SUMMER STUDENT PROGRAMME 2025

รายงานการเข้าร่วมโครงการนักศึกษาภาคฤดูร้อนเซิร์น
ระหว่างวันที่ 8 มิถุนายน - 31 สิงหาคม 2568
ณ เซิร์น กรุงเจนีวา สมาพันธรัฐสวิส

นายสิทธิพล คำดา

วิศวกรรมหุ่นยนต์และปัญญาประดิษฐ์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

Acknowledgements

I wish to express my deepest gratitude to Her Royal Highness Princess Maha Chakri Sirindhorn for the distinct honor of being selected to participate in the prestigious 2025 CERN Summer Student program.

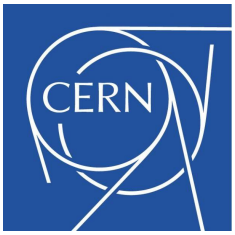
My sincere thanks are extended to the Information Technology Foundation, under the Initiative of Her Royal Highness Princess Maha Chakri Sirindhorn, and to Dr. Norraphat Srimanobhas for their crucial roles in organizing and facilitating this collaboration with CERN. This invaluable experience would not have been possible without their support.

I am profoundly grateful to my supervisors, Doga Elitez, Jonathan Niklas Renusch, and Dr. Paul Gessinger-Befurt. Their mentorship and invaluable guidance were instrumental in deepening my understanding of the academic research process, computer science, and AI engineering, and in sharpening my professional skills. I also extend my thanks to the ATLAS experiment and the Experimental Physics Department at CERN for providing consistent support and a stimulating and welcoming research environment.

This work was made possible by the generous funding and organization provided by the Synchrotron Light Research Institute, and the Program Management Unit for Human Resources & Institutional Development, Research and Innovation (PMU-B). I am thankful for this prestigious program, which offered not just an opportunity to study science and engineering, but the invaluable experience of contributing to a world-class research laboratory. It has truly been, as CERN states, an opportunity "in a place like nowhere else on Earth."

Finally, I extend my heartfelt thanks to my family and friends for their unwavering support throughout my education and my life.

Contents



CERN Summer Student Report
August 2025
sittiponkumda@gmail.com

Secondary Vertex Reconstruction at The High-Luminosity LHC with MaskFormer Architecture

Author: Sittipon Kumda
Supervisors: Doga Elitez, Jonathan Niklas Renusch,
Dr. Paul Gessinger-Befurt
CERN, CH-1211 Geneva, Switzerland

Keywords: Secondary Vertex Reconstruction, High-Luminosity LHC, MaskFormer, Machine Learning

Abstract

With the upcoming era of the High-Luminosity LHC, secondary vertex reconstruction will face significant computational challenges in dense environment, highlighting the need for novel and efficient identification algorithms. This report introduces a new approach, adapting the transformer-based MaskFormer architecture from computer vision to reframe SV finding as an end-to-end instance segmentation problem. A dedicated data processing pipeline was developed to convert simulated ROOT data into a uniform HDF5 format suitable for a multi-task model designed to simultaneously perform SV classification, track assignment (masking), and vertex property regression. The model, trained on a realistic, class-imbalanced dataset, demonstrated excellent performance on the classification and track assignment tasks, achieving results comparable to a baseline trained on a perfectly balanced reference sample. However, the vertex regression task proved highly sensitive to this imbalance, showing significant underperformance compared to the reference. These findings establish the query-based transformer architecture as a powerful and viable new direction for tackling the combinatorial challenge of vertex reconstruction. The results also highlight that achieving precise regression in a multi-task setting on imbalanced data is a key challenge, suggesting that future work must prioritize the optimization of the loss function weights and target scaling strategies to unlock the full potential of this approach.

1 Introduction

1.1 Background

The reconstruction of secondary vertices (SVs) is an essential task in high-energy physics experiments at CERN. These displaced vertices are key signatures in a wide range of physics analyses, from studies within the Standard Model to searches for long-lived particles in Beyond the Standard Model theories. While numerous established algorithms exist for SV reconstruction, the upcoming High-Luminosity Large Hadron Collider (HL-LHC) era presents a significant new challenge. The expected increase in the number of simultaneous proton-proton collisions, a phenomenon known as pileup, will drastically increase the combinatorial complexity of the detector environment, demanding more robust and efficient reconstruction techniques.

In response to such challenges, Machine Learning has emerged as a powerful paradigm. A particularly promising approach is the use of transformer-based architectures like MaskFormer[1]. Originally designed for the computer vision task of instance segmentation, MaskFormer excels at identifying complex, non-linear correlations in high-dimensional data by learning to group related inputs into distinct object instances. This inherent capability for grouping and pattern recognition makes the architecture exceptionally well-suited for secondary vertex (SV) reconstruction, a task that lacks a simple, deterministic solution and instead relies on interpreting intricate patterns from detector inputs such as particle tracks and jets.

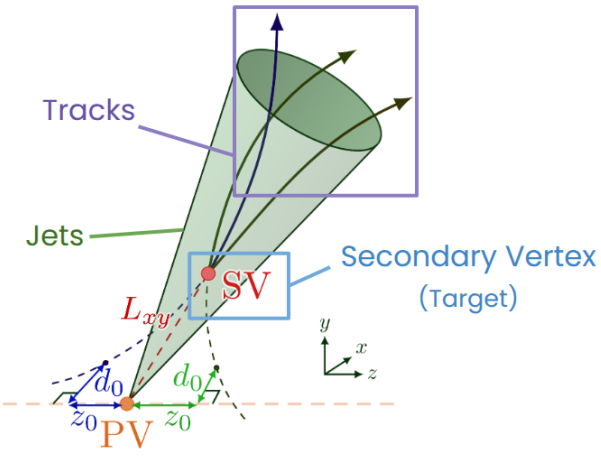


Figure 1: The visualization of track and secondary vertex inside a jet

1.2 Objectives

The ultimate goal of this project is to develop and validate a machine learning pipeline for SV reconstruction, tailored to the high pile-up environment of the HL-LHC. The goal is for this algorithm to effectively process track and jet data to achieve high performance in SV finding and fitting.

However, as official HL-LHC simulation datasets are still in development, the work presented in this report constitutes a foundational proof-of-concept. The dataset employed is a preliminary simulation designed to approximate the expected detector conditions. Therefore, the immediate objectives of this report are to:

- Establish a complete, end-to-end machine learning pipeline, from data pre-processing to model evaluation.
- Demonstrate the functionality of the pipeline on the available dataset.
- Conduct a preliminary evaluation of the model's performance to confirm its potential for more advanced studies with future datasets.

