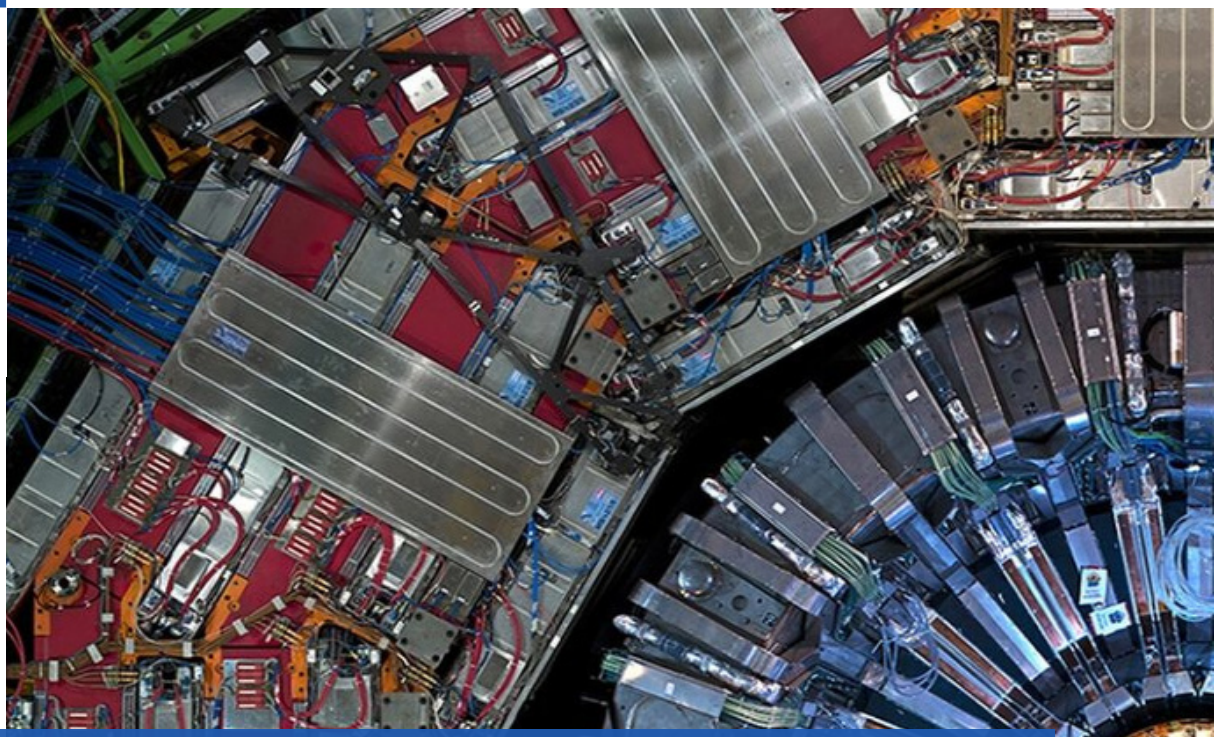
GOOGLE
SUMMER OF CODE 2023

APRIL 4, 2023



Graph Neural Networks

for End-to-End Particle Identification with the CMS Experiment

Prepared for
Google Summer of Code 2023

PRESENTED TO:

Machine Learning for Science (ML4SCI)

PRESENTED BY:

Khalid Bagus Pratama Darmadi S.Si

Graph Neural Networks for End-to-End Particle Identification with the CMS Experiment

Khalid Bagus Pratama Darmadi, S.Si
Universitas Gadjah Mada, Indonesia

April 4th, 2023

Abstract

The CMS Experiment at the Large Hadron Collider (LHC) requires accurate identification and reconstruction of particles, jets, and event topologies for effective searches for new physics. The End-to-End Deep Learning (E2E) project within the CMS Experiment aims to develop innovative deep learning approaches for these tasks. This proposal focuses on developing end-to-end graph neural networks (GNNs) for particle (tau) identification and integrating them with the CMSSW inference engine in both offline and high-level trigger systems of the CMS Experiment. The project will involve the development, testing, and benchmarking of GNN models for low-momentum tau identification and their inference on GPUs.

Contents

1 Introduction

Particle identification and reconstruction are crucial for the analysis of collision events at the LHC, as they provide insights into the underlying physics processes. Traditional reconstruction algorithms have limitations in terms of computational efficiency and accuracy, and deep learning-based methods have shown great promise in improving these aspects. The E2E project within the CMS Experiment explores the potential of deep learning techniques for particle identification and reconstruction tasks, with a focus on end-to-end learning approaches.

My motivation for participating in this project is to contribute to the advancement of deep learning techniques in high-energy physics and to gain hands-on experience in developing and implementing graph neural networks for particle identification. I am confident that my background in computer science, programming skills in C++, Python, and PyTorch, and experience in machine learning make me a suitable candidate for this project.

2 Personal Details

- Name: Khalid Bagus Pratama Darmadi, S.Si
- Affiliation: Department of Physics, Universitas Gadjah Mada
- Graduation Status: Graduated, Bachelor of Science with Honors
- Email: khalidbagus@gmail.com
- Phone: (+62) 82248155093
- Country of Residence: Indonesia
- Timezone: Eastern Indonesia Time (UTC+9)
- Local Residence: Papua, Indonesia

3 Biographical Information

As a passionate and committed individual hailing from rural Papua, Indonesia, I am eager to apply my expertise in machine learning and physics to the domain of particle detection and classification. My solid foundation in physics, combined with my experience in data science and machine learning, positions me to effectively address intricate and demanding problems in particle physics. My unique background has imbued me with a strong work ethic and a deep appreciation for the opportunities to learn and grow in my field.

Recently, I received a stipend from the TensorFlow team, which will enable me to pursue the TensorFlow Professional Certification on Coursera and take the TensorFlow Certification Exam in the near future. This opportunity will further demonstrate my proficiency in deep learning and its practical applications.

In addition to my machine learning experience, I hold a certificate as an Android developer, showcasing my versatility and ability to adapt to various technological domains.

Currently, I am engaged in Google's Isolated Sign Language Recognition competition, which entails using TensorFlow Lite to classify isolated American Sign Language (ASL) signs. This project has further refined my deep learning skills and solidified my understanding of TensorFlow. My background in electrical engineering and proficiency in MATLAB, acquired during my time as an electrical engineering student at Institut Teknologi PLN in Indonesia, have provided me with a strong foundation in data processing and analysis.

Additionally, my bachelor's degree in physics with honors from Universitas Gadjah Mada and my internship at a local particle accelerator in Yogyakarta have equipped me with a robust understanding of particle physics and its practical applications. My experience in 2D & 3D modeling of cyclotrons with EM simulation has also enabled me to visualize and analyze complex data.

