# Fully Convolutional Network for Jet Reconstruction

A thesis presented for the degree of
Master of Science in Computer Science

Figure 1: Photo by Josef Kristofoletti [12].

**Algelly Malik**

Supervisor: **Prof. Anna Sfyrla** & **Prof. Alexandros Kalousis**
Co-Supervisor: **M.Sc. Leon Bozianu**

UNIVERSITÉ DE GENÈVE

ATLAS EXPERIMENT

CERN

Department of Computer Science
University of Geneva
May 2024

**Abstract**

The Large Hadron Collider (LHC) at CERN generates enormous amounts of data through proton collisions, necessitating efficient data processing in the ATLAS experiment. With the High-Luminosity LHC (HL-LHC), handling the increased data volume becomes critical. This thesis explores using Fully Convolutional Networks (FCNs) for jet reconstruction to enhance the High-Level Trigger (HLT) system by eliminating the computationally intensive Topo-Cluster creation step, aiming to reconstruct jets directly from calorimeter data. FCNs are suited for processing spatial data, preserving its structure throughout the network. Various FCN models were developed, trained on synthetic datasets simulating ATLAS conditions, to directly reconstruct jets and predict their transverse momentum ($p_T$). The models were evaluated using metrics like Intersection over Union (IoU) for position accuracy and $p_T$ error rate. Results show that FCNs can effectively reconstruct jets, with the best model achieving a 96.3% match rate for target jets above 40 GeV. Increasing model complexity generally improves performance. FCNs could be a alternative to the current jet reconstruction process, offering faster and accurate event selection at the HL-LHC.

**Keywords:** Fully Convolutional Networks, Jet Reconstruction, High-Luminosity LHC, ATLAS Experiment, High-Level Trigger System, Particle Physics, Data Processing, Topo-clustering, Intersection over Union (IoU), $p_T$ Error Rate

**Acknowledgements**

*I would like to express my deepest gratitude to my supervisor, Prof. Anna Sfyrla, and M.Sc. Leon Bozianu, for their unwavering guidance, support, and encouragement during our weekly meetings. Their expertise and insights have been invaluable in shaping this project. I am also profoundly grateful to Prof. Alexandros Kalousis for his valuable feedback, suggestions, and expertise in the field of machine learning. His knowledge has been instrumental in the development of this thesis.*

*I want to thank my family and my beloved partner for their endless support, understanding, and encouragement throughout this journey.*

# Contents

# 1 Introduction

No introduction is needed for the world's most powerful particle accelerator: the Large Hadron Collider (LHC) at the European Organization for Nuclear Research (CERN). This exceptional research instrument accelerates protons to speeds close to that of light and collides them, producing a wide variety of subatomic particles.

One of the major experiments conducted at the LHC is the ATLAS experiment, which uses a detector of the same name to detect and measure particles from proton collisions, the famous particle jets. With a beam crossing event every 25 ns and collisions at an average rate of 40 MHz, the ATLAS detector generates a massive amount of data to process. It's 60 terabytes of data generated every second. The challenge is clear: how to efficiently manage and analyze this massive data flow to extract relevant information and advance research in particle physics?

This is the role of the ATLAS detector's trigger system, which is designed to filter and select the most interesting events to record for further analysis. The trigger system operates in two stages: the first-level trigger, called the Level-1 Trigger, which uses hardware to quickly analyze data and decide whether to retain an event, and the second-level trigger, called the High-Level Trigger (HLT), which uses software to perform a more detailed analysis of the events selected by the Level-1 Trigger.

The HLT uses a process in two steps: the creation of Topo-Clusters, which group energy deposits in the detector into clusters, and the reconstruction of jets from these clusters with an algorithm called the Anti-$k_t$ algorithm. This process is computationally intensive and requires significant resources to handle the large volume of data generated by the ATLAS detector. Actually, 55,000 CPU cores are used by this second-level trigger to process the data.

Moreover, with the arrival of the High-Luminosity LHC (HL-LHC) [4] planned for 2029, the LHC will see its luminosity increased by a factor of 10 compared to its original design. This increase means a multiplication of collisions and thus of jets to analyze, which poses a considerable challenge for the current trigger systems. The HLT will need to be optimized to handle this increased data flow efficiently while maintaining high accuracy in event selection.

It is in this context that we propose to examine the performances of Fully Convolutional Networks (FCN) for particle jet reconstruction. The goal is to evaluate the potential of FCNs to reconstruct jets without the step of Topo-Cluster creation, which could significantly speed up the process and increase the accuracy of event selection.

FCNs are models of neural networks designed to efficiently process spatial data while preserving the spatial structure of the data throughout the network. Because of the simplicity of their architecture, FCNs are particularly well suited for high-performance computing resources such as Field-Programmable Gate Arrays (FPGAs), which could be an interesting solution to optimize the processing of the vast amount of data generated by the ATLAS detector.

In this report, we will first present the ATLAS detector and its trigger system, then introduce the concept of Fully Convolutional Networks and explain why they are particularly well suited for jet reconstruction. We will detail the architecture of the FCN models we have developed, the training strategies we have used, and the loss functions we have tested. We will also present the data we have used, the preprocessing and postprocessing we have applied, and the results of our experiments. Finally, we will discuss the potential of FCNs to enhance the performance of the ATLAS trigger system at the HL-LHC.

# 2 ATLAS

The ATLAS detector at CERN is an impressive structure, cylindrical in shape, extending about 25 meters in length and standing 44 meters tall, with a weight of approximately 7000 tons [2]. It comprises several sub-detectors, each designed to detect and measure different particle types including electrons, photons, muons, and hadrons.

## 2.1 Calorimeter Overview

Central to our discussion is the Calorimeter System [1] within the ATLAS detector, divided into the Liquid Argon (LAr) Calorimeter and the Tile Hadronic Calorimeter. These systems are pivotal in measuring the energies of various particles produced during proton collisions.

### 2.1.1 Liquid Argon Calorimeter

Surrounding the Inner Detector, the LAr Calorimeter measures the energy of electrons, photons, and hadrons. It utilizes layers of dense metals like tungsten, copper, or lead to absorb and transform incoming high-energy particles into showers of lower-energy particles. These secondary particles ionize the liquid argon placed between the metal layers, generating electric currents that are subsequently measured to determine the energy of the impacting particles.

This component is exquisitely designed to capture electrons and photons through an accordion-shaped structure featuring a honeycomb pattern, which ensures comprehensive detection coverage. The entire system is maintained at a chilling -184°C to keep the argon in liquid form, with specialized vacuum-sealed cables transmitting the electronic signals to the analysis electronics.

### 2.1.2 Tile Hadronic Calorimeter

Encasing the LAr Calorimeter, the Tile Hadronic Calorimeter is tasked with assessing the energy of hadronic particles. Constructed from layers of steel interspersed with plastic scintillating tiles, it captures particles that the LAr Calorimeter does not fully absorb. When hadronic particles strike the steel, they initiate a particle shower, causing the scintillators to emit photons that are transformed into electric signals, proportional to the energy of the original particles. Comprising roughly 420,000 scintillator tiles, this is the heaviest component of the ATLAS experiment, weighing nearly 2900 tonnes.

## 2.2 Trigger System

As previously mentioned, the ATLAS detector produces a vast amount of data, with 60 terabytes generated every second. To manage this data flow, the detector is equipped with a Trigger System [3] that filters and selects the most interesting events from a physics perspective.

The trigger system operates in two distinct phases:

- Level-1 Trigger: This first-level trigger is hardware-based and rapidly analyzes data from the calorimeters and the muon spectrometer. Within 2.5 microseconds, it decides whether an event should be retained for further analysis. If an event is selected, it is then passed to the second-level trigger.
- High-Level Trigger (HLT): The second-level trigger is software-based and performs a more detailed analysis of the events selected by the Level-1 Trigger. Operating from a network of 55,000 CPU cores, this trigger employs complex algorithms composed of two main steps: the creation of Topo-Clusters and the reconstruction of jets. The Topo-Clusters group energy deposits in the detector into clusters, while the jet reconstruction algorithm uses the Anti-$k_t$ algorithm to reconstruct jets from these clusters. The HLT processes approximately 100,000 events per second, ultimately selecting about 1,000 events per second.

Thanks to these two levels of triggering, the data flow is significantly reduced, achieving a selection rate of one event in every 40,000. This reduction allows for the efficient storage and analysis of the most relevant events. However, with the anticipated increase in collision rates at the HL-LHC, the second-level trigger will need to be optimized to handle a larger volume of events efficiently while maintaining high accuracy in event selection.

Currently, with the actual algorithms used for jet reconstruction, the only solution to increase the processing capacity is to enhance the number of CPU cores, which would significantly increase both the costs and complexity of the system. Therefore, there is a pressing need to rethink and optimize the jet reconstruction process to ensure high efficiency and accuracy in event selection.

# 3 Fully Convolutional Network

In this section, we discuss using FCNs to improve jet reconstruction within the ATLAS detector. We start by explaining why FCNs are well-suited for handling spatial data, a critical component in jet reconstruction. Next, we outline the architecture of our FCN models, emphasizing how their design helps simplify the data processing. We then describe the training methods and loss functions we use to fine-tune the performance of our models. Finally, we examine the potential advantages of using FCNs, such as improving the second-level trigger system, and discuss the need for thorough testing to confirm these benefits.

## 3.1 Advantages of Fully Convolutional Networks

Fully Convolutional Networks (FCNs) offer several advantages, especially for jet reconstruction tasks. We highlight four key benefits that make FCNs particularly well-suited for this application.

Firstly, FCNs have fewer parameters than traditional deep neural networks because their parameters are not directly tied to the input size. For example, a convolutional layer with a 3x3 kernel, 64 input channels, and 64 output channels has $3 \times 3 \times 64 \times 64 = 36,864$ parameters, regardless of the input size. In contrast, a fully connected layer for an input of size 128x128 with 64 neurons would require $128 \times 128 \times 64 = 1,048,576$ parameters, showing how FCNs manage parameters more efficiently for large inputs. Obviously, this reduction in the number of parameters has a direct impact on the capacity of the model. A smaller model have less capacity to learn complex patterns. However, in the context of jet reconstruction, the patterns to learn are not as complex as in other tasks like image recognition because the data is more structured and the patterns are more regular.

The second advantage is that FCNs are particularly well-suited for processing spatial data. The convolutional layers in FCNs are designed to capture spatial features and preserve the spatial structure of the data. This can be particularly useful for jet reconstruction tasks, where the spatial distribution of energy deposits in the detector is crucial for accurate jet reconstruction. Moreover, FCNs have proven efficient in image segmentation tasks [11], which are similar to jet reconstruction as both involve spatial data.

Additionally, FCNs are ideal for parallel processing. Their convolutional layers can be processed simultaneously, making them compatible with high-performance computing resources like GPUs or FPGAs. However, it's important to note that while FCNs may be compatible with FPGAs, their inputs might not be. For instance, a

large image can be too big to fit into a single FPGA buffer, which means the entire image must be read in before processing can begin, leading to an increase in the number of clock cycles required. Despite this, the parallel processing capability of FCNs allows for rapid and scalable data handling, which is crucial for managing the high data rates expected at the High-Luminosity Large Hadron Collider (HL-LHC).

Finally, FCNs are particularly effective for FPGA implementation. The simpler and more compact architecture of FCNs makes them well-suited for FPGA implementation. By using libraries like HLS4ML [6], FCN architectures can be quickly transformed into FPGA-compatible code, significantly reducing both development time and effort.

## 3.2 Architecture

Our FCN models exclusively use convolutional layers to preserve the intrinsic spatial properties of the data. The models vary in complexity from 500 to nearly 200,000 parameters, with changes in the number of layers and channels designed to explore the impact of network depth on performance.

To accommodate the limited memory of FPGAs, some models are designed with fewer parameters (between 500 and 3,000), essential for effective FPGA implementation. Across all models, we use the LeakyReLU activation function to handle non-linearities efficiently, avoiding the risk of dead neurons common with the standard ReLU function in deep networks.

All our convolutional layers use a 3x3 kernel size with padding of 1 and a stride of 1. We mix pooling layers and transposed convolutions to condense and expand data, aiming to enhance the network's ability to interpret spatial features and reduce noise. The pooling layers use a 2x2 kernel size and a stride of 2, while the transposed convolutional layers use a 2x2 kernel size and a stride of 2 to restore the original input dimensions.

All convolutional and transposed convolutional layers are followed by batch normalization to stabilize and accelerate training [7]. The final layer of each model is a single convolutional layer outputting transverse momentum ($p_T$) values for jet reconstruction, allowing direct comparison with actual jet data. Residual connections are strategically placed in models to bypass pooling layers and transposed convolutional layers, maintaining spatial information across these layers [9].

We now present the architecture of our FCN models, starting with the simplest model and gradually increasing complexity to explore the impact of network depth on performance.

### 3.2.1 Model 1: Smallest Model

Model 1, our baseline configuration, comprises exactly 537 parameters. Figure 2 illustrates the architecture of this model, which consists of 6 convolutional layers, going from 2 channels to 4 channels. The pooling layer is a max pooling layer, reducing the spatial dimensions by half.

In order to restore the original input dimensions, a transposed convolutional layer is included. The model also features a residual connection that goes around the pooling layer, linking the output of the first layer to the input of the fifth layer.

