Contents

Anomalous Jet Detection using Unsupervised Learning

Dhiraj Khanal

May 18, 2023

Abstract

This study presents an analysis of data from the CMS detector at CERN's Large Hadron Collider (LHC), focusing on the use of machine learning techniques for jet identification and anomaly detection. The data are produced by advanced clustering algorithms, such as anti-kt (AK), for organizing particles into jets based on their proximity, momentum, and other properties. The Particle Flow (PF) candidates algorithm was utilized to derive key fields like transverse momentum (p_T), pseudorapidity (η), and azimuthal angle (ϕ) from the 4-momentum vector.

A model trained on QCD patterns was employed for classifying jet types, with Mean Squared Error (MSE) and Kullback-Leibler (KL) divergence metrics used to compare their reconstruction errors. A distinct separation between QCD and W-Jets was observed in the MSE and KL loss distributions. Anomaly detection was carried out using an autoencoder (AE) with the loss function serving as an anomaly score.

The model performance was evaluated with Receiver Operating Characteristic (ROC) curves and Area Under the Curve (AUC) scores, which showed improvement in AUC with increasing epochs, with a final score of 0.87. Anomaly detection efficiency was demonstrated even under conditions of class imbalance and different contamination levels, demonstrating the model's stability and robustness. These findings highlight the potential of machine learning in high-energy physics for efficient data analysis and discovery.

1 Motivation

1.1 The Standard Model

The Standard Model(SM) entails our current understanding of elementary particle physics which is laid on experimental observation and theoretical advancement that relates to particles and their interactions. The SM describes two classes of spin-1/2 called fermions and spin-1 called leptons. These particles carry three fundamental forces: the electromagnetic force, the strong force, and the weak force propagated by spin-1 vector boson. The electromagnetic force is

propagated by photons(γ), the strong force by gluons(g), and the weak force by W and Z bosons ($W^{\pm}, Z$). Quarks carry color charge and can interact via the strong force, the electromagnetic force and the weak force. There are six quarks classified into three generations each of down-type and up-type. Up-type has up u, charm c and top t quarks with charge $+2/3e$, and down-type has down d, strange s, bottom b with charge $-1/3e$. Similarly, leptons are also classified into three generations of charged leptons- electrons e, muon μ, tau τ with charge $-e$, and corresponding uncharged neutrinos ν_e, ν_μ, ν_τ, which interact through the electromagnetic force and the weak force. Each fermion has corresponding anti-particle with opposite charge denoted by a bar on the top, for example $\bar{e}$ for positron. Figure 1 shows the description of these classes of particles along with theorized particle(like graviton).

In 2012, the existence of the Higgs boson was confirmed by experiments conducted at the Large Hadron Collider (LHC) at the European Organization for Nuclear Research (CERN) in Geneva [4], [6]. Until then the existence of Higgs boson was theorized in 1964 but was a missing piece of the SM. The Higgs boson is a scalar boson and other particles acquire mass by interacting with the Higgs field.

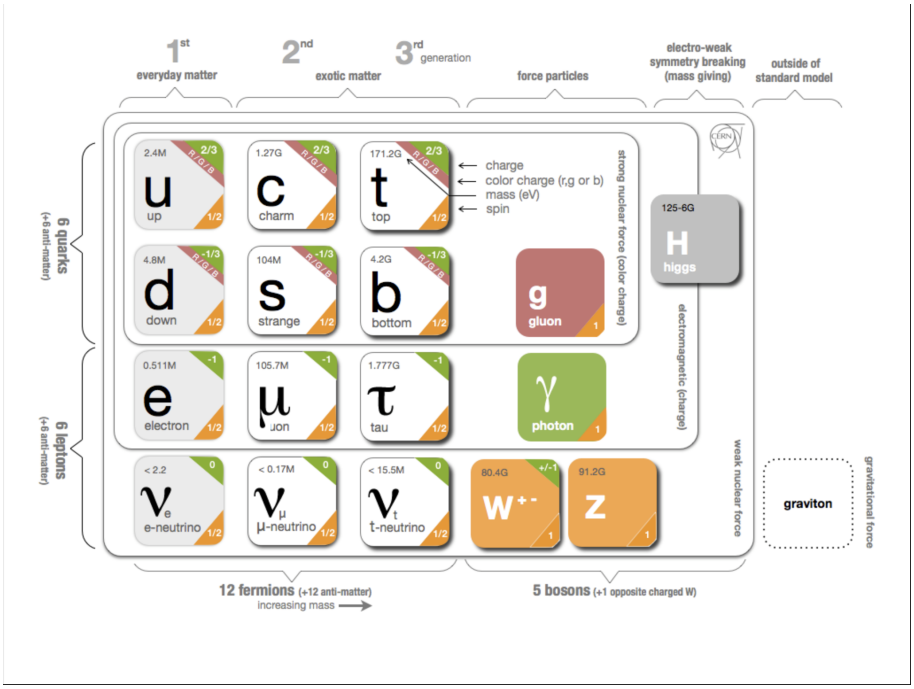


Figure 1: Constituent Particles and the force carriers in the Standard Model. "graviton" is a theorized carrier of gravity which currently does not exist in the SM.[1]

1.2 Beyond the Standard Model and Large Hadron Collider

Beyond the Standard Model (BSM) physics encompasses theoretical and experimental pursuits aimed at expanding the framework of the Standard Model. The discovery of the Higgs boson revealed discrepancies between the SM and observations, highlighting several shortcomings such as the inability to explain dark matter, the baryon asymmetry of the universe, and the smallness of the cosmological constant. These and other phenomena give rise to the hypothesis that physics beyond the Standard Model must exist.

The LHC serves as a critical instrument for exploring BSM physics. Its high-energy collisions generate new particles, offering a unique platform to investigate the properties and interactions of particles not accounted for by the Standard Model. The ATLAS and CMS experiments at the LHC play a vital role in measuring and devising experimental techniques to confirm or rule out a variety of BSM models.

In Run III, which commenced in July 2022, the LHC has been collecting collision data of considerable statistical significance[2]. Statistically speaking, BSM models can be examined and verified by detecting anomalous distributions, often called signals in the context of physics, amidst the extensive backdrop of collider events explained by Standard Model interactions. New physics would be represented by data that deviates from the Standard Model's expectations. If new physics is present at the scales currently being probed by the LHC, it might be more elusive than anticipated, making it challenging to discover through conventional means and without substantially increasing the number of analyses. In this scenario, model-independent approaches, such as unsupervised machine learning techniques, become invaluable, supplementing traditional, signal-driven search methods.

2 Introduction & Background

At CERN different experiments have different collaboration which is focused on data and techniques specific to the type of Collaboration. This research is a CMS Collaboration.

