamma$ ) mediates this force. The theory has  $U(1)$  gauge symmetry, making sure that there is electric charge's local conservation. Because of the **Abelian** nature of  $U(1)$  symmetry, it allows only one force mediator.

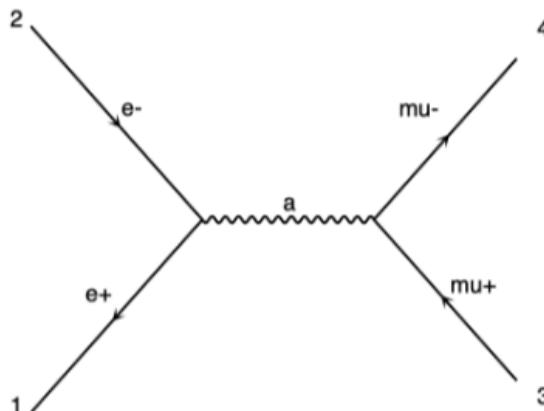


Figure 1.3: Feynman diagram for QED interactions

### Quantum Chromodynamics

Quantum chromodynamics is a **non-Abelian** gauge theory. It describes how particles with colour charge interact through the **strong nuclear force**. Quarks and gluons possess the colour charge property, describing their QCD interactions. We do not see any naturally occurring coloured particles, which says that colourless particles are stable in nature, and when we say colourless particles, we mean the colour charge being zero; for example, protons, neutrons, and mesons are the particles with zero net colour charge. In quantum chromodynamics (QCD), colour confinement, often called confinement, is the phenomenon that color-charged particles (such as quarks and gluons) cannot be isolated and, therefore, cannot be directly observed in normal conditions below the **Hagedorn** temperature of approximately **2 terakelvin** (corresponding to energies of roughly **130–140 MeV** per particle). There are three types of colours: red, blue and green which have absolutely no relation visible colours. The  $SU(3)$  gauge

symmetry ensures the colour charge conservation, unlike QED, where there is only one force mediator, there are 8 different types of gluon to mediate the strong nuclear force. Gluons carry two different type of colour charge; one for emission and one for absorption

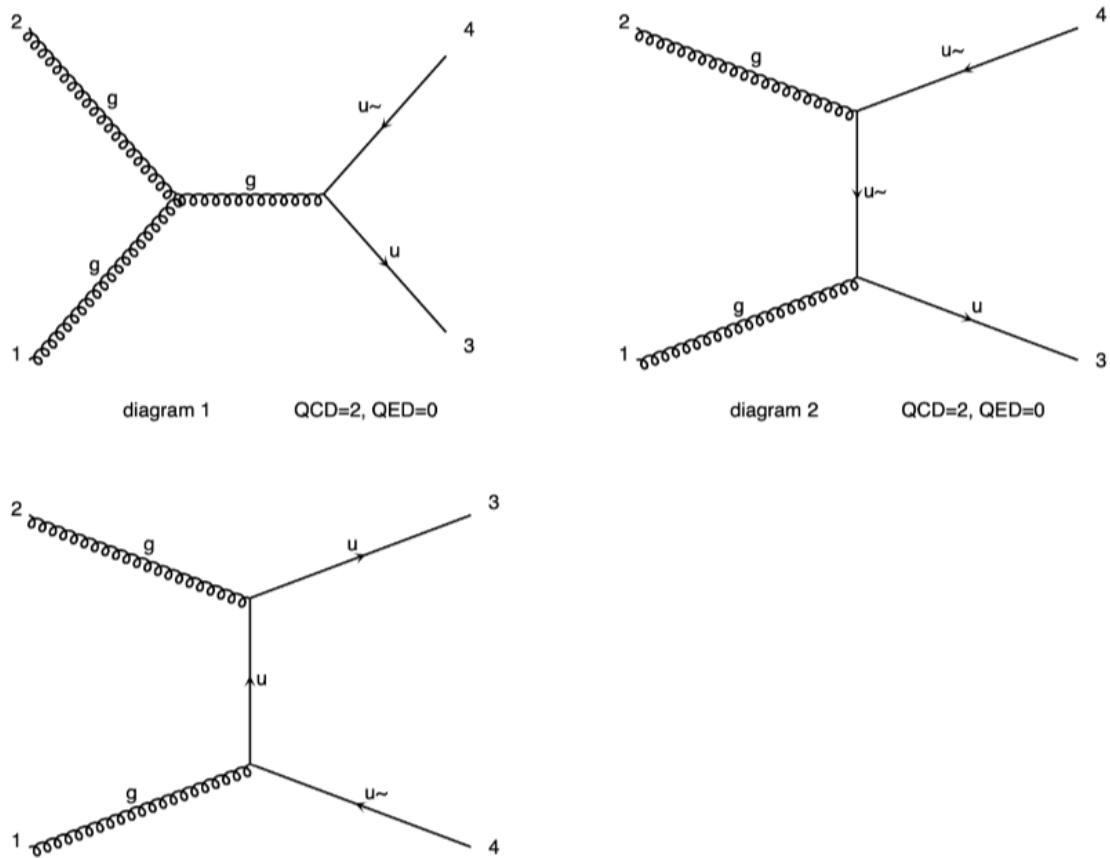


Figure 1.4: Feynman diagram for QCD interactions

## Weak Interactions

Three massive Bosons mediate weak interactions: the W boson ( $W^\pm$ ), and the Z boson; the W boson has a unit electric charge, and the Z boson is neutral, and the mass of W boson is about 90 GeV and the mass of Z boson is about 80 GeV. Since the these bosons are massive, it instantaneously decays into other particles such as quarks, leptons or photons. Weak interactions define particle's flavor changing nature and these interactions are described by  $SU(2)_L$  gauge symmetry, where **L** describes that the symmetry only applies to the particles with left-hand chirality i.e. particles with their spins and momenta pointing in the opposite directions)

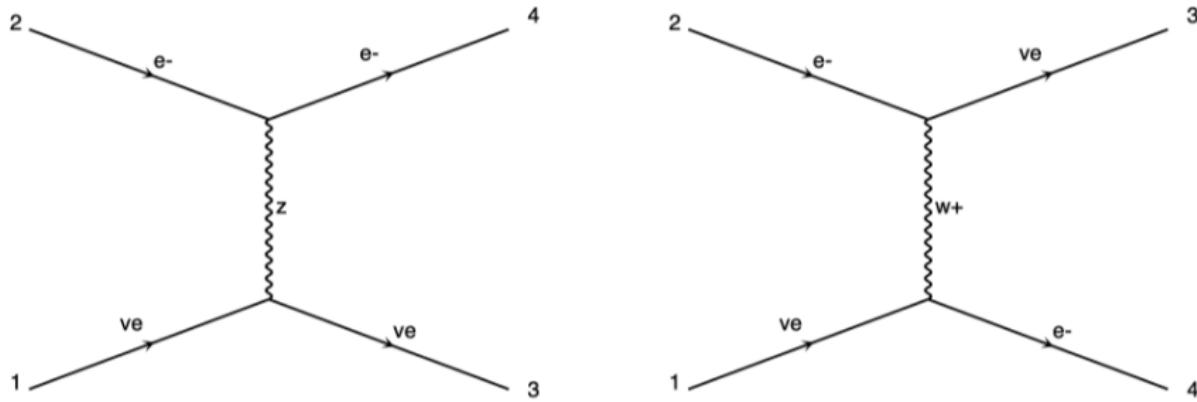


Figure 1.5: Feynman diagram for Weak interactions

### Higgs Interactions

Higgs field provides mass to all the other elementary particle, it's a scalar boson so the spin is 0, and it is massive with the mass of **125GeV**. It interacts with particles in different way and the strength of the interaction which is called coupling, determines the mass a particle will acquires. Since it is massive, it decays almost instantly into other particles.

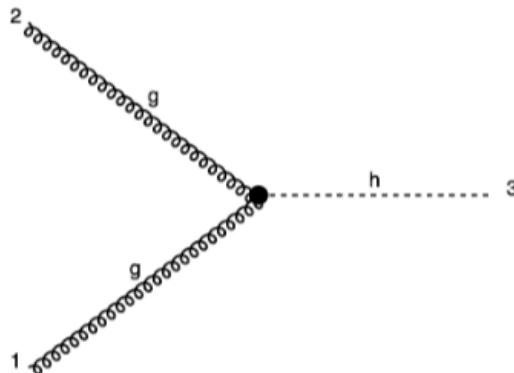


Figure 1.6: Feynman diagram for Higgs interactions

## 1.4 Jets

As we discuss the study of particle interactions across different energy scales, we now focus on a more comprehensive representation of these interactions—the phenomenon of **Jets**. Jets play a crucial role in high-energy physics because they provide us with the observable signatures of bounded particle quarks and gluons, which cannot be observed individually due to the nature of Quantum Chromodynamics (QCD); recently, it has become exciting to study the jet because using machine learning techniques we can learn more about jet sub-structures and based on its sub-structures we can do machine learning based anomaly detecting.

In this section on **Jets**, we are going to discuss the fundamental properties and significance of

jets. We will discuss **jet algorithms**, which are essential tools for reconstructing and identifying jets in experimental data, and examine the **structure of jets**. This robust framework allows us to probe the underlying partonic interactions. Jet substructure techniques have become particularly important in distinguishing standard QCD jets from those coming from exotic processes, such as boosted heavy particles or potential new physics beyond the Standard Model. By understanding jets in detail, we can improve the accuracy of collider data analysis, and enhance our ability to search for new fundamental particles and interactions.

But the above definition of jets is ill-defined because it fails to answer the simple question of whether two particles are part of the same jet or belong to separate jets, since jets are nothing but a spray of particles. Therefore, it becomes a fundamental problem to define a jet. One relies on the jet definition, which tells us how to reconstruct jets from the final state of hadrons, and this is where **Jet Algorithm** comes into the picture.[3]

### Jet Algorithm

The jet algorithm is the formula with some free parameters that allow us to reconstruct jets from the spray of particles; a typical parameter always present in the jet definition is the *jet radius*. Jet radius provides us with the rapidity-azimuth plane, which tells us whether two particles are part of the jet; it also tells us about the jet's recombination scheme using the jet's kinematics properties obtained from its constituents. The jet algorithms are divided into two categories: **sequential algorithms** and **cone algorithms**.

#### 1.4.1 Generalized $k_t$ algorithm

It takes particles generated in the event as the list of objects. The list is then fed into the following formula:

$$d_{ij} = \min(p_{t,i}^{2p}, p_{t,j}^{2p})\Delta R_{ij}^2 \quad (1.1)$$

$$d_{iB} = p_{t,i}^{2p}R^2 \quad (1.2)$$

where  $R$  is the free parameter called as **jet radius** and  $\Delta R_{ij}$  is the geometric distance in the rapidity angle plane, Eq (1.1), the measures the distance between two particles is calculated. Similarly, Eq (1.2) defines a particle's distance from the beam.

These distance measures determine whether two particles should be clustered into the same jet. Here,  $d_{ij}$  represents the relative distance between two particles  $i$  and  $j$ , while  $d_{iB}$  measures the distance of a particle  $i$  to the beam. The parameter  $p$  controls the type of jet clustering algorithm, with common choices being  $p = 1$  for the  $k_t$  algorithm,  $p = 0$  for the Cambridge/Aachen algorithm, and  $p = -1$  for the anti- $k_t$  algorithm. The algorithm iteratively clusters particles until all are assigned to jets or classified as background.[3]

### Jet Substructure

Distinguishing jets coming from various physical processes requires an understanding of jet substructure. Jets from the disintegration of heavy particles like W, Z, Higgs bosons, or top

quarks have several hard cores. In contrast, due to gluon radiation, QCD jets from quarks and gluons usually have a soft, single-core structure. Finding these characteristics makes it possible to tag enhanced items more accurately and lessens background contamination in new physics searches.

### 1.4.2 Prong Finders

Prong finders categorize a jet's interior structure by locating several hard cores. This method aids in distinguishing between ordinary QCD jets, which are usually single-pronged, and those that come from heavy particle decays. For example, top quark jets are three-pronged, W, Z, and Higgs boson jets are two-pronged, and more complicated new physics situations, such as a resonance decaying into two Higgs bosons, could result in four-pronged jets.

By using prong finders, we can improve the identification of boosted objects, which helps in precision measurements and potential new physics discoveries at high-energy collider experiments.[\[3\]](#)

## 1.5 BSM physics

The Standard Model (SM) has been remarkably successful in describing the known fundamental particles and their interactions. However, it leaves several critical questions such as the nature of dark matter, the origin of neutrino masses, and the origin of mass. Newton had discovered that weight is proportional to mass, and Einstein taught us that energy is related to mass by the infamous equation  $E = mc^2$ . These gaps motivate the search for new physics beyond the Standard Model (BSM). BSM theories propose extensions like supersymmetry, extra dimensions, and new gauge bosons (e.g.,  $Z'$ ) that can be explored at current and future colliders. In this section, we will discuss simulated BSM processes using tools like MadGraph, aiming to understand potential signals and their experimental signatures in high-energy collisions.[\[4\]](#)

### Boosted Resonances in New Physics Searches

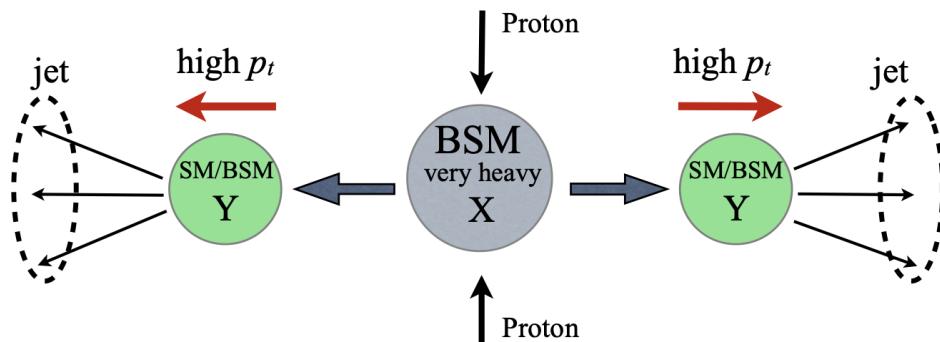


Figure 1.7: Pictorial representation of proton-proton collision forming a BSM particle which decays to a SM/BSM particles[\[5\]](#)

### 1.5.1 Beyond the Standard Model: Heavy Gauge Boson $Z'$

We extend our discussion by considering a Beyond the Standard Model (BSM) scenario that includes a heavy, neutral gauge boson, commonly denoted as  $Z'$ . Such a boson is predicted in many BSM theoretical models. The  $Z'$  boson can interact with Standard Model particles and may decay into fermions, such as quarks and leptons. If the mass of the  $Z'$  is very large, its decay products will be highly energetic or **boosted**. At the Large Hadron Collider (LHC), searches have been performed for  $Z'$  bosons decaying into top quark pairs. Although no such particle has been discovered yet, the data has allowed physicists to set lower limits on the mass of a potential  $Z'$ , excluding masses below approximately 2.5 TeV. Looking toward future high-energy colliders like the Future Circular Collider-hh (FCC-hh), which will operate at a center-of-mass energy of 100 TeV, the direct production of much heavier  $Z'$  bosons becomes possible.[\[6\]](#)

# Chapter 2

## Collider Physics and Experiments

### 2.1 Introduction

In high-energy physics, particle colliders are the most important experimental source for us because they provide testable results for various theoretical models at different energy scales. Many kinds of particle accelerators are based on the energy scale and geometry. Nonetheless, in particle accelerators, hadron colliders, electron-positron colliders, and linear colliders are still active systems. The most powerful and famous of such colliders is the **Large Hadron Collider (LHC)**, which identified the Higgs boson in 2012. As discussed in the previous section, theoretical models expect new fundamental particles at higher energy scales. Colliders are the only source to test these models because they have provided remarkably accurate confirmation of the Standard Model's (SM) predictions. These theories are usually simulated using sophisticated software replicating different detector conditions before investigations. Making insightful theoretical comparisons and evaluating experimental results requires understanding how these simulation tools operate. The foundations of collider physics and important facts of collider experiments will be covered in this chapter.

### 2.2 Exploring Colliders

In a collider experiment, two beams of particles travelling in opposite directions collide at high energies to study fundamental particles and their properties. Colliders can have either a **linear** or **ring** geometry; each comes with it's own advantages and limitations.

**Linear accelerators**, such as the Stanford Linear Accelerator Center (SLAC), are able to achieve high energies with the tiniest energy loss because of the linear geometry. However, their linear design limits the maximum attainable energy, as there is a limit to the acceleration path of the particle.

**Ring accelerators**, such as the Large Hadron Collider (LHC), are more compact, using a circular geometry that allows particles to circulate continuously, achieving energy with each cycle, so there is no limit to the acceleration path. However, there is a problem as charged particles travel through the ring; they experience energy loss because of synchrotron radiation caused by the magnetic fields designed to keep them on a curved path. This energy loss limits the maximum possible energy the particle obtains, especially for lighter particles like electrons.

### 2.2.1 Important Features

One the key aspects of a collider is COM energy and luminosity. Two beams of particles are accelerated to high energies  $E_1$  and  $E_2$ , assuming that the COM energy at a small crossing angle is approximately  $2\sqrt{E_1 E_2}$ . (Generally, studies are done between particles of equal mass, giving COM energy  $E_{cm} = 2E_b$ ) To investigate high-energy physics phenomena, a large number of events is required, along with sufficient energy. A limited number of events would not provide adequate statistical significance or enough opportunities for meaningful physics analyses.

The number of events can be determined using the following equation, which depends on the total time, instantaneous luminosity, and cross-section.

$$N_{events} = \sigma \int \mathcal{L} dt$$

where,  $\int \mathcal{L} dt$  is called the integrated luminosity.

$\sigma$  is the cross-section is a measure of the probability that a specific process will take place in a collision of two particles, and it depends on two things COM energy and theoretical model.

High-luminosity colliders are essential for the search for Beyond Standard Model (BSM) particles, as luminosity directly determines the rate of collisions. Since BSM phenomena typically have low cross-sections, achieving a high luminosity is crucial to increasing the probability of observing rare events.

## 2.3 Large Hadron Collider

The Large Hadron Collider (LHC) is a proton-proton collider with a ring-shaped geometry, which allows us to study of Quantum Chromodynamics (QCD) interactions at high energies. Currently, the collider operates at a center-of-mass (COM) energy of 13 TeV. There are plans to upgrade the LHC to the High-Luminosity LHC (HL-LHC). The High-Luminosity Large Hadron Collider (HL-LHC) is an upgrade to the LHC that aims to increase the number of collisions by a factor of 5-7.5[7, 8]. Now we are going to discuss how does a collision event looks like inside the LHC.

### 2.3.1 Collision event

A collision event is mainly divided into four parts: **Hard Scattering**, **Parton Shower**, **Hadronization** and **Detection and Clustering**.

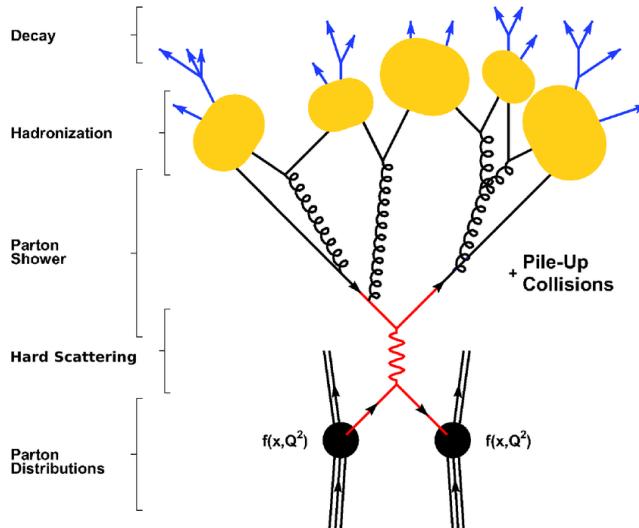


Figure 2.1: Visualization of a Hadronic Collision at the LHC[9]

- **Hard Scattering**

As described in the (ref images), the first of the process is the hard scatter; in this step, a bunch of protons at high energy scales approach each other and collide; this collision takes place at very short distances, which allows us to use QFT to understand the physics of the collision.[10]

- **Parton Shower**

In this stage of the process, because of the hard collision, there is an emission of a large number of gluons, quark-antiquark pair which is called **parton shower** and as these partons are at low energies and moving away, we cannot perform QFT calculations.[10]

- **Hadronization**

After the parton shower procedure, the particles undergo hadronization; in this step, particles recombine together to form stable particles. The resulting hadrons are categorized into two main groups: **baryons**, which consist of three quarks ( $qqq$ ), and **mesons**, which are composed of a quark-antiquark pair ( $q\bar{q}$ ). These particles now pass through a hadron calorimeter to give observables such as  $p_T$ ,  $\eta$  and  $\phi$ .[10]

- **Detection and Clustering**

This is the final stage, where the hadrons from the hadronization, along with leptons and photons, reach the detector and are measured by various components. In the next section, when we talk about computational work, we will discuss that this stage can be described using detector simulation software, which accounts for various detector efficiencies. After the detection, clustering algorithms are applied to group hadronic objects into jets, which substitute the original partitions.[10]

### 2.3.2 Detectors

The Large Hadron Collider (LHC) generally consists of four main collision detectors: **ATLAS**, **CMS**, **LHCb**, and **ALICE**. The CMS detector has a **cylindrical geometry**, with multiple layers of specialized components arranged in a concentric structure. Each layer of the detector plays an extremely crucial role in detecting and identifying particles produced in collisions by measuring their properties, such as momentum, energy and pseudorapidity.

- **Bending particles**

A powerful solenoid magnet bends the particle formed after the collision event. Bending particles is extremely important for the following reasons [10]

1. To know the particle's charge since positive and negative charges move in opposite directions.
2. To know the momentum of particles, since particles with high momentum would deviate less compared to low momentum particles.