1.3 Pipeline Overview

The end-to-end machine learning pipeline developed for this study comprises three principal stages: data format conversion and pre-processing, model training, and performance evaluation. Each stage is designed to be a modular component, facilitating independent development and testing. The overall workflow is illustrated in Figure 2.

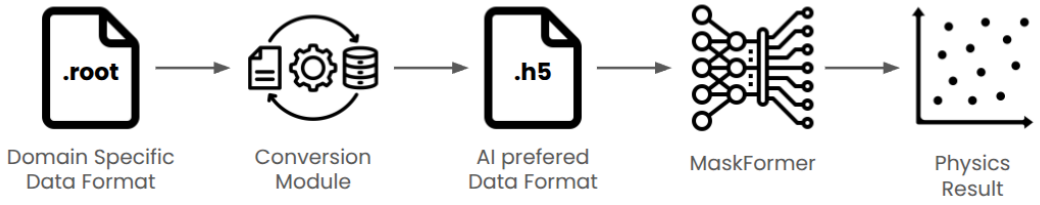


Figure 2: Schematic of the end-to-end machine learning pipeline, detailing the workflow from initial data conversion to final performance evaluation.

2 Adaptation of MaskFormer

2.1 MaskFormer Architecture

The original MaskFormer architecture, introduced by Cheng et al. in 2021[1], consists of three main modules: a pixel decoder, a transformer decoder, and segmentation heads. A schematic of this architecture is shown in Figure 3.

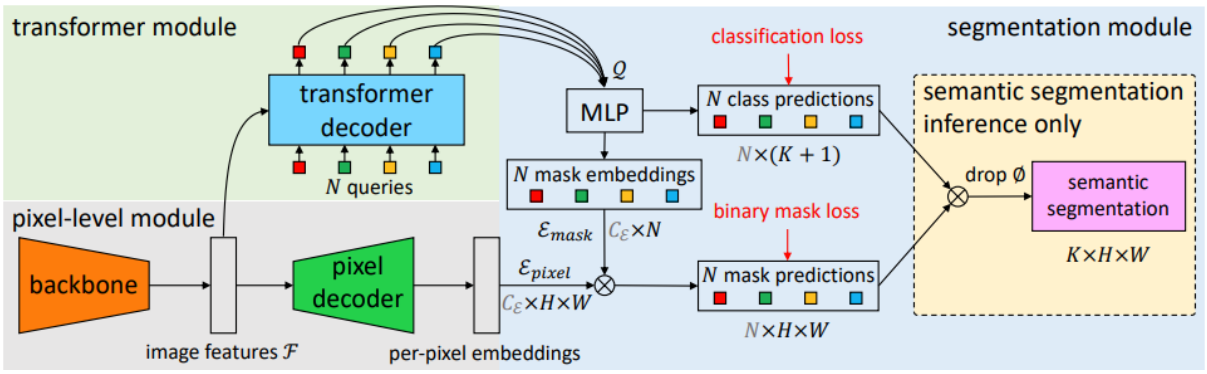


Figure 3: Overview of the original MaskFormer architecture.

The role of each module is described below:

Feature Embedding This module is a CNN-based feature extractor, often using a bottleneck design. A backbone network first extracts a low-resolution feature map

from the input image. This feature map is then fed to the transformer decoder, while a separate network generates high-resolution per-pixel embeddings used for the final mask prediction.

Transformer Decoder This is a standard transformer decoder module that processes a set of N learnable embeddings, known as object queries. It performs cross-attention between these queries and the image feature map from the pixel decoder. This process yields a set of N per-query embeddings, which serve as the final representations for potential objects or segments.

Segmentation Heads This final stage consists of two separate feed-forward networks (MLPs). One network takes the per-query embeddings to perform a classification, predicting the class label for each of the N queries. The second network is used to generate the final binary masks from the per-pixel embeddings.

2.2 Proposed Architecture for SV Reconstruction

In the context of secondary vertex reconstruction, two primary tasks must be accomplished. The first, *SV finding*, involves assigning constituent tracks to their originating secondary vertex within a jet. The second, *SV fitting*, uses these assigned tracks to calculate the properties of the vertex. These physics tasks can be framed as distinct machine learning problems: SV finding is analogous to an instance segmentation task, where each "instance" is an SV and the "pixels" are the tracks. SV fitting, in contrast, is a classic regression task. This framing makes the MaskFormer architecture, which excels at segmentation, a highly advantageous starting point for this work.

The proposed architecture, a novel adaptation of MaskFormer introduced by Samuel Van Stroud (2024)[2], is designed to perform these tasks simultaneously. A schematic of this model is shown in Figure 4, and its main components are detailed below.

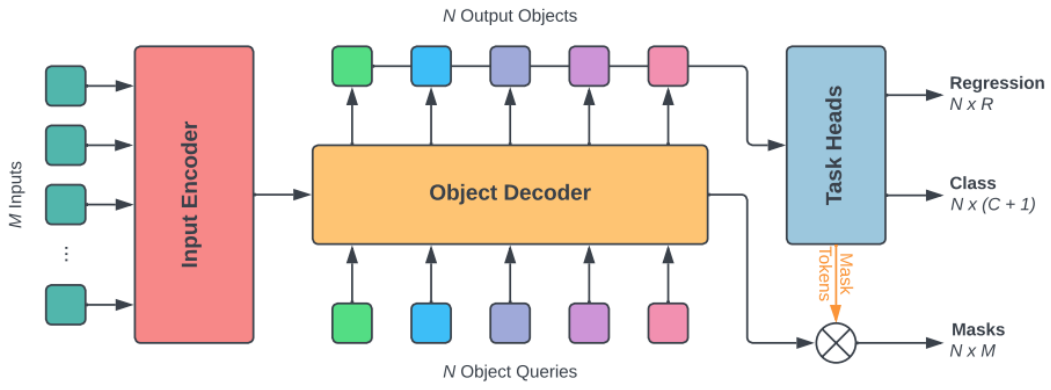


Figure 4: Overview of the architecture for SV reconstruction, adapted from MaskFormer.

The primary adaptations to the original MaskFormer model are as follows:

Input Encoder The original CNN-based pixel decoder is replaced with a transformer-based encoder. This module is designed to process an input set of unordered track and jet kinematic features as presented in the Figure 1, producing a contextualized embedding for each track.

Transformer Decoder This module remains largely unchanged. It performs cross-attention between the contextualized track embeddings (from the Input Encoder) and the N learnable object queries. The output is a set of N embeddings, each representing a candidate secondary vertex.

Task Heads The original segmentation heads are replaced by three distinct, task-specific heads that take the SV candidate embeddings as input:

- **Classification Head:** A feed-forward network (FFN) that predicts the class of each SV candidate (originating from a b-quark, c-quark, or light-quark fragmentation).
- **Masking Head:** An FFN that produces the track-to-vertex association mask, assigning tracks to each SV candidate.
- **Regression Head:** A new, dedicated FFN that predicts the precise geometric properties, Angular displacement from Jet-Axis (dR) and Radial flight length from Primary Vertex (L_{xy}), for each SV candidate.