2.1 CMS detector

The CMS detector is situated at one of the four collision points of the LHC, the largest and most powerful particle accelerator ever built. As a general-purpose detector, CMS is designed to observe any new physics phenomena that the LHC might reveal. The detector acts like a giant, high-speed camera, capturing 3D "photographs" of particle collisions from all directions up to 40 million times per second. By identifying (almost) all the stable particles produced in each collision, measuring their momenta and energies, and piecing together this information, the detector can recreate an "image" of the collision for further

analysis.

The 14,000-tonne CMS detector is shaped like a cylindrical onion, with several concentric layers of components as shown in Figure 2. These components help prepare "photographs" of each collision event by determining the properties of the particles produced in that particular collision. The particles are bent with a powerful solenoid magnet that can produce a magnetic field of around 4 Tesla[3]. Using a silicon tracker the path of any charged particles are identified when it interacts electromagnetically with the silicon. There are two energy measuring detectors with two kinds of calorimeters: the Electromagnetic Calorimeter (ECAL) and the Hadron Calorimeter (HCAL). ECAL is the inner layer of the two and measures the energy of electrons and photons by stopping them completely. Hadrons, which are composite particles made up of quarks and gluons, pass through the ECAL and are detected by HCAL. CMS also detects muons as they do not interact with caloriemeters so they are identified with special sub-detectors[3].

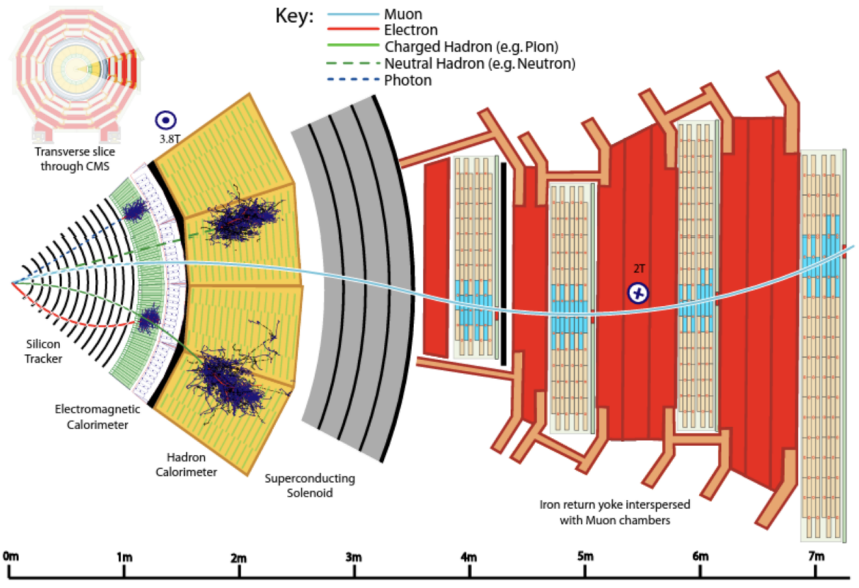


Figure 2: Transverse slice of CMS detectors showing where different class of particles are detected. For example: Muons are detected in the outermost chamber and where as electrons are stopped at the Electromagnetic Caloriemeter.[11]

2.2 *pp* Collision to Jets

In high-energy physics(HEP) experiments like proton-proton (*pp*) collisions are done to analyze the distributions of final-state quarks and gluons. However,

due to the confinement of color charge, quarks and gluons are not observed directly as final-state particles; instead, they manifest as colorless hadrons. These transformation of these quarks and gluons into hadrons is called hadronization. During this process, a collimated spray of particles is produced. A collimated spray refers to a narrow, focused, and directed stream of particles that emerge from the collision point. Collimated spray that are spatially-grouped collections are called jets, which are well-defined, easy to measure and calculate, and closely correspond to the final-state quarks and gluons. They offer valuable information about the initial partons (quarks and gluons) that produced them[10]. However, identifying and labeling jets is challenging due to the subsequent decays and hadronizations that happen in close proximity in the detectors. Clustering algorithms are used to organize the particles into jets based on their proximity, momentum, and other properties, making it easier to organize the jets. One such algorithm is anti-kt (AK) which uses a jet radius parameter R of 0.4 for so-called AK4 jets and a jet radius of 0.8 for AK8, which is shown in Figure 3[5]

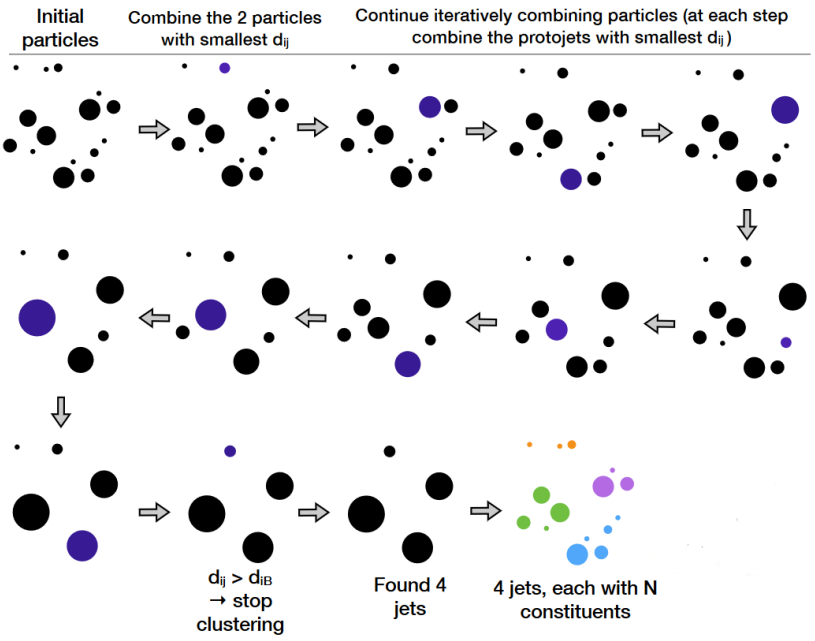


Figure 3: Clustering algorithm called anti-kt (AK). First define distance for every particle d_{ij}. Then iterate: If min$\{d_{ij}\}$ < min$\{d_{ib}\}$; merge particles i and j. If min$\{d_{ij}\}$ >min$\{d_{ib}\}$ jet i is complete [14].

2.3 4-momentum vector to η, ϕ, pT

Particle Flow (PF) candidates algorithm are the result of a anti-kt reconstruction algorithm that combines information from various sub-detectors in the CMS

experiment described in the section 2.1. Figure 3 describes AK algorithm where initial particles are combined to form 4 jets based on the proximity of the particles. Key fields of PF candidates include their transverse momentum (pT), pseudorapidity (η), and azimuthal angle (ϕ).