- **Tracking**

As a particle passes through these silicon trackers, it deposits some of its energy in the trackers, which helps to reconstruct the trajectories of the particles; therefore, these trackers play a very important role in identifying the particle and measuring its momentum.[10]

- **Energy measurement**

Knowing what the particle is is important to get information about the particle's energy. To identify the particular goal is to measure its energy, and to that, calorimeters are placed in the collider; The CMS detector has two energy calorimeters, ECAL(Electro–Particles like photons and electrons deposit their energy in ECAL. Particles like quarks and gluons deposit their energy in the HCAL, and in this way, we can measure the energy content.[10]

- **Muon Detector**

Muons pass through all the calorimeters. Therefore, it needs to be detected separately.[10]

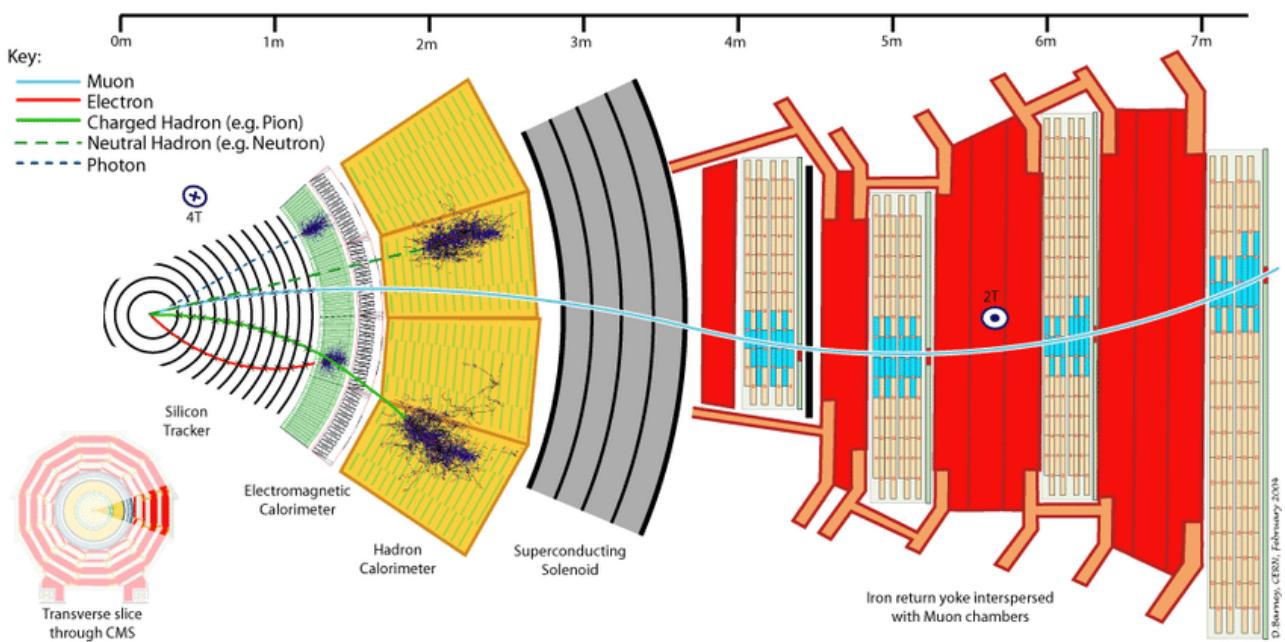


Figure 2.2: CMS detector cross-section[11]

Since different particles possess varying charges and masses, their interactions with the detector occur in distinct ways. Neutral particles, for example, do not experience electromagnetic forces and thus travel in straight-line trajectories until they interact with the detector material. Charged particles, conversely, are influenced by the detector's magnetic field, causing their paths to bend. Additionally, different particles interact with specific detector components depending on their nature.[10]

- **Photons ( $\gamma$ )**

Photons are neutral bosons and because of that reason, it travels in straight lines and do not leave tracks in the inner tracking system. While moving through the calorimeters it only deposits its energy in ECAL via pair production ( $\gamma \rightarrow e^+e^-$ ) or Compton scattering.[10]

- **Neutral Hadrons**

Neutral hadrons, such as neutrons ( $n$ ) and neutral kaons ( $K_L^0$ ) interact strongly with the hadronic calorimeter (HCAL). These particles also travel straight paths and deposit their energy in HCAL.[10]

- **Charged Hadrons**

Charged hadrons, such as protons ( $p$ ), charged pions ( $\pi^\pm$ ), and kaons ( $K^\pm$ ), interact strongly with the HCAL, causing hadronic showers, as observed in **Fig. 8**. Since they carry electric charge, they experience bending due to the detector's magnetic field, leaving identifiable curved tracks in the silicon tracker and depositing energy in it. The curvature of these tracks allows for precise momentum measurements using the tracker system.[10]

- **Muons ( $\mu^\pm$ )**

Muons are heavier counterparts of electrons and interact only weakly with the ECAL and HCAL. Due to their relatively large mass and weak interactions with hadronic and electromagnetic components, they pass through the ECAL and HCAL with minimal energy loss. Eventually, they reach the muon chambers, where they are detected. Their presence in the muon system and tracks in the inner detector confirms their identification.[10]

## 2.4 Collider Physics

The theory of Special Relativity explains the physics of particle collisions. Because of special relativity, collision events are invariant under the transformation of the Lorentz group. 4-vector  $p$  can be described by the energy and the 3-momentum of mass,  $p = (E, \vec{p})$ , which can also be written as  $p^2 = E^2 - |\vec{p}|^2$ ;  $p^2$  is Lorentz invariant. Particle's velocity is  $\beta = \frac{|\vec{p}|}{E}$ , it is a Lorentz scalar. This is a brief introduction to collider physics; in the following subsection, we will discuss certain specifics of colliders and their physics.

### 2.4.1 Madlestram Variables

Madlestram variables are the quantities that give energy, momentum, and angles of particles in a scattering process in a Lorentz-invariant fashion. They are used for scattering processes of two particles to two particles.[12]

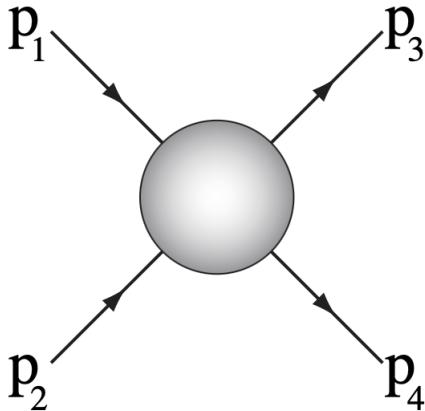


Figure 2.3: Two particles with momenta  $p_1$  and  $p_2$  interact, producing two outgoing particles with momenta  $p_3$  and  $p_4$ .[12]

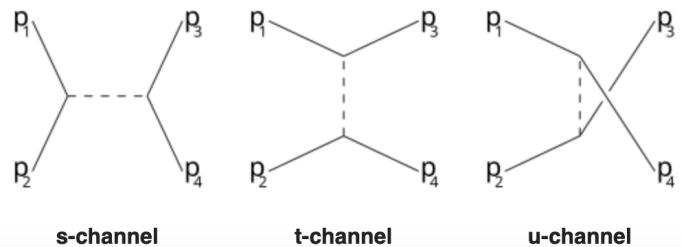


Figure 2.4: The letters s, t, and u are also used in the terms s-channel (timelike channel), t-channel, and u-channel (both spacelike channels).

Figure 2.5: Here is the pictorial representation of the Madlestram variables.

- **S channel**

$$(p_1 + p_2)^2 = (p_3 + p_4)^2$$

- **T channel**

$$(p_1 - p_3)^2 = (p_2 - p_4)^2$$

- **U channel**

$$(p_1 - p_4)^2 = (p_3 - p_2)^2$$

These variables are essential in calculating various properties of the interaction of particles, for example,  $E_{cm}$ , which is  $\sqrt{s}$  or cross-section.

### 2.4.2 Pseudorapidity

**Pseudorapidity**( $\eta$ ) is an approximation to Rapidity. Since it can be readily computed from the cartesian angle between the particle direction above or below the beamline and has a direct relationship to detector components, this number is occasionally used in place of rapidity. For massless particles, this quantity is exactly equal to rapidity; but, in order to comprehend pseudorapidity, we must first comprehend rapidity. Angles to the axis of the collision beams are well-defined for particles traveling near the speed of light and are expressed using . Rapidity. It becomes challenging to discuss these values when particles move at high speeds because angles that are specified in more conventional methods can increase or decrease. For particle trajectories straight to the beam, rapidity is zero; for those at an angle to the beam, it can be either positive or negative.[13]

$$y = \frac{1}{2} \ln \left( \frac{E + p_z}{E - p_z} \right) = \ln \left( \sqrt{\left( \frac{E + p_z}{E - p_z} \right)} \right)$$

$$\begin{aligned} \ln \left( \sqrt{\left( \frac{E + p_z}{E - p_z} \right)} \right) &= \ln \left( \sqrt{\left( \frac{E + p_z}{E - p_z} \right) \left( \frac{E + p_z}{E + p_z} \right)} \right) = \ln \left( \sqrt{\left( \frac{(E + p_z)^2}{E^2 - p_z^2} \right)} \right) \\ \therefore y &= \ln \left( \frac{E + p_z}{m} \right) = \tanh^{-1} \left( \frac{p_z}{E} \right) \end{aligned}$$

where **m** is the transverse mass.

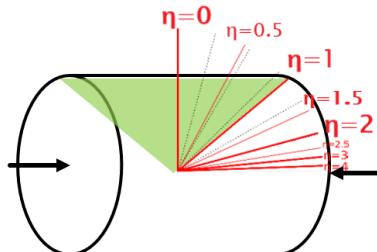


Figure 2.6: Pictorial representation of pseudorapidity.[1]

Rapidity differences are invariant to the Lorentz boost in z-direction.  
for  $p \gg m$ , the formula of rapidity can be written as,

$$y \approx -\ln \left( \tan \left( \frac{\theta}{2} \right) \right)$$

Since it is tough to measure the rapidity of the highly relativistic particle, we use pseudorapidity more commonly.

# Chapter 3

## Computational Framework and Simulations

### 3.1 Introduction

To model and simulate collision events, it is essential to employ various tools to manage the different phases of event generation and reconstruction. In this section, we will provide a concise overview of some of the tools utilized in our project and explain their functions. In the first subsection, we will discuss the essential software for experimental high-energy physics, and after that, we will talk about simulating jet images.

### 3.2 Madgraph

**MadGraph5\_aMC@NLO** is a framework designed to provide all the components required for SM and BSM phenomenology, including creating hard events, identifying cross-sections, and applying various tools pertinent to event manipulation and analysis for different theoretical models? This means that the processes can be simulated for any user-defined Lagrangian, and the NLO accuracy is in the case of models that facilitate such computations. It is also possible to retrieve matrix elements at the tree level and one-loop-level. In addition to these, this framework allows us to obtain Feynman diagrams for the different processes.[14] (**All the Feynman diagrams in this thesis report are made using Madgraph**)

### 3.3 PYTHIA

**PYTHIA** is a program for generating high-energy physics collision events, i.e., describing collisions at high energies between electrons, protons, photons and heavy nuclei. It contains theory and models for several physics aspects, including hard and soft interactions, parton distributions, initial- and final-state parton showers, multiparton interactions, fragmentation and decay. It is largely based on original research but also borrows many formulae and other

knowledge from the literature. As such, it is categorized as a general-purpose Monte Carlo event generator.[15, 16]

## 3.4 Delphes

**Delphes** is a framework for fast simulation of a generic collider experiment designed to emulate the response of a detector to high-energy physics collision events. It includes parametrized simulations of tracking systems, calorimeters, and muon chambers, providing a realistic reconstruction of final-state objects such as jets, electrons, muons, and missing transverse energy. Delphes integrates theoretical models and experimental efficiencies, allowing users to quickly analyze event samples generated by Monte Carlo programs. As such, it is a flexible and efficient tool for phenomenological studies and detector performance assessments.[17, 18]

## 3.5 ROOT

**ROOT** is a unified software package for storing, processing, and analysing scientific data, from its addition to the final visualisation, which is highly customisable. It is dependable, effective, well-maintained, simple to use, and accessible. It aims to optimise the number and significance of scientific findings produced per unit expenditure of computational and human resources. At the Large Hadron Collider experiments, ROOT demonstrated scale by offering a very effective data model storage system: Columnar ROOT format is used to write exabytes of scientific data. In addition to a general parallel processing framework called RDataFrame that can significantly speed up analysis by utilising multi-core and distributed systems, ROOT includes statistical modelling, curve fitting, minimisation, and histogramming capabilities in an arbitrary number of dimensions. These features simplify setting up a data analysis system that can query and process the data interactively or in batch mode.[19]

## 3.6 Jet Image Generation Pipeline

As discussed previously in the context of computational tools for collider event simulation, in this section we are going to elaborate on the method generating jet images from simulated collider data. The entire process involves multiple stages, utilizing a combination of software frameworks widely used in high-energy physics.

### 3.6.1 Event Generation Using MadGraph

The process begins with using **MadGraph5\_aMC@NLO**, a powerful tool for generating parton-level events. In this thesis we are going to discuss extensivly on the QCD and top jet process and if time permits we will talk about the Higgs process. For now let's discuss the generation of the top jet image.

This process describes proton-proton collisions resulting in a top-antitop pair. The top

quark subsequently decays into a  $W$  boson and a bottom quark, and the  $W$  boson decays hadronically into a quark-antiquark pair. Thus, the resulting jet image is expected to exhibit a 3-pronged structure corresponding to the decay products.

$$p\ p \rightarrow t\ t\sim, \text{ followed by } t \rightarrow W\ b \text{ and } W \rightarrow q\ q\sim$$

### 3.6.2 Parton Showering and Detector Simulation

Within the `MadGraph` framework, the event is further processed using `Pythia` for parton showering and hadronization, followed by `Delphes` for detector simulation. The output of this stage is a `.hepmc` file, which contains the complete event record including reconstructed jets and associated particle information.

### 3.6.3 Conversion to ROOT Format

For the further data handling, the generated `.hepmc` file is converted to a `.root` file using the `DelphesCMS` card. This format is widely used in high-energy physics for storing and analyzing large datasets. The ROOT file contains structured data that can be accessed programmatically using ROOT’s C++ API.

### 3.6.4 Data Extraction Using C++

In order to extract relevant physical observables such as transverse momentum ( $p_T$ ), pseudorapidity ( $\eta$ ), and azimuthal angle ( $\phi$ ) from the `.root` file, we utilize a custom C++ script. This script interfaces with the ROOT file, retrieves the necessary jet constituents, and writes the data into a `.out` text file.

### 3.6.5 Jet Image Construction

With the kinematic information ( $p_T$ ,  $\eta$ , and  $\phi$ ) available in the `.out` file, we proceed to generate jet images using a Python-based preprocessing pipeline. The Python script, inspired by an open-source implementation available on GitHub [20], discretizes the  $\eta$ - $\phi$  plane into a 2D grid and fills each pixel with the corresponding transverse momentum content, thereby constructing grayscale jet images.

This multi-stage pipeline allows us to convert high-level simulated collider data into a computer-vision-compatible format, which can then be utilized for machine learning-based jet classification and anomaly detection tasks.

The Python code begins by loading the `.out` file using `pandas`. The file is assumed to contain rows labeled either as `jet_x` (where `x` is the jet number) or as constituents of that jet. The columns extracted are:

- `Type` – either the jet header or constituent

- $p_T$  – transverse momentum
- $\eta$  – pseudorapidity
- $\phi$  – azimuthal angle

Next, the script separates out all rows corresponding to jets using:

```
jet_only = df[df["Type"].str.startswith("jet_")]
```

It then iterates over each jet, extracting the constituents' kinematic properties. Each jet is expected to have up to 200 constituents, so if fewer are present, the constituent list is padded with zeros. The data is flattened and stored into a new DataFrame for easy access.

**Initial Visualization:** Here we are going to discuss the jet image simulation of a **Top** jet. To understand the raw spatial distribution of jet constituents, 2D histograms of  $\eta$  and  $\phi$  (weighted by  $p_T$ ) are plotted using `matplotlib`. The first set of plots shows the raw distribution for the first few jets.

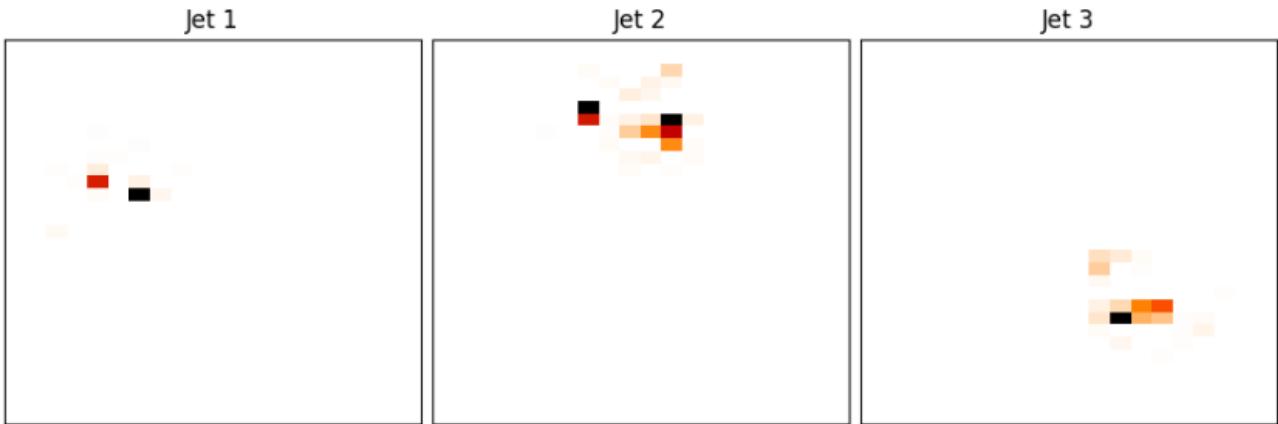


Figure 3.1: Raw jet images, where each jet is represented as a heatmap in the  $\eta$ - $\phi$  plane, with pixel intensity proportional to the transverse momentum ( $p_T$ ).

### Jet Preprocessing Steps:

To make the images invariant to translation and rotation, and to align jets for machine learning purposes, several preprocessing transformations are applied:

1. **Translation:** Each jet's  $\phi$  values are shifted such that the constituent with the maximum  $p_T$  lies at  $\phi = 0$ . The same is applied to scale  $p_T$  so that the highest- $p_T$  constituent is normalized to 1.

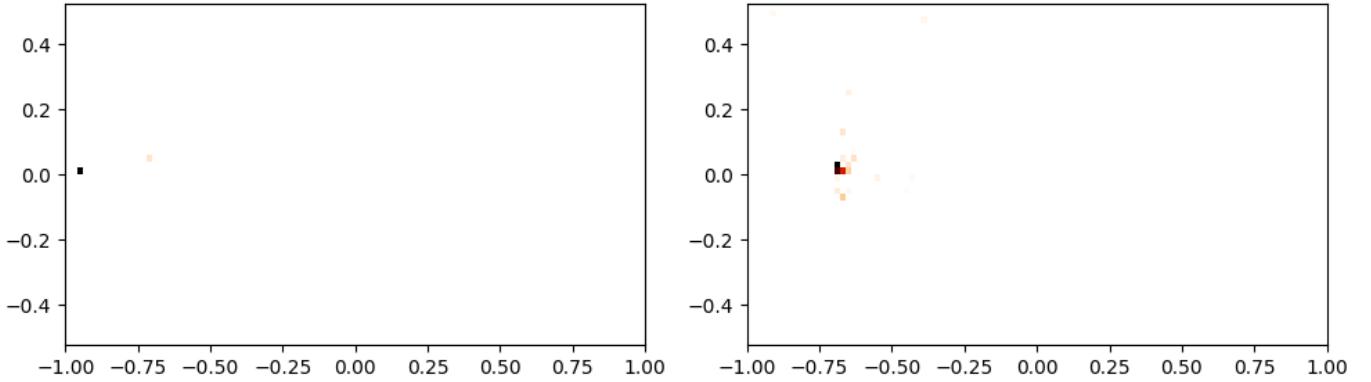


Figure 3.2: Jet image representations after preprocessing. The constituent with the highest transverse momentum ( $p_T$ ) in each jet has been translated to  $\phi = 0$  and its  $p_T$  normalized to 1. Each 2D histogram shows the spatial distribution of jet constituents in the  $(\eta, \phi)$  plane, weighted by normalized  $p_T$  values. This standardization centers and scales jets for consistent input to machine learning model

2. **Centering:** The center-of-mass in  $\eta$  and  $\phi$  is computed as:

$$\eta_{\text{centroid}} = \frac{\sum p_T \cdot \eta}{\sum p_T}, \quad \phi_{\text{centroid}} = \frac{\sum p_T \cdot \phi}{\sum p_T}$$

Then, all constituents are shifted such that the jet is centered at  $(0, 0)$ .

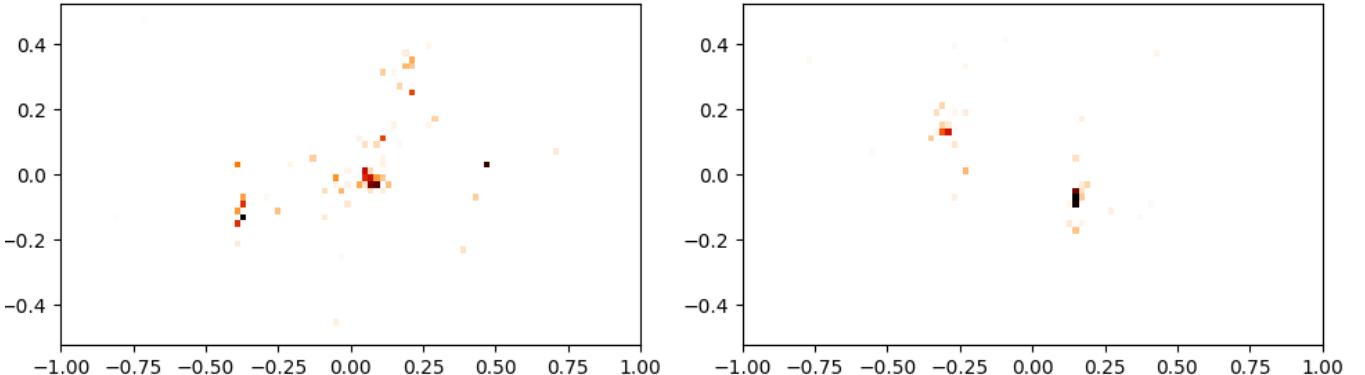


Figure 3.3: Side-by-side visualization of two preprocessed jet images. The  $(\eta, \phi)$  coordinates represent constituent positions, colored by their normalized transverse momentum ( $p_T$ ). Each image has been centered around the constituent with maximum  $p_T$  at  $\phi = 0$ , and all  $p_T$  values have been normalized, enabling consistent comparison across jets.

3. **Rotation:** A principal component analysis (PCA)-like method is used to find the orientation of the jet's energy distribution. The covariance matrix is constructed using second moments:

$$u_{ij} = \frac{\sum p_T \cdot \eta^i \cdot \phi^j}{\sum p_T}$$

The eigenvector corresponding to the largest eigenvalue defines the rotation angle  $\theta$ , and all constituents are rotated accordingly.

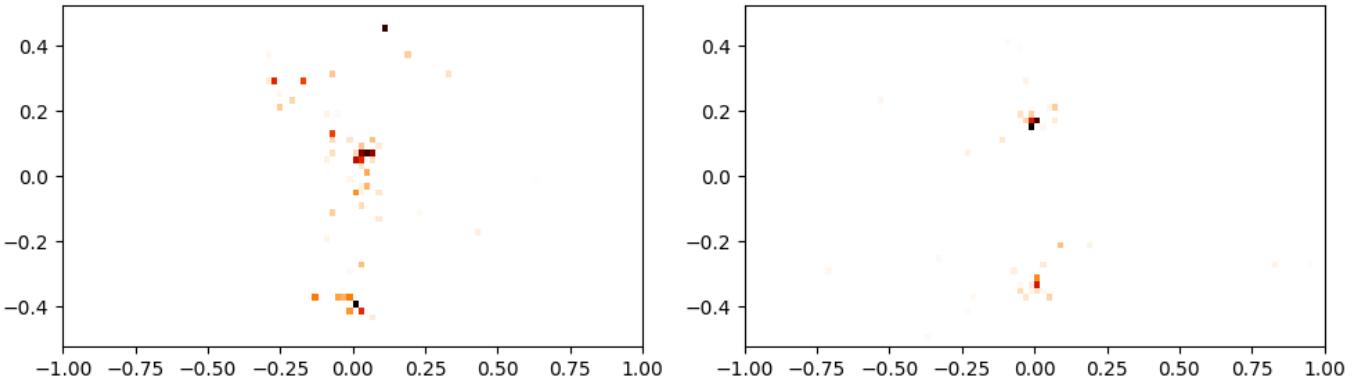
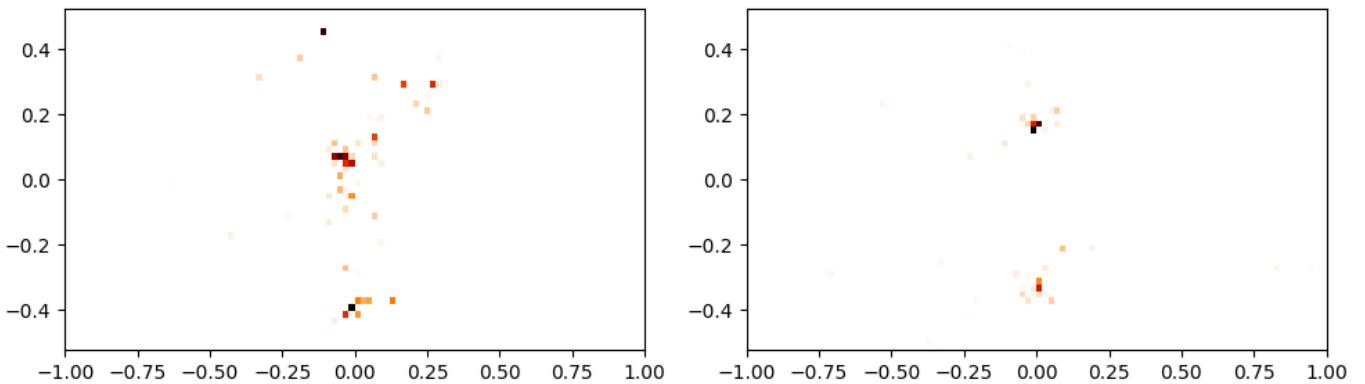


Figure 3.4: Jet images after rotational alignment using the principal axis of energy flow. The jets are rotated such that their maximum variance direction (principal component) lies along a fixed orientation, improving consistency across samples. This step enhances the performance of machine learning models by removing rotational ambiguity.