I am confident that my background and expertise make me an exceptional candidate for this project. As someone who has overcome the challenges of coming from a rural area, I am especially motivated to contribute my skills and enthusiasm to the field of particle detection and classification, ultimately advancing research and analysis in this important domain.

4 Objectives and Goals

The primary objectives and goals of this project are to improve the particle identification process in the CMS experiment by developing end-to-end graph neural networks (GNNs) and implementing a CMSSW inference engine for use in reconstruction algorithms in offline and high-level trigger systems. The key points of focus include:

- Developing end-to-end GNNs for low-momentum tau identification, enhancing the experiment's ability to detect and analyze tau particles.
- Testing and benchmarking the performance of the GNN models, ensuring their efficiency and accuracy in particle identification tasks.
- Integrating the GNN models into the CMS experiment's reconstruction algorithms, optimizing the experiment's processing pipeline for particle identification.
- Implementing a CMSSW inference engine that leverages the developed GNN models, ensuring seamless integration and compatibility with the existing CMS infrastructure.

5 Benefits to Community

In addition to my academic background and professional experience, I am also deeply committed to giving back to my community. As someone who has roots in a rural area in Papua Indonesia, I understand the importance of providing access to education and resources to underprivileged communities. I believe that my expertise in machine learning and physics can be a valuable resource for the people in my community and beyond.

For example, I am eager to share my knowledge and experience with others through workshops and training programs. I see this as a way to empower individuals and communities with the skills and knowledge they need to tackle complex problems and make meaningful contributions to their fields. I am also passionate about using technology and data science to address social and environmental issues, and I believe that my work in particle detection and classification can play a role in this effort.

Overall, I am confident that my background and experience, combined with my commitment to community service, make me an ideal candidate for this project. I am eager to bring my skills and enthusiasm to this project and make a meaningful impact in the field of particle detection and classification.

6 Methodology

To achieve the project objectives and goals, a structured methodology will be employed, encompassing key phases such as data acquisition, model development, evaluation, and optimization. This approach, combined with close communication with mentors and the CMS experiment team, ensures a successful and efficient solution for end-to-end graph neural networks in particle identification.

1. **Literature Review and Feedback**:
   Review relevant GNN literature and discuss findings with mentors and the CMS experiment team.

2. **Data Preparation**:
   Acquire and preprocess necessary data, ensuring quality and relevance with mentor feedback.

3. **Model Development and Iteration**:
   Design and optimize GNN models for low-momentum tau identification, adjusting based on team input.

4. **Model Evaluation**:
   Test GNN models against benchmarks and existing methods, refining the model based on results and feedback.

5. **CMSSW Inference Engine Implementation**:
   Develop a compatible CMSSW inference engine, incorporating mentor and team feedback during development.

6. **Integration and Optimization**:
   Integrate GNN models and CMSSW inference engine into the CMS pipeline, optimizing for efficiency and performance with continuous team input.

7. **Documentation**:
   Document the development process, sharing with mentors and the team for review and feedback.

7 Deliverables

The following deliverables are expected from this project:

1. **End-to-end graph neural network (GNN) model for tau identification:** A fully developed and tested GNN model that effectively identifies low-momentum tau particles in the CMS experiment.

2. **Optimized GNN model for GPU inference:** An optimized version of the GNN model designed for improved performance on GPU platforms, with benchmark results comparing the optimized and original models.

3. **Integration with the CMSSW inference engine:** The developed GNN model will be integrated into the CMS experiment's CMSSW inference engine, ensuring compatibility and seamless operation.

4. **Comprehensive test suite:** A set of tests to ensure the GNN model's performance, functionality, and robustness, as well as the successful integration with the CMS experiment's systems.

5. **Documentation:** Detailed documentation covering the GNN model's design, implementation, and optimization, as well as instructions for integration, usage, and maintenance. This will also include inline code comments for better maintainability and understanding.

6. **Final project report:** A summary report outlining the project's objectives, methodology, results, and any lessons learned throughout the development process. This report will be helpful for future reference and for sharing the project's outcomes with the wider research community.

8 Related Work

The use of end-to-end deep learning methods for particle identification and reconstruction is a rapidly growing area of research in high energy physics. This project aims to contribute to this effort by developing graph neural networks for tau particle identification in the CMS experiment.

In recent years, several studies have been conducted to explore the application of end-to-end deep learning techniques in particle physics. A notable example is the work described in [1], where the authors proposed a novel end-to-end image-based classifier that leverages low-level simulated detector data to distinguish between signal and background processes in proton-proton collision events. The classifier was applied to the decay of the Higgs boson into two photons, demonstrating its ability to learn from the angular distribution of the photons, their intrinsic shapes, and the energy of their constituent hits.

Another study [2] described the construction of end-to-end jet image classifiers for quark-versus-gluon jet discrimination. The authors used multi-detector images that correspond to true maps of the low-level energy deposits in the detector, giving the classifier direct access to the maximum recorded information about the jet. The resulting classifier was found to be competitive with state-of-the-art jet classifiers based on particle-based algorithms.

The current project builds on these previous works by proposing the use of graph neural networks for tau particle identification. GNNs have recently gained popularity in various fields, including computer vision and natural language processing, due to their ability to handle structured and irregular data. Our project aims to explore the use of GNNs in the particle physics domain and demonstrate their potential in solving particle identification problems.

In this project, we will develop end-to-end graph neural networks for tau particle identification and evaluate their performance on a benchmark dataset. Our approach will leverage the inherent structured nature of particle data and the information encoded in the relationships between particles to achieve accurate and robust tau identification.

To achieve this goal, we will first perform a thorough analysis of the available data to determine the best representation for the graph inputs. This will involve exploring different graph structures and node features, as well as methods for handling missing data.

Next, we will implement and train multiple GNN models, using a combination of graph attention networks and Chebyshev layers to process the graph inputs. The models will be trained and evaluated on a benchmark dataset, and the best-performing model will be selected for further analysis.

Finally, we will perform a comprehensive evaluation of the selected model, including a comparison with state-of-the-art tau identification methods and an assessment of its robustness to various challenges, such as missing data and particle misidentification. The resulting model will be a valuable contribution to the ongoing research and analysis in the field of particle detection and identification.

Furthermore, the development of the graph-based model will add to the existing body of research in particle physics and contribute to the advancement of the field. The application of GNNs for particle identification has been limited in the past, but this project aims to demonstrate the potential of graph-based methods in solving complex classification problems. The results from this project will also provide insights into the effectiveness of GNNs for other particle identification tasks in the CMS experiment and potentially other experiments as well.

In conclusion, we are confident that our proposed graph-based approach will bring significant advancements to the field of particle identification. With the combination of our expertise in deep learning, graph neural networks, and particle physics, we believe that we have the necessary skills and knowledge to successfully complete this project. Our proposed approach has the potential to deliver a powerful and effective tool for particle identification, and we look forward to contributing to the ongoing research in this field.