The transverse momentum (p_T) of a jet is the component of its momentum perpendicular to the beam axis, calculated as $p_T = \sqrt{p_x^2 + p_y^2}$, where p_x and p_y are the momentum components in the x and y directions. The pseudorapidity (η) describes the angular distribution of particles with respect to the beam axis. The azimuthal angle (ϕ) is the angle in the transverse plane, measured from a reference direction (usually the positive x-axis). It ranges from 0 to 2π and describes the orientation of the particle in the x-y plane.

3 Machine Learning

3.1 Introduction to Machine Learning (ML)

ML is an algorithm that learns from the data by an optimization algorithm like Gradient Descent by minimizing empirical loss function, which measures the difference between the predicted output and the actual output (ground truth) in a supervised learning setting. The loss function is minimized iteratively by adjusting the weights of the model using the gradient (partial derivative) of the loss function with respect to the model's parameters. The learning rate, a hyperparameter, determines the step size taken during each update. Broadly, ML can be classified into: Supervised Learning and Unsupervised Learning. In supervised learning, the model is trained on a labeled dataset, where the input data is associated with corresponding target outputs (ground truth). The objective is to learn a mapping from inputs to outputs that generalizes well to unseen data. Supervised learning tasks include regression and classification. In unsupervised learning, the model is trained on an unlabeled dataset, without any target outputs. The goal is to discover underlying patterns or structures in the data, such as clustering, dimensionality reduction, or anomaly detection. In this research, we adopt unsupervised learning to train a model to learn the distribution of QCD(light) jets. If the model is able to learn the QCD distribution well, then non-QCD jets will appear out-of-distribution for the model, which can be considered an anomaly or a signal.

3.2 Overview of AutoEncoders

Autoencoders are unsupervised learning neural networks models which are primarily used for dimensionality reduction and feature learning. They consist of two main components: an encoder and a decoder. The encoder maps the input data to a lower-dimensional representation, while the decoder reconstructs the input data from this lower-dimensional representation. The key idea is to learn an identity function by compressing the representation of the input data by

learning more efficient compressed representation and dimensionality reduction as described in [8].

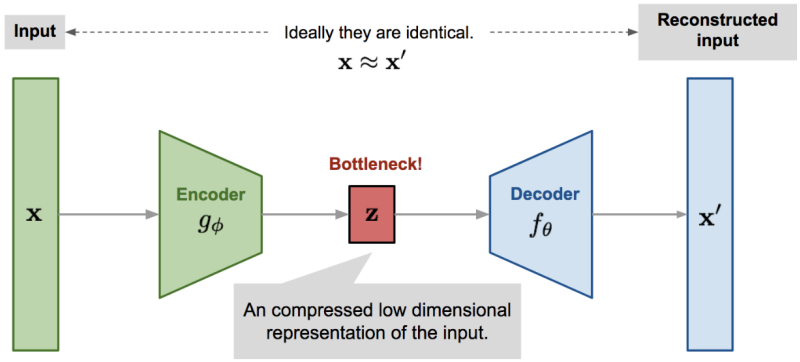


Figure 4: Auto Encoder Network Architecture. Encoder learns the parameters ϕ of the encoding function $g(X)$. Decoder learns the parameters θ of the decoding function $f(Z)$[12].

Given an input dataset $\mathbb{D} = \{x^1, x^2, ..., x^N\}$, where each x^i is data point of d dimensions, $x^i = [x_1^i, x_2^i, ..., x_d^i]$ the encoder learns a function $g_\phi(X)$ that maps the input, x^i to a lower-dimensional representation Z, where $Z = g_\phi(X)$. The decoder then learns a function $f_\theta(Z)$ that attempts to reconstruct the input data from the compressed representation Z. The objective of training an autoencoder is to minimize the reconstruction error, which is typically defined as the mean squared error (MSE) between the original data and its reconstruction:

$$L(X, f_\theta(g_\phi(X))) = \frac{1}{N}\sum_{i}^{N}\|X_i - g_\phi(f_\theta(X_i))\|^2 = \frac{1}{N}\sum_{i}^{N}\|X_i - X_i^{'}\|^2$$

Under this prescription, we have the following algorithm for AutoEncoder.

Algorithm 1 Autoencoder

Require: Dataset $x(1), \ldots, x(N)$
Ensure: encoder f_θ, decoder g_ϕ
1: $\phi, \theta \leftarrow$ Initialize
2: **repeat**
3: $\quad L \leftarrow \sum_{i=1}^{N} \|x(i) - g_\phi(f_\theta(x(i)))\|$ ▷ reconstruction error
4: $\quad \phi, \theta \leftarrow$ Update using Gradient Descent
5: **until** convergence of parameters ϕ, θ

Feature	Description
Dataset	QCD Jets NanoAOD, W+ Jet NanoAOD
Size	QCD: 480,000 events (500 GB), W+: 120,000 events (120 GB)
Event type	$p\bar{p}$ collision
Attributes	η, ϕ, p_t
HT range	500 - 700 GeV
Generation tools	MadGraph, Pythia8
Configuration	TuneCP5 13TeV

4 Methodology

4.1 Dataset Description

The dataset used in this study is supplied by CMS and is derived from NanoAOD(Nano Advanced Optical Disc) files containing Monte Carlo simulations of Jets. NanoAOD files are a highly compressed, ntuple-like format containing per-event information. The main Tree in these files is named "Events," with physics objects features grouped via naming conventions and sharing the same array dimensionality in EDM (Event Data Model) format[13].

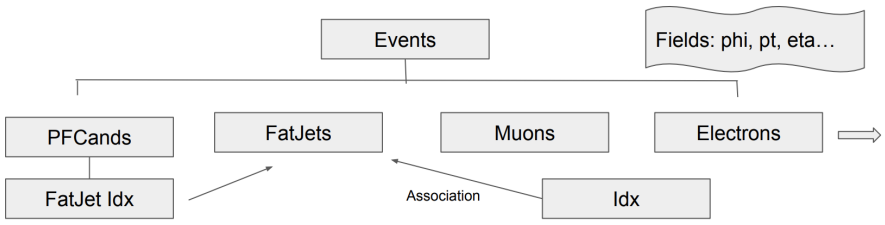


Figure 5: NanoAOD file tree structure. The particle candidates are associated with jets through a unique index. Manual association of the particles with jets is necessary to extract Jet image variables.