4. **Flipping:** To resolve any symmetry ambiguities, jets are flipped along the  $\eta$  and  $\phi$  axes so that most of the  $p_T$  is located in the positive quadrant.



After all preprocessing steps, the jet images are much more standardized and aligned.

### Grid Mapping:

The  $\eta$  and  $\phi$  coordinates are mapped onto a 2D pixel grid, where the pixel size and range are configurable. Each jet becomes an image of shape (180, 180), where the intensity of each pixel is the  $p_T$  deposited at that grid location.

The mapping is done using:

$$z[n, y_{\text{index}}, x_{\text{index}}] = p_T$$

Where  $n$  is the jet number, and  $x_{\text{index}}, y_{\text{index}}$  are calculated based on minimum distance to grid bins.

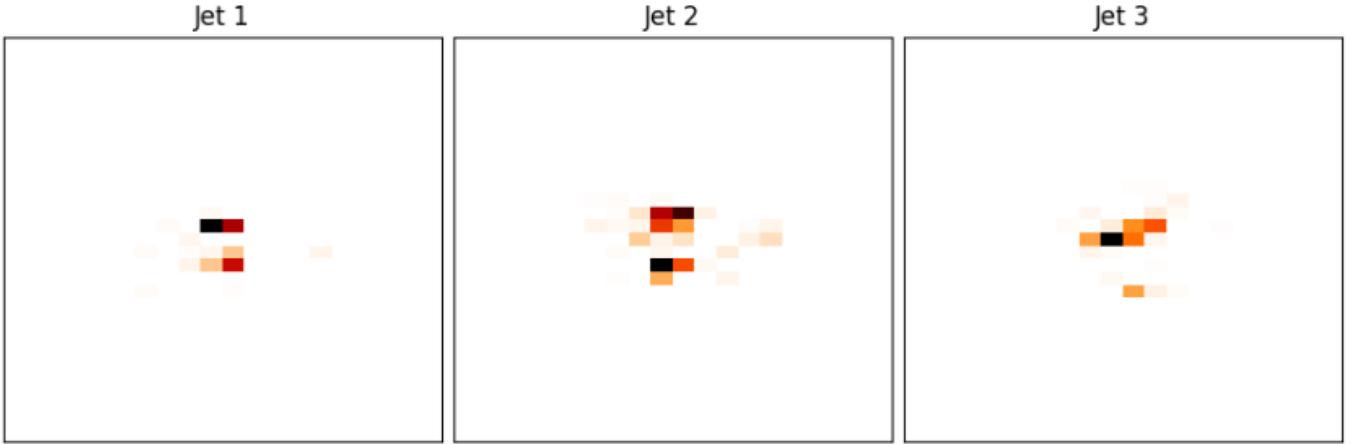


Figure 3.6: Visualization of 20 jet images represented in the  $\eta$ – $\phi$  plane. Each image is constructed using a 2D histogram where the color intensity reflects the transverse momentum ( $p_T$ ) of constituents. These jet images serve as the raw input to preprocessing and machine learning models.

#### Cropping and Normalization:

To reduce image size and enhance features near the center, each  $(180, 180)$  image is cropped to a smaller region (e.g.,  $(40, 40)$ ) centered on the jet axis. Optionally, normalization can also be applied so that each image sums to 1.

$$z_{\text{crop}} = \frac{z_{\text{centered region}}}{\sum z_{\text{centered region}}}$$

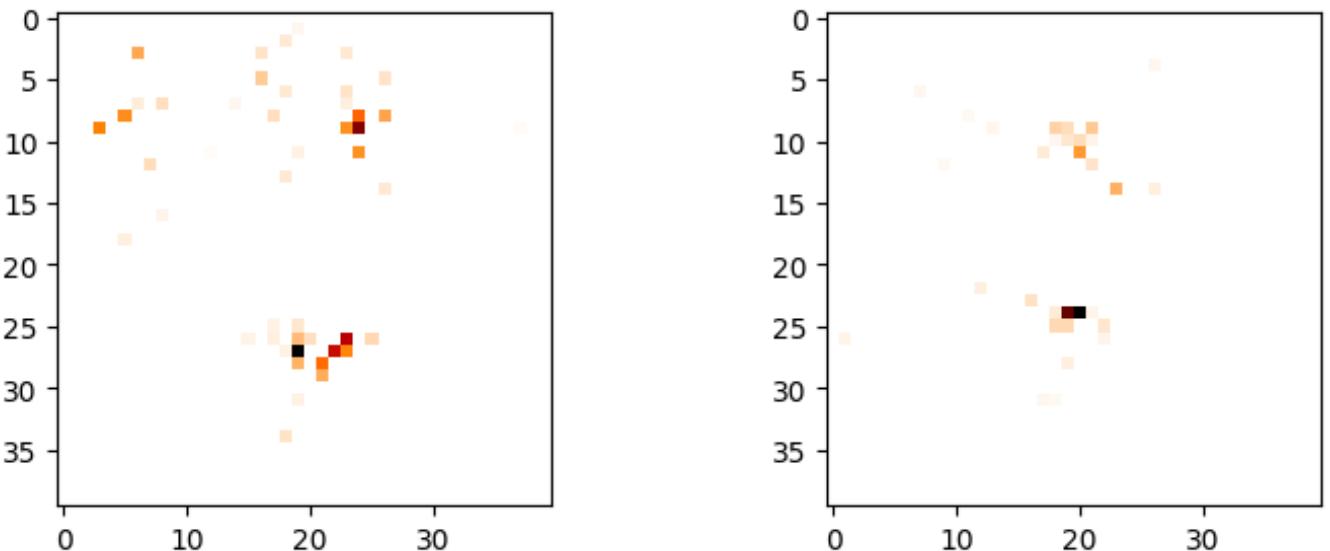


Figure 3.7: Construction of jet images from calorimeter data. Constituent particles' transverse momenta are projected onto a 2D  $\eta$ – $\phi$  grid, forming a pixelated image. The image is then cropped to a fixed central region and normalized such that the total intensity sums to 1.

**Averaging:**

To understand the average topology of jets, the cropped images are averaged across the dataset:

$$z_{\text{average}} = \sum_{i=1}^N z_i$$

This average image helps visualize common features such as prong-like structures in top jets.

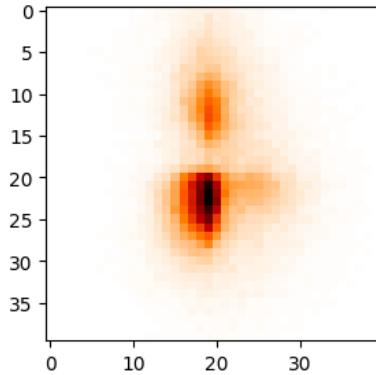


Figure 3.8: Average jet image computed from the training dataset. The image highlights the typical spatial distribution of transverse momentum ( $p_T$ ) across the  $\eta$ - $\phi$  plane. High-intensity regions correspond to more frequently occupied calorimeter cells by jet constituents.

This full preprocessing pipeline provides clean, standardized jet images that retain physics-relevant features while minimizing nuisance variables such as translation or orientation. These jet images can then be used as inputs to convolutional neural networks or other machine learning classifiers.

Preprocessing Steps		
Step	Operation	Motivation
1	Centering in $\phi$	Shift the jet such that the constituent with the highest $p_T$ lies at $\phi = 0$ . This removes rotational ambiguity.
2	Normalizing $p_T$	Scale all pixel intensities by the maximum $p_T$ in the jet. This focuses learning on the shape of the jet rather than its energy scale.
3	Cropping to fixed size	Extract a fixed-size window around the jet core. This ensures uniform image dimensions and focuses on the most relevant region of the jet.
4	Rotating along principal axis	Rotate the jet so that its principal axis aligns along a fixed direction. This removes random orientation differences between jets.
5	Flipping	Apply horizontal/vertical flipping based on energy distribution to enforce reflectional symmetry, e.g., jet always "leans" right or up.
6	Grid discretization	Convert sparse 4-vector data into a fixed-size 2D array (jet image), making the data compatible with convolutional neural networks (CNNs).

Table 3.1: Motivation Behind Jet Image

## 3.7 Problem definition

We began our discussion by exploring the fundamental particles, their fundamental properties, and the interactions that govern them under the physics of the Standard Model. This foundational understanding of these fundamental particles enabled us to study how these particles behave in the particle colliders. We discussed the working principles of these colliders, their technical aspects, and how these colliders help us testify to the current theory of particle physics. Building upon this, the current chapter will discuss the simulation of particle collisions at colliders. We focus on generating observables from these collisions and, subsequently, on constructing jet images derived from them. These jet images are served as inputs for modern machine learning-based classification techniques. Our current setup of simulating jets primarily operates at relatively low energy scales, where jet substructures are reasonably well-separated, and classification is comparatively straightforward. However, a substantial challenge arises when we move to higher energy scales (i.e. jet  $p_T > 10$  TeV), especially in the context of boosted objects. As discussed in the previous section, we have already demonstrated the production and representation of top jet images. When jets become highly energetic, they tend to be more collimated. This collimation causes the individual prongs of the jet (such as those resulting from the decay products of a top quark) to appear much closer together, often merging into a single structure in the detector. Consequently, distinguishing a boosted top jet from others, such as those arising from a QCD background, becomes considerably more difficult. This

thesis is centred around addressing that very challenge. We aim to apply anomaly detection techniques—specifically, convolutional autoencoders to identify top jets based on their high reconstruction error. The premise is that jets that deviate from the norm (e.g., standard QCD jets) will have a lower reconstruction error. Top jets, being structurally more complex, will result in higher errors when reconstructed by an autoencoder trained on background-like jets. Following this anomaly detection approach, we will critically evaluate the hypothesis that as the energy scale increases and the jets become more boosted, it becomes increasingly complex to distinguish between different types of jets due to their overlapping substructure. The main reason to explore this regime of collider physics is to find any signatures of BSM particles that are expected at higher energy scales.

### 3.7.1 Feynman Diagrams for Signal and Background Processes

In this section we are going to briefly discuss the Feynman diagrams of signal and background process.

#### Background

**QCD process** QCD dijet events arise from strong interactions between partons (e.g.,  $qq \rightarrow qq$ ,  $qg \rightarrow qg$ ) via gluon exchange

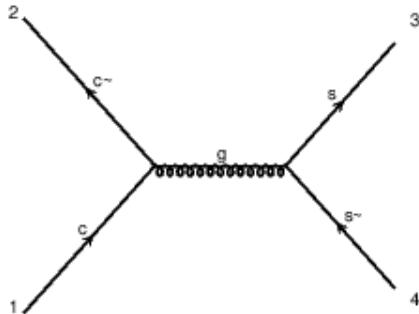


Figure 3.9: Feynman diagram for the QCD process, where two high energy protons collide to give two quark jets

#### Signal

**Top process** Top quark pair production ( $t\bar{t}$ ) is a key process at the LHC, predominantly occurring via gluon-gluon fusion. Each top quark decays almost exclusively into a W boson and a bottom quark. When both W bosons decay hadronically into quark-antiquark pairs, the resulting final state consists of six jets.

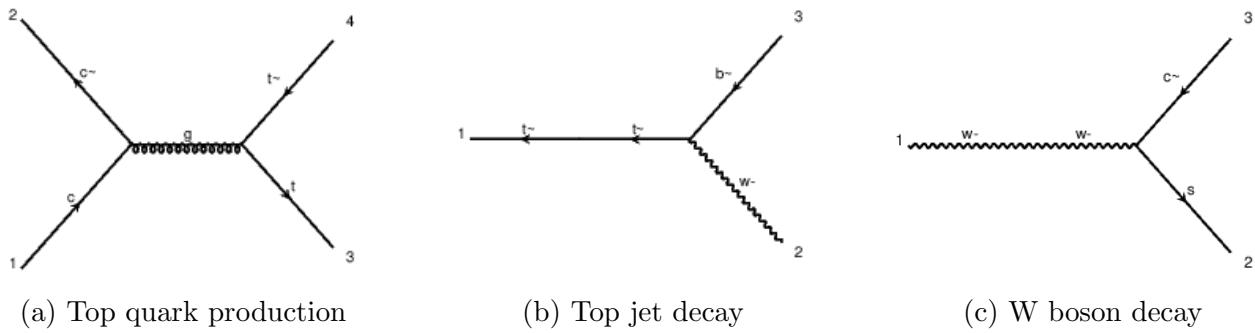


Figure 3.10: Representative Feynman diagrams showing (a) top quark pair production via gluon fusion, (b) top quark decay into a W boson and a b quark, and (c) subsequent W boson decay into a quark-antiquark pair.

**Higgs process** In the signal process considered, a Higgs boson is produced in association with a Z boson ( $pp \rightarrow hZ$ ). The Higgs decays into a pair of bottom quarks ( $h \rightarrow b\bar{b}$ ), while the Z boson decays invisibly into neutrinos ( $Z \rightarrow \nu\bar{\nu}$ ).

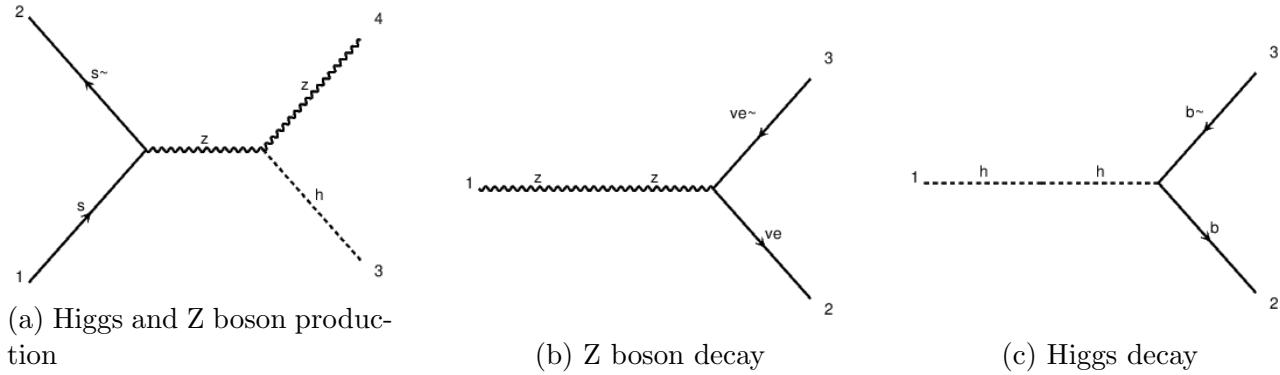


Figure 3.11: Representative Feynman diagrams showing (a) top quark pair production via gluon fusion, (b) top quark decay into a W boson and a b quark, and (c) subsequent W boson decay into a quark-antiquark pair.

# Chapter 4

## Machine learning

As one of the most advancing fields in recent years, machine learning allows models to perform specific tasks without explicit programming. Its applications have expanded across various domains of physics, including particle physics, where it is used for data analysis, anomaly detection, and simulation enhancement.[\[21\]](#)

### 4.1 Introduction

Machine learning has numerous applications, and one explored in this thesis is jet tagging—an open problem in experimental high-energy physics. The primary goal is identifying the parton (such as a gluon, boson, or potential BSM particle) that originated the detected jets. As discussed in previous chapters, jets exhibit high-level features ( $p_T$ , jet mass, N-subjectness, etc.) and low-level features. These features collectively form a d-dimensional input vector  $x \in R^d$  in the machine learning model.

### 4.2 Deep learning

Deep learning is a kind of machine learning that uses artificial neural networks to learn from data. A neural network is a machine-learning model miming the human brain's ability to identify patterns and make decisions. The magic of a neural network is that you will provide X and Y for some examples in the training set, and it will figure out all the other things by itself. X describes the features, and Y specifies the kind of output. For example, I want to buy a tennis racquet, but many options are based on various features.

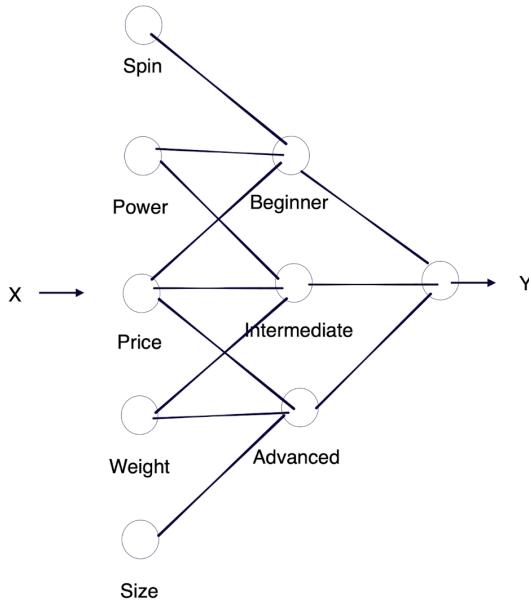


Figure 4.1:  $X$  is a set of tennis racquets with the features described on the neurons, and we also have a hidden layer in the neural network that represents the player's ability;  $Y$  will give the best racquet for a player based on the requirement and the level of their game. (It is a **fully connected neural structure**; however, I have not connected each neuron in the diagram to clarify and avoid visual clutter.)

### 4.2.1 Logistic Regression

Logistic regression is one of the simplest and most widely used algorithms for binary classification tasks in supervised machine learning. It predicts the probability of an event occurring and is used for tasks where the outcome has two possible classes. As a binary classification problem, the final output of logistic regression can be either 0 or 1, often representing a true/false or cat/dog scenario.[\[22\]](#)

In the context of logistic regression, we define the feature matrix  $X$  as follows:

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix}$$

Each row represents a feature vector for a training example, and each column represents a specific feature. The target vector  $Y$  is given by:

$$Y = \begin{bmatrix} y_1 & y_2 & \dots & y_m \end{bmatrix}$$

Where  $y_i$  represents the label associated with the  $i$ -th training example, with  $y_i \in \{0, 1\}$ , where one could represent "ca", "nd zero could represent "do" in a simple classification task.

The goal of logistic regression is to predict the probability  $\hat{y}$  for each training example,

such that  $\hat{y} \approx y$ . We express the predicted value  $\hat{y}$  using the logistic (sigmoid) function:

$$\hat{y} = \sigma(w^T x + b)$$

where  $\sigma$  is the sigmoid function:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

And  $w$  and  $b$  are the parameters (weights and bias) of the model.

We must minimise the loss function to train the logistic regression model, quantifying how well the model's predictions match the actual labels. The binary cross-entropy loss gives the loss function for a single training example:

$$L(\hat{y}, y) = -(y \log \hat{y} + (1 - y) \log(1 - \hat{y}))$$

The cost function, which is the average of the loss function over all  $m$  training examples, is:

$$J(w, b) = \frac{1}{m} \sum_{i=1}^m L(\hat{y}^{(i)}, y^{(i)})$$

where  $L(\hat{y}^{(i)}, y^{(i)})$  is the loss for the  $i$ -th training example.

**Gradient Descent** is a popular optimization technique for minimising the cost function. In gradient descent, the model parameters are updated iteratively in the direction of the steepest descent of the cost function. The steps for minimizing the cost function using gradient descent are as follows[22]:

1. Compute the gradient (partial derivative) of the cost function concerning the parameters  $w$  and  $b$ :

$$\frac{\partial J}{\partial w}, \quad \frac{\partial J}{\partial b}$$

2. Update the parameters using the learning rate  $\alpha$ :

$$w := w - \alpha \frac{\partial J}{\partial w}$$

$$b := b - \alpha \frac{\partial J}{\partial b}$$

3. Repeat the above steps iteratively until convergence, i.e., when the change in the cost function becomes sufficiently small or stops decreasing significantly.

The key to effective gradient descent is choosing an appropriate learning rate ( $\alpha$ ). If the learning rate is too large, the algorithm may overshoot the minimum of the cost function. If the learning rate is too small, converging the algorithm may take too long.

Following these steps, the logistic regression model can learn optimal parameters that

minimize the cost function, ultimately enabling accurate predictions for binary classification tasks.

### 4.2.2 Convolutional Neural Networks

A Convolutional Neural Network (CNN) is a deep learning algorithm well-suited for image recognition and processing tasks. But why can't we use DNN for image data? for a 28x28 image, it's okay we can use DNN, but when we are dealing with 4k resolution images, the number of pixels is in the order of  $10^7$ . Feeding such high-dimensional input into a fully connected DNN would result in large number of parameters, making the model computationally inefficient and prone to overfitting.