2.3 Competitive Architectures

While Graph Neural Networks (GNNs) are a strong baseline for reconstruction tasks, their reliance on iterative message passing across a predefined graph can be a limitation. In contrast, the transformer-based architecture provides a more direct and powerful solution. Its attention mechanisms inherently operate on a dynamic, fully-connected graph of all input tracks, allowing a set of SV queries to perform a global, top-down grouping of tracks into vertices in a single, end-to-end step.

2.4 Simulation tools and Data format

A detailed description of the physics event generation and detector simulation is beyond the scope of this report. However, the primary software tools employed to produce the dataset are summarized here. Each component is a standard tool in the High-Energy Physics community, and their specific roles in the data generation pipeline were as follows:

- The event generation, detector simulation, digitization, and dataset production were performed within the ACTS framework.
- The Open Data Detector (ODD) [3] was used to define the detector geometry.
- Event generation was carried out with Pythia8 [4], while detector simulation employed Fatras [5] with the ODD geometry, both integrated into ACTS [6].
- Track reconstruction, navigation, jet–track matching, truth labeling, secondary vertex reconstruction, and plotting were handled using ACTS libraries, whereas FastJet [7] was employed for jet clustering.
- The final datasets were written to ROOT [8] files using ACTS as the overarching toolkit.

The raw data retrieved from the detector simulation is stored in ROOT files, a domain-specific format widely used in high-energy physics. Within these ROOT files, simulation data is typically organized into separate branches, containing event-wise non-homogeneous 2-dimensional arrays. This structure, while efficient for physics analysis, is generally incompatible with standard machine learning frameworks, which often expect uniform, vectorized inputs.

The non-homogeneous nature of this data, where array sizes can vary significantly from one event to another, poses a direct challenge for ingestion by neural networks. A conceptual visualization of this non-uniform data state is shown in Figure 5, highlighting the need for a dedicated conversion and pre-processing step to transform the contents into a structured, machine learning-compatible data format.

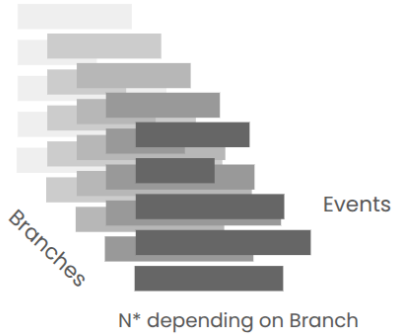


Figure 5: Conceptual visualization of the non-homogeneous data structure.

The details of conversion module will be elaborated in the Conversion Module sections.

2.5 Machine Learning preferred data format

Machine learning frameworks, such as PyTorch[9] used in this project, typically require a uniform, vectorized data format to perform essential matrix algebra for operations like gradient optimization.

HDF5 is used in this project because it stores data in dense, multi-dimensional, and homogeneous arrays. This structure allows for direct and efficient ingestion by machine learning libraries. Furthermore, HDF5 is a self-describing format that supports a hierarchical structure, which is excellent for organizing large datasets, and it is optimized for fast I/O slicing, enabling efficient training on datasets that may be too large to fit into memory.

The simulation data from ROOT file is restructured into Jet-wise uniformed array. These arrays are then systematically stored within a single HDF5 file, organized under a hierarchical group structure to maintain clarity and ease of access. For each jet, the data is partitioned into three distinct groups:

- **Jets:** An array containing the kinematic properties of the jet itself.
- **Tracks:** A fixed-size 2D array containing the properties of every track associated with the jet. Jets with fewer tracks than the maximum are padded.
- **SV:** A fixed-size 2D array containing the truth-level properties of the secondary vertices within the jet.

2.6 Conversion Module

A dedicated conversion module was developed to systematically restructure the event-wise, non-homogeneous arrays from the source ROOT files into the structured, jet-wise arrays

required by the model. This module simultaneously applies pre-processing steps according to physics-based requirements. The operation of this module was illustrated in Figure 6.

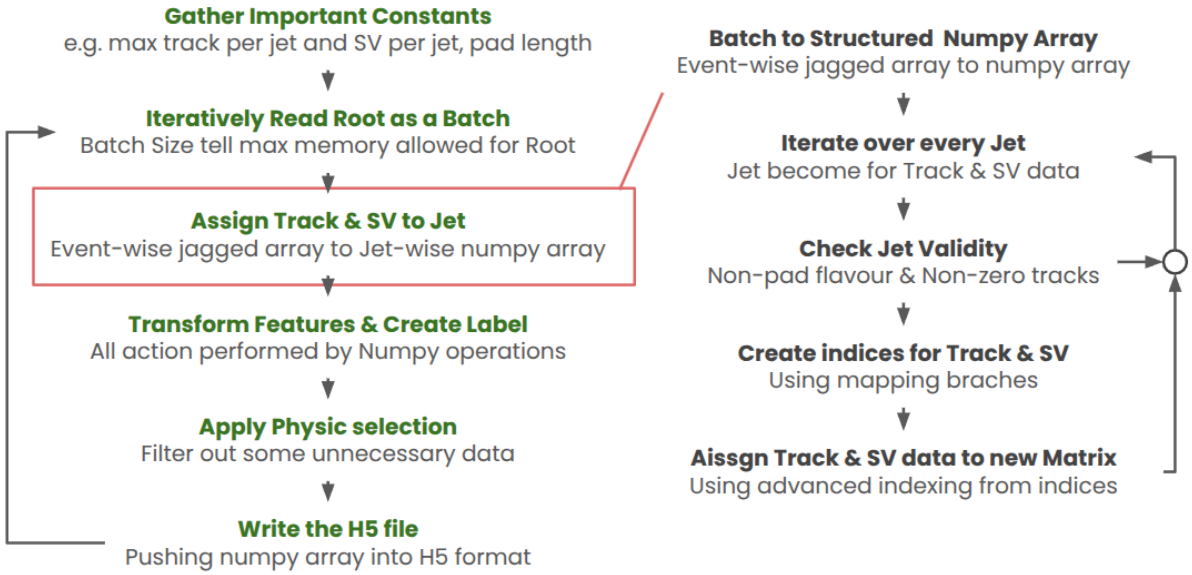


Figure 6: High-level diagram of the data conversion module's operation.

Developed in Python, the module is designed for batch processing to ensure both memory and time efficiency when handling large datasets. The core of the algorithm processes each batch of events within a single main loop. Inside this loop, instead of slow, iterative appending, advanced NumPy indexing techniques are used to directly map and assign track and SV data into pre-allocated, fixed-size, jet-wise arrays. This approach provides stable and predictable memory usage while leveraging the highly optimized, vectorized nature of NumPy. By minimizing native Python loops, this strategy mitigates the performance overhead typically associated with interpreted languages like Python.