The dataset consists of two types of simulated Jet events: QCD Jets NanoAOD with approximately $480,000$ events (500 Gigabytes) and W+ Jet NanoAOD with around $120,000$ events (120 Gigabytes). Each event is a $p\bar{p}$ collision and includes various particle attributes like pseudorapidity η, azimuthal scattering angle ϕ, and transverse momentum p_t, which are in the detector's coordinate system. The sample contains processes with a specific range of HT (Hadron Transverse momentum) from 500 to 700 GeV. The events in this file are generated using the MadGraph (Matrix Element generator) and Pythia8 (parton shower and hadronization) tools, following the TuneCP5 13TeV configuration.

4.2 Dataset Preparation

The Jets are pre-selected with specific criteria: Jet.pt $>$ 200GeV and $-2.0 < \eta < 2.0$. Overlapping Jets with leptons are removed by selecting those without nearby electrons and muons ($\Delta R > 0.4$) for candidates.pt $> 20GeV$. Gen-matching is performed by selecting Jets with overlap $\Delta R < 0.4$ with matched gen, using the coffea tool. A simple Galilean transformation is used to convert the coordinates from the detector's frame to the jet's frame, resulting in $\Delta\eta$ and $\Delta\phi$ values.

4.3 Feature Extraction: Root to η, ϕ, p_t Jet

This process involves converting the four momenta of jet constituents into two-dimensional jet images. After pre-processing the root files and transforming the η and ϕ variables to the jet frame, the jet is properly centered. The η and ϕ axes are fixed within a range of -0.8 to +0.8 with a binning of 0.05. This choice of range and binning ensures that the jet constituents are captured within a suitable region ($\Delta R < 0.8$). The color axis represents the fractional transverse momentum (p_t/p_J), which is calculated as the ratio of the transverse momentum (pt) of each constituent to the total transverse momentum (pJ) of the jet. To normalize the jet images, each image is divided by its total transverse momentum, ensuring that the pixel intensities sum to one. This normalization step allows for a fair comparison between jets with different transverse momenta.

With a granularity of 0.05 x 0.05 per pixel, the resulting jet image contains $(1.6/0.05)^2$ = 32 x 32 = 1024 pixels. This granularity is chosen to provide sufficient resolution for capturing the jet constituents' features while maintaining a manageable image size for the training process. A two-dimensional transverse momentum histogram is created for the jet constituents in the $\Delta\eta$-$\Delta\phi$-plane. This histogram provides a visual representation of the distribution of the jet constituents within the selected region. With this the files are saved in a matrix representation (32 x 32 x 1 x N) in numpy(.npy) format which is easier for analysis. The code for the CNN Auto-encoder is written in Python, using the deep learning library Keras with the TensorFlow backend[7] [9].

4.4 Auto Encoder Network Architecture

The autoencoder model architecture is designed for jet images with size 32x32. The architecture can be divided into two main parts: the encoder and the decoder.

In the encoder part, the input layer takes input images of shape (32, 32, 1). A 2D convolutional layer with 10 filters, kernel size of (4, 4), and padding set to 'same' is applied, followed by a parameterized ReLU activation function with an alpha value initialized by a random normal variable. Another 2D convolutional layer with 5 filters, kernel size of (4, 4), and the same padding and activation settings is applied. Then, an average pooling layer with a pool size of (2, 2) and strides of (2, 2) is applied to reduce the spatial dimensions. Two dropout

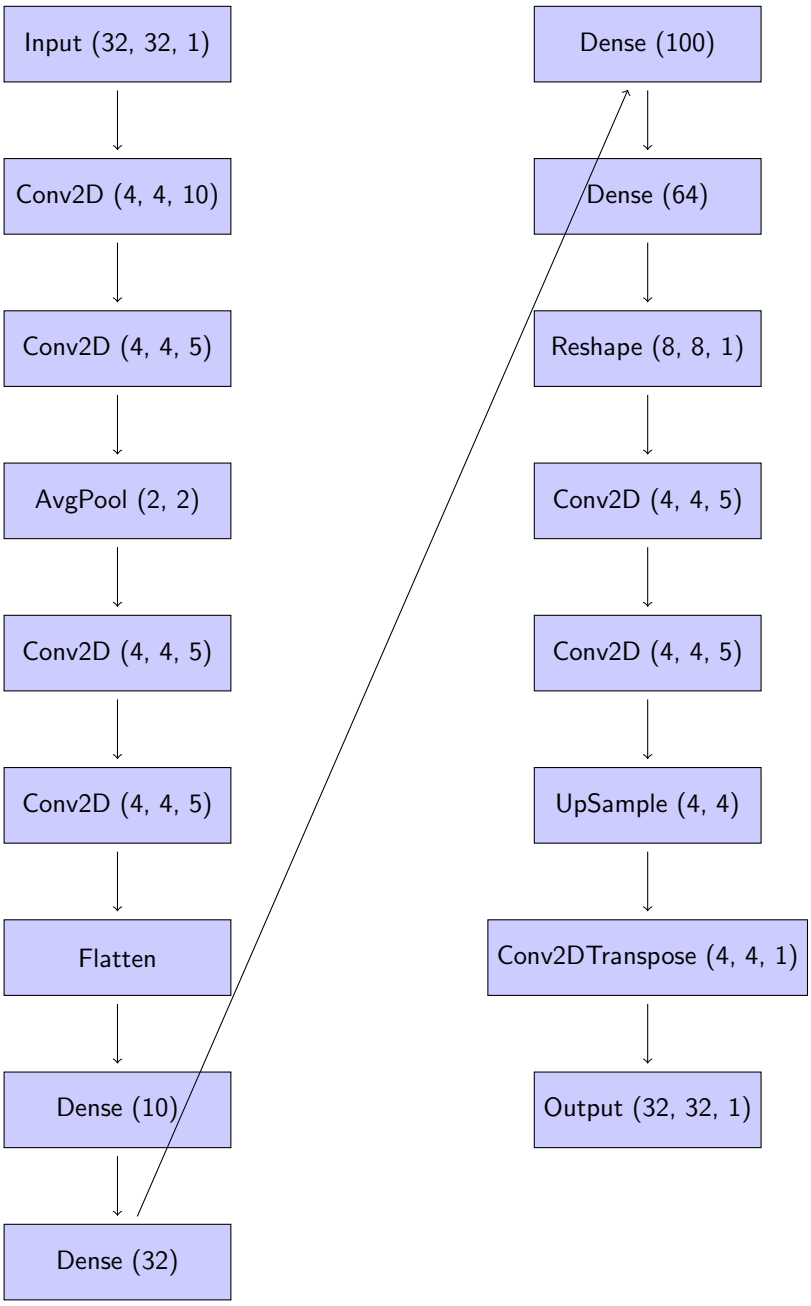
Feature	**Description**
Selection criteria	Jet.pt > 200 GeV, $-2.0 < \eta < 2.0$
Overlap removal	$\Delta R > 0.4$ for candidates.pt > 20 GeV
Gen-matching	$\Delta R < 0.4$ with matched gen
Tools	Coffea
Coordinates	$\Delta\eta$, $\Delta\phi$
Jet image range	$-0.8 \leq \Delta\eta \leq 0.8$, $-0.8 \leq \Delta\phi \leq 0.8$
Binning	0.05
Normalization	Pixel intensities sum to 1
Image size	32 x 32 pixels
Output format	Numpy (.npy)

Table 1: Summary of preprocessing and feature extraction details

layers consisting of convolutional layers with 5 filters, kernel size of (4, 4), and the same padding and activation settings are applied sequentially. The encoder is then flattened and passed through two dense layers with 10 and 32 neurons, respectively, and the same parameterized ReLU activation function.