**What makes CNN so unique that it can handle such extensive data and doesn't blow up?**

CNNs handle large image data efficiently without blowing up in terms of computational cost or memory usage because of their architectural design—particularly local connectivity, parameter sharing, and downsampling (pooling). [23]

CNN architecture

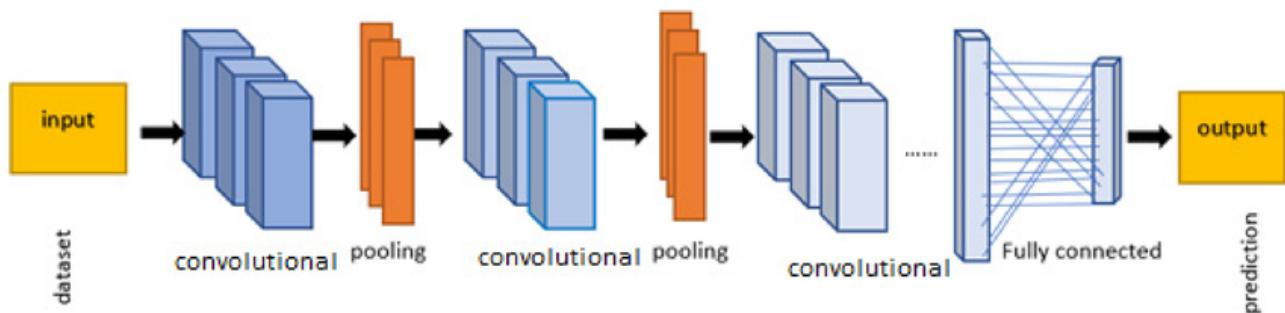


Figure 4.2: CNN architecture[24]

1. **Filters and Strides:** Filters (kernels) are small matrices that slide over the input to detect specific features. The *stride* controls how much the filter moves across the input. By increasing the stride, the spatial dimensions of the output are reduced, which lowers computational load while preserving essential features.[22, 25]
2. **Pooling Layers:** Pooling operations, such as max pooling, reduce the spatial dimensions of the input data, retaining the most significant features. This helps to decrease memory usage and computational complexity.[22, 25]
3. **Padding:** Padding is applied to the input image to maintain spatial dimensions when necessary and prevent the loss of edge information during convolution. The padding ensures the filter can operate across the entire image, especially in deeper layers.[22, 25]

Convolution Arithmetic

## Notation

- $H_{\text{in}}, W_{\text{in}}$  – Height and width of the input feature map
- $H_{\text{out}}, W_{\text{out}}$  – Height and width of the output feature map
- $C_{\text{in}}$  – Number of input channels
- $C_{\text{out}}$  – Number of output channels (i.e., number of filters)
- $K$  – Kernel (filter) size (assumed square:  $K \times K$ )
- $S$  – Stride (step size for sliding the filter)
- $P$  – Padding (number of pixels added around the input)
- Parameters – Total number of trainable weights in a layer
- FLOPs – Floating point operations (proxy for computational cost)

1. **Filters and Strides:** Filters detect features such as edges or textures. Stride controls movement across the input.[22, 25]

$$H_{\text{out}} = \left\lfloor \frac{H_{\text{in}} + 2P - K}{S} \right\rfloor + 1, \quad W_{\text{out}} = \left\lfloor \frac{W_{\text{in}} + 2P - K}{S} \right\rfloor + 1$$
$$\text{Parameters} = (K \times K \times C_{\text{in}} + 1) \times C_{\text{out}}$$
$$\text{FLOPs} = K \times K \times C_{\text{in}} \times C_{\text{out}} \times H_{\text{out}} \times W_{\text{out}}$$

2. **Pooling Layers:** Pooling reduces spatial size and retains dominant features (e.g., edges in max pooling).[22, 25]

$$H_{\text{out}} = \left\lfloor \frac{H_{\text{in}} - K}{S} \right\rfloor + 1$$
$$W_{\text{out}} = \left\lfloor \frac{W_{\text{in}} - K}{S} \right\rfloor + 1$$

3. **Padding:** Padding preserves boundary information by expanding the input. Affects output size.[22, 25]

If  $P = \frac{(K - 1)}{2}$ , then output size is preserved (for odd  $K$ )

These design features enable CNNs to process large-scale data efficiently, making them ideal for image classification and object detection tasks while minimizing computational bottlenecks and overfitting.

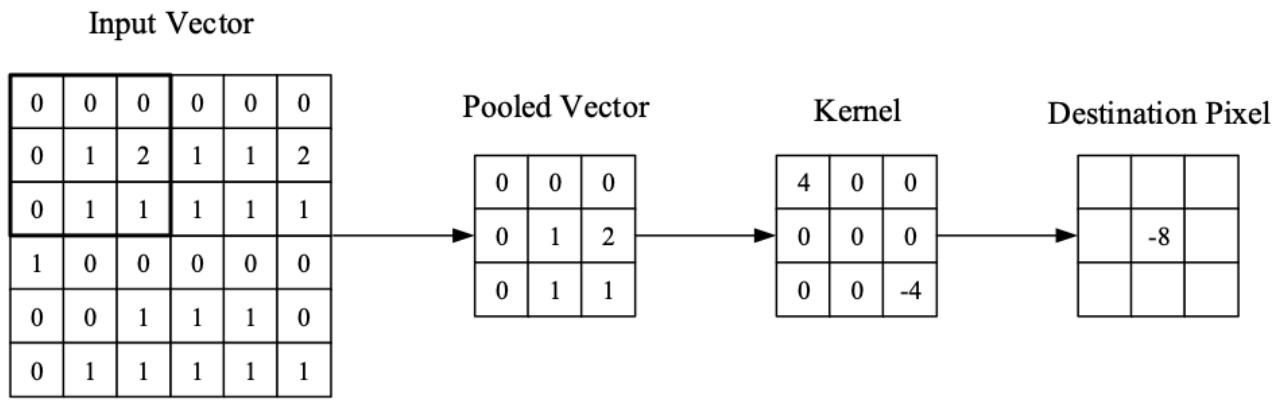


Figure 4.3: Example of convolution operation after pooling. As the input vector undergoes the pooling layer, its spatial dimensions get reduced, resulting in a pooled vector. A kernel is then convolved with the pooled vector to compute the destination pixel. The kernel underlines directional features, and the final value ( $-8$ ) reflects the weighted sum of overlapping elements.[23]

## 4.3 Supervised and Unsupervised learning

### 4.3.1 Supervised Learning

In supervised learning, we provide our ML model with a labelled dataset; the goal of this type of learning is to correctly map the input with the output, i.e. after training, our model should be able to correctly map a function  $f : X \rightarrow Y$ , where  $X$  represents the feature space (data), and  $Y$  represents the label space (target/output).

We have to provide the input-output pairs  $\{(x_i, y_i)\}$  to the model, from which it will learn to adjust its parameters while minimizing the error between the predicted output and the actual output, and once it is trained well, we can predict labels for unseen data  $x_{new}$ .

1. Regression: When the labels  $y$  of a provided dataset are continuous values, such as predicting the best tennis racquet based on the provided features like the level of the player, cost, the spin and the racquet's power.

2. Classification: When the labels  $y$  are discrete, such as classifying images of animals such as 'cat', 'dog' or 'bird'.

The supervised learning models are beneficial in the cases like image classification or speech recognition. The only thing we need for these type of models is high-quality data, which is extremely important for model training.

### 4.3.2 Unsupervised Learning

Unlike supervised learning, here we provide the ML model with unlabelled data; this type of learning aims to find the hidden patterns underlying the dataset, finding the link between input and predicted output.

Standard techniques in unsupervised learning include:

1. Clustering: Grouping similar data points
2. Anomaly Detection: Identifying some rare or unusual data points that tend to differ significantly from the normal dataset.

Unsupervised learning models are widely used in image compression and anomaly detection applications. These models are helpful because this learning trains the models in such a way that they can discover insights from raw data without human supervision.

## 4.4 AutoEncoders

Autoencoders are a type of ML model, that are designed to learn efficient representations of data by compressing input data into a lower-dimensional space and then reconstructing it. Since they operate in an unsupervised manner requiring only the data itself, these models are particularly useful for anomaly detection. When trained on a typical dataset, autoencoders become very good at reconstructing that particular data. However, they tend to struggle with rare or unusual inputs, result in giving higher reconstruction errors. This property makes them a powerful tool for identifying anomalies using analyzed reconstructed errors. In this study, we apply machine learning techniques to explore autoencoder architecture. Mainly this thesis will focus on finding the anomalies using the reconstructed errors of the jet images. Our work aims to contribute to ongoing efforts in the search for phenomena beyond the Standard Model (BSM).

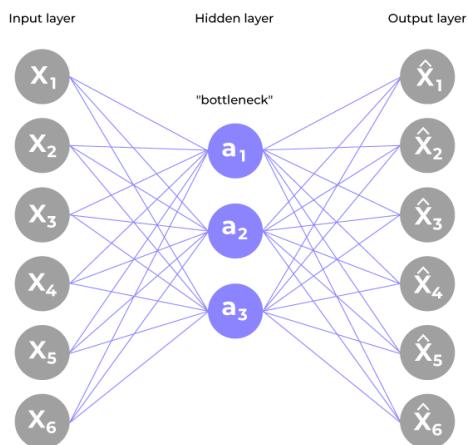


Figure 4.4: Autoencoder architecture

### 4.4.1 Encoders

An encoder takes the input data and compresses the data through a series of hidden layers that progressively. Throughout the reduction of the dimensions, it preserves the important features of the input data. The final hidden layer of the encoder, known as the **bottleneck**

layer or **latent space**, contains a compact and information-rich representation of the input. To explain mathematically, the encoder is a function, which can be written as:

$$h_i = g(x_i)$$

where  $x_i \in \mathbb{R}^n$  is the input and  $h_i \in \mathbb{R}^q$  is the latent representation, with  $q < n$  [26].

#### 4.4.2 Decoders

The decoder takes the latent representation of the data from the bottleneck layer and attempts to reconstruct the original input. This is achieved by gradually increasing the dimensionality through hidden layer, last layer is called the output layer that matches the dimensionality of the input data. The goal of the decoder is to generate an output  $\tilde{x}_i \in \mathbb{R}^n$  such that:

$$\tilde{x}_i = f(h_i) = f(g(x_i))$$

It tries to minimize the error between the input and reconstructed data, and anything with a high reconstruction error can be flagged as an anomaly or unusual data. [26].

### 4.5 Activation Function of the Output Layer

In autoencoders, the output layer's activation function plays a very crucial role in reconstructing the input data. The choice of this function depends on the nature and scale of the input data, because it directly affects how well a model can reconstruct the original input data [26].

Below are some of the most commonly used activation functions in the output layer:

#### 4.5.1 ReLU (Rectified Linear Unit)

The ReLU activation function is defined as:

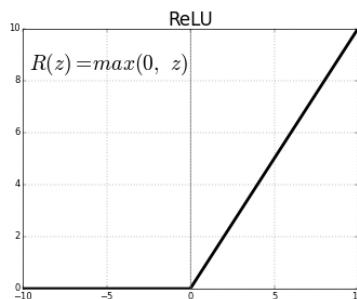


Figure 4.5: Graph for the ReLU activation function[27]

$$\text{ReLU}(x) = \max(0, x)$$

- **Range:**  $[0, \infty)$

- **When to use:** Best suited for input data that contains only non-negative values and spans a wide range (e.g., sensor data, some types of images).
- **Limitation:** Cannot represent negative values, so if the input  $x_i$  has negative components, ReLU may introduce significant reconstruction errors.

**Note:** Use ReLU when the input data is non-negative and unbounded.[27, 28]

### 4.5.2 Sigmoid

The sigmoid activation function is defined as:

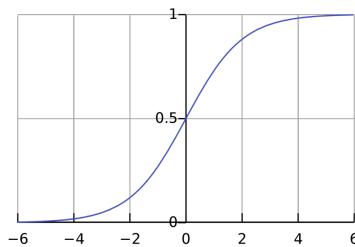


Figure 4.6: Graph for the ReLU activation function[27]

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

- **Range:**  $(0, 1)$
- **When to use:** Ideal for binary input data or normalized data that has been scaled between 0 and 1.
- **Example:** In the MNIST dataset, pixel values range from 0 to 255. By dividing by 255, the input values fall into the  $(0, 1)$  range, making sigmoid a suitable activation.

**Note:** Sigmoid is a good choice when working with normalized data in the  $(0, 1)$  range.[27, 28]

### 4.5.3 Linear

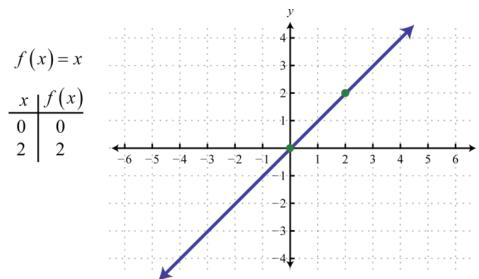


Figure 4.7: Graph for the linear activation function[27]

The linear activation function is simply the identity function:

$$f(x) = x$$

- **Range:**  $(-\infty, \infty)$
- **When to use:** Best for unbounded data that can contain both negative and positive values.
- **Advantage:** No restriction on output range, allowing for more flexibility in reconstructing real-valued continuous data.

**Note:** Linear activation is often used in regression tasks or when inputs are not scaled or normalized.[27, 28]

#### 4.5.4 Tanh (Hyperbolic Tangent)

The tanh activation function is defined as:

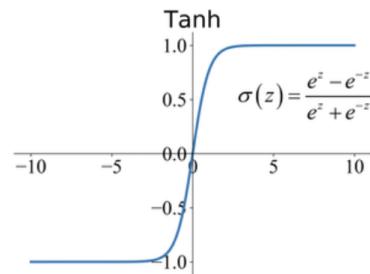


Figure 4.8: Graph for the tanh activation function[27]

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

- **Range:**  $(-1, 1)$
- **When to use:** Suitable for inputs that have been normalized to this range.
- **Comparison:** Like sigmoid, tanh is smooth and differentiable, but it is centered around 0, making it better for handling data with both positive and negative values.

**Note:** Use tanh when your input data is normalized to fall between  $-1$  and  $1$ .[27, 28]

## Summary Table

Activation Function	Range	Use Case
ReLU	$[0, \infty)$	Non-negative unbounded data
Sigmoid	$(0, 1)$	Normalized input in $(0, 1)$
Linear	$(-\infty, \infty)$	Real-valued unbounded input
Tanh	$(-1, 1)$	Normalized input in $(-1, 1)$

Table 4.1: Activation functions for the output layer and their typical use cases.

Choosing the appropriate activation function for the output layer is essential, as it ensures that the reconstructed data  $\tilde{x}_i$  is within a meaningful range and aligns with the distribution of the original input  $x_i$  [26].

## 4.6 Loss Function and Training Objective

The central objective of an autoencoder is to minimize the difference between the input data  $x_i$  and the reconstructed output  $\tilde{x}_i$ . This is typically achieved using a **reconstruction loss** function [26].

Commonly used loss functions include:

### 4.6.1 Mean Squared Error (MSE)

Mean Squared Error is one the most used loss function for the autoencoders, especially when dealing with the image type input data. It calculates the average of the squares of the differences between the input and reconstructed data.

$$\text{MSE} = \frac{1}{n} \sum_{j=1}^n (x_{ij} - \tilde{x}_{ij})^2$$

Where:

- $x_{ij}$  is the  $j^{th}$  feature (e.g., pixel) of the input data  $x_i$ ,
- $\tilde{x}_{ij}$  is the  $j^{th}$  feature of the reconstructed data  $\tilde{x}_i$ ,
- $n$  is the total number of features (e.g., pixels per image).

**Mathematical Insight:** In this part, we follow the mathematical framework presented by Michelucci and in their work on autoencoders [26]. Let us define the MSE over the entire dataset as:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{M} \sum_{i=1}^M \|x_i - \tilde{x}_i\|^2$$

Where  $M$  is the total number of observations in the dataset, and  $\|\cdot\|$  denotes the Euclidean norm.

To demonstrate that MSE is minimized when the reconstruction exactly matches the input (i.e.,  $\tilde{x}_i = x_i$ ), we can derive the loss with respect to  $\tilde{x}_{ij}$ :

$$\frac{\partial \mathcal{L}_{\text{MSE}}}{\partial \tilde{x}_{ij}} = -\frac{2}{M}(x_{ij} - \tilde{x}_{ij})$$

Setting the derivative to zero:

$$\frac{\partial \mathcal{L}_{\text{MSE}}}{\partial \tilde{x}_{ij}} = 0 \Rightarrow \tilde{x}_{ij} = x_{ij}$$

This confirms that the MSE is minimized when each output feature exactly matches its corresponding input.

We also verify the second derivative to ensure this is indeed a minimum:

$$\frac{\partial^2 \mathcal{L}_{\text{MSE}}}{\partial \tilde{x}_{ij}^2} = \frac{2}{M} > 0$$

Thus, the MSE loss function has a unique global minimum when  $\tilde{x}_i = x_i$ , validating its use for training autoencoders to replicate input data as closely as possible.

**Note:** Mean Squared Error is one the most used loss function for the autoencoders, especially when dealing with the image type input data. It calculates the average of the squares of the differences between the input and reconstructed data.

### 4.6.2 Mean Absolute Error (MAE)

MAE, or  $L_1$  loss, calculates the average of the absolute differences between the original and reconstructed data.

$$\text{MAE} = \frac{1}{n} \sum_{j=1}^n |x_{ij} - \tilde{x}_{ij}|$$

MAE tends to produce blurrier reconstructions than MSE but is less sensitive to very large differences in individual features.

As we move ahead, we will observe the reconstruction of jet images using MSE, MAE and SSIM, giving us the experimental evidence for using MSE over MAE and SSIM.

### 4.6.3 Structural Similarity Index (SSIM)

While MSE and MAE compare pixel-wise differences, SSIM is designed to capture perceptual similarity between images, accounting for luminance, contrast, and structure. SSIM is especially useful for image data, as it aligns more closely with human visual perception.

SSIM between two image patches  $x$  and  $\tilde{x}$  is computed as:

$$\text{SSIM}(x, \tilde{x}) = \frac{(2\mu_x\mu_{\tilde{x}} + c_1)(2\sigma_{x\tilde{x}} + c_2)}{(\mu_x^2 + \mu_{\tilde{x}}^2 + c_1)(\sigma_x^2 + \sigma_{\tilde{x}}^2 + c_2)}$$

Where:

- $\mu_x, \mu_{\tilde{x}}$  are the mean values of  $x$  and  $\tilde{x}$ ,
- $\sigma_x^2, \sigma_{\tilde{x}}^2$  are their variances,
- $\sigma_{x\tilde{x}}$  is their covariance,
- $c_1$  and  $c_2$  are small constants to stabilize the division.

The SSIM score ranges from  $-1$  to  $1$ , where  $1$  indicates perfect similarity. In practice, we often use  $1 - \text{SSIM}$  as a loss term (i.e., maximize similarity by minimizing dissimilarity).

### 4.6.4 Final Objective

The final training objective is to minimize the chosen loss function over all samples in the dataset:

$$\min_{f,g} \langle \Delta(x_i, f(g(x_i))) \rangle$$

As we move ahead, we will observe the reconstruction of jet images using MSE, MAE and SSIM, giving us the experimental evidence for using MSE over MAE and SSIM.

## 4.7 Model Evaluation Metrics

In the context of anomaly detection and classification tasks in high-energy physics, evaluating the performance of a model is crucial. The following tools and metrics are commonly used to assess how well a model distinguishes between signal and background events.

### 4.7.1 Receiver Operating Characteristic (ROC) Curve

The Receiver Operating Characteristic (ROC) curve represents a ML model's ability to separate signal from the background. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR):

$$\text{TPR} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}, \quad \text{FPR} = \frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}}$$

The area under the ROC curve (AUC) is a value that describes the performance; a perfect classifier yields an AUC of 1.0, while a random classifier corresponds to an AUC of 0.5 (Basically random choosing). ROC AUC tells us about how well a model to classify the data.

### 4.7.2 Signal Efficiency vs. Background Rejection

In high-energy physics, describing model performance in terms of signal efficiency and background rejection is often more intuitive. Signal efficiency is equivalent to the TPR, while background rejection is defined as:

$$\text{Background Rejection} = 1 - \text{FPR}$$

A plot of signal efficiency versus background rejection provides a useful trade-off curve to select operating points based on physics goals.

This metric is especially important in anomaly detection, where the signal (anomalies) is rare and maintaining high efficiency is critical while still filtering out background events effectively.

### 4.7.3 Confusion Matrix

The confusion matrix is a tabular summary that shows how well the classifier performed by comparing actual and predicted class labels. It is defined as:

	Predicted: Background	Predicted: Signal
Actual: Background	True Negative (TN)	False Positive (FP)
Actual: Signal	False Negative (FN)	True Positive (TP)

Table 4.2: Confusion matrix for binary classification.

From the confusion matrix, the following evaluation metrics can be derived:

- **Accuracy:**  $\frac{TP+TN}{TP+TN+FP+FN}$
- **Precision:**  $\frac{TP}{TP+FP}$
- **Recall (TPR):**  $\frac{TP}{TP+FN}$
- **F1 Score:** Harmonic mean of precision and recall.

These metrics provide a more detailed insight into classification performance, especially when dealing with class imbalance — common in anomaly detection tasks in collider experiments.

## 4.8 Reconstructing MNIST Data

To explain the concept of autoencoders in practice, we will reconstruct images from the Fashion MNIST dataset using Keras. The dataset consists of 28x28 grayscale images (784-dimensional vectors). A basic autoencoder is structured as follows:

- **Encoder:** Compresses the input data from 784 dimensions to a 32-dimensional latent representation.
- **Decoder:** Reconstructs the original image from the 32-dimensional encoded data.
- **Autoencoder:** Combines both encoder and decoder into a unified model trained for image reconstruction.

Separate models for the encoder and decoder can be used for feature extraction and visualization. The model is trained using the Adam optimizer and the MSE loss function.

## 4.9 Visualization

The following figure shows a visual representation of the encoding and decoding process. The top row shows the original input images, the middle row shows the input data in the latent space, and the bottom row shows the reconstructed images generated by the decoder.

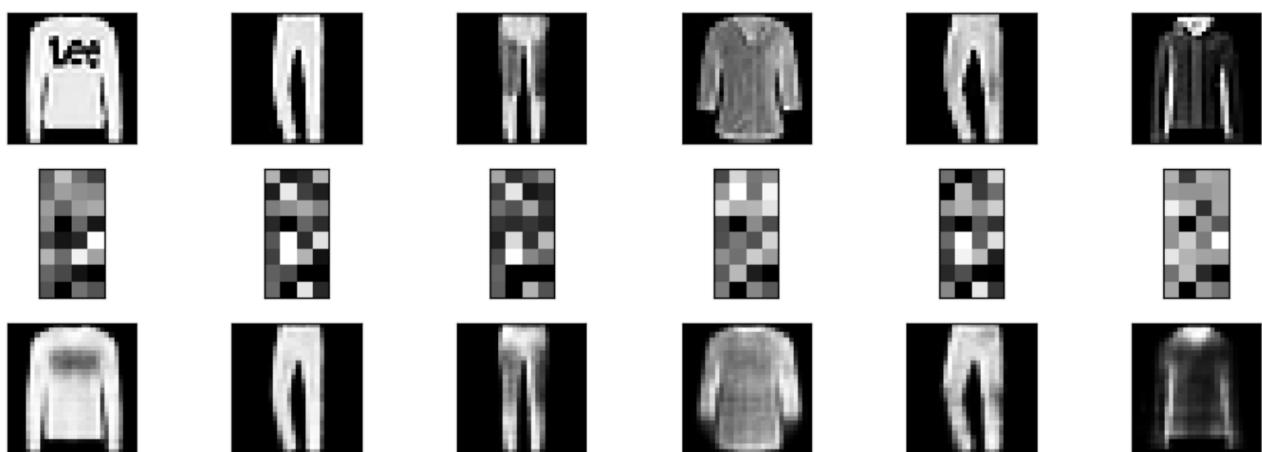


Figure 4.9: Reconstruction of MNIST dataset

## 4.10 Anomaly Detection

One of the main applications of the autoencoder is the use of reconstruction error for analysis and flagging the anomalies in the data as it generally performs poorly for the anomalous data, which results in high reconstruction error. Any data point with a significant loss than this threshold can be identified as an anomaly. This method is particularly helpful in domains like fraud detection, industrial monitoring, and medical imaging, where normal patterns dominate and anomalies are rare but significant. In the next section, we will discuss the results of particle jet image anomaly detection using autoencoders.