We would like to express our sincere gratitude to the mentors for providing us with this exciting opportunity. We are eager to work with you and contribute to the CMS experiment in a meaningful way. We believe that our passion for machine learning and particle physics, combined with our technical skills, will make us valuable contributors to the project. Thank you for considering our application. We look forward to the opportunity to discuss our proposal further.

9 Timeline (350 hours)

This timeline outlines the project's milestones and tasks, beginning with understanding relevant technologies and designing the GNN model. Next, the focus is on implementation, testing, and refinement, followed by optimization and integration with the CMS experiment's systems. The final stages involve validation, documentation, and addressing any unforeseen challenges.

9.1 Overview

The project timeline consists of the following key phases:

Before May 4: Understand relevant technologies and review documentation.

May 4 - May 28: Design GNN model and establish project milestones.

May 29 - July 9: Implement, test, and refine the GNN model.

July 10 - July 14: Midterm evaluations.

July 14 - September 4: Optimize GNN inference, integrate model, and ensure compatibility.

September 5 - October 1: Validate the solution, resolve issues, and prepare documentation.

October 2 - November 5: Address unforeseen challenges, if needed.

9.2 Milestones

This detailed timeline provides a comprehensive breakdown of the project's milestones and tasks, which are essential for a successful completion. Each milestone represents a crucial phase in the project, including understanding relevant technologies, designing the GNN model, implementation, testing, and refinement.

Further milestones cover optimization, integration with the CMS experiment's systems, and final validation. The last stages ensure that thorough documentation is provided, and any unforeseen challenges or issues are addressed effectively.

Before May 4:

- Gain familiarity with the CMS experiment, E2E project, graph neural networks, and PyTorch.
- Review the relevant papers and documentation in the project description.

May 4 - May 28 (Community Bonding Period):

- Design the end-to-end GNN model for tau identification.
- Discuss the project plan and milestones with mentors and the organization.

May 29 - July 9 (Weeks 1-6, 175 hours):

- Implement the end-to-end GNN model for low-momentum tau identification.
- Test the initial model, gather performance metrics, and refine it.

July 10 - July 14 (Midterm evaluation):

- Submit midterm evaluations.

July 14 - September 4 (Weeks 7-14, 140 hours):

- Benchmark and optimize the end-to-end GNN inference on GPU.
- Compare the optimized model's performance to the original one.
- Integrate the model into the CMSSW inference engine.
- Test the integration and ensure compatibility with CMS experiment's systems.

September 5 - October 1 (Weeks 15-18, 30 hours):

- Perform final testing and validation of the integrated solution.
- Resolve any remaining issues and prepare documentation for ease of use.

October 2 - November 5 (Weeks 19-22, 5 hours):

- This period will be left unused unless unforeseen challenges or delays arise that require additional time to address. If needed, adjustments and refinements to the project will be made during this time.

10 Risks and Mitigation Strategies

The following are the identified risks associated with the project and their corresponding mitigation strategies:

1. **Unfamiliarity with the CMS experiment and related technologies.**
   Mitigation: Proactively review relevant papers, documentation, and resources during the preparation and community bonding period. Communicate with mentors and the organization to clarify any doubts or questions.

2. **Difficulty in designing and implementing the GNN model.**
   Mitigation: Consult mentors, research papers, and relevant resources for guidance. Break down the model design and implementation into smaller, manageable tasks to simplify the process.

3. **Model underperforms or does not meet expectations.**
   Mitigation: Continuously evaluate and refine the GNN model based on performance metrics and feedback from mentors. Explore alternative approaches and techniques if needed.

4. **Challenges in optimizing the GNN model for GPU inference.**
   Mitigation: Research GPU optimization techniques and seek advice from mentors or the community. Perform iterative optimization and benchmarking to achieve the desired performance improvements.

5. **Integration issues with the CMSSW inference engine and the CMS experiment's systems.**
   Mitigation: Follow best practices for integration and work closely with mentors and the CMS experiment team to address any compatibility or integration issues that arise.

6. **Unforeseen challenges or delays affecting the project timeline.**
   Mitigation: Maintain regular communication with mentors to promptly identify and address any issues. Adjust the project scope or timeline as needed, in consultation with mentors and the organization.

11 Task Discussion

11.1 Task 1

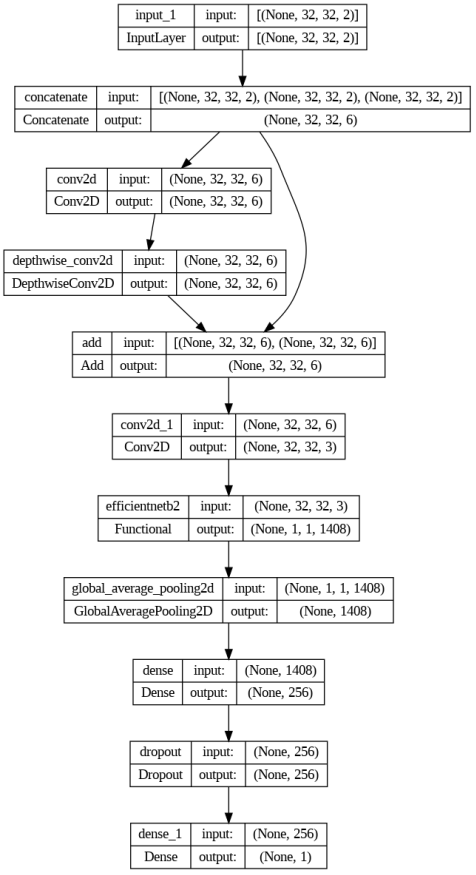


Figure 1: CNN Model layers scheme for Task 1 to classify electron/photon.

In the proposed project for Google Summer of Code, we aim to develop a deep learning model to accurately classify particles as either electrons or photons based on their interactions with a calorimeter. The input data is represented by 32x32 matrices with two channels, representing hit energy and time. To address this challenge, we propose a hybrid model architecture that combines a residual block, channel reduction, and the EfficientNetB2 model. We proposed 2 models using PyTorch and TensorFlow, where each of the model has a similiar architecture.

Preprocessing: The preprocessing stage involves two primary steps: normalization and quantization. Normalization is essential in preparing the data to be fed into the neural network, as it ensures that all input features are on a similar scale. This facilitates the learning process and can help improve convergence speed. In this case, the data is normalized by dividing each element by the maximum value along the sample, width, and height dimensions, keeping the channel dimension intact. After normalization, quantization is applied to re-

duce the data's precision to 8-bit unsigned integers. This step can help reduce memory consumption and computation time without significantly affecting the model's performance.