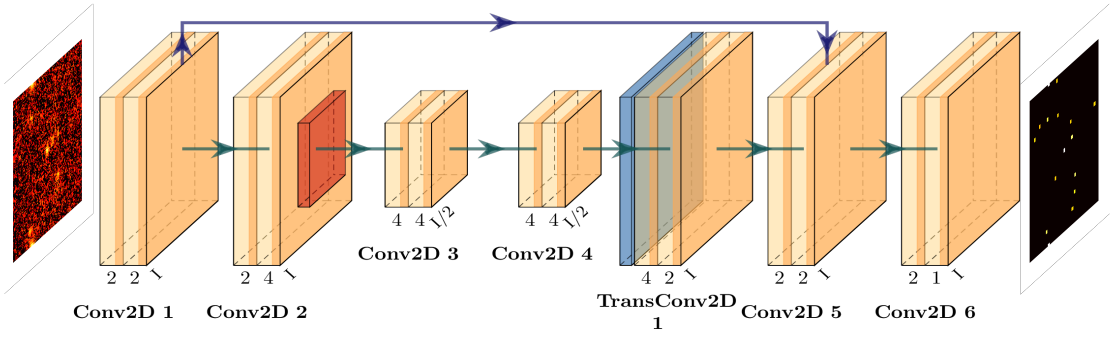



Figure 2: Model 1: Smallest Model

This model serves as a baseline for comparison with more complex models, to evaluate the impact of network depth on performance. Its simplicity makes it particularly suitable for initial testing and insights into the effectiveness of FCNs for jet reconstruction.

### 3.2.2 Model 2: Small Model

Model 2, labeled as the small model, incorporates 1053 parameters. Figure 3 illustrates the architecture of this model.

It consists of 7 convolutional layers, each with 4 channels. The model use pooling to the architecture with the first one being an average pooling layer and the second a max pooling layer.

To counteract the dimension reduction from pooling, two transposed convolutional layers are included. The model also includes two residual connections that go around the part of the network that includes reduction of spatial dimensions.

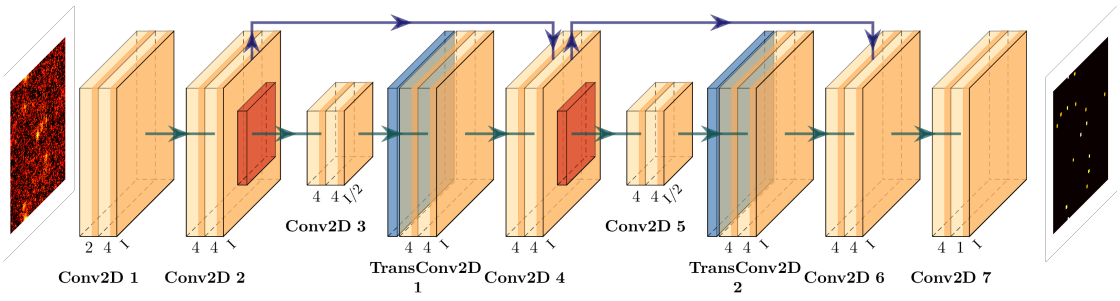



Figure 3: Model 2: Small Model

This model is designed to explore the impact of additional layers and pooling on jet reconstruction performance, providing insights into the optimal network configuration for efficient event selection.

### 3.2.3 Model 3: Medium Model

Model 3, our medium model, is designed with 1,793 parameters. The architecture of this model is depicted in Figure 4.

It comprises 7 convolutional layers, varying from 4 to 8 channels. The model includes three pooling layers, two average and one max pooling. To restore the original input dimensions, three transposed convolutional layers are included.

The model features only one residual connection, linking the output of the first layer to the input of the seventh layer.

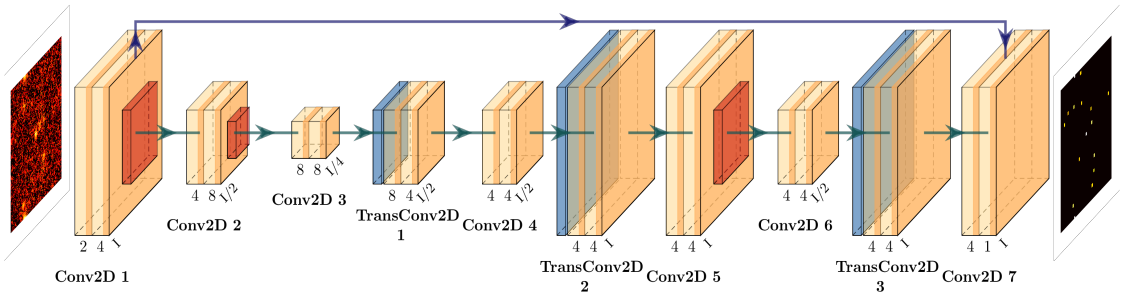



Figure 4: Model 3: Medium Model

This model is developed to continue exploring the impact of bigger models on the task of jet reconstruction. The additional layers and channels aim to enhance the network's ability to capture complex spatial features.

### 3.2.4 Model 4 and 5: Large Models

Models 4 and 5, while sharing the same structural architecture, differ in the configuration of their convolutional layers: Model 4 is designed with 4 channels per layer, resulting in 3,039 parameters, whereas Model 5 doubles the channel count to 8 per layer, totaling 11,331 parameters. The architecture of Models 4 is depicted in Figure 5, with Model 5 following a similar structure.

Each model is structured with 20 convolutional layers. The layering includes three layers of pooling (two average and one max) to manage spatial dimensions effectively. Additionally, three transposed convolutional layers counteract the dimension reduction from pooling.

There are two strategically placed residual connections: the first extends from the output of the second layer to the input of the eleventh layer, and the second from the twelfth layer to the sixteenth layer. As before, these connections are designed to maintain spatial information across the network, optimizing feature transfer and integrity.

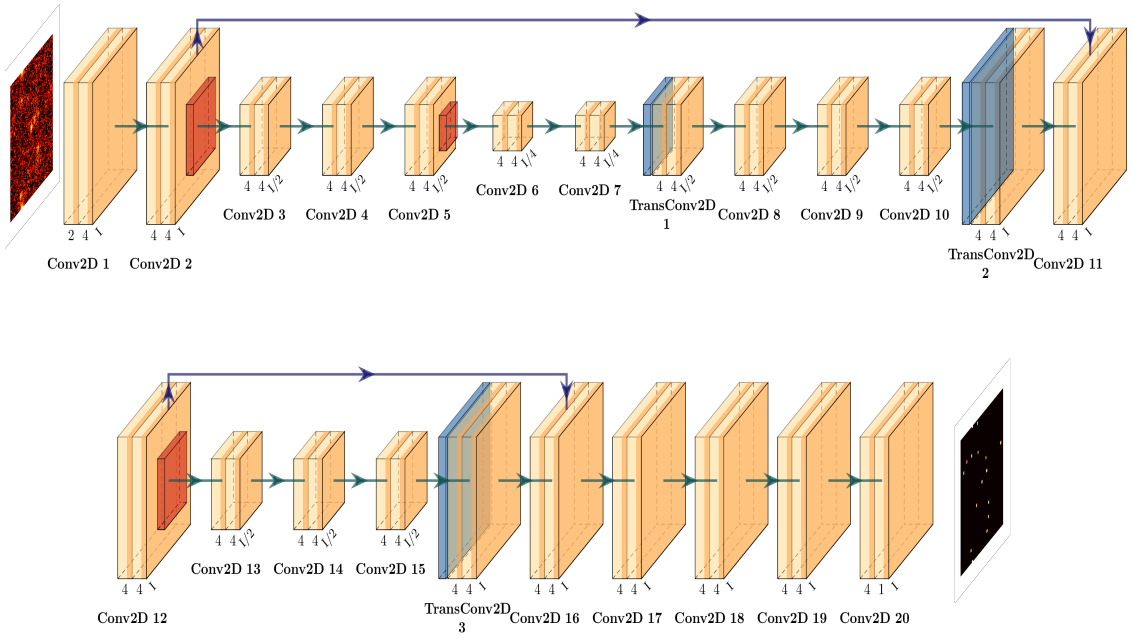



Figure 5: Model 4: Large Model

### 3.2.5 Model 6: Biggest Model

The biggest model, Model 6, is the most complex configuration, featuring 185,347 parameters. The architecture of this model is illustrated in Figure 6. It comprises 20 convolutional layers, with each layer having 32 channels. The model includes four pooling layers (two average and two max pooling) to manage spatial dimensions effectively. To restore the original input dimensions, four transposed convolutional layers are included. The model also features two residual connections, going around the two parts of the network that include reduction of spatial dimensions.

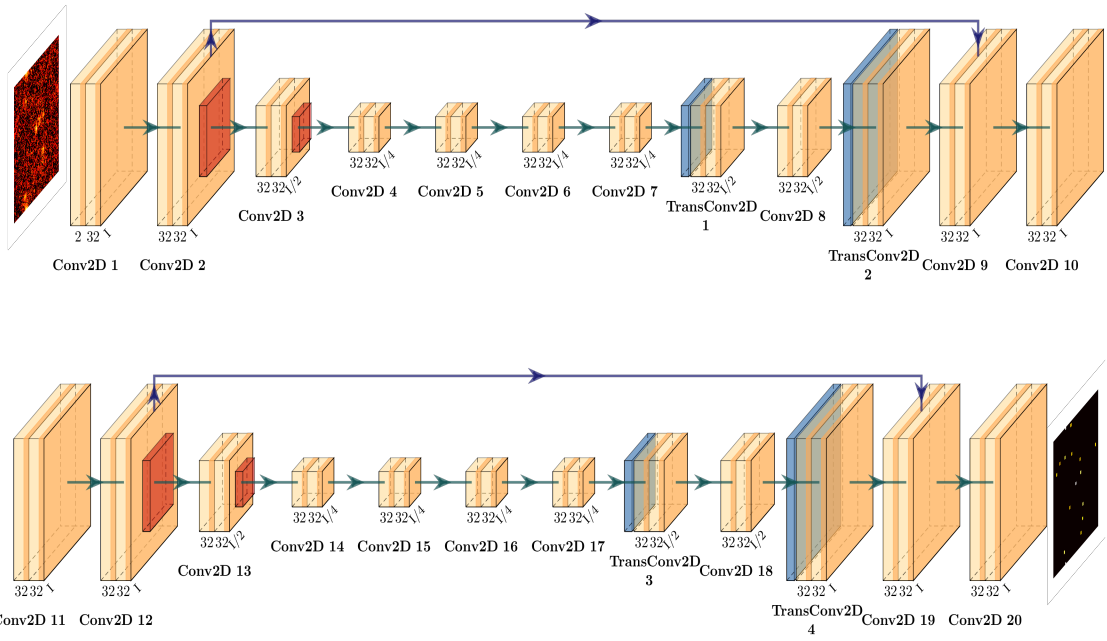



Figure 6: Model 6: Biggest Model

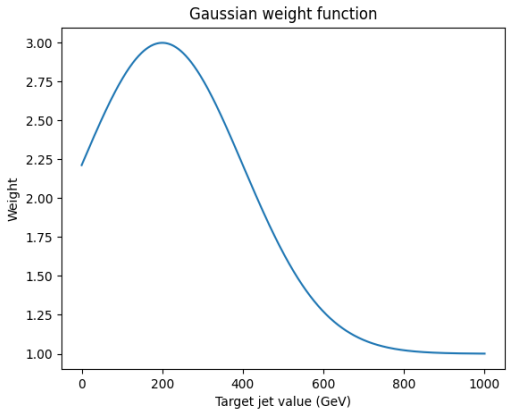
We decided to stop at this model size both for simplicity's sake and in an exploratory spirit. Given that the goal is the rapid reconstruction of jets, we aimed to limit the complexity of the model to ensure efficient implementation on FPGAs. However, it is important to note that this model is already too large for most of the available FPGAs, presenting a significant challenge in terms of practical deployment.

## 3.3 Loss Functions

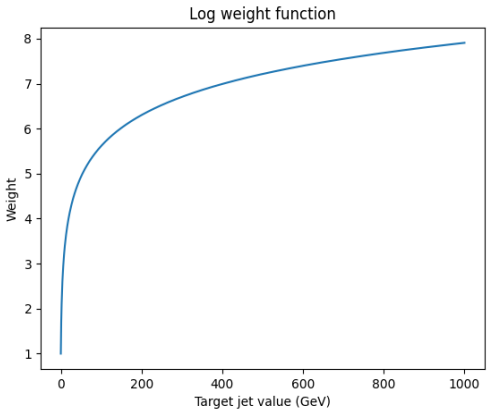
To optimize our FCN models, we experimented with several loss functions to enhance jet reconstruction.

**Tested Loss Functions:**

- **Mean Squared Error (MSE)**: Calculates the average of squared differences between predicted and actual values, penalizing larger errors more significantly.
- **Mean Absolute Error (MAE)**: Computes the average of absolute differences, reducing the impact of outliers.
- **Log Weighted MAE and MSE**: Modified versions where each pixel error is weighted by the logarithm of the target value, emphasizing higher $p_T$ values. A constant value of 1 is added to the logarithm for weight adjustment. See Figure 7b.
- **Gaussian Weighted MAE and MSE**: Each error is weighted by a Gaussian function centered around 200 GeV with a standard deviation of 200 GeV, emphasizing errors near 200 GeV. The Gaussian function is doubled and a constant of 1 is added. See Figure 7a.