# Chapter 5

## Anomaly Detection in Jet Images Using Autoencoders

### 5.1 Introduction

In the previous chapter, we introduced the concept of autoencoders, discussing the architecture, loss functions, activation functions, and typical applications in data compression and feature extraction. This chapter extends that discussion by understanding and applying it to a physics-driven task anomaly detection in jet images obtained from particle collisions at different energy scales. Jet images are two-dimensional plots of energy deposits recorded by detectors, typically in the  $\eta$ - $\phi$  (pseudorapidity–azimuthal angle) plane. These images contain spatial patterns that correspond to the underlying jet substructure. By treating these jet images as conventional images, we can utilize the power of convolutional neural networks (CNNs) to perform both classification and anomaly detection tasks. Our goal in this section is to utilize an autoencoder architecture based on CNNs to identify anomalous jet structures using the reconstruction error of the images. Specifically, while doing semi-supervised learning, we train the autoencoder exclusively on Quantum Chromodynamics (QCD) jet images, which we consider a set of regular events. After training, we evaluate the model’s ability to reconstruct jet images originating from top quark decays; these are treated as **anomalies**. The central hypothesis depends on the jet substructure as QCD jets, primarily from gluon radiation and hadronization, typically have a **two-pronged** structure. In contrast, top jets resulting from hadronic decays of the top quark ( $t \rightarrow bW \rightarrow bq\bar{q}'$ ) are expected to display a more complicated **three-prong structure**. Since the autoencoder is trained only on the simpler two-prong QCD jets, it learns to reconstruct such structures efficiently. However, when presented with three-prong top jets, the model struggles to reconstruct them accurately, leading to a higher reconstruction error. This difference in reconstruction quality serves as the key to identifying anomalous jets. This method allows us to flag top jets as anomalies based on their structural deviation from the QCD-like norm. However, as we will explore in the subsequent sections, several nuances and limitations are associated with this approach. In particular, we will discuss the effect of jet energy scales, overlap in

substructure due to boosted kinematics, and the potential failure modes of autoencoders in discerning subtle anomalies. These discussions will help establish the strengths and constraints of using autoencoders for physics-motivated anomaly detection tasks.

### 5.1.1 Jet Images and Physics Motivation

The dataset used in this study comprises  $10^5$  simulated QCD jet images for four different values of transverse momentum ( $p_T$ ) intervals: 450 GeV, 750 GeV, 1500 GeV, and 5000 GeV. These images are preprocessed as discussed in Chapter 3.

As the energy of the jet increases, the decay products become more collimated—i.e., they appear closer together in the detector’s image plane. This phenomenon presents a significant challenge: distinguishing between two-prong QCD jets and three-prong top jets becomes increasingly difficult at higher  $p_T$  values due to overlapping or merged features in the images. Because of this overlapping, the three-prong structure will either look like a two or one-prong structure and it will be similar for QCD, which will result extreme similarities in normal and anomaly data and because of that reconstruction error will be very similar for both and it would be difficult to flag top jets as anomalies.

### 5.1.2 Autoencoder-Based Anomaly Detection

This section will briefly discuss the technicalities of the autoencoder that we will use to detect anomalies.

Let  $x$  be the input jet image and  $\hat{x}$  be its reconstruction by the autoencoder. The reconstruction error is computed using the Mean Squared Error (MSE):

$$\text{MSE}(x, \hat{x}) = \frac{1}{N} \sum_{i=1}^N (x_i - \hat{x}_i)^2,$$

where  $N$  is the total number of pixels in the image.

We can set an optimal threshold to classify jets as normal or anomalous by comparing the distribution of reconstruction errors for QCD and top jets. The performance of the anomaly detection system is evaluated using standard metrics such as the Area Under the ROC Curve (AUC), Precision-Recall curves, and confusion matrices.

### 5.1.3 Chapter Overview

In the subsequent sections, we will:

1. Present the detailed architecture of the convolutional autoencoder used in our study.
2. Show training performance, loss convergence, and reconstructions for QCD jets.
3. Analyze reconstruction error distributions across different  $p_T$  bins.

4. Evaluate anomaly detection performance using ROC and precision-recall curves.
5. Discuss the significance of our findings in the context of current research in unsupervised anomaly detection in high-energy physics.

Through this work, we aim to demonstrate the potential of convolutional autoencoders as powerful tools for model-independent anomaly detection, while also acknowledging the challenges posed by increasingly collimated jet structures at higher energies.

## 5.2 Convolutional Autoencoder Architecture

In this work, we used a convolutional autoencoder that takes a jet image of  $40 \times 40 \times 1$  as input. The architecture contains an encoder, bottleneck (i.e. the latent space) and the decoder. The encoder contains three convolutional layers with increasing filter sizes, 16, 32, and 64, respectively, followed by a max-pooling layer after each filter that reduces the spatial dimensions by a factor of two. This progressively compresses the input image into a compact latent representation of size  $5 \times 5 \times 64$ .

The decoder then mirrors this process to upscale the reconstructed image size to the original. It starts with applying a convolutional layer with 64 filters, which is followed by a series of upsampling layers and convolutional layers that gradually increase the spatial dimensions and reduce the number of filters simultaneously. The final layer is a single-filter convolutional layer with a linear activation function that outputs the reconstructed image, again of size  $40 \times 40 \times 1$ .

This architecture is designed to learn and identify meaningful low-dimensional structures of jet images and capture slight differences in their structure, which is crucial for anomaly detection.

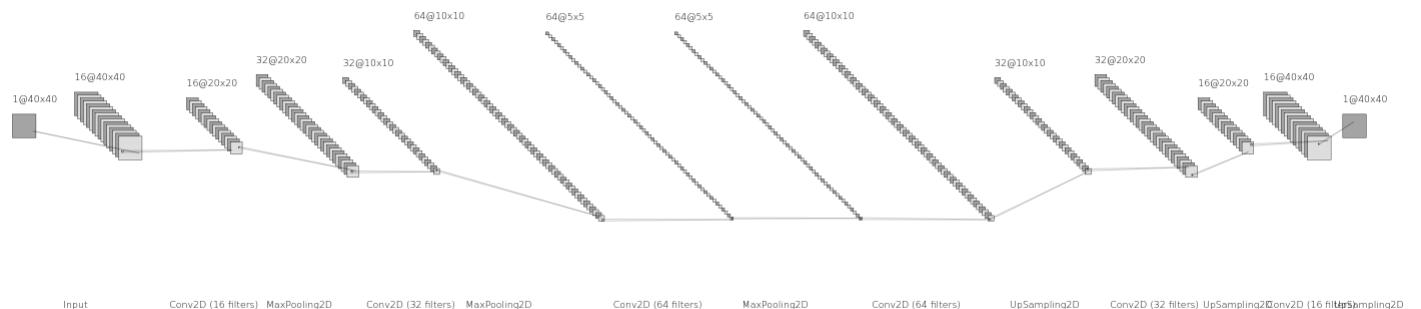


Figure 5.1: Architecture of the convolutional autoencoder used for anomaly detection in jet images. The input is a grayscale image of size  $40 \times 40 \times 1$  that passes through the encoder composed of successive Conv2D and MaxPooling2D layers, progressively reducing the spatial dimensions while increasing feature depth, leading to a compressed latent representation. The decoder then reconstructs the original image using Conv2D and UpSampling2D layers. This reconstruction is used to differentiate between QCD and anomalous (Top) jets based on the reconstruction error.

### 5.2.1 Training Setup

The model was trained using the **Adam** optimizer, with **ReLU** activations in hidden layers and a **linear** activation at the output. The loss function was the **Mean Squared Error (MSE)** which measures pixel-wise reconstruction quality. The dataset consists of  $10^5$  normalized QCD jet images, with pixel values scaled to  $[0, 1]$  to ensure faster convergence and numerical stability, which is suitable for pixel-wise image reconstruction tasks. Training was conducted over 50 epochs

## Top Tagging

As discussed in the preceding sections, our objective is to detect Top jets as anomalies using a convolutional autoencoder. This section presents the model’s results and outlines the methodology employed in the semi-supervised learning framework.

### 5.2.2 Training Setup and Model Configuration

The autoencoder was trained on a dataset of  $10^5$  QCD jet images, each of resolution  $40 \times 40 \times 1$ . These grayscale jet images were generated using standard simulation tools, as detailed in chapter 3. For evaluation, we use 50,000 QCD jet images from the test set and an unseen dataset of Top jets. Although labels are not used during training, they are employed in the evaluation phase to measure the model’s performance in distinguishing QCD from Top jets via reconstruction error.

### 5.2.3 Reconstructions of the Jet Images

As discussed in the preceding sections, we now turn our attention to the reconstruction of jets at a transverse momentum of  $p_T = 450$  GeV. In this section, we aim to provide experimental validation of the hypothesis introduced at the beginning of the chapter—namely, that jets originating from top quark decays, due to their complex substructure, are more difficult to reconstruct using an autoencoder trained solely on QCD jets.



Figure 5.2: Sample QCD jet images used as input to the autoencoder. Grayscale, resolution:  $40 \times 40 \times 1$ .



Figure 5.3: QCD jet images at  $p_T = 450$  GeV. These images represent typical two-prong structures.

Figures 5.2 and 5.3 show original and reconstructed QCD jet images, respectively. As observed:

- The reconstructions retain the original jets' core structure and if we observe closely the bright pixels are reconstructed well compare to the dim pixel.
- Minor blurring is seen in low-activity peripheral regions, which is expected, as the model focuses on minimizing the overall MSE by prioritizing dominant features.

Overall the autoencoder has done good job in reconstructing the QCD jet images on which it was trained on.

Now let's discuss the top jet image reconstruction.



Figure 5.4: Top jet images at  $p_T = 450$  GeV. These images represent typical three-prong structures.



Figure 5.5: Reconstructed Top jet images from the trained autoencoder.

As seen in Figures 5.4 and 5.5, the autoencoder performs poorly on Top jets. This is expected, as the model was not trained on these images. Consequently:

- The reconstructed Top jets are missing key structural features.
- Energy deposits are often blurred or misrepresented.
- As the structure of top jets are complicated than the qcd jets, autoencoder models performs poorly in reconstructing the top jet images.

This discrepancy demonstrates the model’s effectiveness in anomaly detection, it reconstructs familiar QCD-like structures well while failing to reproduce anomalous Top-like patterns.

### 5.3 Model Evaluation and Anomaly Scoring using MSE

After training the convolutional autoencoder on QCD jet images, the next important step is to evaluate its performance and evaluate its ability to detect anomalous jet images based on its structures specifically, those arising from top quark decays. Since the autoencoder is optimized to reconstruct jets that follow the QCD (normal) jet images data, we expect it to perform poorly when reconstructing jets with different substructures, such as three-prong top jets. This difference in reconstruction quality forms the basis for anomaly detection.

To quantify the reconstruction quality, we use the Mean Squared Error (MSE) between the original and reconstructed jet images. We have used MSE error function for knowing how well the autoencoder captures the features of a given input. Lower values indicate better reconstruction (and thus higher similarity to the training distribution), while higher values suggest that the input deviates from the learned QCD patterns and may be considered anomalous.

Before analyzing the anomaly scores, we first present the convergence of the loss during training, which reflects how well the model has learned to represent and reconstruct QCD jets. To comprehensively evaluate the performance of the trained autoencoder in identifying anomalies, we employ both quantitative metrics (like ROC-AUC, signal efficiency vs. background rejection) and qualitative visualizations (such as reconstruction error maps and scatter plots of anomaly scores).

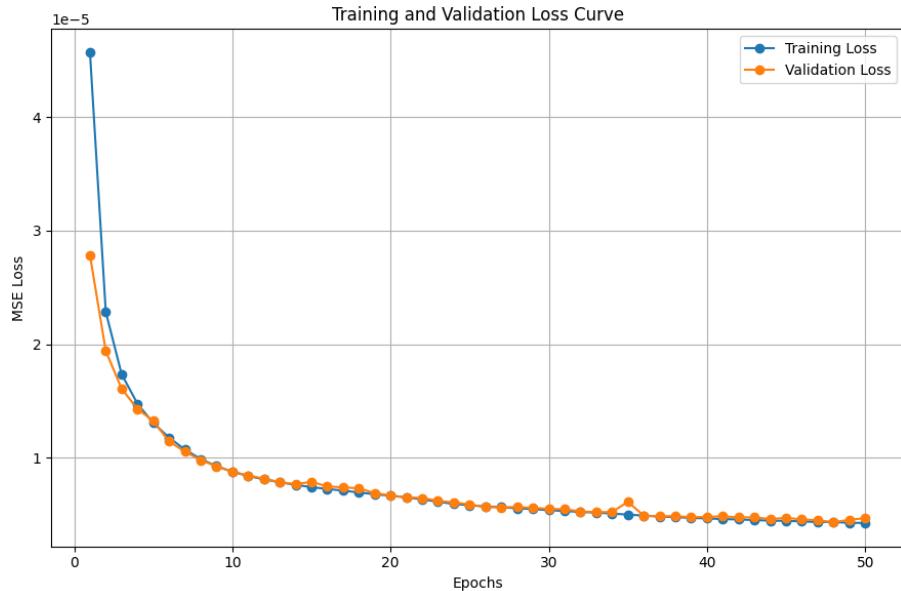


Figure 5.6: Training and validation loss curve for the convolutional autoencoder over 50 epochs. The loss is measured using Mean Squared Error (MSE) between the input and reconstructed QCD jet images. The smooth convergence of both training and validation loss indicates that the model has learned to effectively reconstruct QCD-like jets without significant overfitting.

Before moving forward in the chapter with the anomaly detection analysis, we feel that it is important to mention that our study on  $p_T = 450$  GeV jets will be conducted using three different types of reconstruction loss functions: **Mean Squared Error** (MSE), **Mean Absolute Error** (MAE), and **Structural Similarity Index Measure** (SSIM). Each loss function captures different aspects of the reconstruction quality such as MSE penalizes more significant errors more severely, MAE provides a more balanced treatment of reconstruction deviations, and SSIM is sensitive to structural and perceptual differences in images.

By evaluating the performance of our model under these different three loss functions, it will enable us to identify which loss is most effective for our model at distinguishing anomalous jets from the background. This comparative analysis will help us determine an optimal loss function for anomaly detection task.

## Qualitative MSE Maps

To better understand how the reconstruction error is distributed spatially across the jet image, we visualize input–reconstruction–error for both QCD and top jet classes. These visualizations contain the original input jet image, the corresponding output produced by the autoencoder using MSE loss, and a pixel-wise difference map (i.e., the reconstruction error) between the two.

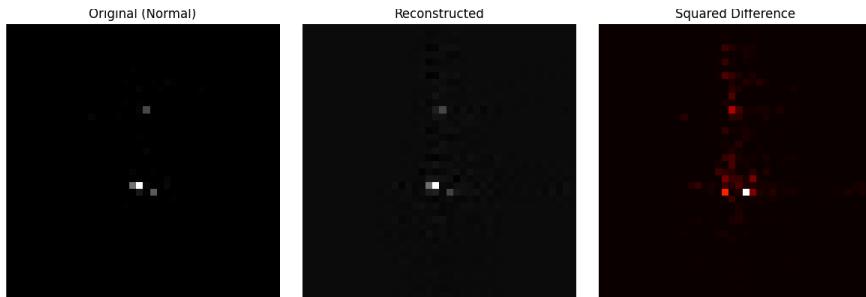


Figure 5.7: Left: original QCD image; Middle: reconstruction; Right: pixel-wise squared error.

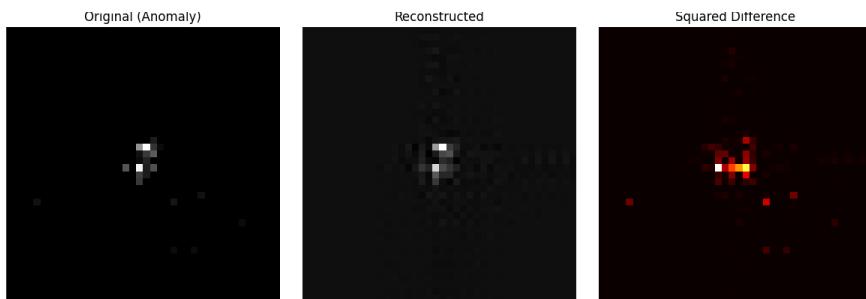


Figure 5.8: Left: original Top image; Middle: reconstruction; Right: pixel-wise squared error.

These visualizations show that the QCD reconstruction aligns closely with the original, resulting in minimal error. In contrast, the Top jet reconstruction fails to preserve the fine-grained substructure, especially in high-activity regions, yielding significantly higher pixel-wise error.

## Distribution of MSE Scores

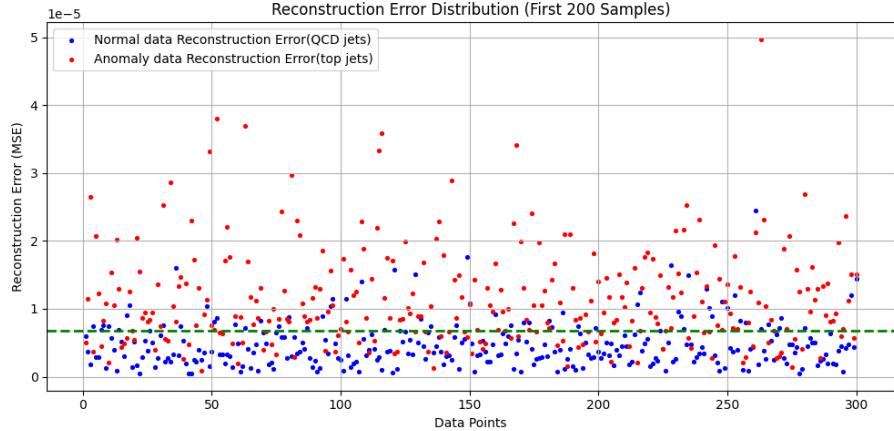


Figure 5.9: Scatter plot of Mean Absolute Error (MSE) scores for QCD and Top jets. The red line indicates the anomaly threshold.

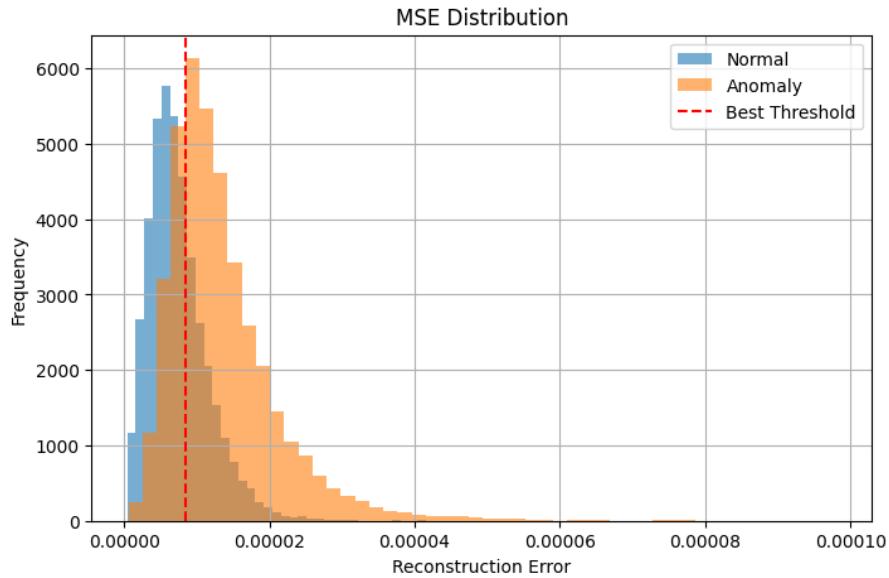


Figure 5.10: Scatter plot of Mean Absolute Error (MSE) scores for QCD and Top jets. The red line indicates the anomaly threshold.

Figure 5.9 presents a scatter plot showing the anomaly scores (MSE in this case) for individual jet images. The QCD jets are densely clustered around low error values, while the Top jets show a broader spread with many points lying beyond the threshold, validating our hypothesis that anomalous jets are harder to reconstruct, and we can use the reconstruction error to flag anomalies.

We now evaluate the model's anomaly detection ability based on reconstruction error statistics. To comprehensively evaluate the performance of the trained autoencoder in identifying anomalies, we employ both quantitative metrics (like ROC-AUC, signal efficiency vs. background rejection) and qualitative visualizations (such as reconstruction error maps and scatter plots of anomaly scores).

## Confusion Matrix Analysis

To quantitatively evaluate the performance of our anomaly detection model, we present the **confusion matrix** in Figure 5.11. This matrix summarizes the classification results based on a chosen threshold on the reconstruction error.

In the context of anomaly detection using jet images, we adopt the standard confusion matrix terminology with the following interpretations:

- **True Negative (TN):** A jet from QCD origin (normal) is correctly classified as normal by the model.
- **False Positive (FP):** A QCD jet (normal) is incorrectly classified as anomalous, i.e., it is misidentified as a top jet.
- **False Negative (FN):** A jet from top quark decay (anomalous) is incorrectly classified as normal, i.e., the model fails to detect it as an anomaly.
- **True Positive (TP):** A top jet (anomalous) is correctly identified as anomalous by the model.
- **True Negatives (TN):** 33,048 QCD jets correctly identified as normal.
- **False Positives (FP):** 9,197 QCD jets misclassified as anomalous.
- **False Negatives (FN):** 10,000 top jets misclassified as normal.
- **True Positives (TP):** 30,000 top jets correctly identified as anomalous.

This notation is essential for evaluating model performance in distinguishing between normal (QCD) and anomalous (top) jets. High values of TP and TN indicate a strong classifier, whereas high FP or FN values suggest areas for improvement in either sensitivity (recall) or specificity (precision).

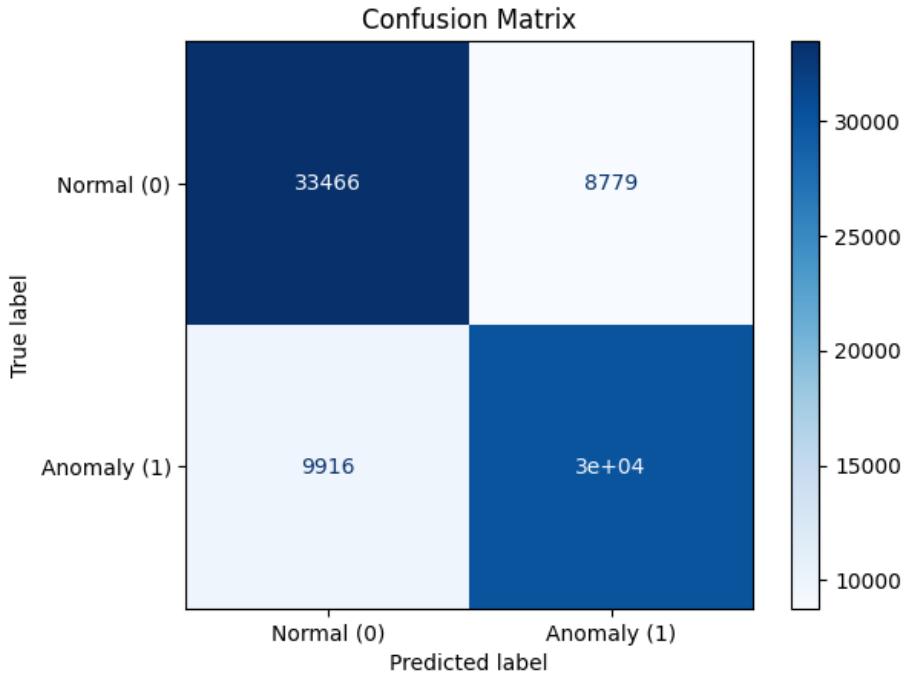


Figure 5.11: Confusion matrix for anomaly detection using reconstruction error on 450 GeV jets. The autoencoder was trained on QCD (normal) jets and tested on both QCD and top (anomalous) jets.