The conversion module successfully processes ROOT files into the HDF5 format, demonstrating excellent performance and scalability. The processing time scales with the number of events; for a sample of 250k events (containing 1.6M jets), the operation took approximately 6 minutes, while a larger sample of 1M events (64M jets) was processed in approximately 30 minutes.

Data integrity is preserved throughout this process. A direct comparison between the original ROOT file and the final HDF5 file confirms that the conversion and pre-processing steps have not introduced biases. **Figure 7** validates the distributions for key jet and track features, while **Figure 8** and **Figure 9** demonstrate that the track-to-jet and SV-to-jet assignments, respectively, are perfectly preserved. All validation plots shown were produced using a sample of 250,000 events.

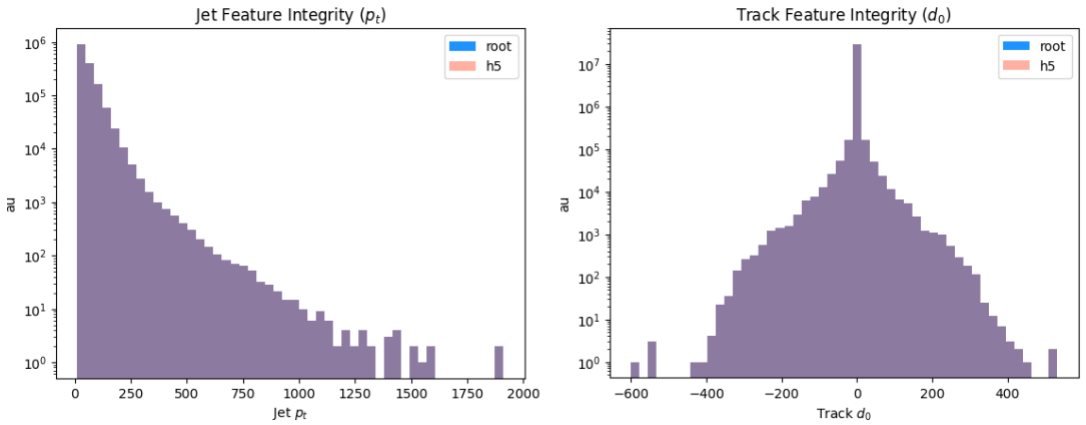


Figure 7: Validation of jet and track feature distributions, comparing the source ROOT data with the processed HDF5 data.

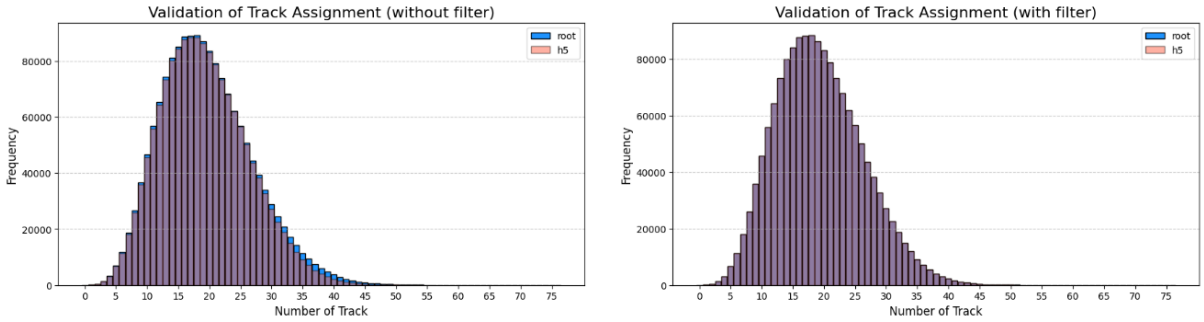


Figure 8: Validation of the track-to-jet assignment, showing a perfect correlation between the number of tracks per jet in the source and processed files.

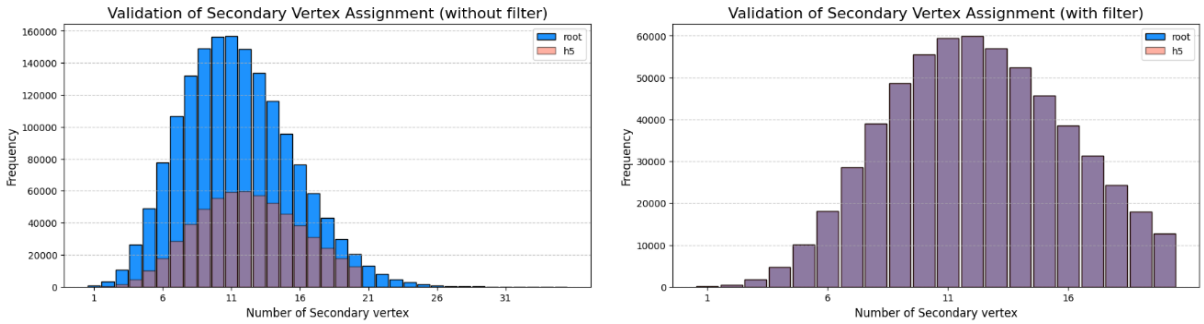


Figure 9: Validation of the SV-to-jet assignment, showing a perfect correlation in the number of secondary vertices per jet. Note that the distribution's shape is affected by the exclusion of light-flavor jets during pre-processing.

During data pre-processing, the following filtering and selection criteria were applied:

- **Jet Filtering:** Jets were removed if they were identified as background (with a truth flavor label of -99) or if they contained no reconstructed tracks.
- **Track Selection:** For each jet, a maximum of the 50 tracks with the highest transverse momentum (p_T) were retained.

- **Secondary Vertex Limit:** A limit was placed on the number of secondary vertices stored per jet. While this was set to 20 for the validation figures to retain 95% of the data, a tighter limit of 5 is planned for the final training dataset. Additionally, SVs within light-jets are flagged and excluded from certain calculations.

2.7 Dataset Summary

The final dataset used for this work is derived from a simulation of **1M events**, which initially contained approximately 64M jets. It should be noted that while this dataset is suitable for the development and validation of our reconstruction algorithm, it is a preliminary sample and has not yet undergone full physics validation.

After applying the selection criteria described in the previous section, the total number of jets available for the study was reduced to approximately **12.6M jets as a final number** used to produce the plots in this report. This final dataset was then partitioned into three distinct subsets for training, validation, and testing. A detailed summary of the dataset splits and properties are provided in Table 1 and Table 2 respectively.