The decoder part begins with the encoded 32-dimensional representation, followed by two dense layers with 100 and 64 neurons, respectively, and the same parameterized ReLU activation function. The output of the 64-neuron dense layer is reshaped to an 8x8x1 tensor. Two dropout layers consisting of convolutional layers with 5 filters, kernel size of (4, 4), and the same padding and activation settings are applied sequentially. Next, an upsampling layer with a size of (4, 4) is applied to increase the spatial dimensions. Finally, a 2D transposed convolutional layer with 1 filter and a kernel size of (4, 4) with the same padding is applied which results in (32, 32, 1) output shape.

The autoencoder model is defined by connecting the input layer to the last layer of the decoder. The model is compiled with the Adam optimizer and Mean Squared Error (MSE) loss. An early stopping callback is defined to monitor the validation loss and stop the training if it doesn't improve for 20 consecutive epochs.


Input (32, 32, 1)
Conv2D (4, 4, 10)
Conv2D (4, 4, 5)
AvgPool (2, 2)
Conv2D (4, 4, 5)
Conv2D (4, 4, 5)
Flatten
Dense (10)
Dense (32)
Dense (100)
Dense (64)
Reshape (8, 8, 1)
Conv2D (4, 4, 5)
Conv2D (4, 4, 5)
UpSample (4, 4)
Conv2DTranspose (4, 4, 1)
Output (32, 32, 1)

5 Results

5.1 Average Plot

The average plot of 10,000 images from each class (Figure 6) highlights differences between average images of two jet classes: light jets (left) and merged W+ jets (right). The fractional transverse momentum (p_T/p_J) is plotted on a logarithmic scale to effectively represent intensity variations across the jet images. It reveals that both QCD jets and W+ jets have a concentrated intensity in the central pixels, progressively getting less as we move away from the center. The major difference between is that merged W+ jets show a distinct satellite structure. This difference in distribution can be learned by encoder-decoder models for jet classification.

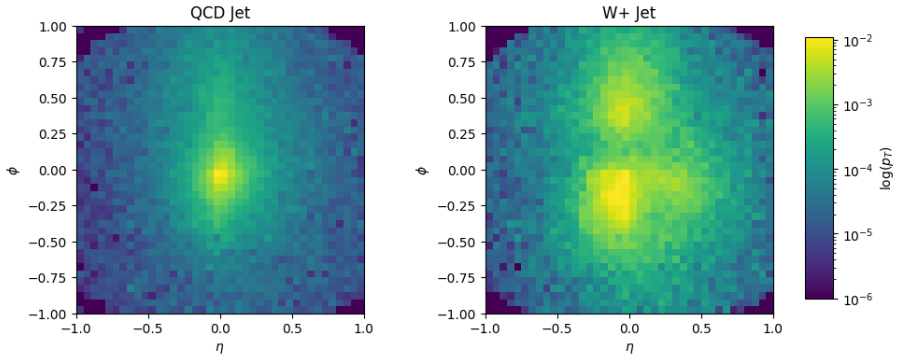


Figure 6: Average of 10,000 jets. Both have energy concentrated at the origin. W+ jet has prong structure but QCD does not which is captured in this average plot.

5.2 Training

During the training process, over $200,000$ light jets were used for training, and $50,000$ light jets were used for validation. The model was trained for around 90 epochs, as the reconstruction loss (defined by the mean squared error) started to converge. The early stopping callback was set with a patience of 20 epochs. To evaluate the model's performance at different epochs, the model was saved every 10 epochs. The convergence of training loss, represented by the mean squared error per epoch, is shown in the appendix.

We first load the matrix representative of the jet image, normalize it, and pass it through the trained encoder (g_ϕ) to obtain the latent representation $g_\phi(X) = Z$. Then, we pass the latent representation through the trained decoder (f_θ) to get the reconstructed image $f_\theta(g_\phi(X)) = f_\theta(Z)$. We then calculate the squared error image as $\|\text{jet} - \text{decoded}\|^2$ and plot it for every 10 epochs. We

compare the squared error of an examplary jet as $\sum \|\text{jet} - \text{decoded}\|^2$ with the reconstruction loss of the model every 10 epochs.

In Figure 7, we observe that after 30 epochs of training, the reconstructed image starts resembling the input jet. The model begins to reconstruct the central pixels, and the squared error is dominated by these central pixels, which generally have higher pixel values as the jet originates radially from the center. Both reconstruction losses and squared error losses are decreasing as the we train the model for more epochs. At epoch 70, the squared error is dominated by the outer pixels with lower intensities. Although the central pixels are reconstructed more accurately, the loss function penalizes higher pixel values more than lower pixel values, causing the model to gradually focus on the brightest pixels. This behavior highlights the trade-offs in the reconstruction process when using auto-encoder.