From the confusion matrix, we compute the following performance metrics:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{30000 + 33048}{82145} \approx 0.768,$$

$$\text{Precision (for anomalies)} = \frac{TP}{TP + FP} = \frac{30000}{30000 + 9197} \approx 0.765,$$

$$\text{Recall (for anomalies)} = \frac{TP}{TP + FN} = \frac{30000}{30000 + 10000} = 0.75,$$

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \approx 0.757.$$

These results show that the model is reasonably effective in detecting top jets as anomalies based on their reconstruction error. However, the presence of a noticeable number of false positives and false negatives suggests potential areas for improvement, such as optimizing the threshold, using alternative loss functions, or incorporating additional features into the anomaly scoring.

Metric	Value
Accuracy	0.768
Precision	0.765
Recall (True Positive Rate)	0.75
F1 Score	0.757

Table 5.1: from the obtained confusion matrix 5.11 we are calculating accuracy, precision, recall and the F1 score values

## ROC Curve Analysis

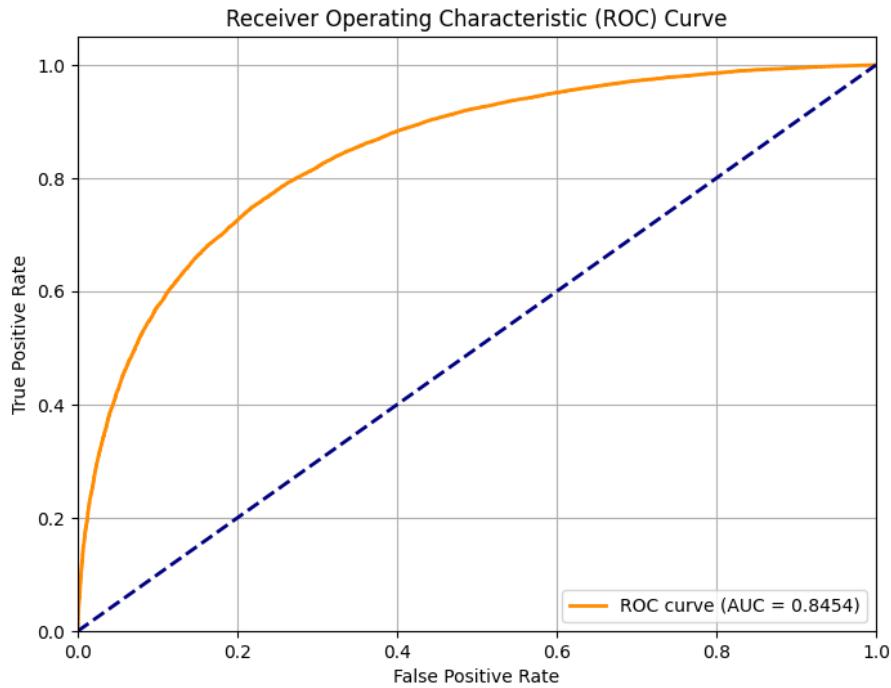


Figure 5.12: ROC curve obtained using MSE-based anomaly score. Area Under Curve (AUC) = 0.8454.

The ROC curve in Figure 5.12 shows strong separability between the two classes. An AUC of **0.8454** confirms that the model has strong capacity to flag the top jets as anomalous based on high reconstruction error and this suggests that our autoencoder is working well in identifying QCD and top jets.

## Signal Efficiency vs. Background Rejection

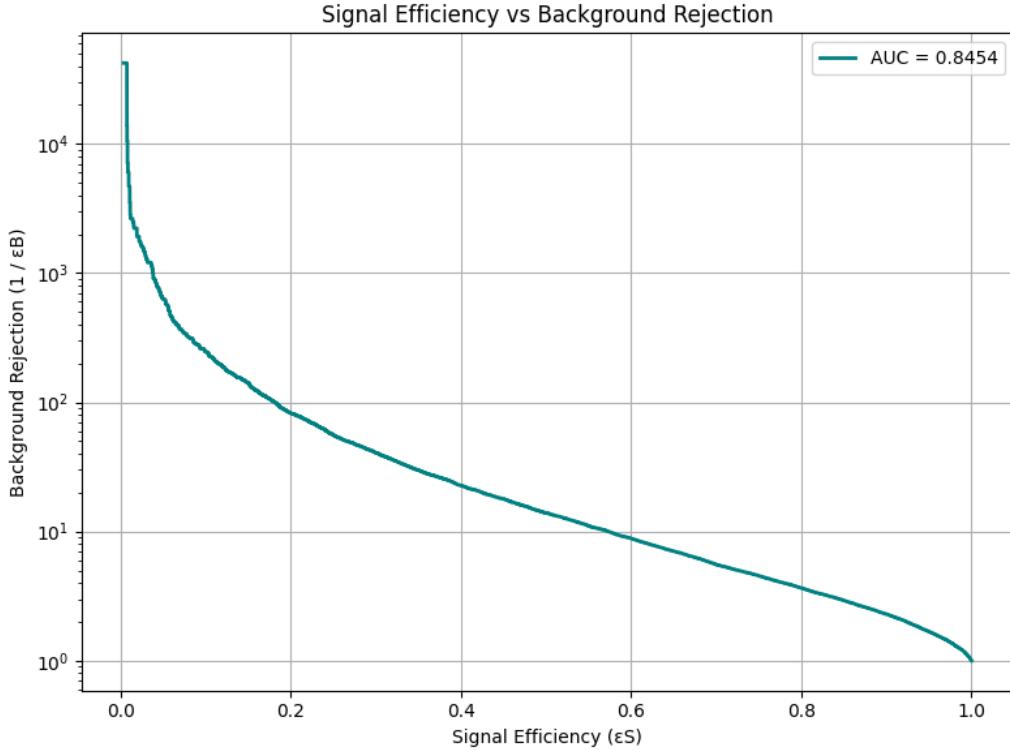


Figure 5.13: Signal efficiency ( $\epsilon_S$ ) vs. background rejection ( $1/\epsilon_B$ ). The red marker indicates the chosen anomaly threshold.

The plot in Figure 5.13 illustrates the tradeoff between detecting Top jets (true positives) and rejecting QCD background. The optimal threshold is approximately  $6.78 \times 10^{-6}$ . This minimal value reflects the low overall MSE distribution — typical for well-trained autoencoders — and emphasizes the importance of careful normalization and threshold tuning. We further analyze the quality of reconstructions using two different types of error functions MAE and SSIM .

## 5.4 Model Evaluation and Anomaly Scoring using MAE

### Visualizing Jet Reconstructions and Errors

These visualizations contain the original input jet image, the corresponding output produced by the autoencoder using MAE loss, and a pixel-wise difference map (i.e., the reconstruction error) between the two

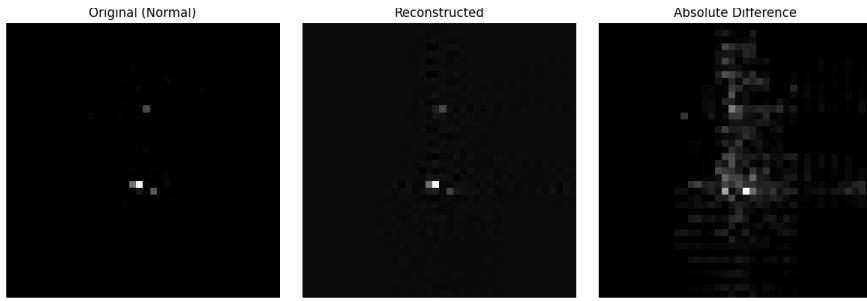


Figure 5.14: Normal QCD jet: original (left), reconstructed (middle), and MAE heatmap (right).

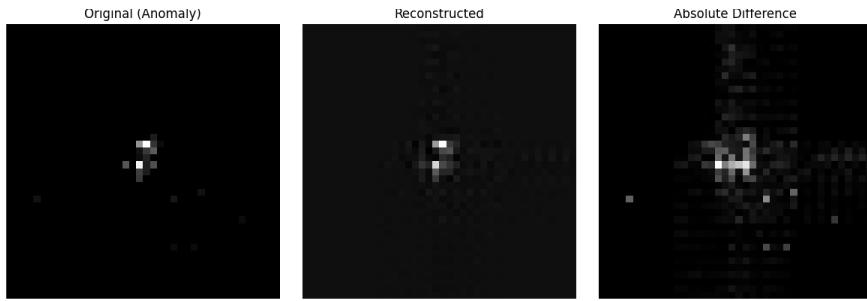


Figure 5.15: Anomalous Top jet: original (left), reconstructed (middle), and MAE heatmap (right).

In Figure 5.14, MAE also does a good job in reconstructing the bright pixels but it creates a lot of fuzzy effects near the bright pixels for both QCD and Top jet images. The QCD jet image reconstruction still retains key details, with minimal error concentrated in low-activity zones but in case of the Top jets in Figure 5.15 exhibits strong error regions corresponding to substructure features.

## Distribution of MAE Scores

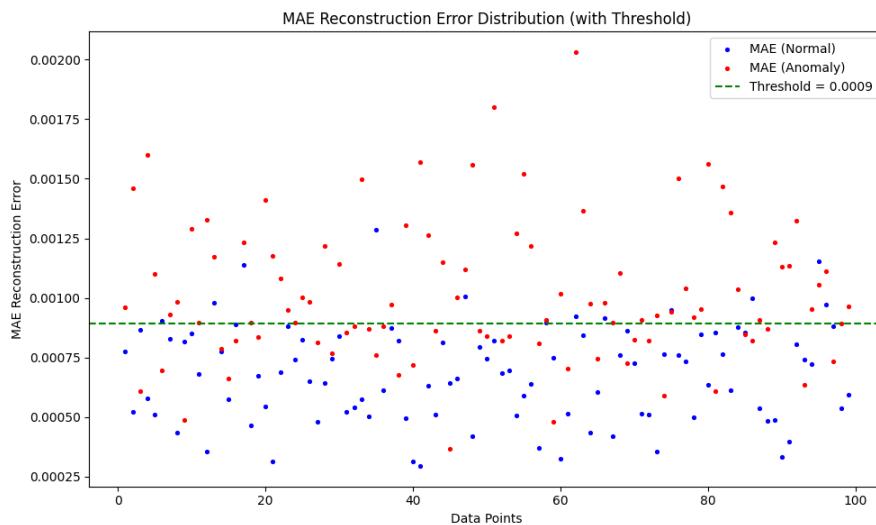


Figure 5.16: Scatter plot of Mean Absolute Error (MAE) scores for QCD and Top jets. The red line indicates the anomaly threshold.

Figure 5.16 presents a scatter plot showing the reconstruction scores (MAE in this case) for individual jet images. The QCD jets are densely clustered around low error values, while the Top jets exhibit a broader spread with many points lying beyond the threshold, validating the hypothesis that anomalous jets are harder to reconstruct. So in this case also MAE is able to flag to the anomalies, to have a better understanding of the model's performance using the MAE loss, we will have a look at the confusion matrix and ROC plots.

## Confusion Matrix for MAE

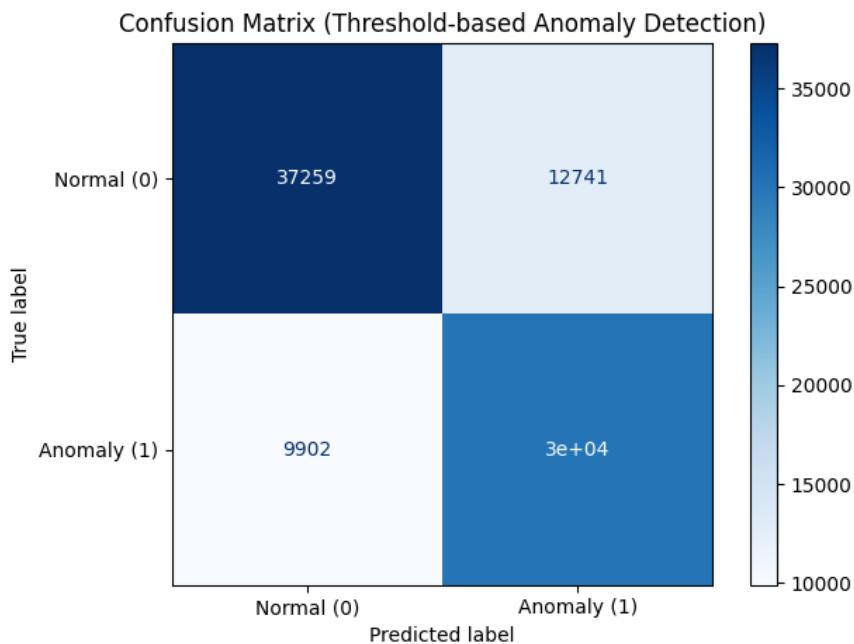


Figure 5.17: Confusion matrix showing the classification of QCD (normal) and Top (anomalous) jets at the chosen threshold.

Figure 5.17 shows the confusion matrix obtained from applying a threshold on the reconstruction error to classify jets as either normal (QCD) or anomalous (top). The entries in the matrix are as follows:

- **True Negatives (TN):** 37,259 QCD jets correctly classified as normal.
- **False Positives (FP):** 12,741 QCD jets misclassified as anomalous.
- **False Negatives (FN):** 9,902 top jets misclassified as normal.
- **True Positives (TP):** 30,084 top jets correctly classified as anomalous.

From this, we calculate the performance metrics:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{30084 + 37259}{90086} \approx 0.7475,$$

$$\text{Precision (for anomalies)} = \frac{TP}{TP + FP} = \frac{30084}{30084 + 12741} \approx 0.7026,$$

$$\text{Recall (for anomalies)} = \frac{TP}{TP + FN} = \frac{30084}{30084 + 9902} \approx 0.7523,$$

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \approx 0.7266.$$

This version of the model achieves a good balance between recall and precision. While it detects a higher number of anomalous top jets (improved recall), it also incurs more false positives, slightly lowering the precision. This illustrates a key trade-off in anomaly detection that we will continue to evaluate with different loss functions and scoring methods.

Metric	Value
Accuracy	0.747
Precision	0.702
Recall (True Positive Rate)	0.752
F1 Score	0.726

Table 5.2: From the obtained confusion matrix 5.17 we are calculating accuracy, precision, recall and the F1 score values

## ROC plot for MAE

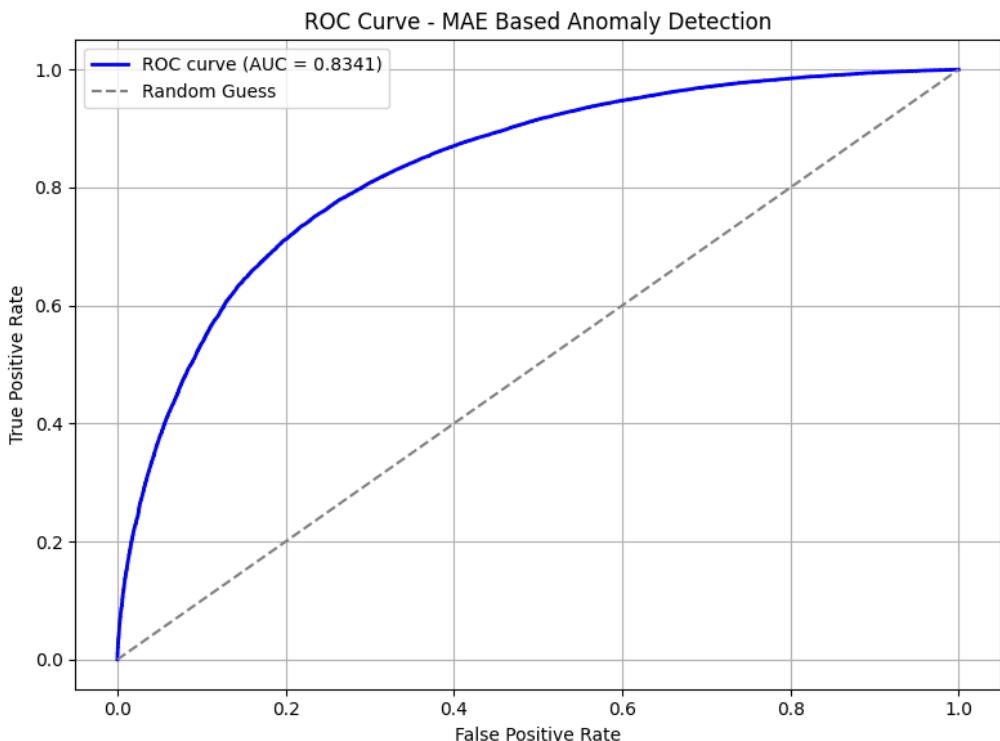


Figure 5.18: ROC plot for MAE loss function

As we can see, the model yields a ROC AUC score of 0.8341, which indicates reasonably good performance in distinguishing anomalous jets from normal ones. However, this value is slightly lower compared to the ROC AUC achieved when using Mean Squared Error (MSE) as the anomaly scoring metric. This suggests that while the current loss function or scoring approach is effective, it may not capture the underlying structural differences between QCD and top jets as well as MSE does

## 5.5 Model Evaluation and Anomaly Scoring using Structural Similarity (SSIM)

### SSIM Visualizations: QCD vs Top

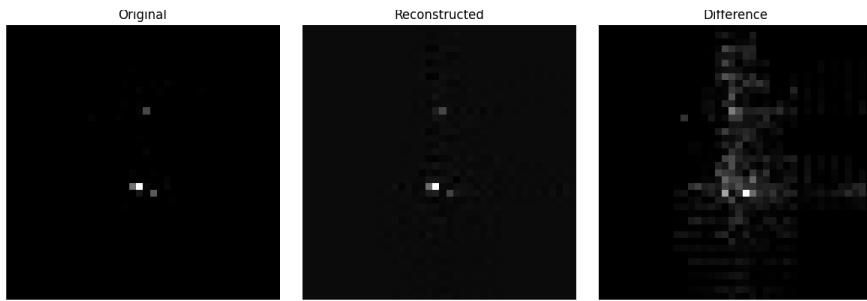


Figure 5.19: SSIM visualization for a QCD jet: original (left), reconstruction (middle), and SSIM error map (right).

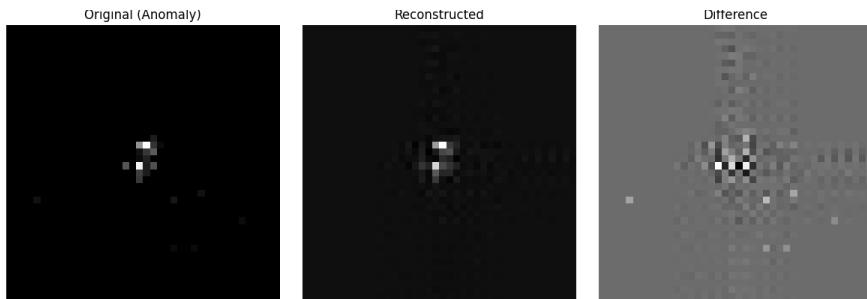


Figure 5.20: SSIM visualization for a Top jet: original (left), reconstruction (middle), and SSIM error map (right).

The SSIM visualizations offer further insight into perceptual discrepancies. For QCD jets (Figure 5.19), the reconstruction retains all important visual structures, with low error concentrated around edges. On the other hand, the Top jet (Figure 5.20) suffers severe structural mismatches in key regions, corroborating its status as an outlier.

While MSE captures pixel-wise intensity differences, it may not fully account for structural distortions. To address this, we also evaluate using the Structural Similarity Index (SSIM), which considers luminance, contrast, and structure.

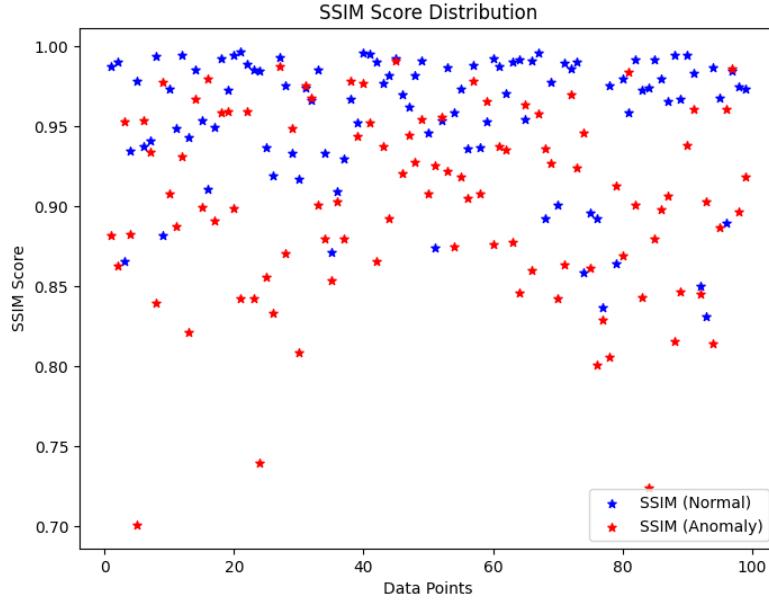


Figure 5.21: Distribution of SSIM scores for QCD and Top jets. Lower SSIM indicates higher anomaly.

Figure 5.21 shows the Structural Similarity Index (SSIM) score distribution for a sample of 100 jets, equally divided between QCD (normal) and top (anomalous) jets. The SSIM metric evaluates the perceptual similarity between the input and the reconstructed jet image, with values ranging from 0 to 1 — where 1 indicates perfect structural similarity.

In this plot, we observe that:

- **Normal (QCD) jets** generally achieve higher SSIM scores, clustering closer to 1. This indicates that the autoencoder is able to reconstruct these jets with high structural fidelity, as expected since the model was trained on QCD data.
- **Anomalous (top) jets**, represented by red stars, show a more spread-out distribution of SSIM scores, with several significantly lower than those of the normal jets. This suggests that the autoencoder struggles to preserve the structural features of jets it has not seen during training, making SSIM a useful metric for distinguishing between the two classes.

This visual distinction reinforces the hypothesis that reconstruction quality, as measured by SSIM, can serve as an effective anomaly detection criterion. In subsequent sections, we compare this method quantitatively against MSE and MAE-based scoring to evaluate its impact on classification performance.

Figure 5.21 shows a clear separation between the SSIM distributions of QCD and Top jets. Most QCD jets have SSIM scores close to 1, indicating highly faithful reconstructions. Top jets, however, show a pronounced shift toward lower values, signaling distorted reconstructions.