(a) Gaussian Weight Function




(b) Logarithmic Weight Function

These loss functions have been tested to evaluate their effectiveness in training FCN models for jet reconstruction. The goal was to identify the most suitable loss function that can accurately capture the characteristics of the data and improve the model's performance.

## 3.4 Training Strategies

We adopted a consistent training strategy across all FCN models to ensure fair comparisons and reliable results.

We experimented with learning rates ranging from 0.02 to 0.001 to find the most effective rate for training. Setting the batch size to 32 allowed us to balance memory usage and training speed efficiently. Utilizing the Adam optimizer, which adaptively adjusts the learning rate, we incorporated a step size of 5 and a gamma of 0.5, meaning the learning rate was halved every 5 epochs.

Our dataset was divided into 80% training, 10% validation, and 10% testing segments, enabling us to evaluate performance at various stages. Although the training was set for up to 1000 epochs, we applied early stopping to prevent overfitting, halting the training if the validation loss did not improve for 10 epochs. This ensured that the model remained generalizable and did not learn irrelevant patterns.

Proper hyperparameter tuning is crucial for optimal model performance and generalizability. While this was not the main focus of our project, consistent hyperparameters were used for simplicity. However, further optimization could lead to improved model performance.

# 4 ATLAS Detector Data and Jet Reconstruction

In this section, we focus on the data used in this project, beginning with an introduction to the raw data and then discussing the treatments applied to both the ATLAS detector data and the reconstructed jet data.

## 4.1 Data Overview

The data used in this project is synthetically generated to mimic the conditions of the ATLAS detector at CERN. It comprises two main categories: ATLAS detector data and reconstructed jet data. There are two datasets used in this project: one without pileup and one with pileup. These datasets are designed to simulate the conditions expected at the ATLAS detector, providing a realistic environment for testing and evaluating our FCN models.

### 4.1.1 No-Pileup

During the initial exploration phase of this project, we chose to work with a no-pileup dataset. Pileup occurs when multiple proton-proton collisions happen simultaneously within the same bunch crossing, leading to overlapping signals in the detector. These overlaps can obscure the primary collision signals, complicating data analysis. By using a dataset free from pileup, we simplify the jet reconstruction process, allowing for more straightforward identification and reconstruction of primary jets from collision events. This approach makes it easier to develop and refine preprocessing methods and neural network models without the additional complexities of pileup.

Using a simplified dataset helps us focus on analyzing the primary collision events, thereby gaining a clearer understanding of the raw data characteristics without the interference caused by pileup. This dataset includes simulated proton collision events at the ATLAS detector, deliberately designed without any additional pileup events. It comprises 10,000 events, each containing simulated data from the ATLAS detector and the corresponding reconstructed jets. This no-pileup dataset serves as an excellent foundation for building and testing our FCN models, accelerating the development process and providing insightful feedback on the models' behavior and effectiveness in jet reconstruction.

### 4.1.2 Pileup

In the subsequent phase of the project, we transitioned to using a dataset with pileup to assess how additional collision events impact jet reconstruction. This more

complex dataset consists of 50,000 events with simulated proton collisions, including an average of 32 interactions per event. Although current ATLAS data has a pileup of 55, and HL-LHC is expected to have approximately 140 pileup interactions, using a dataset with 32 pileup interactions still provides valuable insights and approximates real ATLAS conditions.

In this dataset, signals from primary collision events mix with those from additional pileup events, similar to actual ATLAS data, where multiple collisions frequently occur simultaneously. Working with this dataset allows us to evaluate the impact of pileup on jet reconstruction and assess the performance of our FCN models under more realistic conditions. It enables us to explore the challenges posed by pileup and develop strategies to mitigate its effects on jet reconstruction.

Future work could involve using real data from the ATLAS detector to further refine the models and evaluate their performance under actual conditions. However, for the purposes of this project, the synthetic datasets provide a suitable environment for developing and testing our FCN models.

### 4.1.3 Event Structure

An event at the ATLAS detector involves proton collisions that results in the creation of various particles. These events are captured as a list of detector cells, with each cell holding information about the energy deposited by particles in that specific area.

The ATLAS detector comprises 187,650 cells, and for each cell, we record many parameters, including the $\eta$, $\phi$, and $z$ coordinates, the energy, the transverse momentum, and the sigma of the energy response. These parameters are essential for analyzing the trajectory and energy of the particles produced during collisions, enabling precise reconstruction of the physical events within the detector.

### 4.1.4 Reconstructed Jets

Alongside cell data, we also handle data from reconstructed jets, which are clusters of particles formed during the fragmentation of quarks and gluons. Similar to cell data, jet data for each event is presented as a list, with many parameters, including the $\phi$ and $\eta$ coordinates, the energy, and the transverse momentum of each jet.

The $z$ coordinate is not included for jets because it is assumed that all jets have a radius of 0.4 in the $\phi$ and $\eta$ plane, simplifying their spatial representation and analysis. The size of the jet data list varies depending on the number of jets reconstructed in each event.

## 4.2 Data Analysis

In this section, we analyze the ATLAS detector data and the reconstructed jet data to gain insights into the characteristics of the data and the challenges of jet reconstruction.

### 4.2.1 Distribution of ATLAS Detector Cells

We begin by examining the distribution of cells across different regions of the detector in the Pileup dataset. Analyzing this distribution helps visualize cell density in the detector's various layers, revealing the areas with the highest and lowest concentrations of cells.

The figure 8 shows the distribution of detector cells in the $(\eta, \phi)$ plane, displaying the cell density in different regions of the detector. A highly heterogeneous distribution of cells is observed, with higher densities in the central regions of the detector and a gradual decrease towards the edges. In $\phi$, a periodic distribution of cells reflects the accordion structure of the LAr Calorimeter, while in $\eta$, cell density gradually diminishes towards the ends of the detector.

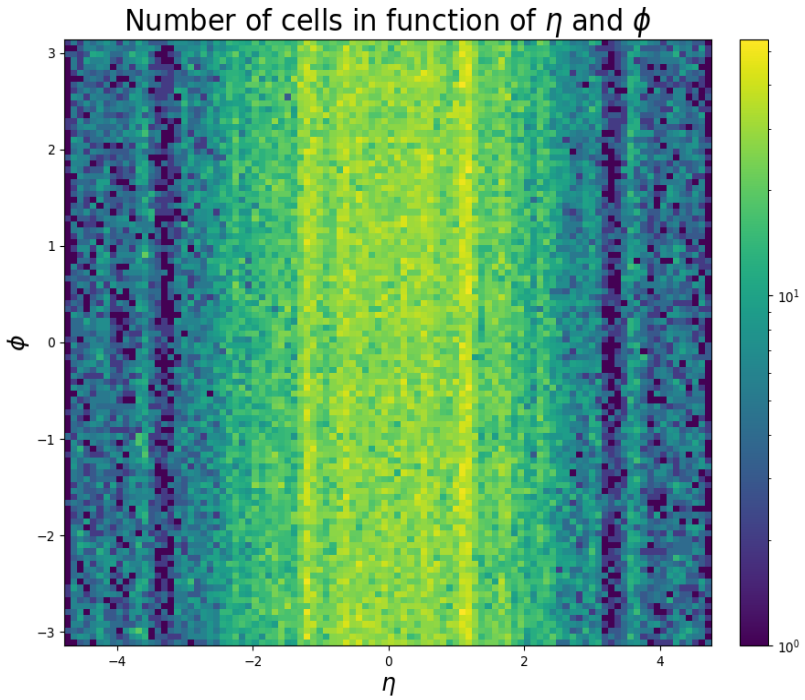



Figure 8: Distribution of ATLAS Detector Cells

### 4.2.2 Distribution of Jet Transverse Momentum

A crucial aspect of ATLAS detector data is the distribution of jet transverse momentum ($p_T$), which represents the primary source of information for reconstructing collision events.

Figure 9a presents the histogram of jet $p_T$, highlighting the distribution of $p_T$ measured in collision events. This visualization helps identify trends and key features of the data, such as the presence of high-$p_T$ jets, the density of events at different $p_T$ scales, and any anomalies or outliers.

A high presence of low-$p_T$ jets is observed, with the majority of events concentrated close to 5 GeV. However, there is also an extended tail distribution towards higher $p_T$ values, indicating the presence of very energetic jets in the data.

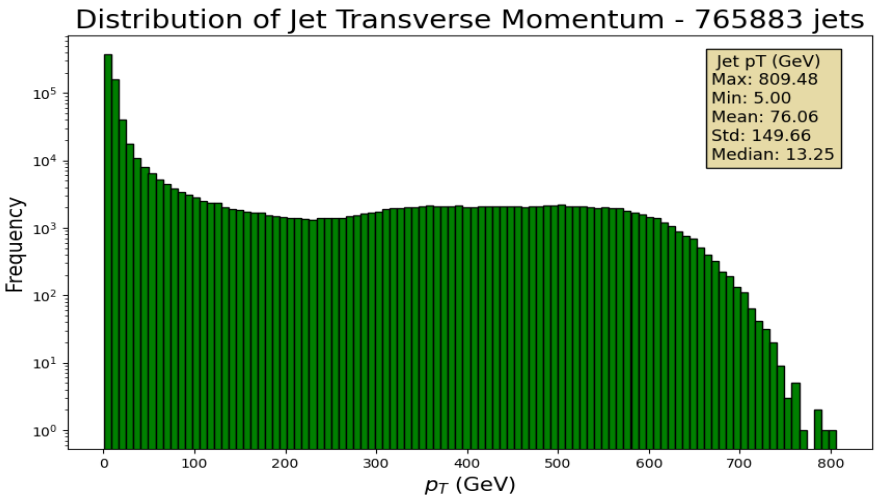
This heterogeneous distribution of jet transverse momentum underscores the importance of designing neural network models capable of efficiently handling this variability and precisely reconstructing jets from these data.

### 4.2.3 Distribution of Transverse Momentum for the Leading Jets

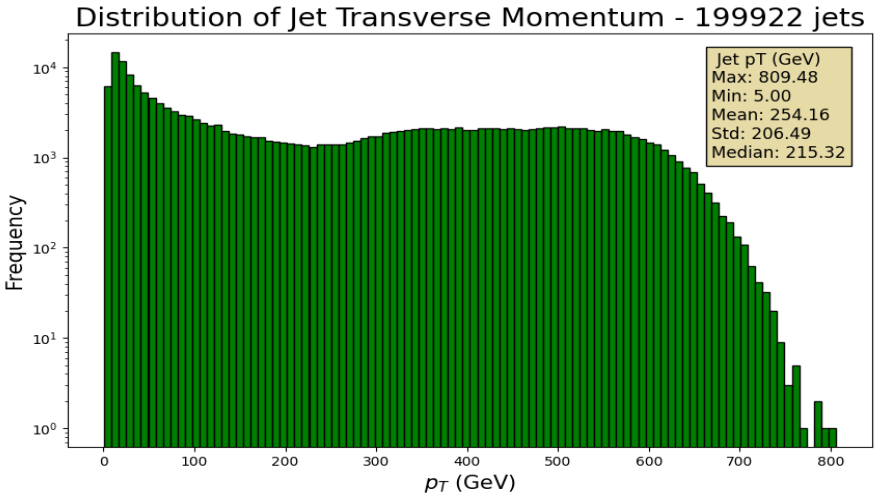
Another interesting analysis is the distribution of transverse momentum ($p_T$) for the leading jet in each event. The leading jet, being the jet with the highest $p_T$ in an event, plays a crucial role in characterizing the collision event and can be used by the trigger system to determine the event's significance.

Figure 9c illustrates the distribution of $p_T$ for the leading jets in each event. The minimum $p_T$ for the leading jet is around 170 GeV, with a mean $p_T$ of approximately 500 GeV. The distribution is skewed towards higher $p_T$ values, with a shorter tail towards lower $p_T$ values. This skewness results from the simulation methodology, which employs Monte Carlo methods.

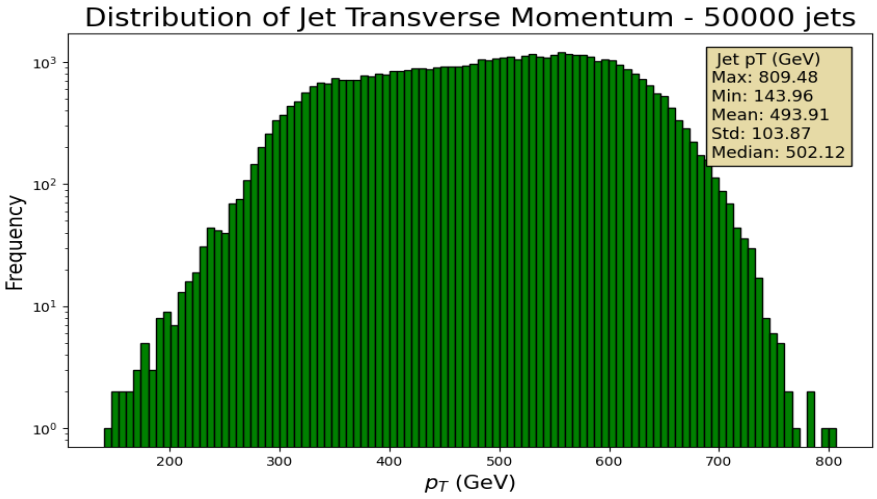
Furthermore, we examined the distribution of $p_T$ for the four leading jets in each event. As shown in Figure 9b, this distribution can be compared with that of the overall jet transverse momentum shown in Figure 9a. The comparison reveals that many low-$p_T$ jets are excluded when focusing on the four leading jets. By considering only the four leading jets, we can focus on the most energetic jets and reduce the data complexity. This approach can be useful for evaluating the performance of the models, as it emphasizes the jets that are most significant for event reconstruction and selection.