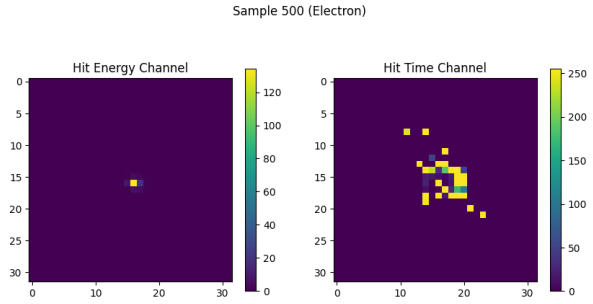


Figure 2: Data visualization for hit energy and time, in interaction with calorimeter on sample classified as electron.

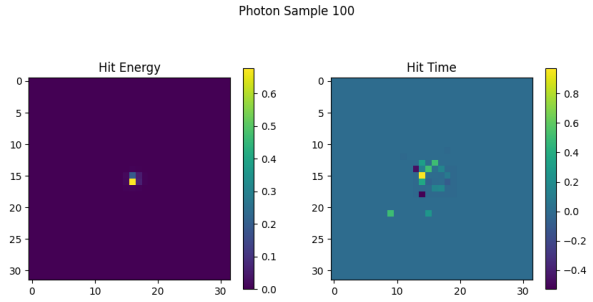


Figure 3: Data visualization for hit energy and time, in interaction with calorimeter on sample classified as photon.

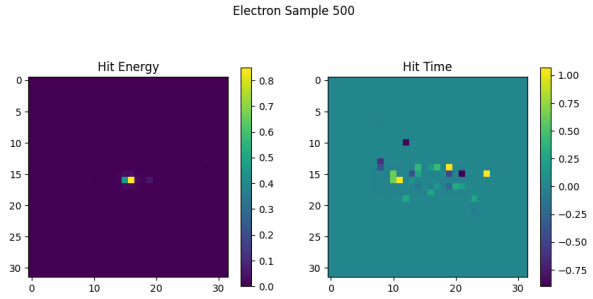


Figure 4: Pixels after being normalized and quantized for sample classified as electron.

Model Architecture: The proposed hybrid model architecture consists of the following components:

1. **Residual block**
   This block is composed of a standard 3x3 convolutional layer followed by a depthwise 3x3 convolutional layer. The output is added back to the input to create the final output of the residual block. Residual blocks are known for their ability to mitigate the vanishing gradient problem, which can occur in deep networks, and improve the overall learning capability of the model[3]. In the context of particle physics, this allows the model to learn more complex patterns and capture the subtle differences between electron and photon interactions with the calorimeter.

2. **Channel reduction**
   Reduces the number of channels to 3 using a 1x1 convolutional layer.

3. **EfficientNetB2**
   A pre-trained model with weights initialized from the ImageNet dataset. EfficientNetB2 is chosen due to its superior performance and efficiency compared to other popular architectures[4]. This choice allows us to take advantage of transfer learning and improve the model's performance in the electron/photon classification task, even though the original pre-training data (ImageNet) is unrelated to particle physics. The model is fine-tuned by setting it to be trainable during the training process, allowing it to adapt to the specific characteristics of the data.

4. **Global Average Pooling layer**
   Applied to the output of the EfficientNetB2. This layer helps reduce the number of parameters and computations, making the model more efficient and less prone to overfitting.

5. **Dense layer**
   256 units with ReLU activation and L2 regularization. This layer serves as a fully connected layer that combines the features learned by the previous layers to make high-level decisions. The L2 regularization is used to prevent overfitting by penalizing large weights.

Training and Optimization: The model is compiled with the binary_crossentropy loss function and the SGD optimizer, which is configured with a learning rate of 0.01 and a momentum of 0.6. These hyperparameters can be further fine-tuned to optimize the model's performance.

Performance Evaluation and Future Improvements: In our initial attempt, the model achieved a ROC-AUC score of 0.63, which falls short of the desired threshold of 0.80. It is crucial to acknowledge that the model's performance is influenced by various factors, including the size of the training data, model complexity, and architectural choices. We trained the model on a limited portion of the available data, which could potentially restrict its ability to learn complex patterns and distinguish between electrons and photons effectively.

Despite this, our model serves as a starting point for the electron/photon classification task and provides a foundation for future improvements. Here are some potential enhancements to boost the model's performance:

- **Data augmentation**
  Implement random transformations (e.g., rotations, flips, and translations) to augment the training data, increasing the diversity of samples during training and enhancing the model's generalization capabilities.

- **Larger training dataset**
  Utilize the entire dataset or a more substantial portion of the data during training, providing the model with more examples to learn from and capture complex patterns, ultimately improving its classification performance.

- **Hyperparameter optimization**
  Conduct a systematic search for optimal hyperparameters (e.g., learning rate, batch size, number of layers, and regularization strength) using techniques such as grid search or Bayesian optimization to fine-tune the model and enhance its performance.

- **Alternative model architectures**
  Investigate different model architectures, such as deeper or wider networks, attention mechanisms, or other state-of-the-art architectures that have demonstrated success in similar tasks.

- **Ensemble methods:**
  Employ ensemble techniques like bagging, boosting, or stacking to combine the predictions of multiple models, capturing a diverse set of patterns and reducing the likelihood of overfitting, which can lead to improved performance.

- **Transfer learning**
  Explore other pre-trained models that may be more suitable for this task or consider using models pre-trained on similar datasets (if available) to enhance the model's performance.

By emphasizing that our model is a starting point and outlining these potential improvements, we demonstrate that the current proposal serves as a solid foundation for further development and refinement in the electron/photon classification task. This approach highlights our commitment to continuous improvement and adaptation to achieve the desired classification performance.

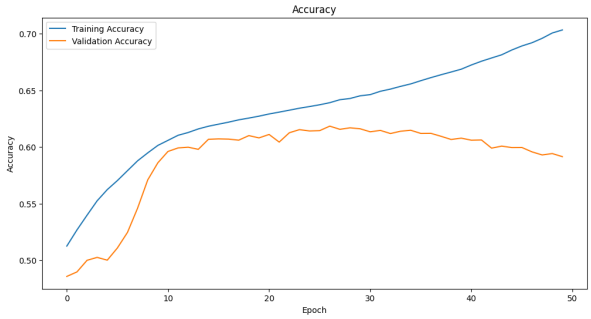


Figure 5: Accuracy plot on 55 epochs for model (TensorFlow) trained on batches of data.

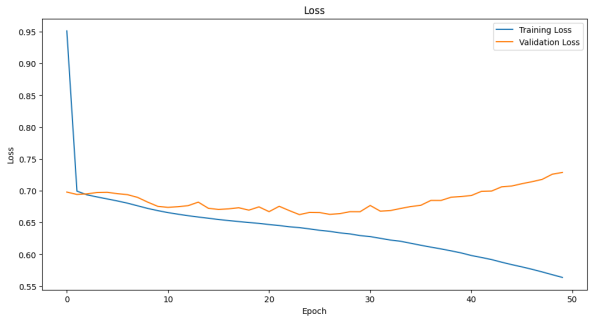


Figure 6: Loss plot on 55 epochs for mode l(TensorFlow) trained on batches of data.