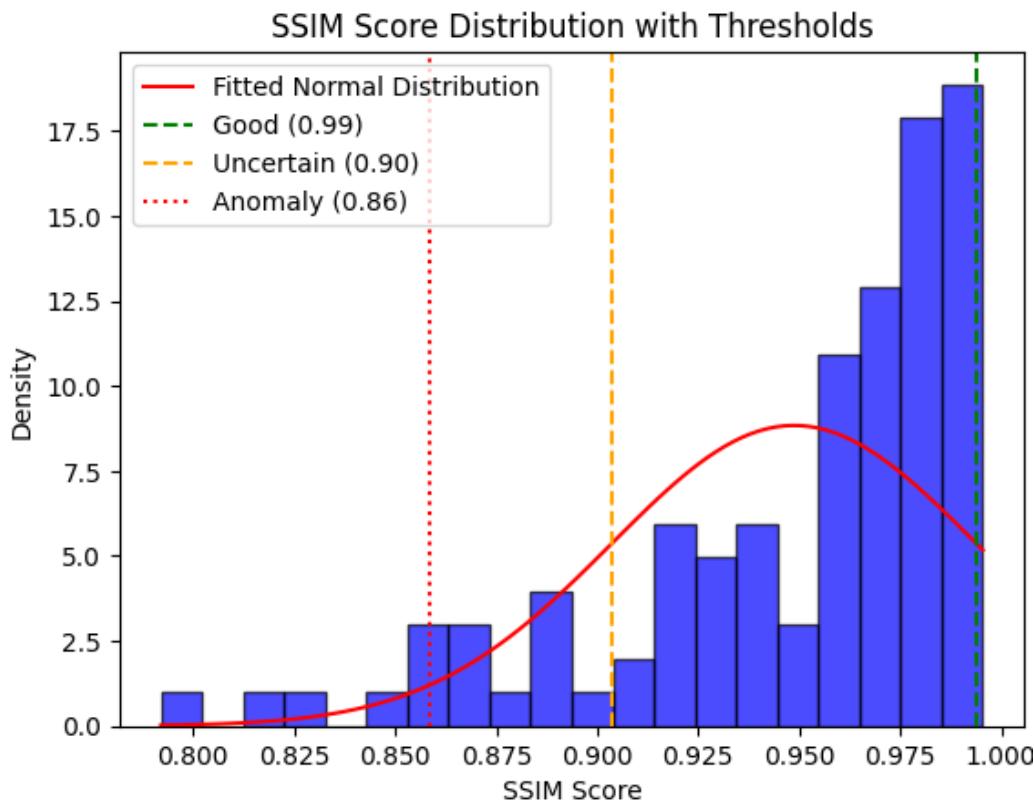


Figure 5.22: SSIM anomaly score with chosen threshold (red line).

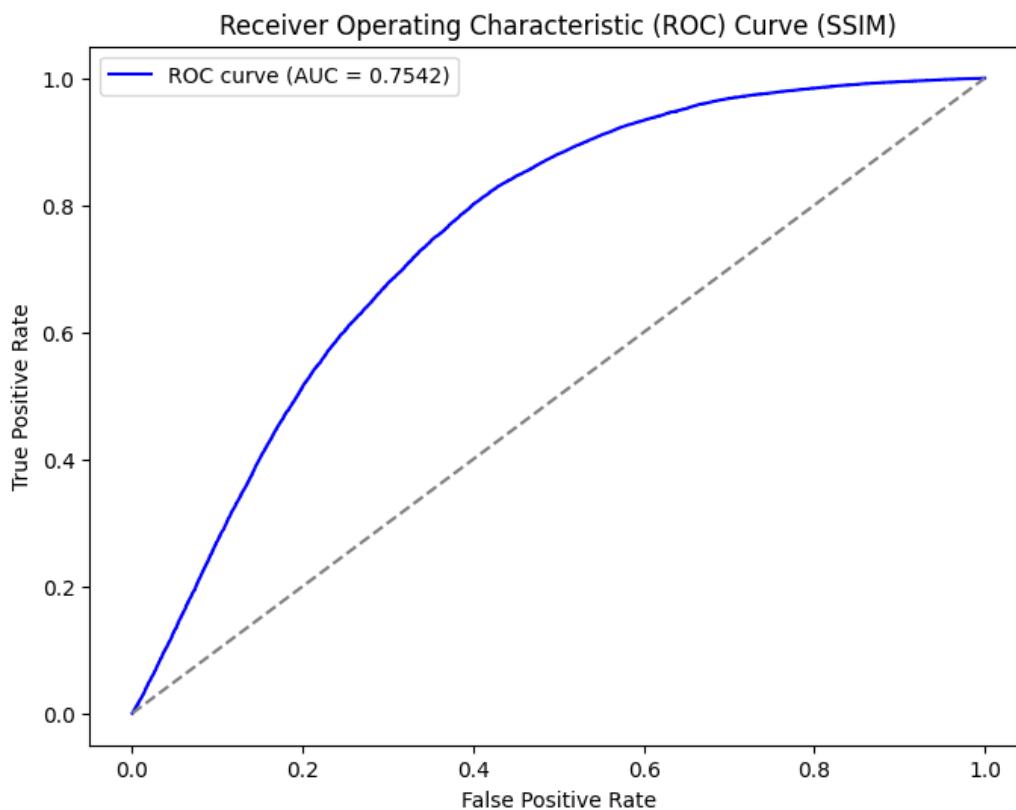


Figure 5.23: RROC curve for anomaly detection using SSIM scores

## SSIM Confusion Matrix

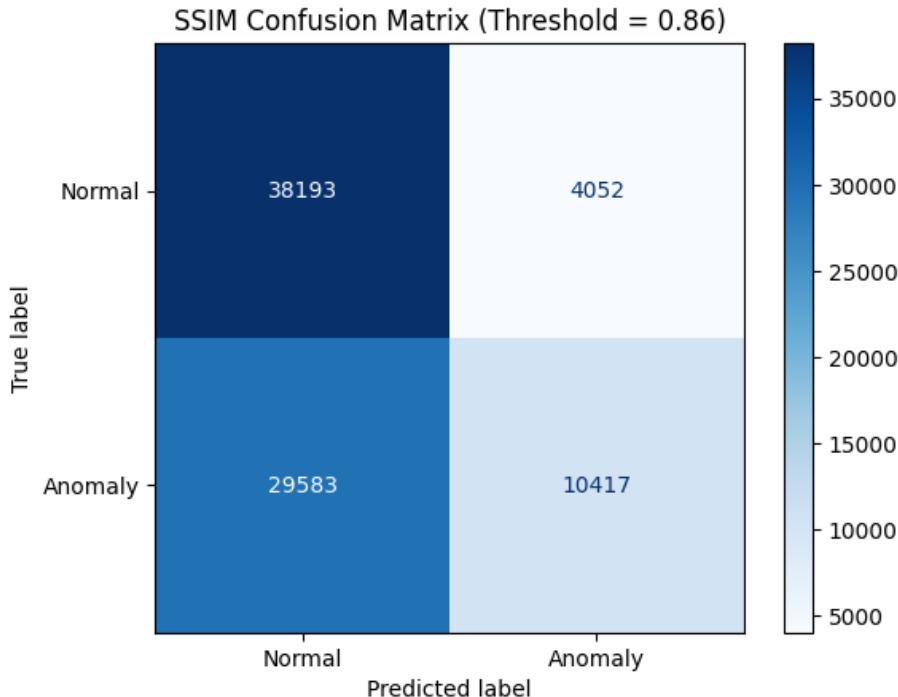


Figure 5.24: SSIM Confusion Matrix

To quantitatively evaluate the performance of our anomaly detection method using the Structural Similarity Index Measure (SSIM), we compute a confusion matrix. A threshold of **SSIM = 0.86** was used to classify an image as anomalous. The results are summarized below:

### Interpretation of Terms:

- **True Negatives (TN):** 38,193 normal images were correctly classified as normal.
- **False Positives (FP):** 4,052 normal images were incorrectly flagged as anomalies.
- **False Negatives (FN):** 29,583 anomalous images were incorrectly classified as normal.
- **True Positives (TP):** 10,417 anomalous images were correctly detected as anomalies.

This confusion matrix highlights that the SSIM-based approach is reasonably effective in detecting anomalies, with a total of 10,417 true positives. However, the relatively high number of false negatives (29,583) indicates that a significant number of anomalies are not being detected. This is a known limitation of reconstruction-based anomaly detection models, particularly when the autoencoder generalizes well to out-of-distribution inputs.

### Optional: Performance Metrics

To further quantify performance, we compute the following metrics:

- **Accuracy:**

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{10417 + 38193}{82245} \approx \mathbf{0.5910}$$

- **Precision:**

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{10417}{10417 + 4052} \approx \mathbf{0.7201}$$

- **Recall:**

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{10417}{10417 + 29583} = \frac{10417}{40000} \approx \mathbf{0.2604}$$

- **F1 Score:**

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 2 \times \frac{0.7201 \times 0.2604}{0.7201 + 0.2604} \approx \mathbf{0.3828}$$

Metric	Value
Accuracy	0.591
Precision	0.72
Recall (True Positive Rate)	0.26
F1 Score	0.382

Table 5.3: From the obtained confusion matrix 5.24 we are calculating accuracy, precision, recall and the F1 score values

These metrics provide additional insight into the trade-off between false positives and false negatives, especially important in evaluating anomaly detection tasks.

The SSIM score scatter plot in Figure 5.21 describes this distinction with an appropriately chosen threshold, the model is able to separate normal and anomalous distributions but if we look at accuracy it is worse than that of MSE and MAE.

These results validate the efficiency of a convolutional autoencoder in a semi-supervised anomaly detection setting for jet substructure analysis using **MSE** error function as it has outstand the MAE, SSIM function. So as we move forward in this section, now we are going to discuss the model evaluation for different  $p_T$ s, as mentioned in the beginning of this section, we are going now going to discuss results for 750 GeV, 1500 GeV and 5000 GeV  $p_T$  using MSE error function. We are going to describe reconstruction error distribution and model's ability to distinguish between QCD and top jets at higher energy scales.

## 5.6 Conclusion

As we progress through this part of the chapter, it becomes very much evident that increasing the transverse momentum ( $p_T$ ) of jets introduces significant challenges for anomaly detection using autoencoders. A clear indication of this trend is the consistent decrease in the ROC AUC

as  $p_T$  increases. This performance drop prompts a deeper analysis of the underlying physical and algorithmic reasons.

To understand the reason behind this phenomenon, we must revisit the jet formation process, as discussed earlier in Chapter 4. Jets are collimated sprays of hadrons originating from the hadronization of high-energy quarks or gluons. When the transverse momentum of the initiating parton increases, the resulting jet becomes increasingly narrow due to Lorentz boosting. This collimation has specific consequences for image-based jet analysis. Since we treat jets as 2D images (representing density deposition across calorimeter pixels), higher  $p_T$  jets result in more compact spatial distributions. At moderate transverse momenta, complex substructures, such as the characteristic three-prong sub-structure of a hadronically decaying top quark, are spatially resolvable. For instance, the decay chain:

$$pp \rightarrow t\bar{t} \rightarrow Wb \rightarrow q\bar{q}'b, \quad (5.1)$$

produces three final-state quarks, which should ideally form three subjets. This leads to a three-prong structure in the jet image. Autoencoders trained only on QCD jets can thus detect these top jets as anomalies by exhibiting higher reconstruction errors.

However, at higher  $p_T$  (e.g., 2000 GeV and beyond), the decay products of the top quark become collimated. The angular separation between the prongs is significantly reduced due to the boost, and the sub-structure of the jets begin to visually overlap in the 2D projection. Eventually, the jet image will look like a single bright core, making it visually similar to QCD jets. This presents a major limitation for the autoencoder model. Since it relies on reconstructing QCD-like jets accurately and registering high reconstruction error for anomalous jets (like tops), its ability to detect anomalies diminishes when the anomalies become visually indistinguishable from the normal class. In effect, the reconstruction error for high-anomaly jets  $p_T$  is no longer significantly different from that of QCD jets, leading to reduced separability in anomaly scoring. This performance degradation is quantitatively reflected in Figure B.9, where the ROC AUC drops to approximately 0.6271 for  $p_T = 500$  GeV. Such a value is only marginally better than random classification ( $AUC = 0.5$ ), highlighting the severe limitations of reconstruction-based semi-supervised learning methods at high transverse momentum.

These observations underscore an important insight: While autoencoders may perform well in the low to moderate  $p_T$  regime, their effectiveness diminishes as the energy scale increases, mainly because we are interested in higher energy scales because we might observe some signature of the BSM physics and BSM particles providing boosted top quark decay or some other process. This calls for alternative approaches, such as including angular features, using high-level physics-inspired variables, or integrating clustering techniques or attention mechanisms to help recover substructure information at high  $p_T$ . The pictorial representation of model evaluation for high  $p_T$  jets has been discussed in the Appendix B.

# Appendix A

## Youden's J Threshold

In binary classification problems, the choice of threshold for decision-making greatly influences the trade-off between true positive and false positive rates. One commonly used metric to determine the optimal threshold is **Youden's J statistic**, also known as **Youden's Index**.

Youden's J is defined as:

$$J = \text{TPR} - \text{FPR} = \text{Sensitivity} + \text{Specificity} - 1, \quad (\text{A.1})$$

where:

- **TPR** (True Positive Rate) or **Sensitivity** is the fraction of actual positives correctly identified.
- **FPR** (False Positive Rate) is the fraction of actual negatives incorrectly identified as positives.
- **Specificity** is defined as  $1 - \text{FPR}$ .

The threshold that maximizes  $J$  is considered optimal, as it represents the point on the ROC curve that is farthest from the diagonal line (random guess). This threshold balances the trade-off between sensitivity and specificity, making it particularly useful in imbalanced datasets or anomaly detection tasks.

In this work, Youden's J threshold is used to binarize the anomaly scores output by the model. The point that yields the highest value of  $J$  is selected as the classification threshold. This allows for consistent evaluation of performance metrics such as accuracy, precision, recall, and F1-score based on a statistically justified criterion.

## Appendix B

### Model Evaluation and Anomaly Scoring for different $p_T$ values

$p_T = 750 \text{ GeV}$

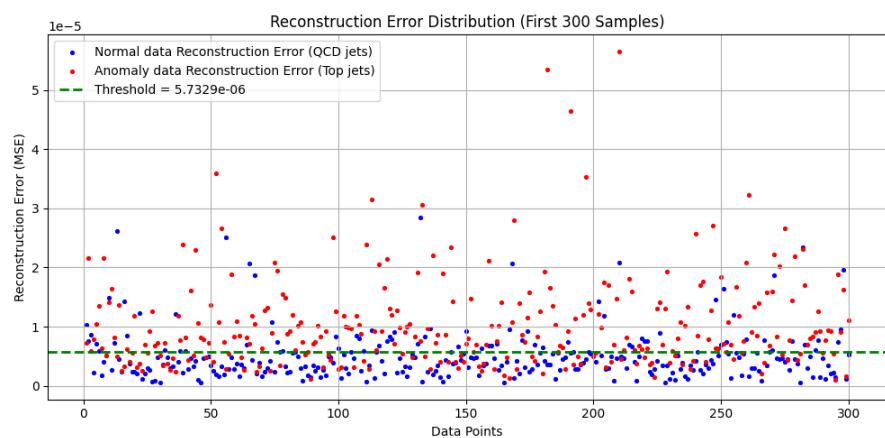


Figure B.1: Scatter plot of Mean Absolute Error (MSE) scores for QCD and Top jets at 750 GeV.

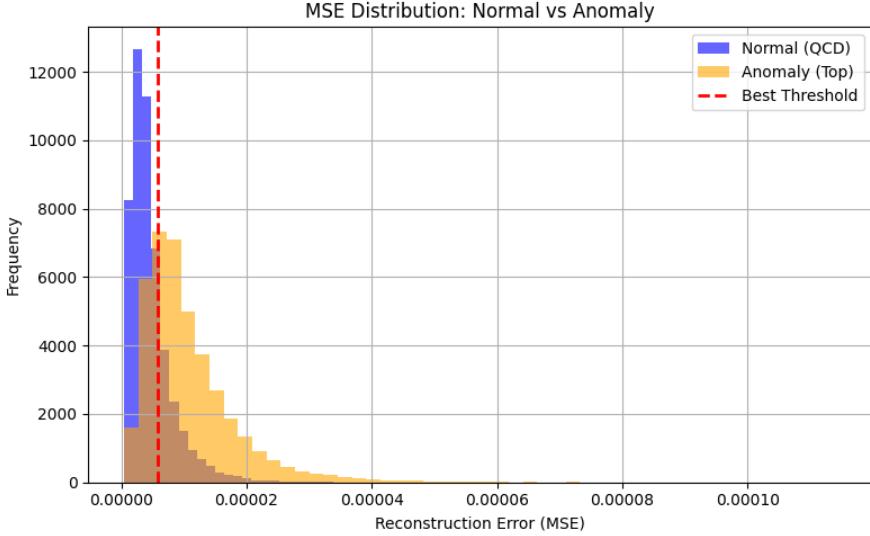


Figure B.2: Histogram of Mean Squared Error (MSE) scores for QCD and Top jets at 750 GeV.

As we can observe, anomaly data reconstruction error has shifted down in the figure B.1 compare to figure 5.9 and similarly for histogram, anomalous data has shifted to the left, low reconstruction error zone.

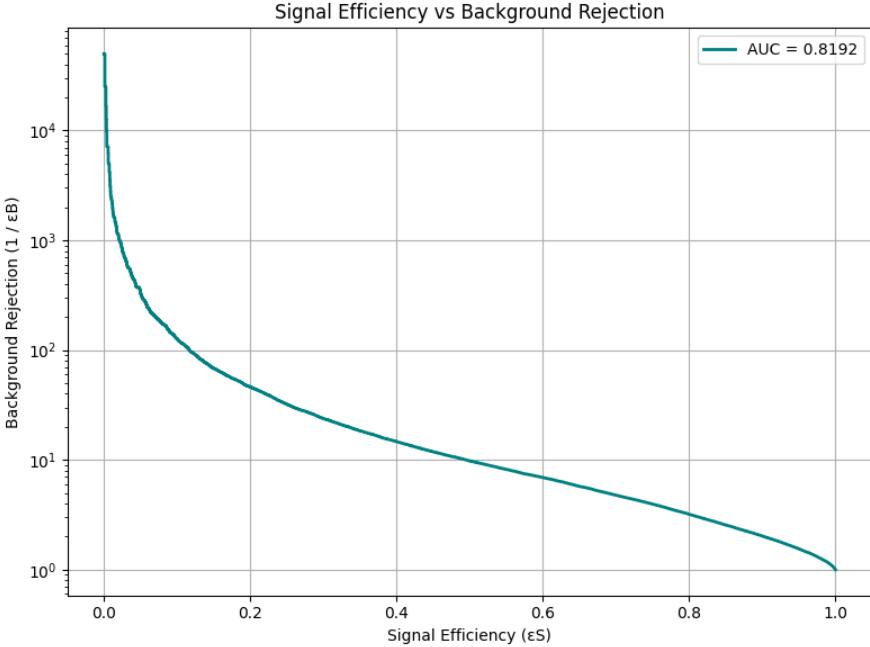


Figure B.3: Signal Efficiency vs Background rejection for 750 GeV  $p_T$  jets.

If we fix the signal efficiency at  $\epsilon_S = 0.6$ , we can compare the current performance with that shown in Figure 5.13. Previously, at this level of signal efficiency, the background rejection was higher—corresponding to a background efficiency of  $\epsilon_B = 0.1$ . This means that out of 10,000 QCD jets, only 1,000 were misclassified as signal (false positives), while the remaining 9,000 were correctly rejected. However, in the current case, we observe a decrease in background rejection, implying an increase in background efficiency. As a result, a larger

## APPENDIX B. MODEL EVALUATION AND ANOMALY SCORING FOR DIFFERENT $p_T$ VALUES

---

number of QCD jets are now being misclassified as signal, which leads to a higher contamination of the signal region by background events.

### $p_T = 1500 \text{ GeV}$

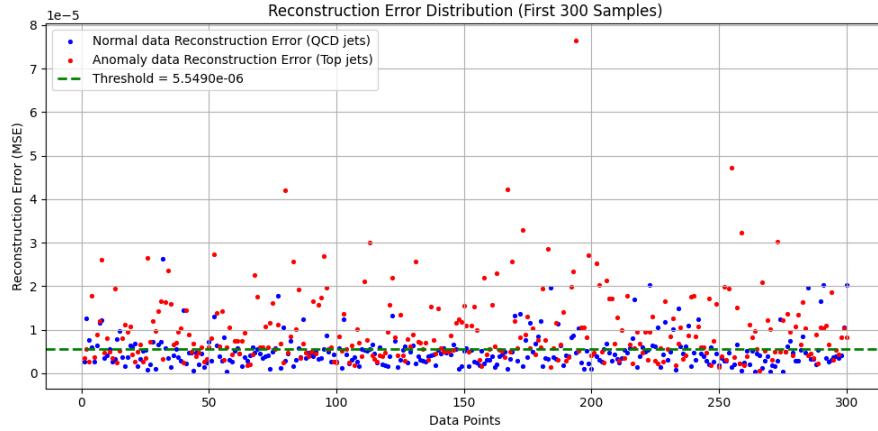


Figure B.4: Scatter of Mean Squared Error (MSE) scores for QCD and Top jets. The red line indicates the anomaly threshold.

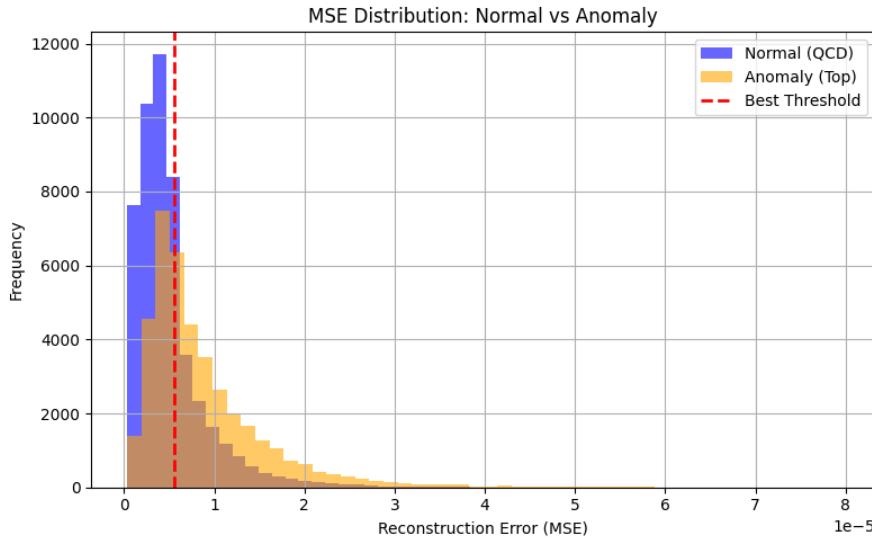


Figure B.5: Histogram of Mean Squared Error (MSE) scores for QCD and Top jets. The red line indicates the anomaly threshold.

APPENDIX B. MODEL EVALUATION AND ANOMALY SCORING FOR DIFFERENT  $p_T$  VALUES

---

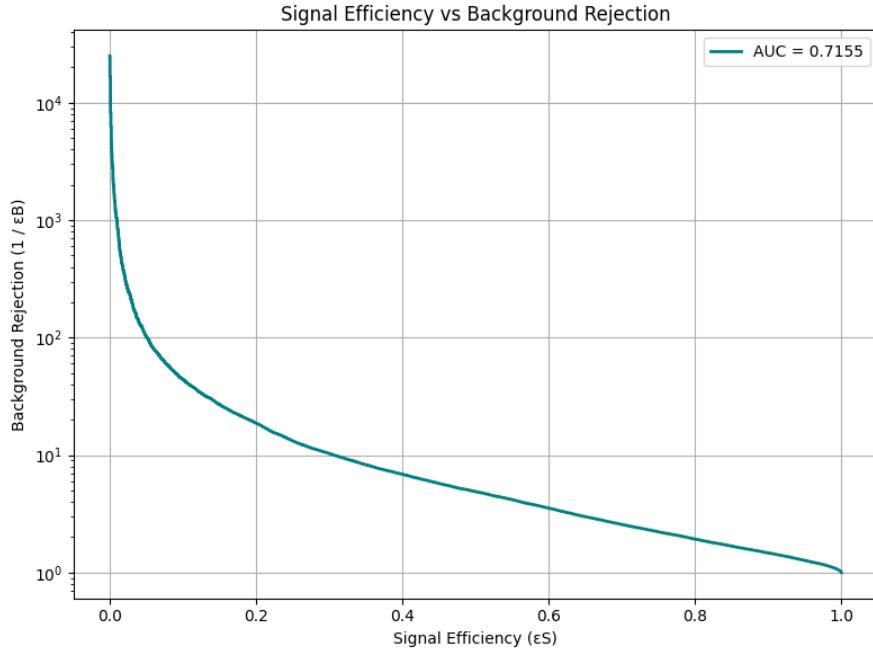


Figure B.6: Signal Efficiency vs Background rejection for 1500 GeV  $p_T$  jets.