(a) All Jets




(b) Four Leading Jets




(c) Leading Jet

Figure 9: Distribution of Jet Transverse Momentum

### 4.2.4 Distribution of Jet Transverse Momentum Based on $\eta$

Another important statistical analysis of ATLAS detector data is the distribution of jet transverse momentum ($p_T$) based on the coordinate $\eta$, which represents the pseudorapidity of particles relative to the beam axis. This distribution helps identify spatial variations of jet $p_T$ across the detector, highlighting regions where jets are more frequent or more energetic. This information is crucial for tailoring data preprocessing, as we will see later.

Figure 10 shows this distribution of jet $p_T$ based on the coordinate $\eta$. A concentration of jets is observed within values of $\eta$ between -2.5 and 2.5, with lower density at the extremities. This concentric distribution of jets allows us to focus on this region to reduce data complexity and facilitate jet reconstruction.

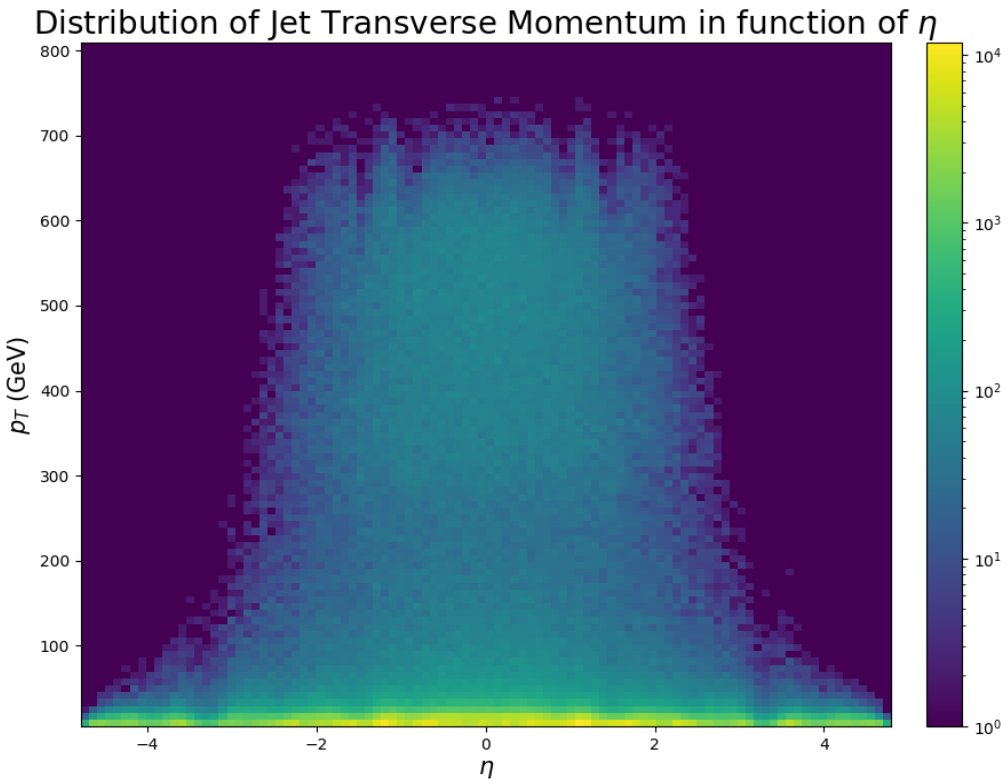



Figure 10: Distribution of Jet Transverse Momentum Based on $\eta$

### 4.2.5 Distribution of Jet Transverse Momentum Based on $\phi$

Finally, the statistical analysis of ATLAS detector data also includes the distribution of jet transverse momentum ($p_T$) based on the coordinate $\phi$, which represents the azimuthal angle of particles relative to the beam axis.

Figure 11 presents this distribution of jet $p_T$ based on the coordinate $\phi$. A certain homogeneity in the distribution of jets along the $\phi$ axis is observed, with a relatively uniform density across the entire range of values. This regular distribution of jets based on $\phi$ indicates a balanced distribution of collision events within the detector, thus facilitating consideration of this dimension during jet reconstruction.

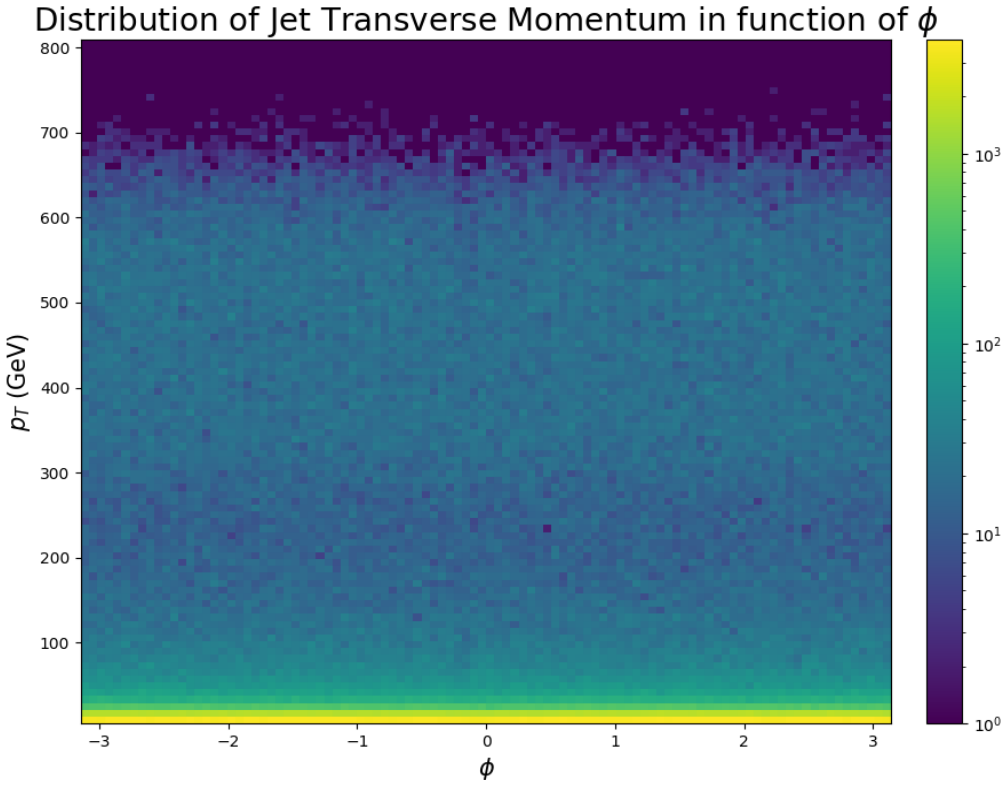



Figure 11: Distribution of Jet Transverse Momentum Based on $\phi$

# 5 Preprocessing

Preprocessing data from the ATLAS detector and jet reconstruction is crucial for training convolutional neural network models. This section explains how raw data is transformed into structured formats suitable for machine learning. The steps include converting ATLAS detector data into 2D images, expanding jet data, and rescaling inputs and targets. These processes simplify the data, reduce its size, and prepare it for neural network training.

## 5.1 Transforming ATLAS Detector Data into 2D Images

A crucial preprocessing step involves converting the raw ATLAS detector data from 187,650 cells into 2D images. This transformation preserves the ($\eta$) and ($\phi$) coordinates of the cells while aggregating the ($z$) dimension. $p_T$ values from cells within specific ($\eta$) and ($\phi$) intervals are summed and assigned to corresponding pixels in the image. This process significantly reduces data dimensionality and provides a comprehensive view of $p_T$ distribution across the detector cells. This transformation produces optimal input for FCN models. Figure 12 illustrates the conversion of ATLAS detector data into a 128x128 pixel 2D image, where each pixel represents the aggregated $p_T$ from corresponding detector cells. In this transformation, each interval corresponds to a size of 0.039 in $\eta$ and 0.049 in $\phi$.

In this transformation, we specifically focus on the $\phi$ range of $[-2.5, 2.5]$, a decision guided by the cell distribution and jet concentration in this region, as previously discussed. This focus simplifies the data while retaining crucial information for jet reconstruction. For the $\phi$ range, we include the full range of $[-\pi, \pi]$ to capture the complete azimuthal spread of particles. This approach ensures a balanced representation of collision events across the detector, as confirmed by the uniform jet distribution along the $\phi$ axis shown in earlier figures.

## 5.2 Transformation of Jet Reconstruction Data

As for the ATLAS detector data, the jet reconstruction data also undergoes a transformation to facilitate the training of neural network models. The jet reconstruction data is initially represented as a list of jets, each characterized by its ($\eta$) and ($\phi$) coordinates, as well as a $p_T$ value. To convert this data into a format suitable for training convolutional neural network models, we transform it into 2D images, where a jet is represented by a pixel with the corresponding ($\eta$) and ($\phi$) coordinates and $p_T$ value.

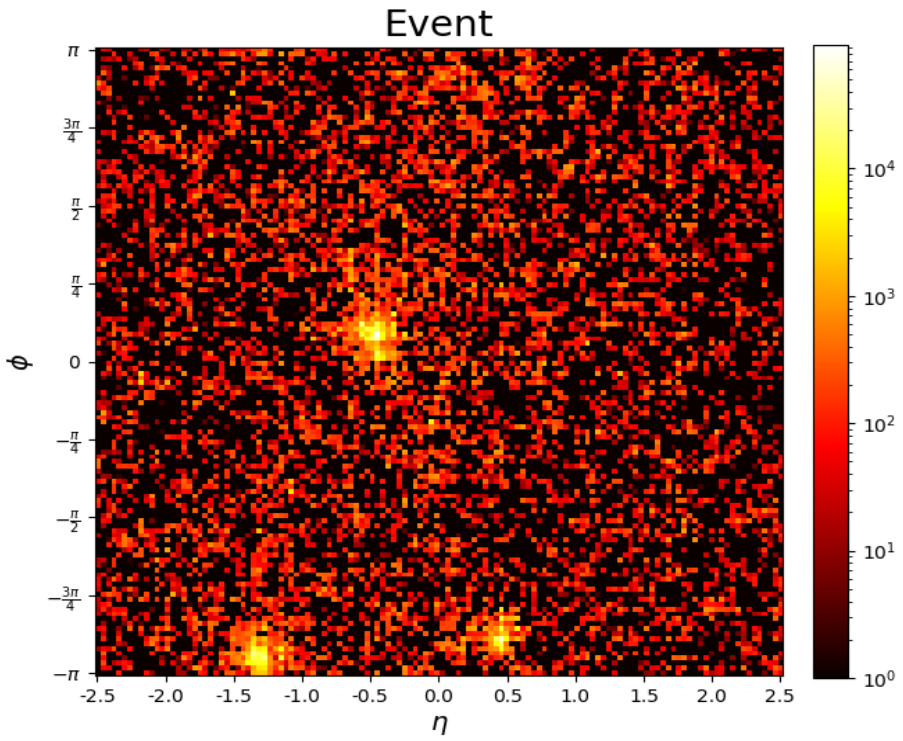



Figure 12: Transformation of ATLAS Detector Data into Images

This transformation allows us to align the input data with the format of the ATLAS detector data, thus ensuring consistency between the input and target data during model training. Moreover, this transformation into 2D images enables the use of FCN models, which are designed to process spatial data efficiently. We can see the result of this transformation in Figure 13a, corresponding to the same event as in Figure 12.

## 5.3 Expansion of Target Data

The transformation of the reconstructed jet data into 2D images presents certain challenges due to the sparse distribution of $p_T$ values across the detector. Indeed, each jet is represented by its ($\eta$) and ($\phi$) coordinates, as well as a $p_T$ value, resulting in images where only a few pixels are active (non-zero) and the rest of the pixels

remain zero. This very sparse distribution of values can complicate the learning process of the model, as most of the useful information is concentrated in a very small number of pixels.

To address this issue and potentially improve training efficiency, a technique of expanding the target data has been adopted. This method involves extending the region of influence of each jet by creating a 3x3 pixel window around the central pixel corresponding to the jet's position. The jet's $p_T$ value is then copied to each pixel in this window. This approach helps to spread the $p_T$ information more uniformly across the image, making the data less sparse and potentially easier for the model to learn from. In the ($\eta$) and ($\phi$) dimensions, this expansion corresponds to a size of 0.117 in $\eta$ and 0.147 in $\phi$.

Figure 13b shows the expansion of the target data for the same event as in Figure 12. This expansion of the target data can be justified by the fact that particle jets are considered to have a certain radius at the edge of the detector. Given that this radius is about 0.4, a 3x3 pixel window in a 128x128 image corresponds to a region smaller than this radius. Therefore, this expansion is a reasonable approximation of the jet's $p_T$ distribution in the detector.

This method requires special attention when multiple jets overlap. When 3x3 windows overlap, the $p_T$ values of adjacent jets can add up in the shared pixels, thus altering the intended initial $p_T$ distribution. It is crucial to manage these overlaps properly to maintain the integrity of the target data and ensure that the model learns the actual $p_T$ distribution in each event reliably. This is one of the reasons why we chose not to augment the target data by more than 3x3 pixels, even though this size does not exactly correspond to the jet radius. By keeping the window size small, we didn't had overlapping jets in the same window for our dataset.