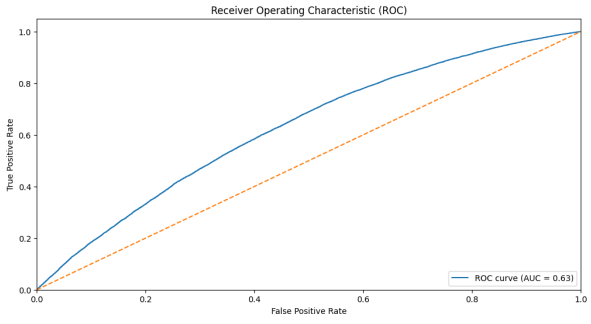


Figure 7: ROC-AUC cuve for task 1 model (TensorFlow), showing value of 0.63.

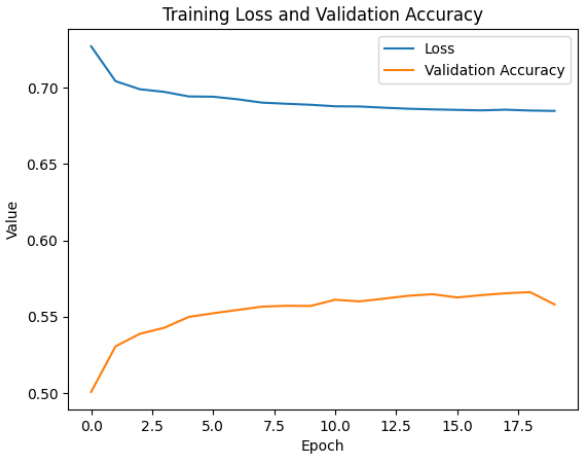


Figure 8: Accuracy plot on 55 epochs for model (PyTorch) trained on batches of data.

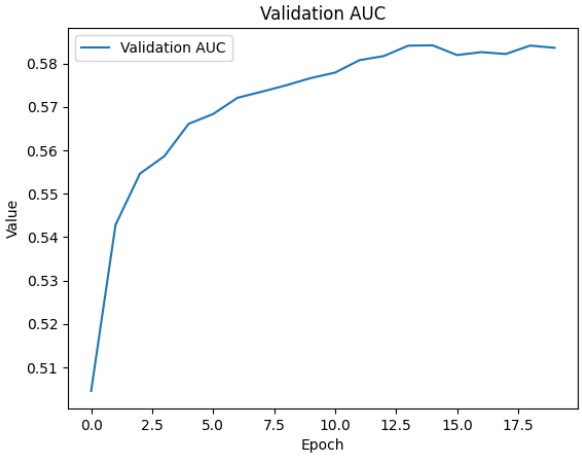


Figure 9: Loss plot on 55 epochs for model (PyTorch) trained on batches of data.

11.1.1 Summary

We aimed to develop a deep learning model for classifying electrons and photons using 32x32 matrices as input data. The model combined a custom residual block with an EfficientNetB2 backbone, utilizing data generators and an RMSprop optimizer during training. Our initial model achieved a ROC-AUC score of 0.63, which was below the desired threshold of 0.80. This performance can be attributed to factors like limited training data and model complexity. However, our model serves as a foundation for future improvements, such as data augmentation, hyperparameter optimization, and alternative architectures. We believe continuous refinement will lead to achieving the desired performance in future iterations.

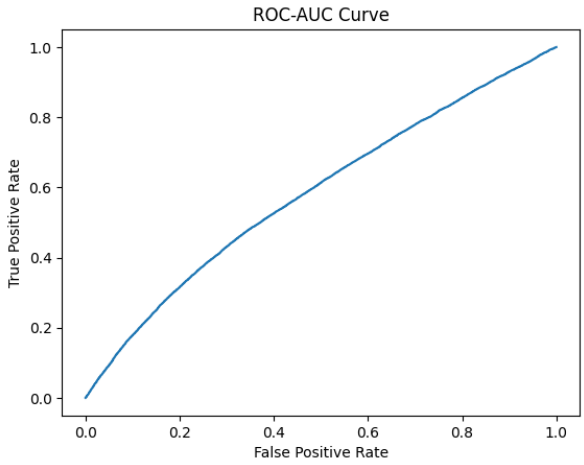


Figure 10: ROC-AUC cuve for task 1 model (PyTorch), showing value of 0.63.

11.2 Task 2

In the provided model visualization, we have designed a deep learning model that combines a Convolutional Neural Network (CNN) for processing the X_jets data (125x125x3 matrices) and a Fully Connected Network (FCN) for processing the additional features, mean transverse momentum (pt) and mass (m0). The goal is to create a robust model that can effectively classify the given particle data into quarks and gluons.

The proposed deep learning model aims to classify particles as quarks or gluons using the information provided in the dataset, which includes the X_jets data (125x125x3 matrices) and additional features, such as mean transverse momentum (pt) and mass (m0). Quarks and gluons are fundamental particles and understanding their interactions is essential to the study of the strong force, one of the four fundamental forces in nature, and Quantum Chromodynamics (QCD), the underlying theory of strong interactions.

The X_jets data represent the energy deposition pattern of particles impinging on a calorimeter, which is an essential detector component in high-energy physics experiments like those conducted at CERN's Large Hadron Collider (LHC). The energy deposition patterns differ between quarks and gluons, providing valuable information for classification.

The additional features, pt and m0, represent the mean transverse momentum and mass of the particles, respectively. These features provide complementary information to the energy deposition patterns for distinguishing between quarks and gluons. In particle physics, different particles exhibit unique combinations of mass and momentum, which help researchers identify the type of particle and understand its properties and behavior.

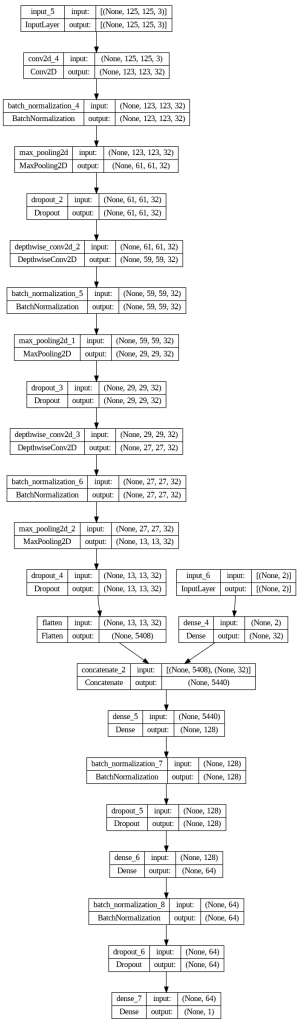


Figure 11: CNN Model layers scheme for Task 2 to classify quark/gluon.