Table 1: Summary of the dataset used for training and evaluation. The table shows the number of jets in each partitioned subset after all filtering and pre-processing steps have been applied, with a breakdown by jet flavor.

Split	**Total Jets**	**B Jets**	**C Jets**	**L Jets**
Training	10 000 000	1 400 000	600 000	8 000 000
Validation	1 300 000	185 000	75 000	1 040 000
Testing	1 300 000	185 000	75 000	1 040 000
Total	12 600 000	1 770 000	750 000	10 080 000

Table 2: Overall statistical properties of the final dataset used in this project. Note: These statistics exclude light-flavor jets (L-Jets) that do not contain any secondary vertices.

Property	**Mean**	**Minimum**	**Maximum**
Tracks per Jet	3.26	1	20
SVs per Jet	1.56	1	5
b-SVs per Jet	1.23	0	5
c-SVs per Jet	0.33	0	5

All input features were transformed using standard normalization. For the regression targets (the SV properties), a logarithmic transformation ($\log(x)$) was used to compress the range of the values. Following this, the resulting distribution was also transformed using the standard normalization.

3 Model Result and Analysis

The MaskFormer model was trained on the final dataset using the same hyperparameter settings as the reference architecture [2]. To benchmark the performance and understand the impact of our more realistic, imbalanced dataset, the model was also trained on the reference dataset from [2], which is a balanced sample of 13 million jets with an equal

number of b-, c-, and light-flavor jets. The model's performance was then evaluated for each of the three primary tasks: classification, track assignment, and vertex regression.

3.1 Secondary Vertex Classification

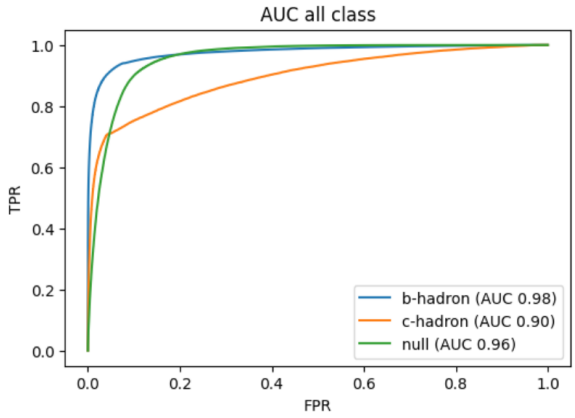


Figure 10: ROC curves for the model trained on our dataset.

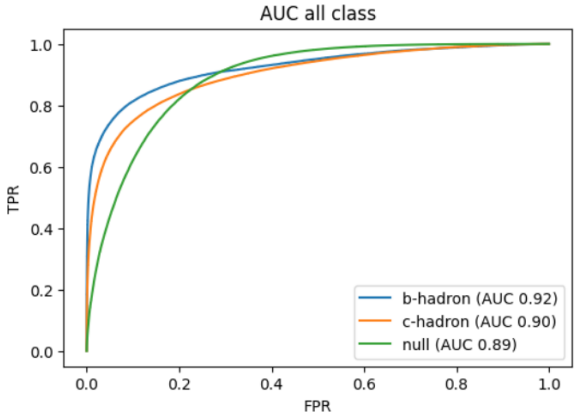


Figure 11: ROC curves for the model trained on the balanced reference dataset.

Despite the significant class imbalance in our dataset (a much smaller proportion of b- and c-jets), the model achieves impressive classification performance, as shown by the ROC curves in Figure 10. This performance is comparable to that of the model trained on the balanced reference dataset (Figure 11). However, a performance degradation is visible for the c-jet classification in our sample, likely attributable to the extremely low statistics for this class.

3.2 Track Assignment (Mask Prediction)

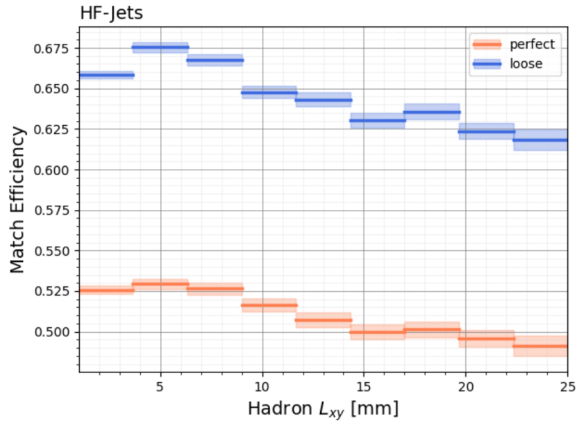


Figure 12: Masking efficiency as a function of transverse displacement (L_{xy}) for the model trained on our dataset.

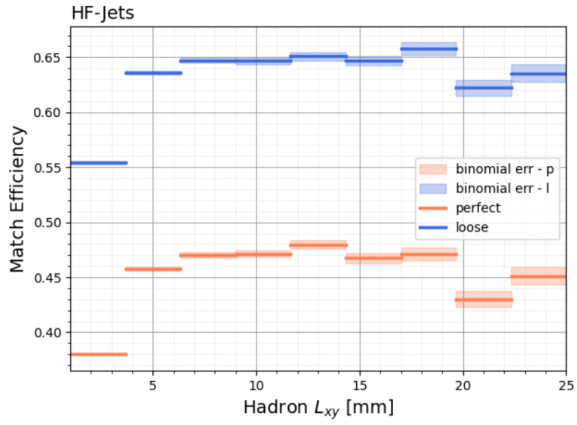


Figure 13: Masking efficiency as a function of transverse displacement (L_{xy}) for the model trained on the reference dataset.

The track assignment performance, or masking efficiency, is comparable between the two models (Figures 12 and 13). The model trained on our dataset shows a slightly higher

'perfect match' efficiency, while the model trained on the reference set exhibits a more stable performance with smaller uncertainty bands.

3.3 Secondary Vertex Regression

The vertex regression task proved to be the most challenging. Figure 14 shows the correlation between the true and predicted vertex properties for the model trained on our dataset. While the model shows some ability to predict the angular displacement (ΔR), it largely fails to capture the transverse displacement (L_{xy}), indicating significant under-performance. In stark contrast, the same model trained on the balanced reference dataset achieves a near-perfect correlation for both properties, as shown in Figure 15. This suggests that the regression task is highly sensitive to the dataset's characteristics, representing a key area for future improvement. The poor regression performance was also likely exacerbated by the use of a preliminary dataset, as time constraints prevented the generation of a more realistic, experimentally-validated simulation sample.