## 5.4 Jets at the Border of the Image

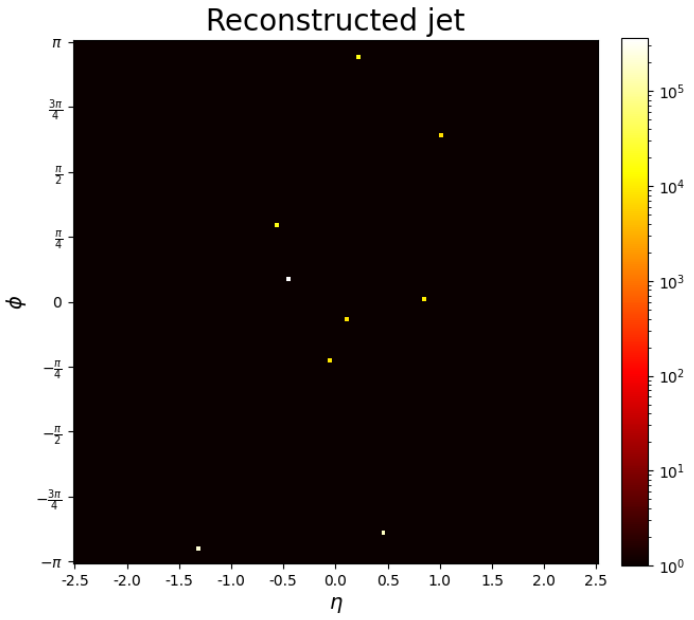
Handling jets located at the edge of the image along the $\phi$ axis requires special preprocessing steps. When a jet is at the edge, its $p_T$ information splits between the beginning and the end of the image. To address this, we initially duplicated the jet's $p_T$ information on both sides of the image. Specifically, for jets with $\phi$ values at pixel 0 or 127, we created a reduced 2x3 pixel version of the jet on both sides. However, this approach proved ineffective later in the project. The model struggled to predict the jet's $p_T$ on one side of the image. The clear solution was to pad the image with the opposite side along the $\phi$ axis, allowing the model to learn the jet's $p_T$ at the border more effectively and improving its prediction accuracy.

Due to time constraints, we were unable to implement this padding solution during the project. To avoid potential bias in our evaluation, we decided to exclude jets at the image borders from the evaluation process. This decision was made to ensure the integrity of our results, acknowledging the limitation while recognizing the importance of addressing it in future work.

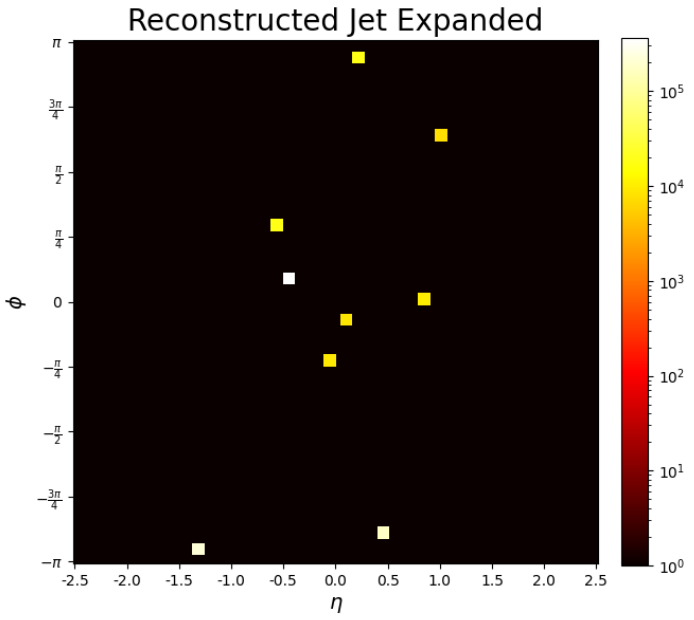
## 5.5 Rescaling of Input and Target Data

An important step in preprocessing ATLAS detector data is the rescaling of input and target data. As previously discussed, the energies of jets vary significantly, ranging from a few GeV to over 800 GeV. This wide range of $p_T$ values can pose significant challenges during model training, affecting both convergence and prediction accuracy. Rescaling is therefore a crucial step in preparing the input and target data of the ATLAS detector for the neural network model.

To address these challenges, we apply a rescaling technique to bring the $p_T$ values of the input and target data to a more manageable range. The raw data contains $p_T$ values in MeV, so we rescale them to GeV to facilitate the model's learning process. This rescaling is achieved by dividing the $p_T$ values by a factor of 1000, effectively converting them from MeV to GeV. This transformation reduces the scale of the $p_T$ values, making them more suitable for training neural network models using loss functions such as Mean Squared Error (MSE) or Mean Absolute Error (MAE).




(a) Transformation




(b) Expansion

Figure 13: Preprocessing of Jet Reconstruction Data

# 6 Postprocessing for Evaluation

After training the models, we implemented a post-processing procedure to interpret the model outputs and refine the analysis of the results. This procedure involves matching jets in both sets of images by position and $p_T$, and identifying jets that are predicted but not matched in the target. These steps help us accurately evaluate how well the model reconstructs jets and detect any discrepancies between the predicted and actual jets.

Given that low-$p_T$ jets are less critical for event selection, we focus on jets with $p_T$ values above a certain threshold. This approach ensures that the model's performance is assessed based on the most relevant jets for event selection. The thresholds are set at 20 GeV and 40 GeV. By concentrating on jets with $p_T$ values above these thresholds, we ensure that the model's performance is evaluated based on the most critical $p_T$ levels. Therefore, when matching jets, if neither the target nor the predicted jet has an $p_T$ value above the threshold, we do not consider them in the evaluation.

## 6.1 Matching Jet Positions

To assess how effectively the model reconstructs jet positions, we employ the Intersection over Union (IoU) metric. This metric quantifies the overlap between the target and predicted jets, serving as an indicator of the model's accuracy in localizing jets. IoU is calculated as the ratio of the intersection area to the union area of the target and predicted jets. A higher IoU value suggests that the predicted jet closely aligns with the target jet, demonstrating precise localization. We use a 9x9 pixel window for calculating IoU to accurately assess how well the model predicts jet positions relative to their actual locations. This window size is crucial because it ensures the model's predictions need to closely align with the target jets to be considered accurate.

For example, if the model predicts a jet covering a 9x9 pixel area, the maximum IoU with a 3x3 target jet would be 9 (the overlapping area) divided by 81 (the total area of the 9x9 window), yielding an IoU of about 0.111. In contrast, if a window of 3x3 pixels were used, the IoU would be 1, even if the predicted jet was significantly larger than the target jet. This could produce a misleadingly high IoU, suggesting a better match than actually exists. This approach ensures that our evaluation reflects the model's ability to localize jets accurately without being overly permissive.

We use IoU thresholds of 0.5 and 0.3. The 0.5 threshold is a common value used in object detection tasks, indicating that a predicted jet is considered a match if its IoU with the target jet is above 0.5. However, given the nature of our task and the potential loss of spatial precision in preprocessing steps, we also consider a lower IoU threshold of 0.3. This threshold provides more flexibility in matching jets while still ensuring that the model's localization of jets is sufficiently accurate.

Moreover, we can justify the use of a lower IoU threshold by considering the form of the jets after the preprocessing steps. The jets are represented as 3x3 pixel areas, which may not perfectly align with the predicted jets. If the model predicts a jet that is shifted by one pixel and doesn't have the exact same form as the target jet, it may still be a good prediction. However, with an IoU threshold of 0.5, this prediction would be considered a mismatch. Therefore, the 0.3 threshold allows for some flexibility in the matching process, reflecting the model's ability to predict jet positions accurately. Another idea would be to use a Scale-Adaptive IoU [10], which would allow for a more flexible matching process for small objects, like the jets in our case. This could be an interesting approach to explore in future work.

## 6.2 Matching Jet Transverse Momentum

To evaluate the model's precision in predicting jet transverse momentum ($p_T$), we utilize the ratio of errors between the $p_T$ values of the target and predicted jets. This measure helps determine how closely the predicted $p_T$ values match the actual values, shedding light on the model's effectiveness in reconstructing jet energies. The $p_T$ error ratio is calculated as follows:

$$p_T \text{ Error Ratio} = \frac{|\text{Predicted } p_T - \text{Target } p_T|}{\text{Target } p_T}$$

Although we expect the model to accurately predict jet positions, we need to evaluate its ability to predict jet energies accurately, even when the positions may not be precise. For this reason, we use again a 9x9 pixel window to compare $p_T$ values. Within this window, we compare the maximum $p_T$ value against the target $p_T$ value. We calculate the $p_T$ ratio as the difference between the predicted maximum and target $p_T$ values, normalized by the target $p_T$ value. This method provides leniency regarding the positional accuracy of the jet, focusing more on the model's ability to accurately predict jet energies.

We then check if the $p_T$ ratio is within a predefined error limit threshold. We set this threshold at 10% and 20% to evaluate the model's performance in predicting jet energies accurately. A 10% threshold indicates that a predicted jet matches well if its $p_T$ value is within 10% of the target $p_T$ value. This threshold ensures that the model's predictions are sufficiently close to the actual $p_T$ values, reflecting its ability to reconstruct jet energies accurately.

## 6.3 Unmatched Predicted Jets

In addition to evaluating matched jets, we also examine unmatched predicted jets. These are jets predicted by the model that do not match any of the target jets. For the previous matching process, we used a 9x9 pixel window centered on each target jet. After calculating the Intersection over Union (IoU) and $p_T$ error ratio for each jet, we remove both the target and predicted jets from this window. Once this process is completed, we're left with unmatched predicted jets that do not correspond to any target jets.

Next, in order to identify unmatched predicted jets, we use a simple algorithm. We find the pixel with the highest value in the predicted image and consider it as a predicted jet. We then remove the 9x9 pixel window centered on this pixel and repeat the process until all pixels in the predicted image are zero. This method allows us to identify all unmatched predicted jets in the image, providing insights into the model's performance in predicting jets that are not present in the target data.

## 6.4 Leading Jet Metrics

Accurately predicting the leading jet is crucial for event selection in the ATLAS detector, as the triggering system relies on the number of jets above a certain $p_T$ threshold. To evaluate the model's performance in reconstructing the leading jets, we consider specific metrics. We computed additional metrics to assess the accuracy of the model in predicting both the leading jet and the four leading jets. The leading jet is the jet with the highest $p_T$ in the event, while the four leading jets are those with the highest $p_T$ values. As discussed in section 4.2.3, focusing on the leading jets allows us to evaluate the model's performance on the most significant jets for event selection.

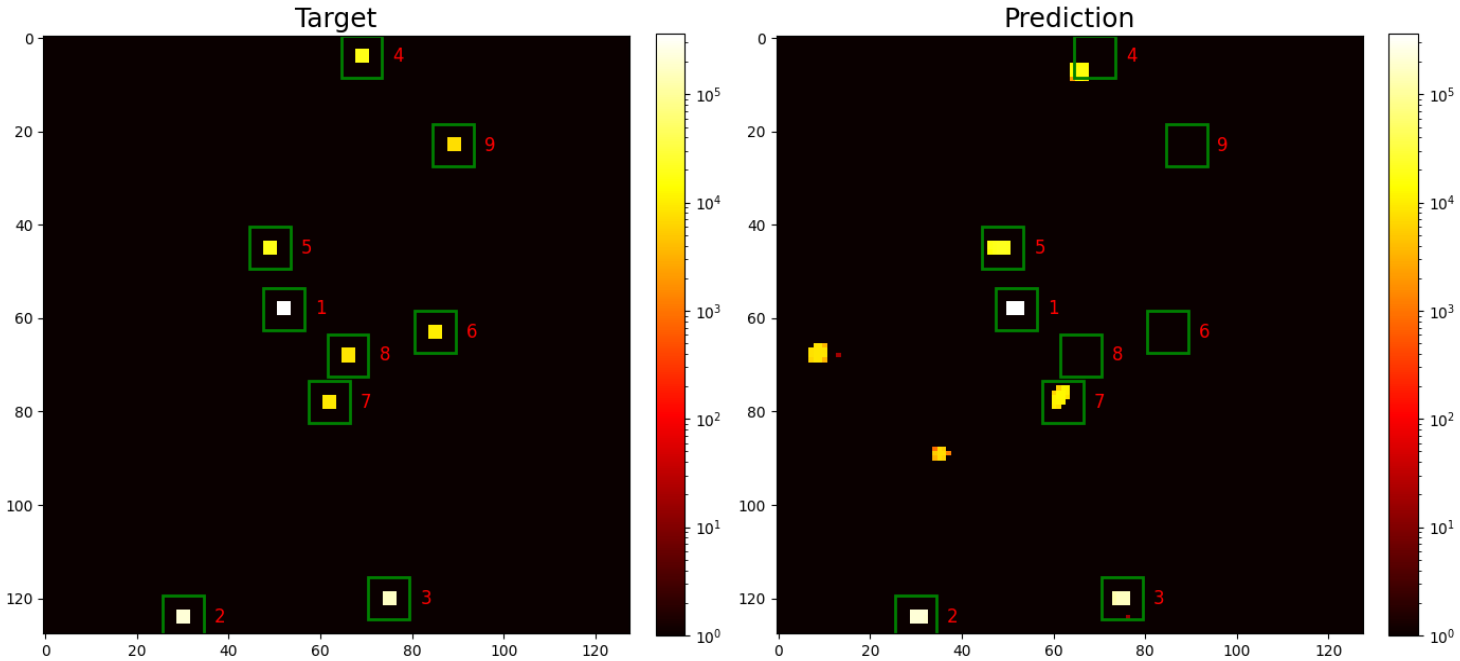
## 6.5 Example of a Predicted Event

This section highlights the matching process between the target and predicted jets in a sample event. The event shown in Figure 14 demonstrates how predicted and target jets are matched using our largest model, Model 6.