By combining the information from the X_jets data and the pt and m0 features, the proposed model architecture can learn to identify the subtle differences in the energy deposition patterns and the characteristic mass and momentum values specific to quarks and gluons. This approach enables the model to capture the complex interactions and behavior of these fundamental particles, which can aid in the development of more accurate theoretical models and contribute to our understanding of particle physics and the strong force.

The successful classification of quarks and gluons using this deep learning model can also have practical implications for high-energy physics experiments, as it can help researchers improve the efficiency of particle identification and event reconstruction in detectors. This, in turn, can lead to more accurate measurements and a deeper understanding of the fundamental processes in nature.

Preprocessing: Before feeding the data into the model, we first normalize the pt and m0 features using the StandardScaler from the scikit-learn library. This step ensures that the features are on the same scale, which is essential for improving the model's performance and training stability. We then split the dataset into training and testing sets (80% for training and 20% for testing) using the train_test_split function.

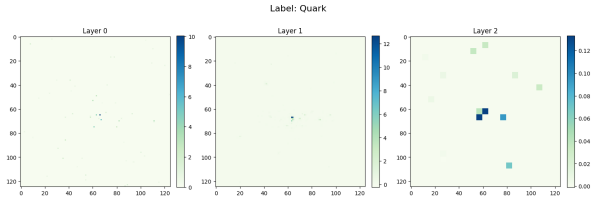


Figure 12: Data visualization of X_jets consisting of 125x125 pixels for 3 channel in which classified as a quark.

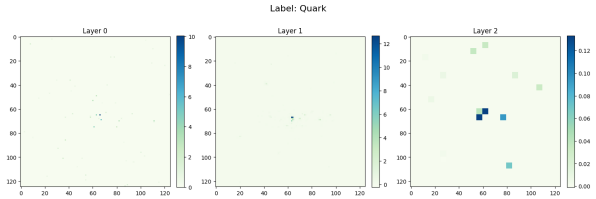


Figure 13: Data visualization of X_jets consisting of 125x125 pixels for 3 channel in which classified as a gluon.

Model Architecture: The model consists of two parts: a CNN for processing the X_jets data and an FCN for processing the pt and m0 features.

1. **CNN for X_jets**
   The CNN processes the input X_jets data through a series of Conv2D, BatchNormalization, MaxPooling2D, and Dropout layers. It uses the GeLU activation function, which is known to improve convergence and model performance. Additionally, we employ DepthwiseConv2D layers, which reduce the number of trainable parameters and computational complexity while maintaining the model's ability to learn meaningful features.

2. **FCN for pt and m0**
   The FCN processes the normalized pt and m0 features with a Dense layer using the GeLU activation function.

3. **Merging CNN and FCN**
   We then concatenate the outputs of the CNN and FCN, followed by a series of Dense, BatchNormalization, and Dropout layers. The final output layer has one neuron with a sigmoid activation function, which classifies the input as either quark (0) or gluon (1).

Training and Optimization: The model is compiled with the binary_crossentropy loss function and the SGD optimizer, which is configured with a learning rate of 0.01 and a momentum of 0.6. These hyperparameters can be further fine-tuned to optimize the model's performance.

Performance Evaluation and Future Improvements: The proposed model achieved an ROC-AUC score of 0.72, which indicates that it has moderate classification performance. While the model is capable of distinguishing between quarks and gluons to some extent, there is room for improvement to achieve higher accuracy and robustness. There are several factors that may have contributed to the current performance, including:

- **Limited training data**
  Due to the large size of the dataset, we only trained the model on a portion of the available data. This might have hindered the model's ability to learn more complex and discriminative features. Training the model on the entire dataset or a larger portion of it could lead to better performance.
- **Model architecture and hyperparameters**
  The current model architecture and hyperparameters might not be optimal for the problem at hand. Experimenting with alternative architectures, such as deeper networks, different types of layers, or a combination of multiple models (e.g., ensemble methods), could improve the model's performance. Additionally, hyperparameter optimization techniques like grid search, random search, or Bayesian optimization can be employed to fine-tune the model's configuration.
- **Data preprocessing and augmentation**
  Exploring different data preprocessing techniques, such as normalization, whitening, or feature extraction, could help the model learn more meaningful representations. Furthermore, data augmentation techniques like random rotations, translations, and flips can be used to artificially expand the dataset, increasing its diversity and enabling the model to generalize better.
- **Regularization and model complexity**
  The model might suffer from overfitting or underfitting, which can be addressed by adjusting the regularization strength (e.g., L1 or L2 regularization) or by changing the model's capacity. For example, reducing the model's complexity could help prevent overfitting, while increasing its capacity might help address underfitting.
- **Transfer learning**
  Leveraging pre-trained models from related domains can provide a good starting point for training, as the model can fine-tune the learned features to the specific quark-gluon classification task. This approach can significantly improve the model's performance and reduce training time.

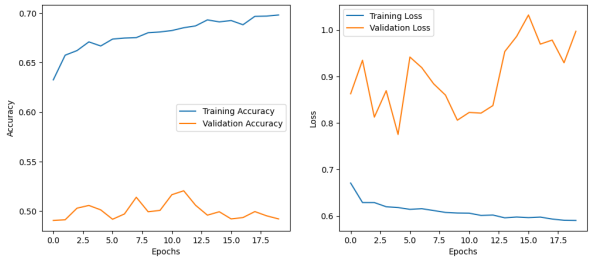


Figure 14: Accuracy and Loss plot in using batches of jet0_run0 data for Task 2 to classify quark/gluon.

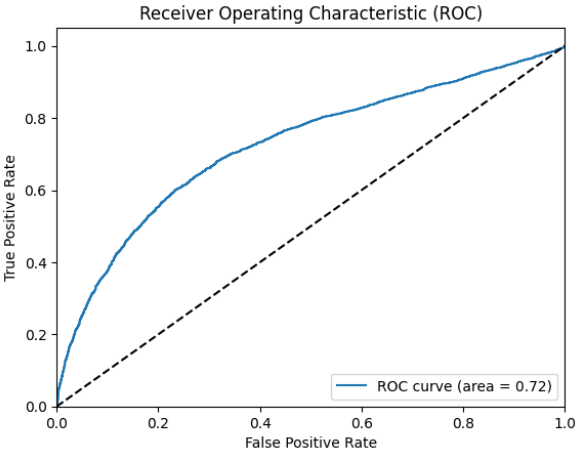


Figure 15: ROC-AUC cuve for Task 2 model (jet0_run0), showing value of 0.72

11.2.1 Summary

In this task, we have developed a deep learning-based solution for the Quark-Gluon Classification problem using a combination of Convolutional Neural Networks (CNNs) and Fully Connected Networks (FCNs). Our model processes the X_jets data (125x125x3 matrices) through a CNN, while an FCN is used to process additional features, mean transverse momentum (pt) and mass (m0). The model has been trained on a portion of the provided dataset, considering its large size, and achieved an ROC-AUC score of 0.72, indicating moderate classification performance.