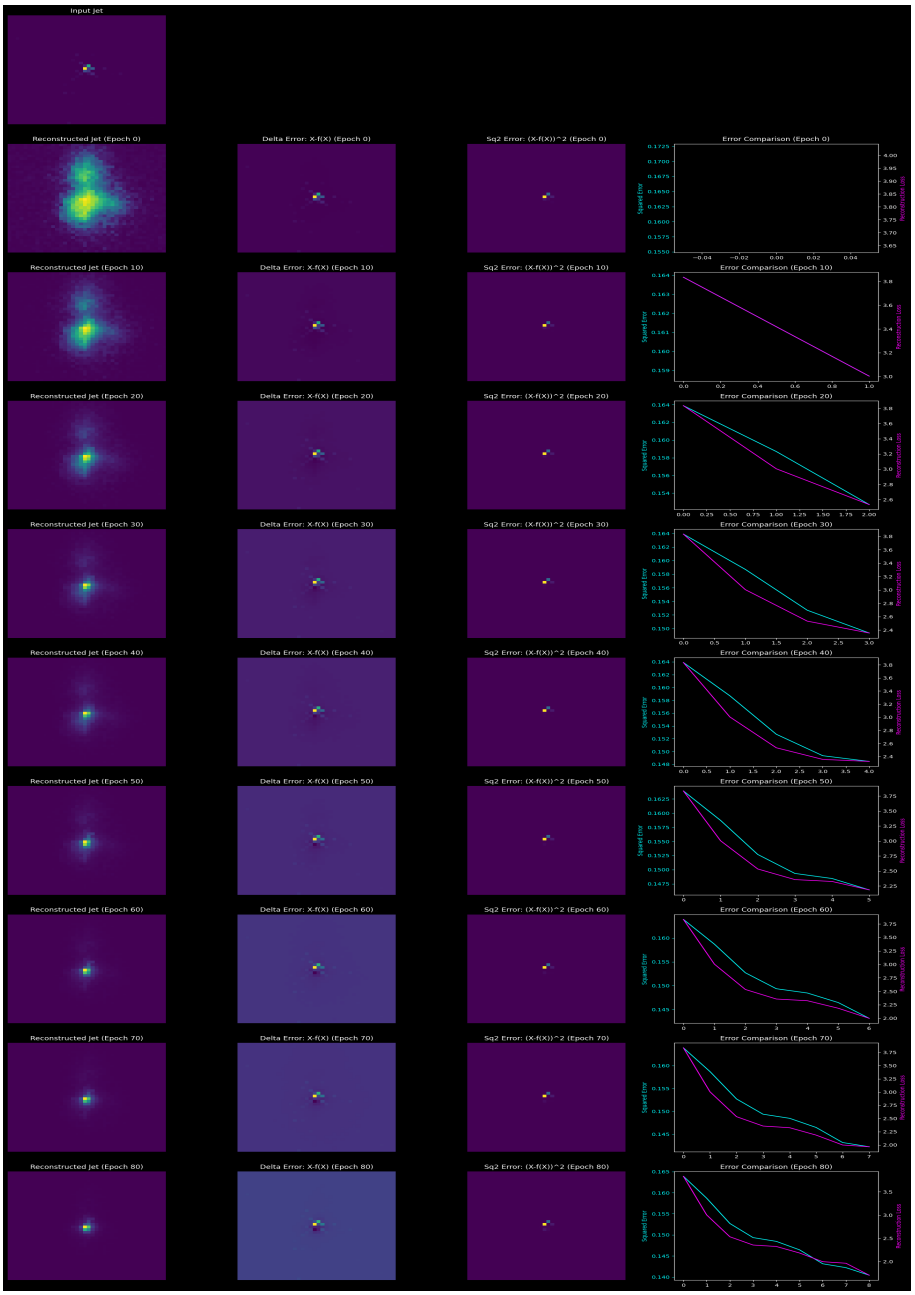


Figure 7: Reconstruction Image of a Light Jet every training epochs Input image is compared with the reconstructed image by comparing at their pixel difference and squared pixel difference.

6 Model Performance

6.1 Distribution under different metrics

To use the model for classifying jet types, we compare their reconstruction errors. Since the model is trained on QCD patterns, non-QCD patterns are expected to have larger Mean Squared Error (MSE) values. Figure 8(top) displays the binned reconstruction losses for QCD (purple) and W-Jet (green) classes. The Kullback-Leibler (KL) divergence metric is used to measure the difference between the QCD probability distribution and a non-QCD reference distribution:

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log \left(\frac{P(x)}{Q(x)} \right).$$

The MSE metric forms distinct Gaussian-like distributions, while the KL divergence metric shows more overlap, indicating the model learns the underlying probability distribution with lower KL loss for most of the QCD jets than for W-Jets. Figure 8(bottom) demonstrates a Kernel Density Estimate (KDE) of the histogram, a non-parametric method estimating the probability density function by placing a kernel at each MSE and KL loss and summing them. This reveals better separation between QCD and W-Jet classes. The KL-divergence metric aligns the peaks of both jet types, with Light Jets having a higher peak than W Jets and W Jets displaying a wider range of higher losses. The model was not trained on the KL metric, but it has somewhat learned to distinguish probability distribution even though it was trained on MSE.

6.2 ROC curve

The scikit-learn Python package was used to create Receiver Operating Characteristic (ROC) curves and for the calculation of Area Under the Curve (AUC) scores. The True Positive Rate (TPR) is calculated as the ratio of correctly classified Light Jets to the total number of Light Jets, while the False Positive Rate (FPR) is calculated as the ratio of incorrectly classified W+ Jets to the total number of W+ Jets. For a random classifier, the AUC is 0.5.

ROC curves and AUC scores help understand the trade-off between TPR and FPR for the classifier. Figure 10 displays ROC curves for Epoch 20, 40, 60, and 80. As the number of training epochs increases, the AUC score improves, indicating better classifier performance. The model ultimately achieves an AUC score of 0.87 at the end of training.

6.3 Anomaly Detection

As the value of the loss function is used as the discriminator between signal and background, for anomaly detection, a trained AE can be also used as an anomaly tagger with its loss function as its anomaly score. An event is tagged as anomaly if the value of the loss function is larger than a given threshold. A naive algorithm for is the following.