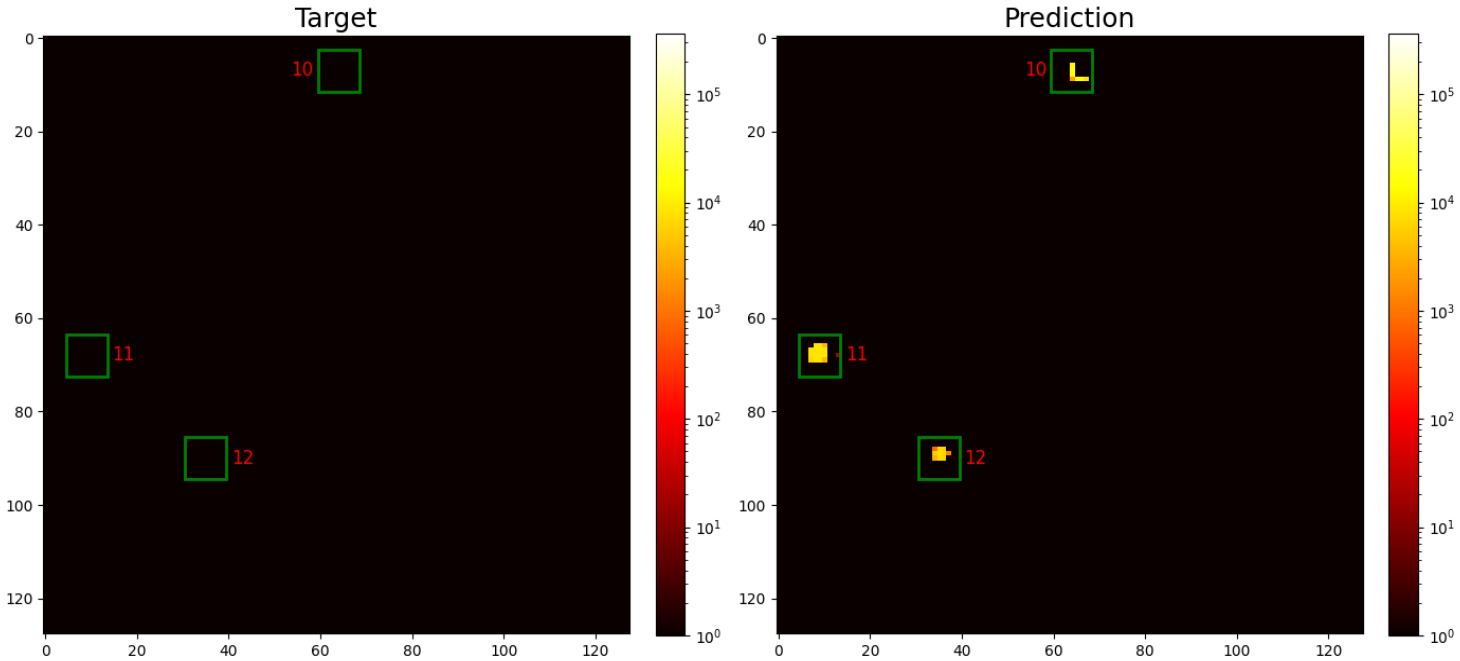
The first step, illustrated in Figure 14a, involves matching the target jets with the predicted jets, if any. The green squares represent the 9x9 pixel windows centered on the target jets. The values of the jets are represented by a color gradient, with the intensity indicating the jet values, as shown with the color bar on the right. Table 1 provides detailed information on each of these matched jets, including the target and predicted jet numbers, the $p_T$ of the target jet, the $p_T$ of the predicted jet, the ratio error in $p_T$ prediction, and the Intersection over Union (IoU) value. As shown, jets with high $p_T$ values are well predicted by the model. The model effectively matches these high-$p_T$ jets with significant overlaps, as indicated by the high IoU values. However, for jets with low-$p_T$ values, the model either did not predict them accurately or failed to match them with the target jets. The second step, depicted in Figure 14b, shows the remaining predicted jets after matching the target jets. These unmatched predicted jets are mostly low-$p_T$ jets.

This aligns with our approach of removing low-$p_T$ values to focus on the most significant jets. By excluding these low-$p_T$ jets, we can better assess the model's performance on the crucial aspects of the event. For example, with a 20% error threshold, 0.5 IoU threshold, and 20 GeV $p_T$ threshold, all the considered jets in this event are matched. However, if we set the error threshold to 10%, we see that the evaluation process would consider jet number 5 as unmatched, as its $p_T$ value is 13.26% off the target value. This discrepancy highlights the importance of setting appropriate thresholds for evaluating the model's performance.

An interesting phenomenon can be observed in the second step: jet number 10 appears as a residual jet after removing jet number 4 in the first matching step. This illustrates some of the difficulties in predicting jet positions. As seen in Table 1, the $p_T$ value for jet number 4 was well predicted with only a 10% error, but its position was not accurately predicted, resulting in an IoU of 0.0. This discrepancy highlights the challenges in precisely localizing jets, even when their $p_T$ values are predicted correctly. Moreover, we can observe that jets number 11 and 12 are predicted by the model but are not present in the target data. These unmatched predicted jets provide insights into the model's performance in predicting jets that are not part of the target data.




(a) Matching target jets




(b) Remaining predicted jets

Figure 14: Example of a Predicted Event with matching process

| Jet | Target | Prediction | Ratio Error (%) | IoU |
|---|---|---|---|---|
| 1 | 363841 | 359325 | 1.24 | 0.75 |
| 2 | 216230 | 217810 | 0.73 | 0.75 |
| 3 | 163681 | 161817 | 1.14 | 0.69 |
| 4 | 18397 | 16532 | 10.14 | 0.0 |
| 5 | 18307 | 20734 | 13.26 | 0.60 |
| 6 | 10269 | 0 | 100.0 | 0.0 |
| 7 | 9387 | 12734 | 35.66 | 0.32 |
| 8 | 8687 | 0 | 100.0 | 0.0 |
| 9 | 7426 | 0 | 100.0 | 0.0 |
| 10 | 0 | 14916 | 100.0 | 0.0 |
| 11 | 0 | 8950 | 100.0 | 0.0 |
| 12 | 0 | 7219 | 100.0 | 0.0 |

Table 1: Matching Process for the Predicted Event from Figure 14

# 7 Results and Discussion

In this section, we present the outcomes of our evaluation process. Initially, we review the loss function that yielded the most favorable results during training. We then show a comparison of the evaluation metrics for the different models, focusing on the IoU metric for jet position evaluation and the ratio of $p_T$ errors for jet $p_T$ evaluation. Results are shown for matching using both criteria (IoU and $p_T$ error), as well as for each criterion separately. Finally, we analyze the unmatched predicted jets to identify any patterns or trends that could provide insights into the model's performance.

## 7.1 Loss Function

In Section 3.3, we discussed several loss functions that we tested during the training of our FCN models. We evaluated these loss functions to find the one that worked best with our dataset and model architecture.

We quickly found that the Log Weighted Mean Absolute Error (MAE) loss function was the most effective for our dataset. This loss function helped the model focus on the higher $p_T$ values, which are more critical for jet reconstruction. By adding a logarithmic weight to the target values, the model could learn the $p_T$ distribution more effectively and achieve better results.

In contrast, the other loss functions did not perform well. In many cases, the model failed to learn anything, and the output was just a black image, composed entirely of zeros. This issue likely arises from the sparse nature of the data, which requires the model to learn how to handle this sparsity. The logarithmic weighting helped the model to focus on areas where the $p_T$ was present, allowing it to learn the $p_T$ distribution more effectively.

This is why we chose to use this loss function for all our models. This adaptation of the loss function highlights the importance of tailoring the model to the specific characteristics of the data. Further work is needed to explore how different loss function adaptations can improve the model's performance and its ability to reconstruct jets accurately.

In the following sections, we will present the results obtained using the Log Weighted MAE loss function, which we found to be the most suitable for our dataset and model architecture.

## 7.2 Comparison of Evaluation Metrics

As previously discussed, we evaluated the models using two primary metrics: the Intersection over Union (IoU) for jet position evaluation and the ratio of $p_T$ errors for jet $p_T$ evaluation. We considered both criteria together and separately to provide a comprehensive assessment of the model's performance.

We will now present the results of this evaluation. The goal is to compare the performance of the different models based on these metrics and identify tendencies or patterns that could help us understand the models' strengths and weaknesses.

### 7.2.1 Performance of Jets Matching

As we say in the section 6, we evaluated the models using two $p_T$ thresholds: 20 GeV and 40 GeV. We also considered two sets of thresholds for the IoU and $p_T$ error criteria: 0.3 and 20%, and 0.5 and 10%. These thresholds were chosen to assess the model's performance under different conditions and evaluate its ability to reconstruct jets accurately.

The Table 2 presents the percentage of target jets matched for all models using these criteria. The percentages represent the number of target jets above the $p_T$ threshold that were matched by the model. The table is divided into two parts, one for a 20 GeV $p_T$ threshold and the other for a 40 GeV $p_T$ threshold. Each part of the table is further divided into two subparts: one for the IoU threshold of 0.3 and $p_T$ error of 20%, and the other for the IoU threshold of 0.5 and $p_T$ error of 10%. The subparts show the percentage of matched jets using both IoU and $p_T$ criteria, only the IoU criterion, and only the $p_T$ criterion.

The first observation is that increasing the $p_T$ threshold significantly improves the percentage of matched jets. For example, for the first model with an IoU threshold of 0.3 and an $p_T$ error of 20%, the percentage of matched jets increases from 76.4% to 89.6%. This increase is expected because there are many low-$p_T$ jets between 20 and 40 GeV, and these jets are harder to predict. This indicates that a important part of the unmatched jets are low-$p_T$ jets.

The second observation is that the percentage of matched jets increases with the complexity of the model. For instance, for the IoU threshold of 0.3 and an $p_T$ error of 20%, the percentage of matched jets goes from 76.4% for the first model to 87.9% for the last model with a 20 GeV $p_T$ threshold. This trend is consistent across all models and thresholds, indicating that more complex models perform better in reconstructing jets. This suggests that further optimization of the model architecture could improve performance.

| Model | IoU = 0.3, $p_T$ error = 20% | | | IoU = 0.5, $p_T$ error = 10% | | |
|---|---|---|---|---|---|---|
| | IoU & $p_T$ | IoU | $p_T$ | IoU & $p_T$ | IoU | $p_T$ |
| 1 | 76.4% | 82.2% | 83.8% | 50.7% | 56.8% | 71.1% |
| 2 | 79.2% | 84.9% | 84.9% | 55.0% | 61.6% | 72.9% |
| 3 | 81.0% | 87.0% | 85.9% | 58.9% | 66.4% | 73.8% |
| 4 | 83.5% | 89.6% | 87.5% | 62.3% | 71.1% | 75.6% |
| 5 | 85.9% | 91.1% | 89.7% | 66.9% | 74.5% | 78.9% |
| 6 | 87.9% | 91.7% | 91.8% | 70.6% | 77.6% | 81.7% |

(a) 20 GeV $p_T$ Threshold

| Model | IoU = 0.3, $p_T$ error = 20% | | | IoU = 0.5, $p_T$ error = 10% | | |
|---|---|---|---|---|---|---|
| | IoU & $p_T$ | IoU | $p_T$ | IoU & $p_T$ | IoU | $p_T$ |
| 1 | 89.6% | 92.9% | 93.3% | 66.5% | 71.1% | 85.6% |
| 2 | 91.5% | 95.0% | 93.7% | 71.3% | 76.4% | 86.5% |
| 3 | 92.8% | 95.5% | 94.5% | 76.1% | 80.7% | 88.0% |
| 4 | 94.6% | 97.1% | 95.7% | 78.3% | 83.2% | 89.5% |
| 5 | 95.6% | 97.8% | 96.4% | 83.4% | 87.3% | 91.8% |
| 6 | 96.3% | 98.0% | 97.1% | 86.1% | 89.1% | 93.4% |

(b) 40 GeV $p_T$ Threshold

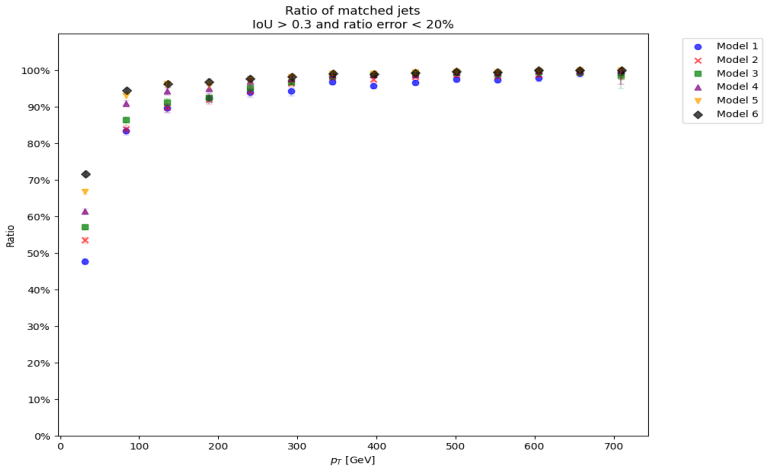
Table 2: Percentage of Target Jets Matched

The third observation is that the IoU threshold has a significant impact on the percentage of matched jets. For example, for the third model with a 20 GeV $p_T$ threshold, the percentage of matched jets decreases from 87.0% to 66.4% when the IoU threshold is increased from 0.3 to 0.5 with only the IoU criterion applied. Similarly, the percentage of matched jets decreases from 85.9% to 73.8% when the $p_T$ error threshold is tightened from 20% to 10% with only the $p_T$ criterion applied.