There is room for improvement in the model's performance, which can be addressed through various strategies such as utilizing the entire dataset or a larger portion for training, experimenting

with alternative model architectures and hyperparameters, exploring different data preprocessing techniques and augmentation, adjusting regularization strength and model complexity, and leveraging transfer learning. By implementing these improvements, we aim to achieve a more accurate and robust quark-gluon classification system, which will significantly contribute to the ongoing research and analysis in the field of particle detection system.

11.3 Additional Task

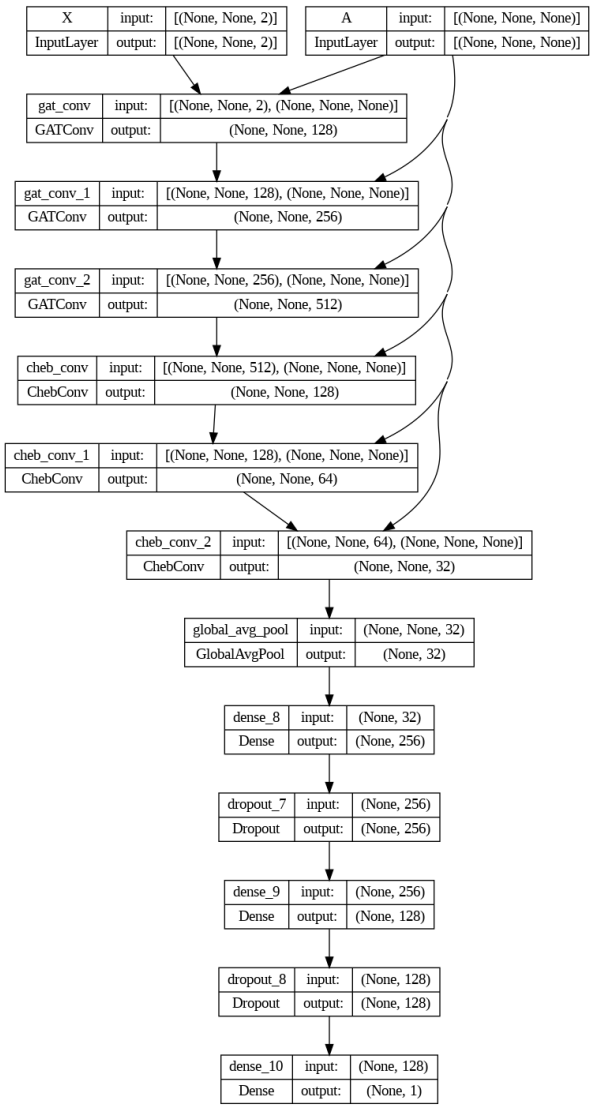


Figure 16: Graph architecture model layers scheme for Additional Task to classify quark/gluon.

In particle physics, understanding the behavior of elementary particles, such as quarks and gluons, is crucial for studying the fundamental forces and structure of the universe. Jets are the observable manifestations of quarks and gluons produced in high-energy collisions [7], such as those occurring at the Large Hadron Collider (LHC). Accurate classification of jets as quarks or gluons helps physicists better interpret experimental data and investigate the underlying processes that give rise to these particle signatures.

A major challenge in classifying jets lies in the complex and high-dimensional nature of the data generated by detectors such as calorimeters. Traditional techniques may struggle to capture the intricate spatial relationships and dependencies between the different features, necessitating the use of advanced machine learning methods. Graph-based

deep learning models offer a promising solution, as they can naturally represent and process data with irregular structures and learn complex patterns in the input features.

In this task, the JetClassifier model is specifically designed to address the unique challenges posed by particle physics data. By converting the point-cloud dataset into a set of interconnected nodes and edges using k-Nearest Neighbors and Delaunay triangulation, we capture the spatial relationships between the points in a manner that is suitable for graph-based learning. This allows the model to effectively learn the distinguishing characteristics of quarks and gluons while preserving the spatial information present in the data.

The use of graph attention layers (GATConv) in the model allows the network to focus on relevant neighboring nodes during the learning process [5]. This is particularly important in particle physics, as certain features may be more informative for classification than others, depending on their spatial context. Additionally, the use of ChebConv layers helps capture the spectral properties of the graph, which can provide valuable insights into the underlying structure of the data[6] and further improve the model's classification performance.

Preprocessing: In the preprocessing stage, the raw data from the calorimeters is first converted into a structured graph representation. The input features, 'X_jets', 'pt', and 'm0', represent the jet images, mean transverse momentum, and mass, respectively. To create the graph, the point-cloud dataset is projected onto a set of interconnected nodes and edges. The nodes are created using the Laplacian and Hessian features from the input data, while the edges are determined by employing k-Nearest Neighbors (k-NN) and Delaunay triangulation techniques. This approach preserves the spatial information in the data and allows the graph-based model to effectively learn the distinguishing characteristics of quarks and gluons. The mean transverse momentum and mass are added as graph-level attributes to provide global context to the model.

Model Architecture: The JetClassifier model is built using the Keras functional API and consists of the following layers:

1. **GATConv layers (Graph Attention Networks)**
   Three GATConv layers with attention mechanisms are stacked in the beginning, which allow the model to selectively focus on relevant neighboring nodes during the learning process. The number of attention heads and output channels are gradually increased in each layer (32, 64, and 128) to capture increasingly complex patterns.

2. **ChebConv layers (Chebyshev Spectral Graph Convolution)**
   After the GATConv layers, three ChebConv layers are added to capture the spectral properties of the graph. The number of output channels is progressively decreased in each layer (128, 64, and 32), which helps learn a hierarchical representation of the graph.

3. **Global Average Pooling layer**
   This layer is used to aggregate the node features into a single graph-level representation by computing the average of the node features. This allows the model to focus on global properties of the graph for the classification task.

4. **FCN**
   A series of fully connected layers (Dense) with varying numbers of neurons (256 and 128) are added after the GlobalAvgPool layer. These layers help the model learn non-linear combinations of the aggregated features and further refine the learned representations.

Training and Optimization: The model is trained using preprocessed graph data split into training and testing sets, with custom dataset and batch loader classes implemented to handle the graph data structure and provide a seamless interface for training and testing. The model is compiled with the RMSprop optimizer for adaptive learning rate updates, BinaryCrossentropy loss function for binary classification, and an accuracy metric to monitor performance. The custom batch loaders ensure the model receives appropriate input data and labels during training, while the choice of optimizer, loss function, and metric provide a tailored solution for the quark-gluon classification problem.