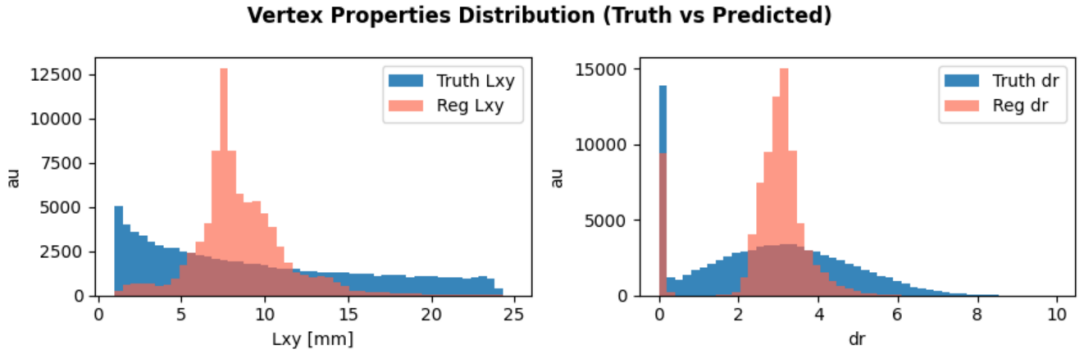


Figure 14: Truth vs. Prediction distributions for vertex properties on our dataset.

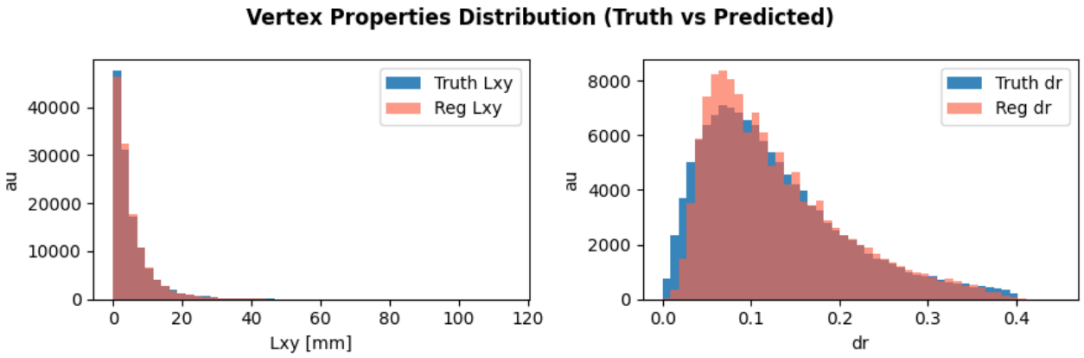


Figure 15: Truth vs. Prediction distributions for vertex properties on the reference dataset.

4 Conclusion

This report addressed the critical challenge of secondary vertex (SV) reconstruction in the high-occupancy environment expected at the High-Luminosity Large Hadron Collider (HL-LHC). We introduced and evaluated a novel approach based on the MaskFormer architecture. A key component of this work was the development of a dedicated data

processing pipeline to convert domain-specific ROOT files into a uniform, jet-wise HDF5 format suitable for modern machine learning frameworks.

The adapted MaskFormer architecture demonstrated significant promise, particularly in the classification and track assignment (masking) tasks. Despite being trained on a preliminary, highly imbalanced dataset with non-optimized hyperparameters, the model achieved classification performance comparable to a baseline trained on a perfectly balanced reference sample. This result validates the strength of the transformer's attention mechanisms for learning powerful representations even with limited statistics for certain classes. However, the study also revealed a key challenge: the vertex regression task proved to be highly sensitive to the dataset's composition. The model struggled on our imbalanced sample but performed exceptionally well on the balanced reference dataset. This discrepancy provides strong evidence that the model's regression capabilities could be significantly improved by training on a more realistic, fully validated physics dataset.

In summary, this work successfully establishes the MaskFormer architecture as a powerful and viable new direction for secondary vertex reconstruction. It excels at the difficult task of correctly grouping tracks to vertices and provides a strong foundation for future research and optimization aimed at delivering a complete, high-performance SV-finding algorithm for the HL-LHC era.

5 Discussion

The results presented in the previous section highlight several key aspects of the adapted MaskFormer's performance on the realistic, imbalanced SV reconstruction dataset. This section discusses the most salient observations, including the challenges of multi-task learning, the impact of data normalization, and a brief study of hyperparameter sensitivity.

5.1 Imbalance in Multi-Task Learning

A key observation during training was the disparity in learning speeds across the different tasks. The losses for the classification and mask prediction tasks converged significantly faster and reached lower plateaus than the loss for the vertex regression. This suggests an imbalance in the gradient magnitudes contributed by each task head, where the regression task may be under-weighted during the optimization process. Consequently, the relative weights of each component in the total loss function are critical hyperparameters that require careful tuning to ensure balanced and effective training across all objectives.

5.2 Impact of Target Normalization on Performance

The choice of normalization for the regression targets was found to have a significant and somewhat counter-intuitive cross-task impact, particularly on the performance of the SV classification head. As shown in Figure 16, standard normalization (Z-score) yielded the most stable and lowest loss for the regression task itself when compared to Min-Max scaling.

However, this choice had a notable effect on the classification task. Figure 17 illustrates that using Min-Max scaling, which bounds the regression targets to a smaller magnitude (e.g., within [0, 1]), resulted in a significantly lower and more stable classification loss. This suggests a delicate interplay between the tasks, where the scale of the regression

targets can affect the stability and performance of the classification head, revealing a crucial trade-off to consider during model optimization.

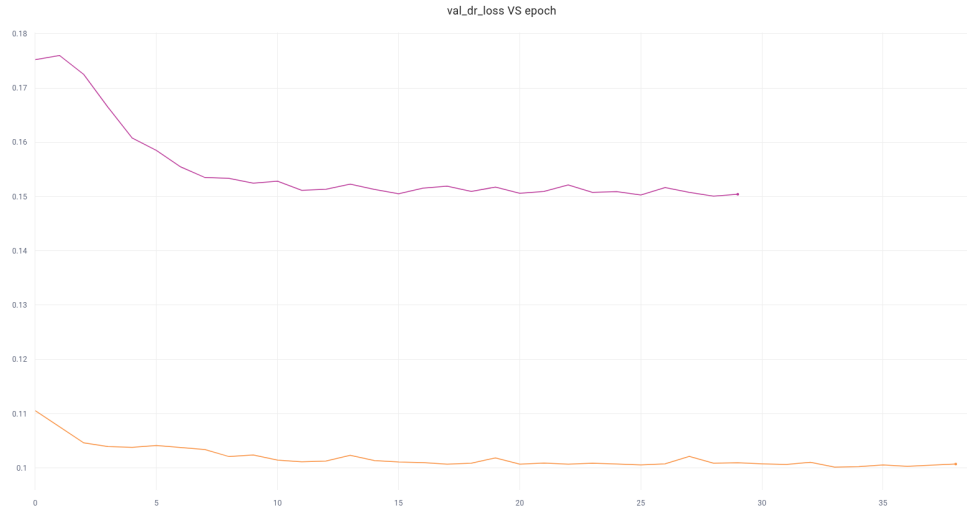


Figure 16: Comparison of the validation loss for the ΔR regression task using two different target scaling methods: Standard Normalization (pink) and Min-Max Scaling (orange).