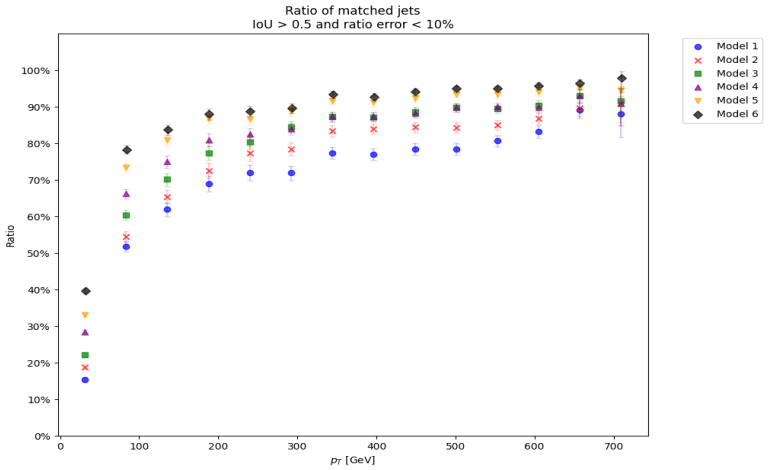
These observations indicate that adjustments to either criterion can substantially affect the matching outcomes. Therefore, these thresholds must be carefully chosen based on the precision requirements and the model's capabilities.

### 7.2.2 Performance of Jets Matching Based on Transverse Momentum

To better understand the performance of the models, we computed the percentage of target jets matched as a function of the target jet $p_T$. The results are shown in Figure 15 for a 20 GeV $p_T$ threshold. As expected, the percentage of matched jets increases with the $p_T$ of the target jet, approaching 100% for the highest $p_T$ levels.




(a) IoU = 0.3, $p_T$ error = 20%




(b) IoU = 0.5, $p_T$ error = 10%

Figure 15: Ratio of Matching Jets as a Function of Target Jet $p_T$ with 20 GeV $p_T$ Threshold

In the previous section, we highlighted that a significant portion of the unmatched jets were low-$p_T$ jets. This is confirmed by the results in Figure 15, where it is evident that most unmatched jets are low-$p_T$ jets. This indicates that much of the loss in evaluation metrics comes from low-$p_T$ jets. This is expected, as low-$p_T$ jets are harder to predict due to their lower signal-to-noise ratio.

Furthermore, Figure 15b clearly shows that increasing the model complexity improves the quality of the predictions. Although this trend is less pronounced in Figure 15a, it is still present, demonstrating the general benefit of more complex models for jet reconstruction accuracy.

### 7.2.3 Performance of Leading Jets Matching

To further evaluate the models, we analyzed their performance in predicting the leading jet, which is the jet with the highest $p_T$ value in the event. The leading jet is crucial for event selection in the ATLAS detector. As before, we computed the percentage of matched jets for the leading jet based on the IoU and $p_T$ criteria. The results are presented in Table 3 for a 20 GeV $p_T$ threshold. The table has the same structure as the previous one. We only show the results for a 20 GeV $p_T$ threshold because there is no leading jet with an $p_T$ value below 40 GeV as we saw in section 4.2.3. This results in the same table for the 40 GeV $p_T$ threshold.

| Model | IoU = 0.3, $p_T$ error = 20% | | | IoU = 0.5, $p_T$ error = 10% | | |
|---|---|---|---|---|---|---|
| | IoU & $p_T$ | IoU | $p_T$ | IoU & $p_T$ | IoU | $p_T$ |
| 1 | 97.31% | 97.68% | 99.27% | 80.50% | 81.77% | 95.91% |
| 2 | 98.40% | 98.79% | 99.50% | 85.49% | 86.32% | 97.47% |
| 3 | 98.86% | 99.37% | 99.32% | 89.52% | 90.66% | 97.78% |
| 4 | 99.18% | 99.38% | 99.60% | 89.35% | 90.12% | 98.23% |
| 5 | 99.43% | 99.65% | 99.60% | 92.87% | 93.53% | 98.63% |
| 6 | 99.47% | 99.66% | 99.71% | 94.59% | 94.97% | 99.06% |

Table 3: Percentage of Leading Target Jets Matched with 20 GeV $p_T$ Threshold

Overall, the models perform well for the leading jet, with a percentage of matched jets close to 100% for all models for the IoU threshold of 0.3 and an $p_T$ error of 20%. This indicates that the models can accurately predict the leading jet's $p_T$ and position. However, the percentage of matched jets decreases when the IoU threshold is increased to 0.5, with the percentage dropping to around 80% for the first model. This highlights the difficulty of predicting the exact position of the jets, even for the leading jet.

Regarding the different models, we observe the same trends as for the general target jet matching. More complex models deliver better performance, though the improvement is less significant. This suggests that the leading jet, having the highest $p_T$ value, is easier to predict than other jets. Moreover, the $p_T$ prediction task is successful for all models. The models are very good at predicting the $p_T$ of the leading jet, with the percentage of matched jets not falling below 95% for any model. This demonstrates that the models can accurately predict the $p_T$ of the leading jet, even when they struggle with position prediction.

### 7.2.4 Performance of the Leading Jets Matching Based on Transverse Momentum

As before, we also analyzed the percentage of matched jets as a function of the $p_T$ of the target jet. The results are shown in Figure 16. As expected, the percentage of matched jets increases with the $p_T$ of the target jet.

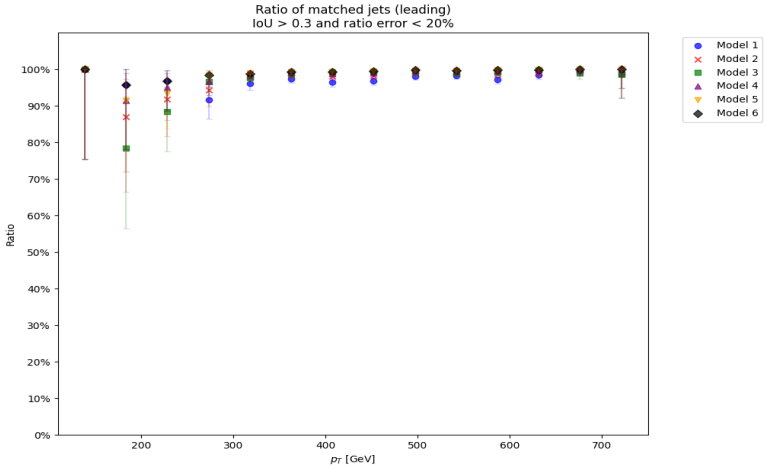
On Figure 16b, we can clearly see that the percentage of matched jets increases with the complexity of the model. This trend is less pronounced in Figure 16a, but it is still present as we saw it in the Table 2. This indicates one more time that more complex models perform better in reconstructing jets.

Given that there are few leading jets with low-$p_T$ values, the results for lower values are less reliable. This sometimes results in unexpected outcomes for smaller values, such as smaller models showing better results.

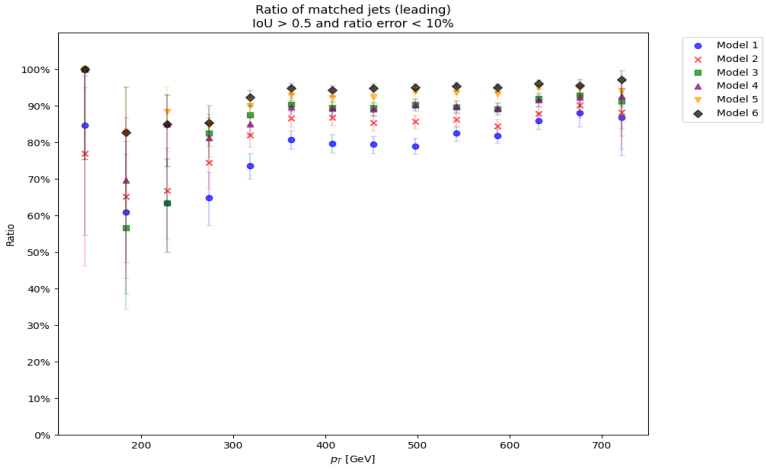
### 7.2.5 Performance of the Four Leading Jets Matching

To further evaluate the models, we analyzed their performance in predicting the four leading jets. The four leading jets are the four jets with the highest $p_T$ values in the event. Again, we show the results in the same format as before, with the percentage of matched jets based on the IoU and $p_T$ criteria for a 20 GeV $p_T$ threshold and a 40 GeV $p_T$ threshold. The results are presented in Table 4.

The same observations can be made for the four leading jets matching as for the last two evaluations. The percentage of matched jets increases with the complexity of the model, and the IoU threshold has a significant impact on the percentage of matched jets. The percentage of matched jets also increases with the $p_T$ threshold, showing that there is a number of low-$p_T$ jets that are part of the four leading jets as we saw in section 4.2.3.




(a) IoU = 0.3, $p_T$ error = 20%




(b) IoU = 0.5, $p_T$ error = 10%

Figure 16: Ratio of Matching Jets for the Leading Jet as a Function of Target Jet $p_T$

| Model | IoU = 0.3, $p_T$ error = 20% | | | IoU = 0.5, $p_T$ error = 10% | | |
|---|---|---|---|---|---|---|
| | IoU & $p_T$ | IoU | $p_T$ | IoU & $p_T$ | IoU | $p_T$ |
| 1 | 85.95% | 89.82% | 91.17% | 61.29% | 66.31% | 81.51% |
| 2 | 87.99% | 91.86% | 91.53% | 66.06% | 71.32% | 82.67% |
| 3 | 89.42% | 92.99% | 92.26% | 70.31% | 75.82% | 83.67% |
| 4 | 91.54% | 94.91% | 93.66% | 73.04% | 79.17% | 85.29% |
| 5 | 92.97% | 95.98% | 94.86% | 77.94% | 83.13% | 87.98% |
| 6 | 94.23% | 96.44% | 96.04% | 81.24% | 85.68% | 89.96% |

(a) 20 GeV $p_T$ Threshold

| Model | IoU = 0.3, $p_T$ error = 20% | | | IoU = 0.5, $p_T$ error = 10% | | |
|---|---|---|---|---|---|---|
| | IoU & $p_T$ | IoU | $p_T$ | IoU & $p_T$ | IoU | $p_T$ |
| 1 | 91.11% | 93.96% | 94.66% | 68.54% | 72.67% | 87.56% |
| 2 | 92.81% | 95.82% | 94.79% | 73.49% | 77.98% | 88.37% |
| 3 | 93.94% | 96.28% | 95.46% | 78.08% | 82.10% | 89.63% |
| 4 | 95.51% | 97.55% | 96.50% | 79.98% | 84.19% | 90.97% |
| 5 | 96.44% | 98.18% | 97.10% | 85.00% | 88.33% | 93.08% |
| 6 | 96.99% | 98.36% | 97.67% | 87.48% | 90.01% | 94.44% |

(b) 40 GeV $p_T$ Threshold

Table 4: Percentage of the four Leading Target Jets Matched

### 7.2.6 Performance of Unmatched Predicted Jets

To better understand the model's performance, we focused on the unmatched predicted jets. This measure includes not only the predicted jets that failed to match with any target jets due to low IoU or high $p_T$ errors, but also those predicted jets that had no corresponding targets at all. Unlike previous measures, this analysis does not consider target jets that were not predicted. This allows us to concentrate solely on the predictions, providing a clearer picture of the model's predictive capabilities. The results are presented in Table 5 for a 20 GeV $p_T$ threshold and a 40 GeV $p_T$ threshold. The table shows the percentage of unmatched jets for all models using different IoU and $p_T$ error thresholds.

| Model | IoU = 0.3, $p_T$ error = 20% | | | IoU = 0.5, $p_T$ error = 10% | | |
|---|---|---|---|---|---|---|
| | IoU & $p_T$ | IoU | $p_T$ | IoU & $p_T$ | IoU | $p_T$ |
| 1 | 29.27% | 23.93% | 22.44% | 53.05% | 47.45% | 34.18% |
| 2 | 24.58% | 19.21% | 19.14% | 47.59% | 41.39% | 30.55% |
| 3 | 22.06% | 16.38% | 17.36% | 43.36% | 36.15% | 29.02% |
| 4 | 19.22% | 13.34% | 15.39% | 39.74% | 31.18% | 26.88% |
| 5 | 16.68% | 11.63% | 13.01% | 35.15% | 27.69% | 23.40% |
| 6 | 14.74% | 11.01% | 11.00% | 31.47% | 24.68% | 20.78% |