Performance Evaluation and Future Improvements: The obtained ROC-AUC score of 0.59 indicates that the current model's performance in classifying jets as quarks or gluons is relatively modest. While this may not be as high as desired, there are several potential reasons for the current performance, as well as suggestions for future improvements to achieve better results. There are several factors that may have contributed to the current performance, including:

- **Limited training data**
  To address this, we plan to use data streaming, distributed or parallel training methods for processing the entire dataset.

- **Model architecture**
  We will explore alternative graph neural network architectures, such as GraphSAGE, GCNs, or GINs, and adjust hyperparameters to find a more suitable model.

- **Feature engineering**
  The model's performance could be affected by

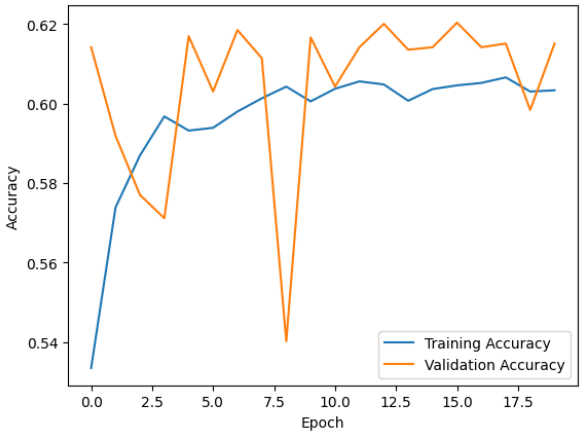


Figure 17: Accuracy plot in using batches of jet0_run0 data for Task 3 to classify quark/gluon.

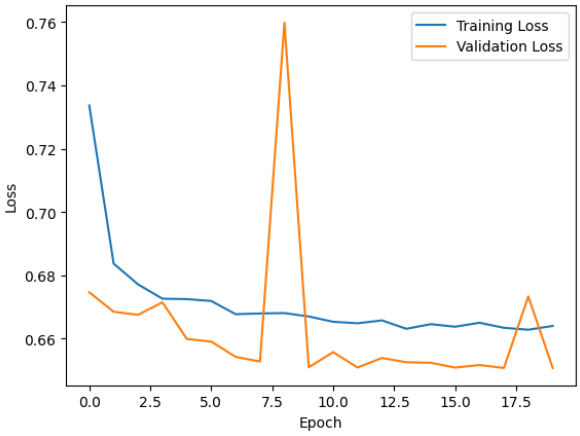


Figure 18: Loss plot in using batches of jet0_run0 data for Task 3 to classify quark/gluon.

the choice of features used to create the graph representation. We will investigate the possibility of incorporating additional features or refining the existing ones to better capture the characteristics of quarks and gluons. Furthermore, we will employ domain knowledge in particle physics to identify and engineer more informative features.

- **Transfer learning**
  If a suitable pre-trained model is available, we plan to leverage transfer learning techniques to improve the model's performance. Fine-tuning a pre-trained model on the specific classification task can help reduce training time and potentially lead to better results.

- **Model ensembling**
  Combining the predictions of multiple models can often result in improved performance. We will train several different models and aggregate their outputs to capitalize on their individual strengths and potentially achieve a higher ROC-AUC score.

11.3.1 Summary

In this task, we proposed a graph-based approach using Graph Neural Networks (GNNs) to classify jets as quarks or gluons. The model was designed with a combination of Graph Attention Networks (GAT) and Chebyshev layers to process the point-cloud dataset transformed into interconnected nodes and edges. Despite the modest performance of the current model, with a ROC-AUC score of 0.59, several promising strategies for future improvements were identified, including efficient data handling, alternative graph-based architectures, feature engineering, transfer learning, and model ensembling.

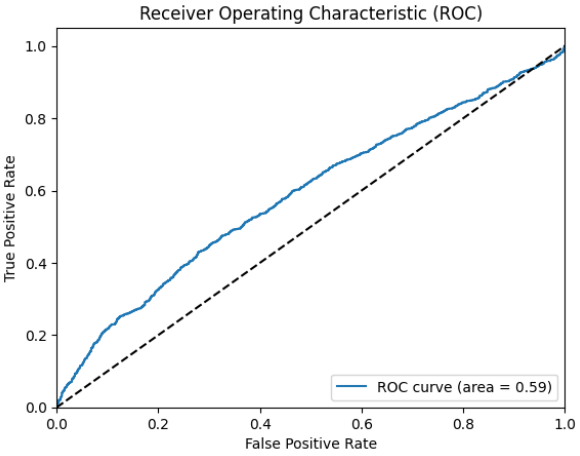


Figure 19: ROC-AUC cuve for Task 3 model, showing value of 0.59.

The graph-based model is expected to outperform the previously used CNN model on the same dataset and classification task when these improvements are implemented. By leveraging domain knowledge in particle physics and employing advanced techniques, we aim to create a more effective and accurate model for jet classification. The successful application of GNNs to this problem will further demonstrate the potential of graph-based methods in solving complex classification tasks in quark-gluon classification problem.

References

[1] M. Andrews, M. Paulini, S. Gleyzer, and B. Poczos. End-to-End Physics Event Classification with CMS Open Data: Applying Image-Based Deep Learning to Detector Data for the Direct Classification of Collision Events at the LHC. *Computing and Software for Big Science*, 4(1), 2020.

[2] M. Andrews, J. Alison, S. An, B. Burkle, S. Gleyzer, M. Narain, M. Paulini, B. Poczos, and E. Usai. End-to-end jet classification of quarks and gluons with the CMS Open Data. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 977, 2020.

[3] Jastrzebski, Stanisław and Arpit, Devansh and Ballas, Nicolas and Verma, Vikas and Che, Tong and Bengio, Yoshua. Residual connections encourage iterative inference. *arXiv preprint arXiv:1710.04773*, 2017.

[4] Li, Hejie and Tan, Ying and Miao, Jiaqing and Liang, Ping and Gong, Jinnan and He, Hui and Jiao, Yuhong and Zhang, Fan and Xing, Yaolin and Wu, Donghan. Attention-based and micro designed EfficientNetB2 for diagnosis of Alzheimer's disease. *Biomedical Signal Processing and Control*, 82, 2023, 104571. Publisher: Elsevier.

[5] Velickovic, Petar and Cucurull, Guillem and Casanova, Arantxa and Romero, Adriana and Lio, Pietro and Bengio, Yoshua and others. Graph attention networks. *stat*, 1050, 20, 2017, 10–48 550.

[6] Mailloux, R. Synthesis of spatial filters with Chebyshev characteristics. *IEEE Transactions on Antennas and Propagation*, 24, 2, 1976, 174–181. Publisher: IEEE.

[7] Marshall, Robin. "Jets: the materialization of quarks and gluons." Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences 404.1827 (1986): 167-188.