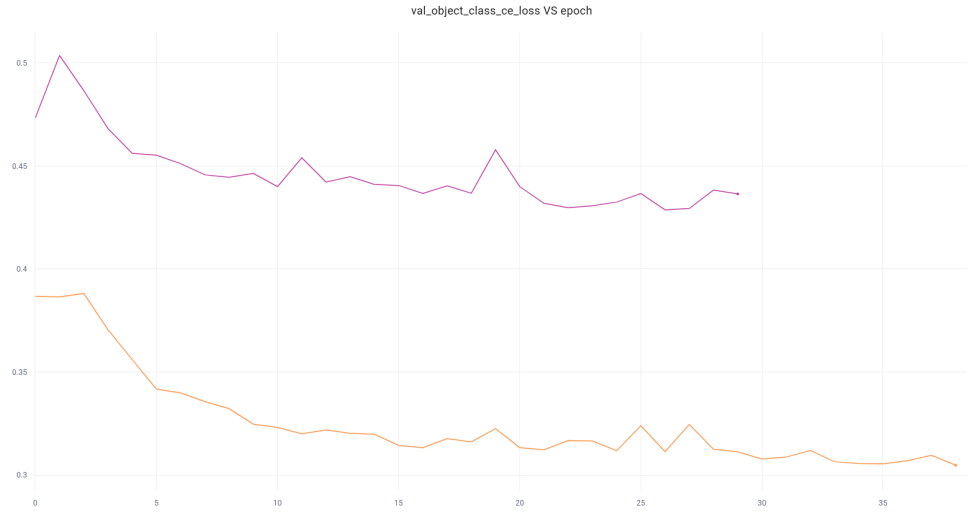


Figure 17: Comparison of the validation loss for the SV classification task, demonstrating the significant performance improvement when using Min-Max scaling for the regression targets.

5.3 Hyperparameter Sensitivity Study

A brief sensitivity study was conducted on key hyperparameters to understand their impact. The model's performance showed little sensitivity to the batch size, with similar results obtained for values of 5,000, 10,000, and 20,000. Conversely, the learning rate had a more noticeable effect. While the regression task was largely insensitive across the tested range, the mask prediction and classification tasks benefited from smaller values. A maximum learning rate of 2.5×10^{-4} yielded the best results compared to 5×10^{-4} and 1×10^{-3}.

5.4 Recommendations for Future Work

Based on the observations in this report, several promising directions for future study emerge. First, in terms of hyperparameter optimization, future work should prioritize a large batch size for efficiency, paired with a relatively small learning rate. The most critical area for tuning, however, appears to be the careful adjustment of the individual loss weights and the choice of target normalization, given the strong coupling observed between the regression and classification tasks.

Beyond parameter tuning, a promising architectural direction is to explore a two-stage training regimen. This could involve a self-supervised pre-training phase where the transformer encoder-decoder is trained on a pretext task, such as masking and reconstructing input track features. Following this, the pre-trained backbone could be fine-tuned with the task-specific MLP heads for classification, masking, and regression. This approach could help the model learn more robust feature representations before tackling the imbalanced, multi-task downstream problem.

References

[1] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. "Per-Pixel Classification is Not All You Need for Semantic Segmentation". In: *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*. 2021. arXiv: 2107.06278. URL: https://arxiv.org/abs/2107.06278.

[2] Samuel Van Stroud. *Secondary Vertex Reconstruction with MaskFormers*. 2024. arXiv: 2312.12272. URL: https://arxiv.org/pdf/2312.12272.

[3] L. Sailer et al. *Open Data Detector*. Version v2.1.1. Oct. 2021. DOI: 10.5281/zenodo.5572213. URL: https://doi.org/10.5281/zenodo.5572213.

[4] Christian Bierlich et al. "A comprehensive guide to the physics and usage of PYTHIA 8.3". In: *SciPost Phys. Codebases* (2022), p. 1. DOI: 10.21468/SciPostPhysCodeb.1. arXiv: 2203.11601.

[5] K Edmonds et al. *The Fast ATLAS Track Simulation (FATRAS)*. Tech. rep. Geneva: CERN, 2008. URL: https://cds.cern.ch/record/1091969.

[6] Xiaocong Ai et al. "A Common Tracking Software (ACTS) for the LHC and future experiments". In: *J. Instrum.* 18.01 (2023), P01021. DOI: 10.1088/1748-0221/18/01/P01021. arXiv: 2209.08693.

[7] Matteo Cacciari, Gavin P. Salam, and Gregory Soyez. "FastJet user manual". In: *Eur. Phys. J. C* 72 (2012), p. 1896. DOI: 10.1140/epjc/s10052-012-1896-2. arXiv: 1111.6097.

[8] R. Brun and F. Rademakers. "ROOT - An object oriented data analysis framework". In: *Nucl. Instrum. Meth. A* 389 (1997), pp. 81–86. DOI: 10.1016/S0168-9002(97)00048-X.

[9] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*. 2019, pp. 8026–8037. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

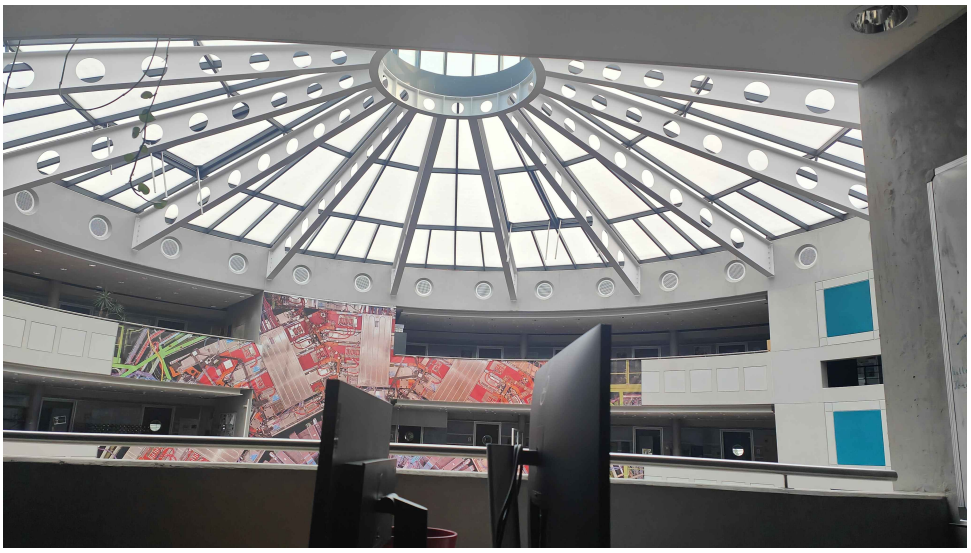
6 Daily Report

Week 1: Project Setup and Initial Validation

Date: 10 June 2025

Log Summary: Commenced the summer student program following the welcome session. An initial meeting was held with supervisors Doga (Physicist) and Jonathan (Technical Student) to establish the project plan, objectives, and key milestones.