$p_T = 5000$  GeV

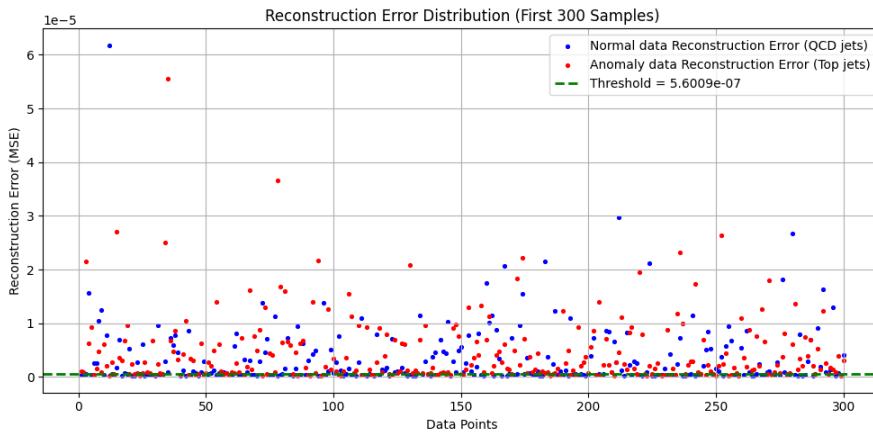


Figure B.7: Scatter plot of Mean Squared Error (MSE) scores for QCD and Top jets. The red line indicates the anomaly threshold.

## APPENDIX B. MODEL EVALUATION AND ANOMALY SCORING FOR DIFFERENT $p_T$ VALUES

---

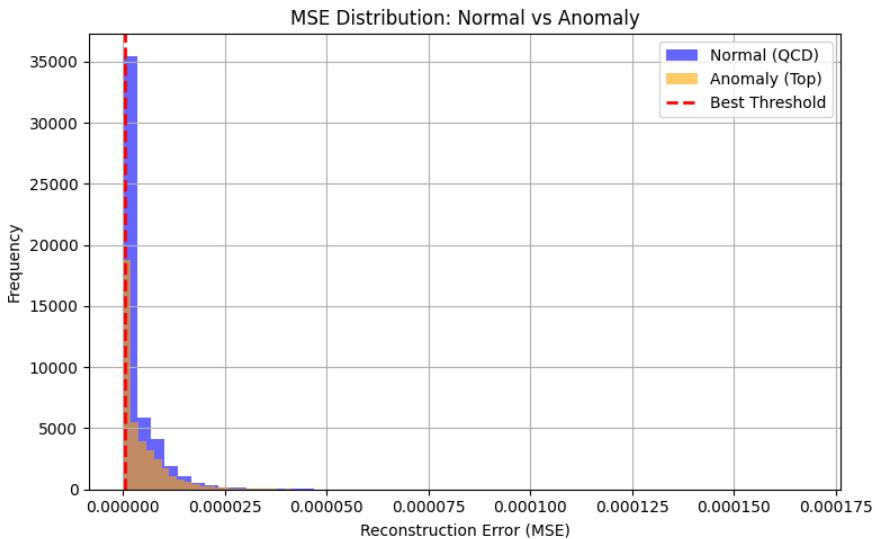


Figure B.8: Histogram of Mean Squared Error (MSE) scores for QCD and Top jets. The red line indicates the anomaly threshold.

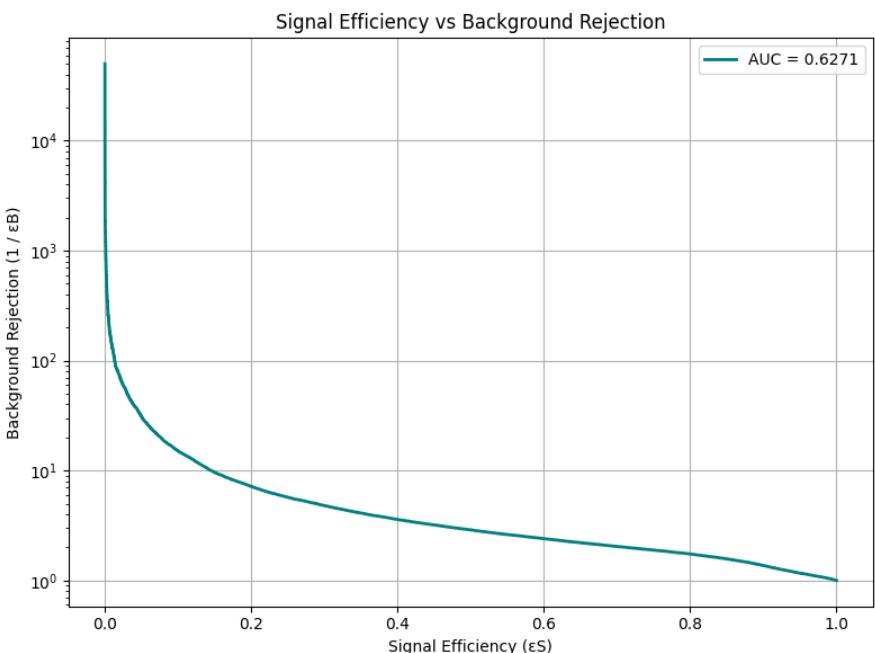


Figure B.9: Signal Efficiency vs Background rejection for 5000 GeV  $p_T$  jets

# Appendix C

## Unsupervised Learning

### Tuning the Bottleneck dimension

In this section, we discuss the role of **bottleneck dimensionality** in unsupervised learning, particularly in the context of jet image analysis. As highlighted by the authors in [29], exploring various bottleneck sizes helps identify the optimal dimensionality at which **QCD jet images can be compressed** effectively while still retaining their essential features. In contrast, **top jet images**, due to their more complex internal structure, tend to lose more information during compression, leading to **higher reconstruction errors**.

This behavior is crucial for unsupervised learning frameworks, where the model is trained on mixed inputs containing both top and QCD jets—without explicit labels. When the bottleneck size is appropriately tuned, the **autoencoder naturally learns to reconstruct QCD jets more effectively** than top jets. As a result, **top jets stand out as anomalies** due to their larger reconstruction errors, enabling a clear separation between the two jet types in the latent space.

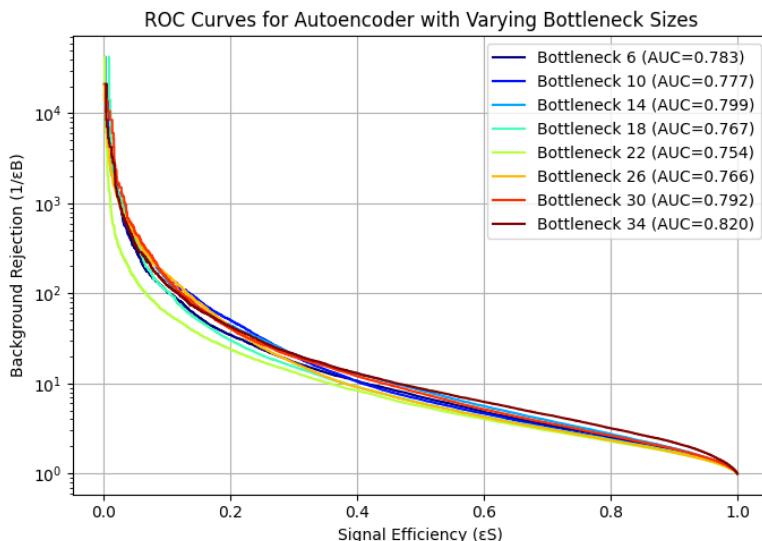


Figure C.1: Model evaluation for various bottle dimentions as described above in the text.

### C.0.1 Autoencoder architecture

Layer Type	Output Shape	Activation
Input	(40, 40, 1)	—
Conv2D (16 filters, 3x3)	(40, 40, 16)	ReLU
MaxPooling2D (2x2)	(20, 20, 16)	—
Conv2D (32 filters, 3x3)	(20, 20, 32)	ReLU
MaxPooling2D (2x2)	(10, 10, 32)	—
Conv2D (64 filters, 3x3)	(10, 10, 64)	ReLU
MaxPooling2D (2x2)	(5, 5, 64)	—
Flatten	(1600,)	—
Dense (bottleneck)	(variable bottleneck size)	ReLU
Dense	(1600)	ReLU
Reshape	(5, 5, 64)	—
Conv2D (64 filters, 3x3)	(5, 5, 64)	ReLU
UpSampling2D (2x2)	(10, 10, 64)	—
Conv2D (32 filters, 3x3)	(10, 10, 32)	ReLU
UpSampling2D (2x2)	(20, 20, 32)	—
Conv2D (16 filters, 3x3)	(20, 20, 16)	ReLU
UpSampling2D (2x2)	(40, 40, 16)	—
Conv2D (1 filter, 3x3)	(40, 40, 1)	Linear

Table C.1: Autoencoder architecture with bottleneck size as a variable parameter.

# Appendix D

## Model Evaluation and Anomaly Scoring for Higgs process

### Introduction

As discussed in the chapter 3 last section about the Higgs process and it's decay to quarks, we now perform the same analysis as we did for top jets and evaluate model based on the ROC score and the confusion matrix. We have simulated jet images producing from the Higgs for  $10^5$  events, but because of low cross-section of Higgs, we are obtaining low number of jets.

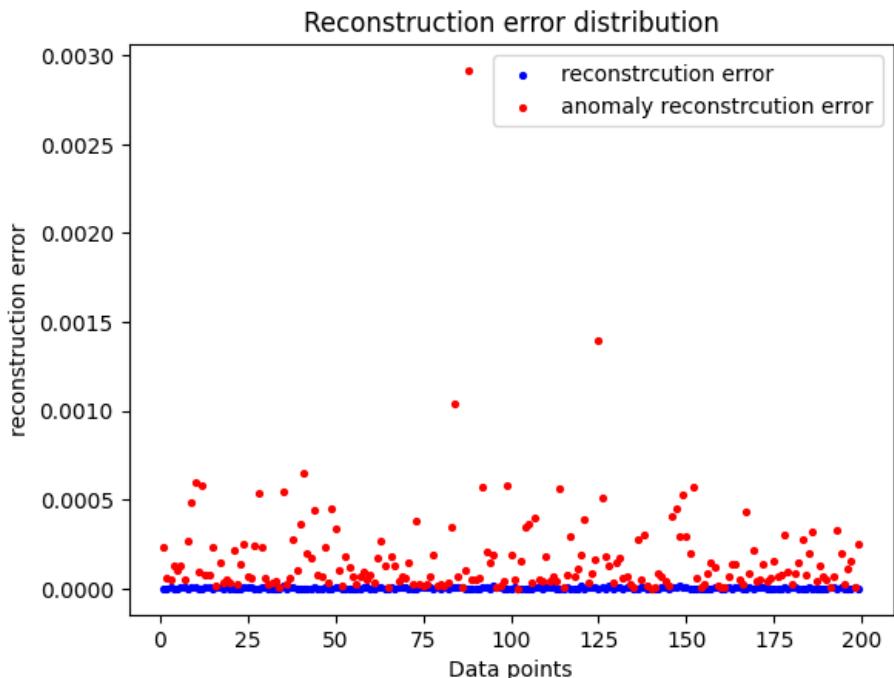


Figure D.1: Scatter plot of Mean Absolute Error (MSE) scores for QCD and Higgs decay jets at 450 GeV.

We can observe that in the scatter plot, the model is not doing a good job in reproducing

## APPENDIX D. MODEL EVALUATION AND ANOMALY SCORING FOR HIGGS PROCESS

---

the Higgs decay jet images which is resulting in high reconstruction error compare to QCD jets.

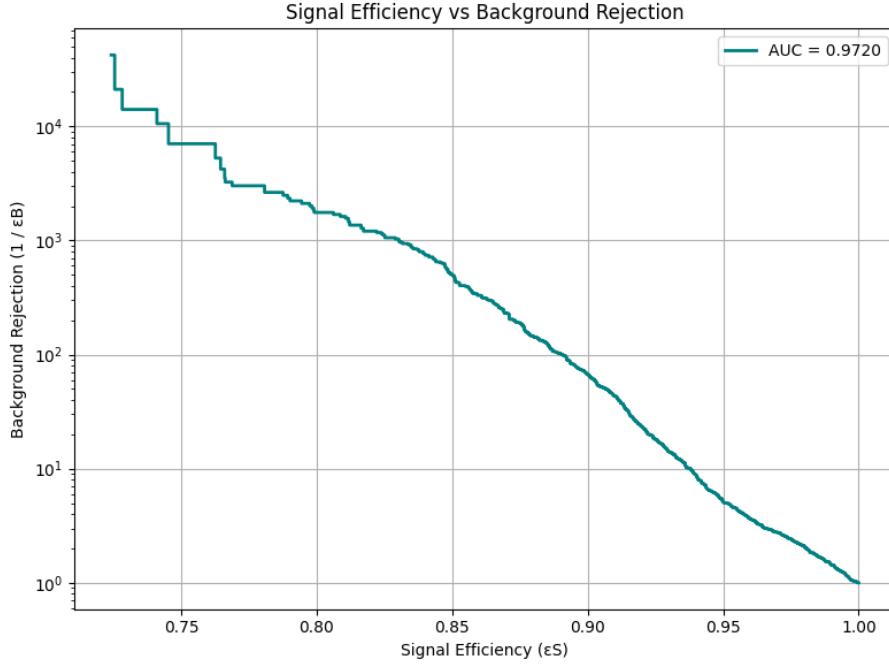


Figure D.2: Histogram of Mean Squared Error (MSE) scores for QCD and Higgs decay jets at 450 GeV.

Figure D.2 shows the curve of signal efficiency ( $\varepsilon_S$ ) versus background rejection ( $1/\varepsilon_B$ ) for the anomaly detection model trained to distinguish Higgs jets (signal) from QCD jets (background).

A high signal efficiency corresponds to correctly identifying Higgs jets, while a high background rejection implies effectively filtering out QCD jets. The curve demonstrates that the model achieves strong performance, with an area under the curve (AUC) of 0.9720, indicating excellent separation between the two classes.

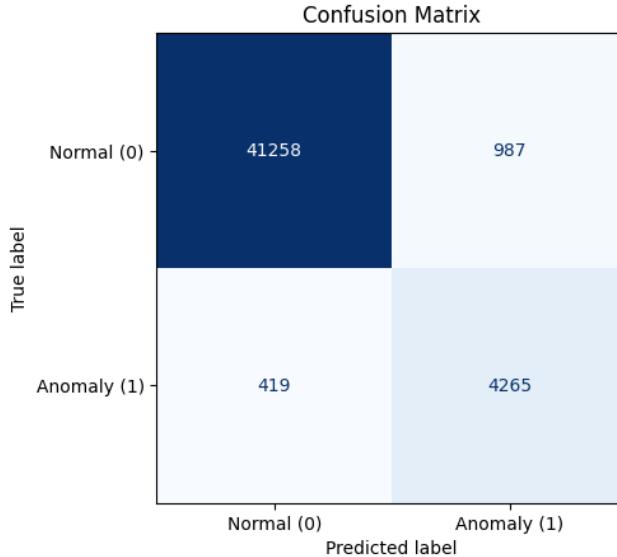


Figure D.3: Signal Efficiency vs Background rejection QCD and Higgs decay jets at 450 GeV.

Metric	Value
Accuracy	0.9700
Precision	0.812
Recall (TPR)	0.910
F1 Score	0.858

 Table D.1: Performance metrics for anomaly detection based on the confusion matrix [D.3](#).

From the output we can easily conclude that, model is doing an excellent job in differentiating between QCD jets and Higgs decay products, because of time constraint we are not extending our discussion to higher  $p_T$  values, we expect that there will be some decrement in the values of ROC, accuracy, precision, recall and F1 score.

# Bibliography

- [1] Tasneem Saleem. “Development of pixel detector for ATLAS Inner Tracker(ITK) upgrade at HL-LHC and Searching for the Standard Model Higgs boson decay into b-quark pair with ATLAS experiment”. In: (Jan. 2019).
- [2] Lumen Learning. *GUTs: The Unification of Forces*. Accessed: 2025-02-24. n.d. URL: <https://courses.lumenlearning.com/suny-physics/chapter/33-6-guts-the-unification-of-forces/>.
- [3] Simone Marzani, Gregory Soyez, and Michael Spannowsky. *Looking Inside Jets: An Introduction to Jet Substructure and Boosted-object Phenomenology*. Springer International Publishing, 2019. ISBN: 9783030157098. DOI: [10.1007/978-3-030-15709-8](https://doi.org/10.1007/978-3-030-15709-8). URL: <http://dx.doi.org/10.1007/978-3-030-15709-8>.
- [4] John Ellis. “Physics Beyond the Standard Model”. In: *Nuclear Physics A* 827.1–4 (Aug. 2009), pp. 187c–198c. ISSN: 0375-9474. DOI: [10.1016/j.nuclphysa.2009.05.034](https://doi.org/10.1016/j.nuclphysa.2009.05.034). URL: <http://dx.doi.org/10.1016/j.nuclphysa.2009.05.034>.
- [5] Michael Spannowsky. *High-pT Taggers*. <https://www.ippp.dur.ac.uk/~mspannow/HighpTTaggers.html>. Accessed: 2025-04-10. n.d.
- [6] Sanchari Bhattacharyya et al. *Tagging ultra-boosted jets at FCC-hh using machine learning techniques*. 2025. arXiv: [2501.06702 \[hep-ph\]](https://arxiv.org/abs/2501.06702). URL: <https://arxiv.org/abs/2501.06702>.
- [7] CERN. *The Large Hadron Collider*. Accessed: 2025-03-05. 2024. URL: <https://home.web.cern.ch/science/accelerators/large-hadron-collider>.
- [8] Wikipedia contributors. *High Luminosity Large Hadron Collider*. Accessed: 2025-03-05. 2024. URL: [https://en.wikipedia.org/wiki/High\\_Luminosity\\_Large\\_Hadron\\_Collider](https://en.wikipedia.org/wiki/High_Luminosity_Large_Hadron_Collider).
- [9] Julia Bauer. “Perspektiven zur Beobachtung der elektroschwachen Produktion einzelner Top-Quarks mit dem CMS Experiment”. In: (Jan. 2010).
- [10] Jai Bardhan. “Revisiting Deep Learning for Particle Physics”. Master’s thesis. International Institute of Information Technology, Hyderabad, 2023. URL: [https://web2py.iiit.ac.in/research\\_centres/publications/view\\_publication/mastersthesis/1269](https://web2py.iiit.ac.in/research_centres/publications/view_publication/mastersthesis/1269).
- [11] Stefanos Dris, C Foudas, and J. Troska. “Performance of the CMS Tracker Optical Links and Future Upgrade Using Bandwidth Efficient Digital Modulation”. In: (Apr. 2010).

- [12] Wikipedia contributors. *Mandelstam variables*. Accessed: 2025-03-07. 2025. URL: [https://en.wikipedia.org/wiki/Mandelstam\\_variables](https://en.wikipedia.org/wiki/Mandelstam_variables).
- [13] ATLAS Collaboration. *Rapidity*. Accessed: 2025-03-07. 2025. URL: <https://atlas.cern/glossary/rapidity>.
- [14] MadGraph5\_aMC@NLO. *MadGraph5\_aMC@NLO*. Accessed: 2025-03-07. 2025. URL: <https://launchpad.net/mg5amcnlo>.
- [15] PYTHIA Collaboration. *PYTHIA Event Generator*. Accessed: 2025-03-07. 2025. URL: <https://www.pythia.org>.
- [16] Torbjörn Sjöstrand et al. “An introduction to PYTHIA 8.2”. In: *Computer Physics Communications* 191 (June 2015), pp. 159–177. ISSN: 0010-4655. DOI: [10.1016/j.cpc.2015.01.024](https://doi.org/10.1016/j.cpc.2015.01.024). URL: <http://dx.doi.org/10.1016/j.cpc.2015.01.024>.
- [17] Delphes Collaboration. *Delphes: A framework for fast simulation of a generic collider experiment*. Accessed: 2025-03-07. 2025. URL: <https://github.com/delphes/delphes>.
- [18] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. “FastJet user manual: (for version 3.0.2)”. In: *The European Physical Journal C* 72.3 (Mar. 2012). ISSN: 1434-6052. DOI: [10.1140/epjc/s10052-012-1896-2](https://doi.org/10.1140/epjc/s10052-012-1896-2). URL: <http://dx.doi.org/10.1140/epjc/s10052-012-1896-2>.
- [19] Yves M. Räth et al. “Settlement relationships and their morphological homogeneity across time and scale”. In: *Scientific Reports* 13 (1 2023), pp. 1–15. DOI: [10.1038/s41598-023-38338-9](https://doi.org/10.1038/s41598-023-38338-9).
- [20] Brian M. Dillon. *QCD-Top Jet Images Preprocessing*. <https://github.com/bmdillon/jet-preprocessing>. Accessed: 2025-04-03. 2021.
- [21] HEP ML Community. *A Living Review of Machine Learning for Particle Physics*. URL: <https://iml-wg.github.io/HEPML-LivingReview/>.
- [22] Swagata Mandal. *Swagata Mandal’s GitHub Profile*. <https://github.com/swagata87>. Accessed: 2025-04-07. 2025.
- [23] Keiron O’Shea and Ryan Nash. *An Introduction to Convolutional Neural Networks*. 2015. arXiv: [1511.08458 \[cs.NE\]](https://arxiv.org/abs/1511.08458). URL: <https://arxiv.org/abs/1511.08458>.
- [24] Saleh Alaraими et al. “Transfer Learning Networks with Skip Connections for Classification of Brain Tumors”. In: *International Journal of Imaging Systems and Technology* 31 (Feb. 2021), pp. 1–19. DOI: [10.1002/ima.22546](https://doi.org/10.1002/ima.22546).
- [25] Vincent Dumoulin and Francesco Visin. *A guide to convolution arithmetic for deep learning*. 2018. arXiv: [1603.07285 \[stat.ML\]](https://arxiv.org/abs/1603.07285). URL: <https://arxiv.org/abs/1603.07285>.
- [26] Umberto Michelucci. *An Introduction to Autoencoders*. 2022. arXiv: [2201.03898 \[cs.LG\]](https://arxiv.org/abs/2201.03898). URL: <https://arxiv.org/abs/2201.03898>.
- [27] Intuitive Tutorial. *Activation Functions in Deep Learning*. Accessed: 2025-04-08. 2023. URL: <https://intuitivetutorial.com/2023/07/23/activation-functions-in-deep-learning/>.
- [28] François Chollet et al. *Keras*. <https://keras.io>. 2015.
- [29] Theo Heimel et al. “QCD or what?” In: *SciPost Physics* 6.3 (Mar. 2019). ISSN: 2542-4653. DOI: [10.21468/scipostphys.6.3.030](https://doi.org/10.21468/scipostphys.6.3.030). URL: <http://dx.doi.org/10.21468/SciPostPhys.6.3.030>.