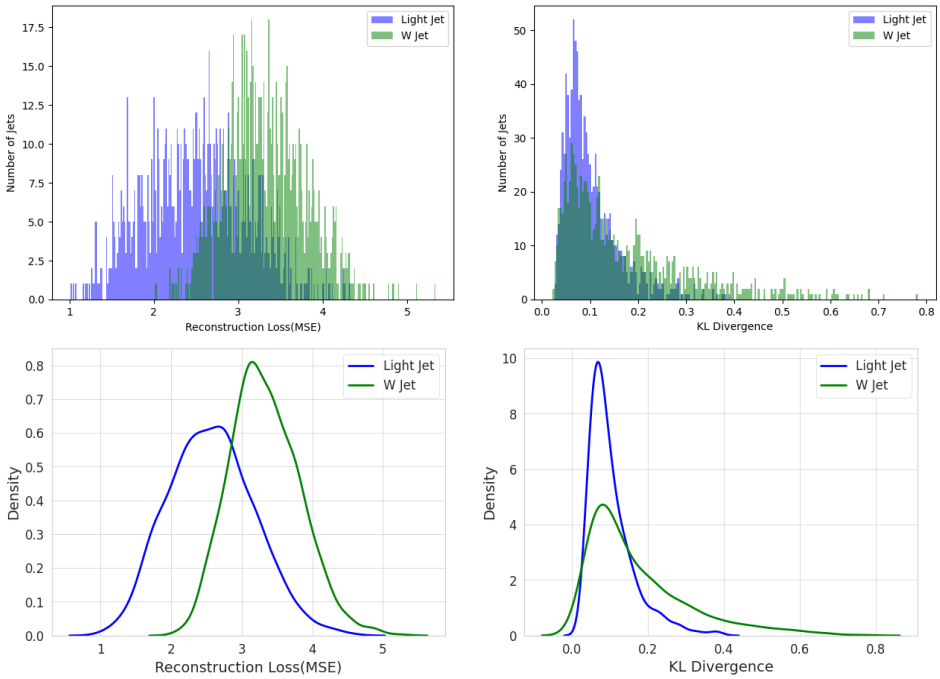


Figure 8: Top: Reconstruction error metrics for jet classification. Left: Distinct Gaussian-like MSE distributions for QCD jets (purple) and W-Jets (green). Right: KL divergence shows higher overlap but lower KL loss for QCD jets compared to W-Jets. Bottom: Extension of analysis in Figure 8 by incorporating a Kernel Density Estimate (KDE) to approximate the probability density functions of the MSE and KL losses.

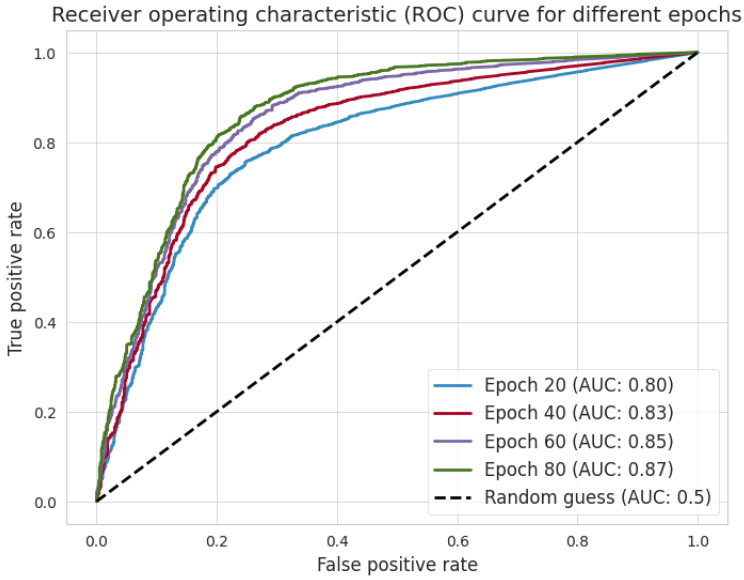


Figure 9: ROC curves for different training epochs (20, 40, 60, and 80) illustrating the classifier's performance. The AUC score improves with increasing epochs. The model achieves an AUC score of 0.87 at the end of training.

Require: Normal Dataset X, Anomalous Dataset Y, threshold λ
$\phi, \theta \leftarrow$ train a AE using X
repeat
error(i) = loss$(X(i), g_\phi(f_\theta(X(i))))$
if error(i) $> \lambda$ then X(i) is anomaly
if not X(i) is not an anomaly
until Iterate through all Dataset

An anomaly tagger should only properly reconstruct jets it has been trained on to be truly model independent, hence if we plot the distribution of MSE error with respect to two classes of jets, we see the pattern shown in Figure 10 plotted for 1000 QCD (blue) and 1000 W-Jets (green). It is clear that W-Jets have higher reconstruction error than Light Jets as it was trained on QCD patterns. Using the naive algorithm above, we can find a threshold value that wells in tagging anomaly from the background. One such threshold line(red) is plotted in the same figure.

Moving towards a more realistic scenario of unsupervised classification, we need to study scenarios with a class imbalance. For this we can consider the dataset with different contamination levels (w/Q): 1.00, 0.25, 0.1, and 0.05.

In Figure 11, the inverse rejection plotted against signal efficiency curve for different contamination levels. For $w/Q = 1.0$, which means there were 1:1 QCD and W-Jets in the sample, the model did the best with the AOC score of

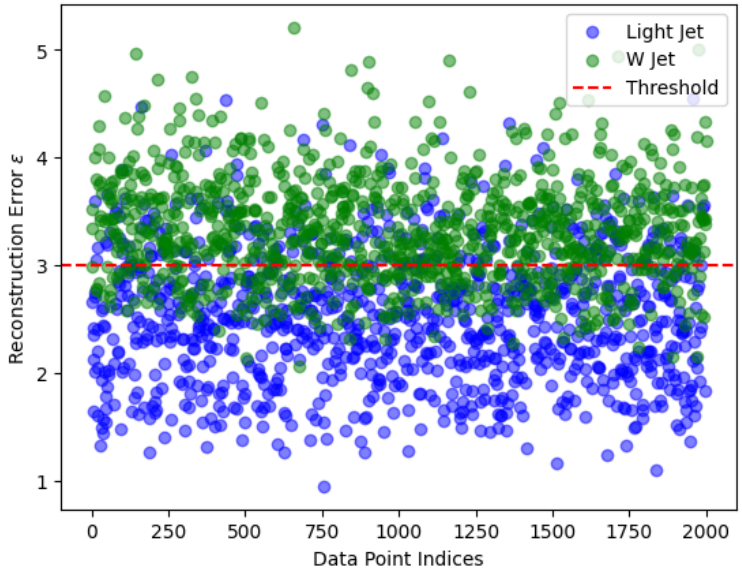


Figure 10: MSE reconstruction errors for 1000 QCD (blue) and 1000 W-Jets (green) using an Autoencoder (AE) as an anomaly tagger. The red line represents the optimal threshold for distinguishing anomalies from background events.

0.84. With decreasing $w/Q = 0.25, 0.1, 0.05$, the AOC scores decrease from 0.84 to 0.83 to 0.82. This slow fluctuations in AOC scores makes it a stable anomaly tagger.