(a) 20 GeV $p_T$ Threshold

| Model | IoU = 0.3, $p_T$ error = 20% | | | IoU = 0.5, $p_T$ error = 10% | | |
|---|---|---|---|---|---|---|
| | IoU & $p_T$ | IoU | $p_T$ | IoU & $p_T$ | IoU | $p_T$ |
| 1 | 15.56% | 12.38% | 12.06% | 37.36% | 32.96% | 19.36% |
| 2 | 11.80% | 8.42% | 9.74% | 31.28% | 26.36% | 16.68% |
| 3 | 9.65% | 6.98% | 7.95% | 25.89% | 21.38% | 14.35% |
| 4 | 7.43% | 4.98% | 6.34% | 23.35% | 18.61% | 12.41% |
| 5 | 5.98% | 3.88% | 5.26% | 17.98% | 13.74% | 9.77% |
| 6 | 5.45% | 3.81% | 4.66% | 15.46% | 12.13% | 7.84% |

(b) 40 GeV $p_T$ Threshold

Table 5: Percentage of Unmatched Predicted Jets

As we can see, the percentage of unmatched jets decreases with the complexity of the model, which is expected. This is important because it indicates that the increase in the percentage of matched jets is not simply due to the models predicting more

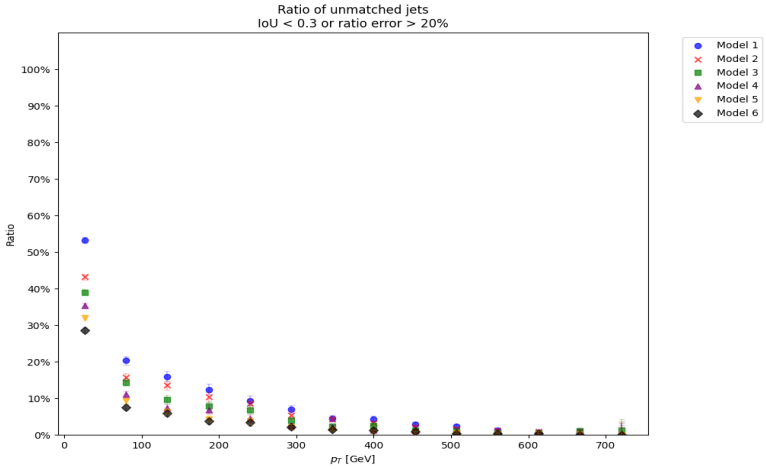
jets. If that were the case, we would see a corresponding increase in unmatched jets. However, the decreasing percentage of unmatched jets with model complexity shows that the models are not just producing more predictions but are making more accurate and relevant predictions. This trend aligns with the increase in the percentage of matched jets, confirming the models' improved performance.

### 7.2.7 Transverse Momentum Distribution of Unmatched Predicted Jets

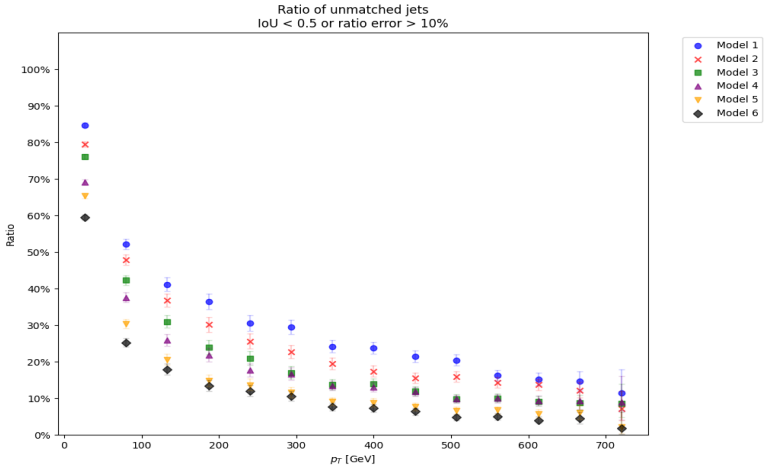
To further analyze the unmatched predicted jets, we examined their $p_T$ distribution. This analysis provides insights into two types of predicted jets: those that fail to match a target jet due to low IoU or high $p_T$ error, and those that do not correspond to any target jet, where the corresponding area in the target is completely empty (all zeros). Understanding these aspects helps identify areas for improvement in the model's predictive capabilities.

The results for the two types of unmatched jets are shown in Figure 17 for a 20 GeV $p_T$ threshold. The $p_T$ distribution of the unmatched jets varies across models. For all models, the $p_T$ distribution of the unmatched jets peaks at low $p_T$ values. This indicates that the models either struggle to predict low-$p_T$ jets accurately or that many predicted low-$p_T$ jets do not correspond to any target jets.

To better understand a model's tendency to predict jets that do not correspond to any target, we examined the frequency of these specific unmatched jets. In Figure 18, we can see that the frequency of these unmatched jets peaks at low $p_T$ values. This indicates that the models tend to predict low-$p_T$ jets that do not correspond to any target jets. This is consistent with previous observations that the models struggle with low-$p_T$ jets, which are harder to predict accurately due to their lower signal-to-noise ratio.




(a) IoU = 0.3, $p_T$ error = 20%




(b) IoU = 0.5, $p_T$ error = 10%

Figure 17: $p_T$ Distribution of Unmatched Predicted Jets with 20 GeV $p_T$ Threshold

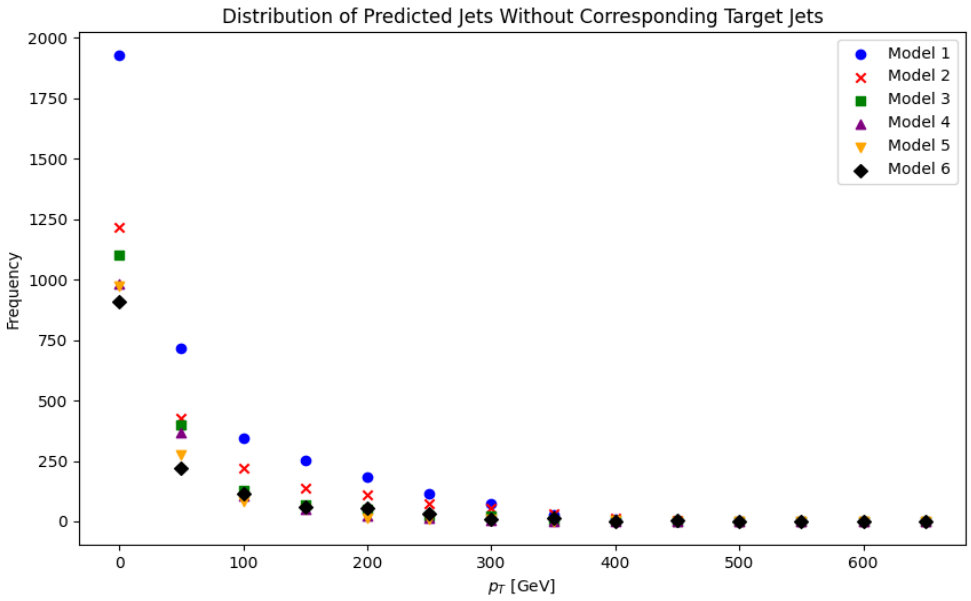



Figure 18: Frequency of Unmatched Predicted Jets that Do Not Correspond to Any Target Jets

# 8 Challenges and Future Research Directions

In this section, we discuss the challenges faced during this study and propose future research directions to improve jet reconstruction in the ATLAS detector. The challenges and future research directions are based on the observations and results presented in the previous sections.

- **Loss of Information**: Converting the input data from the ATLAS detector into images can lead to a loss of information, affecting the model's ability to accurately reconstruct jets. Transforming 3D data into 2D images might result in the loss of some spatial and $p_T$ information. To address this, alternative data representations or preprocessing techniques could be explored to retain more information and improve the model's predictive capabilities.
- **Targets Expansion**: Expanding the target jets in a 3x3 grid around the jet's center could lead to overlap of target jets. Although this was not an issue with our dataset, it could be problematic in other datasets or real-world scenarios. This overlap might affect the model's ability to accurately match predicted jets with target jets. Finding a balance in target expansion to avoid overlap could enhance matching accuracy.
- **Model Architecture**: The choice of model architecture impacts performance and jet reconstruction accuracy. The simple architecture used in this study may not be optimal. Testing custom models and optimizing the architecture, layer configurations, and hyperparameters could improve performance.
- **Matching window**: The size of the matching window used to determine if a predicted jet matches a target jet can affect matching accuracy. We used a 9x9 pixel window, but this may be too large or too small, leading to inaccurate matches. Optimizing the matching window size could improve accuracy by ensuring it is neither too restrictive nor too permissive.
- **Ensemble learning**: Combining the predictions of multiple models using ensemble learning techniques could enhance overall performance [5]. Training models with different architectures or hyperparameters and merging their predictions could yield better results.
- **Hyperparameter tuning**: We used a consistent set of hyperparameters across all models to simplify evaluation, but advanced hyperparameter tuning techniques could optimize performance [8]. Techniques such as grid search, random search, or Bayesian optimization could help find the best hyperparameters for the model.

By addressing these challenges and exploring these future research directions, we can enhance the model's performance and improve the jet reconstruction process in the ATLAS detector. This could lead to more accurate jet reconstruction, improved event selection, and enhanced physics analysis in high-energy physics experiments.

# 9 Conclusion

In this report, we presented the ATLAS detector and the challenges of jet reconstruction in high-energy physics experiments. We introduced FCN models for jet reconstruction and evaluated their performance using various metrics, including Intersection over Union (IoU) for jet position evaluation and the ratio of $p_T$ errors for jet $p_T$ evaluation. We compared the models using different thresholds and criteria, focusing on the target jets, leading jets, and four leading jets. We also analyzed the percentage of matched jets for different thresholds and criteria and examined the $p_T$ distribution of the unmatched predicted jets.

The evaluation process demonstrated that the models successfully learned to reconstruct jets without the Topo-clustering step, showing the potential of FCN models for jet reconstruction in the ATLAS detector. The models showed significant performance improvement with increased complexity, indicating that more complex models could enhance the accuracy of jet reconstruction. However, even very simple models like Model 1 showed acceptable performance, suggesting that further work on hyperparameters or architecture (without increasing complexity) could improve their performance.

All models showed a decrease in the percentage of unmatched jets with increasing target jet $p_T$, indicating that low-$p_T$ jets are more challenging to predict. One interesting avenue mentioned in the previous section is the use of ensemble learning techniques to combine predictions from multiple models and improve overall performance. Some models could be fine-tuned to predict low-$p_T$ jets, while others could focus on high-$p_T$ jets.

Finally, we identified several challenges and future research directions to improve jet reconstruction in the ATLAS detector. By addressing these challenges and exploring these research directions, we could enhance model performance and improve the jet reconstruction process in the ATLAS detector. This could lead to more accurate jet reconstruction, better event selection, and improved physics analysis in high-energy physics experiments.

To conclude, the next steps in this research could involve implementing the proposed model on FPGA, testing it on real data from the ATLAS detector, and comparing its performance with the current jet reconstruction methods. This would provide valuable insights into the model's practicality and potential for deployment in high-energy physics experiments.

# References


[1] ATLAS Collaboration. *The ATLAS Experiment - Discover the Calorimeter*. Consulted: July 1, 2024. 2024. URL: https://atlas.cern/Discover/Detector/Calorimeter.

[2] ATLAS Collaboration. *The ATLAS Experiment - Discover the Detector*. Consulted: July 1, 2024. 2024. URL: https://atlas.cern/Discover/Detector.

[3] ATLAS Collaboration. *The ATLAS Experiment - Discover the Trigger and Data Acquisition (DAQ)*. Consulted: July 1, 2024. 2024. URL: https://atlas.cern/Discover/Detector/Trigger-DAQ.

[4] CERN. *High-Luminosity LHC*. Accessed: July 1, 2024. 2024. URL: https://home.cern/science/accelerators/high-luminosity-lhc.

[5] Thomas G. Dietterich. "Ensemble Methods in Machine Learning". In: *Multiple Classifier Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 1–15. ISBN: 978-3-540-45014-6.

[6] FastML Team. *hls4ml: An Open-Source Codesign Workflow to Empower Scientific Low-Power Machine Learning Devices*. Version v0.8.1. 2023. DOI: 10.5281/zenodo.1201549. URL: https://github.com/fastmachinelearning/hls4ml.

[7] François Fleuret. *Deep Learning Course*. Accessed: July 1, 2024. 2024. URL: https://fleuret.org/dlc/.

[8] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. Chapter 8: Optimization, http://www.deeplearningbook.org/contents/optimization.html. MIT Press, 2016. URL: http://www.deeplearningbook.org.

[9] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *CoRR* abs/1512.03385 (2015). arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385.

[10] Pierre Le Jeune and Anissa Mokraoui. *Rethinking Intersection Over Union for Small Object Detection in Few-Shot Regime*. 2023. arXiv: 2307.09562 [cs.CV]. URL: https://arxiv.org/abs/2307.09562.

[11] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully Convolutional Networks for Semantic Segmentation". In: *CoRR* abs/1411.4038 (2014). arXiv: 1411.4038. URL: http://arxiv.org/abs/1411.4038.

[12] Claudia Marcelloni. "Photos of the massive ATLAS art mural at point 01 at CERN by Josef Kristofoletti, completed." 2010. URL: https://cds.cern.ch/record/1294059.