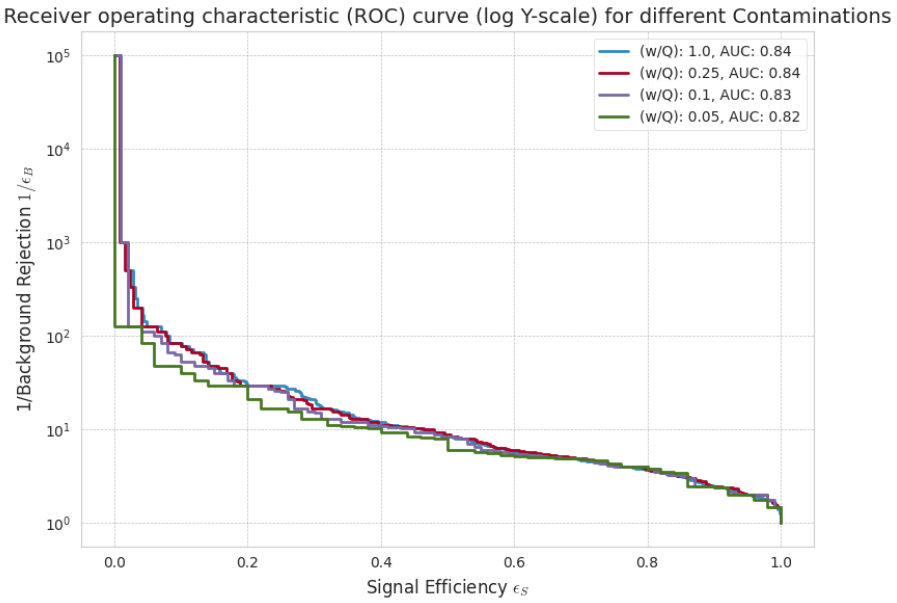


Figure 11: ROC curves for AE trained on samples of QCD Jets contaminated with a fraction of W-Jets.

7 Appendix

The graph visualizes the evolution of the Mean Squared Error (MSE) loss on the validation dataset of light jets for the AutoEncoder architecture, as outlined in the research paper. The x-axis denotes the number of epochs, while the y-axis signifies the MSE loss. It's observed that the MSE loss experiences a significant decline over the epochs, indicating the model's learning progress. The curve of the MSE loss is somewhat steep in the initial epochs, implying a rapid learning phase, but gradually the descent becomes less pronounced, demonstrating that the model is fine-tuning its parameters. Notably, at the 92nd epoch, the curve tends to flatten, signifying that the model has largely converged and additional training does not significantly improve the model's predictive accuracy on the validation dataset. It suggests that the AutoEncoder has succeeded in learning relevant representations from the light jet data and achieved a stable state at the 92nd epoch. The chart presents the Kernel Density Estimates (KDE) of the Mean and Median MSE versus density, which is utilized to determine the anomaly tagging threshold for jet classification. The plot clearly distinguishes between the W jets (indicated by the green label) and the light jets (blue label) based on their respective reconstruction errors (MSE metrics). The x-axis represents the MSE, while the y-axis shows the corresponding density.

For both W jets and light jets, we observe distinct peaks in the KDE, signifying the most frequent MSE values or, more specifically, the Mean and Median.

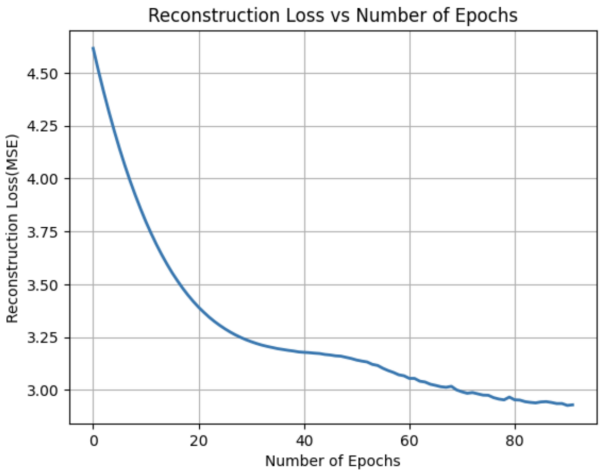


Figure 12: MSE loss on Validation dataset of light jets. For the AE architecture described in the text it converges at the 92nd epoch.

The green density plot corresponding to the W jets is offset to higher MSE values relative to the blue density plot for light jets. This difference in MSE distributions clearly indicates a separation between the two classes.

The delineation between the W jets and light jets in this graph highlights the effective discrimination capability of the AutoEncoder's reconstruction error in jet classification. It implies that the Mean and Median of MSE can be used as effective thresholds for anomaly tagging in this particular context.

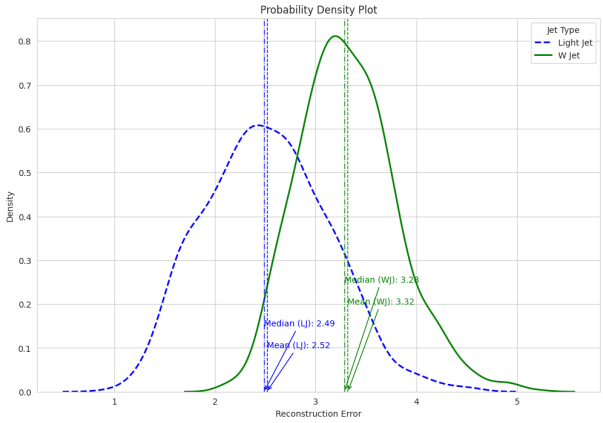


Figure 13: Mean and Medians of MSE vs density using Kernal Density Estimate. This is used to calculate the threshold for anomaly tagging.

References

[1] Cern bulletin 2012/35, 2012. Visited on 2023-04-10.

[2] Run 3, 2022. Visited on 2023-03-10.

[3] Cms detector, 2023. Accessed: 2023-04-10.

[4] Georges Aad et al. Observation of a new particle in the search for the standard model higgs boson with the atlas detector at the lhc. *Physics Letters B*, 716(1):1–29, 2012.

[5] Matteo Cacciari et al. The anti-kt jet clustering algorithm. *Journal of High Energy Physics*, 2008(04):063, 2008. Published: April 16, 2008.

[6] Serguei Chatrchyan et al. Observation of a new boson at a mass of 125 gev with the cms experiment at the lhc. *Physics Letters B*, 716(1):30–61, 2012.

[7] François Chollet et al. Keras. https://keras.io, 2015.

[8] G. E. Hinton. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[9] Google Research and Google Brain Team. Tensorflow: Large-scale machine learning on heterogeneous systems. https://www.tensorflow.org, 2015.

[10] Michael H. Seymour. Jets in qcd. Technical report, CERN, Division TH, CH-1211 Geneva 23, Switzerland, June 1995. CERN-TH/95-176, hep-ph/9506421.

[11] Albert M Sirunyan, CMS Collaboration, et al. Particle-flow reconstruction and global event description with the cms detector. *Journal of Instrumentation*, 12(10):P10003, 2017.

[12] Lilian Weng. Variational autoencoders, 2018. Accessed: 2023-03-02.

[13] CMS Public Wiki. Nanoaod in cms, 2023. Last updated: 2023-01-30; Edited by: Sebastien Wertz; Accessed: 2023-03-14.

[14] CMS Open Data Workshop. Jet and met in cms open data, 2021. Accessed: 2023-04-10.