- **Activities:**
- Configured the development environment, including access to the Next Generation Triggers computing cluster and CERN cloud resources with Kubernetes.
- Familiarized with the Kubernetes command-line tool for cluster operations.
- Established a workspace on the computing cluster, including Python virtual environment setup and importation of the reference dataset.



Date: 11 June 2025

Log Summary: Resolved a PyTorch GPU accessibility issue within the virtual environment by reinstalling the package manager and libraries. The day's objective was to validate the functionality of the reference codebase, ensuring not only error-free execution but also the replication of results consistent with the original paper.

- **Activities:**
- Trained the machine learning model using the original dataset and inspected the initial training plots.
- Identified and debugged an error in the original testing pipeline.
- Corrected and adapted the testing pipeline to function within the CERN computing cluster environment.
- Modified hard-coded path references to be more dynamic.
- Resolved a class referencing conflict within the prediction writer script.

Date: 12 June 2025

Log Summary: Continued the systematic validation of the reference pipeline. The main focus was the interpretation of training plots from an 11-hour training session to ensure the model's learning behavior was as expected.

- **Activities:**
- Inspected the complete set of training plots from the previous day's session. No anomalies were found, but the loss function was observed to plateau after approximately 30 training iterations.
- Implemented an Early Stopping mechanism to optimize training duration when model performance no longer improves.
- Initiated a new training run with the Early Stopping callback implemented.

Date: 13 June 2025

Log Summary: With the initial code validation tasks complete, the focus shifted to building the necessary background knowledge for the next phase of the project. Supervisors provided reading materials covering key concepts in experimental physics.

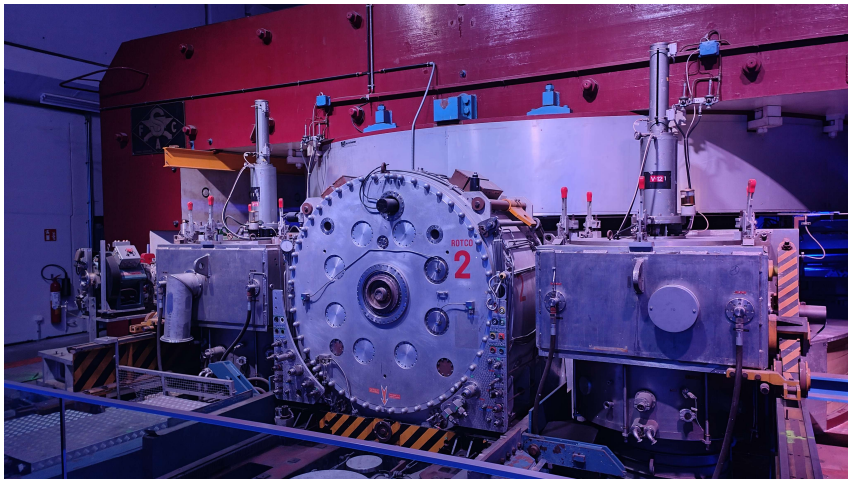
- **Activities:**
- Analyzed the training plots from the Early Stopping-enabled session, confirming a reduction in training time from 11 to 7 hours while maintaining the same final loss value.
- Began a comprehensive review of the provided reading materials on the ATLAS detector, Monte Carlo simulation, and data reconstruction.

Week 2: Background Study and Evaluation Pipeline Reproduction

Date: 16 June 2025

Log Summary: Attended a site visit to the ATLAS experiment and the Synchrocyclotron, featuring an informative presentation on their history, operation, and impact. The visit occupied the afternoon, limiting project development time for the day.

- **Activities:**
- Concluded the review of the reading materials provided by supervisors.
- Held a Q&A session with supervisor Doga to solidify understanding of the physics concepts.



Date: 17 June 2025

Log Summary: Following the consolidation of fundamental knowledge about detector physics, work began on the next project milestone: understanding and reproducing the evaluation metrics.

- **Activities:**
- Learned data manipulation techniques for '.h5' files.
- Analyzed the hierarchical structure of the original dataset.
- Studied the various evaluation plots presented in the reference paper to understand the model's performance metrics.

Date: 18 June 2025

Log Summary: Continued the process of reproducing the evaluation pipeline. In the afternoon, attended a site visit at the CERN data centre, where the presentation focused on the scale of data production and processing at the LHC.

- **Activities:**
- Successfully reproduced the Receiver Operating Characteristic (ROC) plots for hadron flavor classification.

Date: 19 June 2025

Log Summary: The work of reproducing the evaluation pipeline and visualizing the original dataset continued systematically.

- **Activities:**
- Reproduced the probability distribution plots for the mask prediction output.
- Reproduced the efficiency plots of mask prediction as a function of input features.
- Verified the logic and correctness of the reproduced plots against the reference paper.

Date: 20 June 2025

Log Summary: A significant milestone was achieved with the successful reproduction of the entire evaluation pipeline from the reference paper. This confirms a full understanding of the original model's functionality and performance metrics.

- **Activities:**
- Reproduced the residual plots for all secondary vertex properties.
- Prepared a presentation summarizing project progress for the project leader, Paul.
- Discussed the reproduced plot results and outlined the next steps with supervisor Jonathan.

Week 3: New Dataset Integration and Initial Dataloader Development

Date: 23 June 2025

Log Summary: A discussion with supervisors covered feedback on the progress presentation and confirmed the next major step: the development of a data conversion pipeline to process '.root' files into the '.h5' format. The original codebase was noted to have significant undocumented changes compared to the published paper, leading to slower debugging progress.

- **Activities:**
- Adjusted the binning and axis ranges of the reproduced plots to more closely match the reference paper's figures.
- Finalized the progress update presentation, which included the current milestone goals, a brief explanation of the model's I/O, and a comparison of the reproduced and original evaluation plots.

Date: 24 June 2025

Log Summary: Received the new '.root' dataset from supervisor Doga, enabling the start of the data analysis and conversion phase of the project.

- **Activities:**
- Learned to manipulate '.root' files using the 'uproot' library in Python.
- Studied the functionality and data structures provided by the 'uproot' library.

Date: 25 June 2025

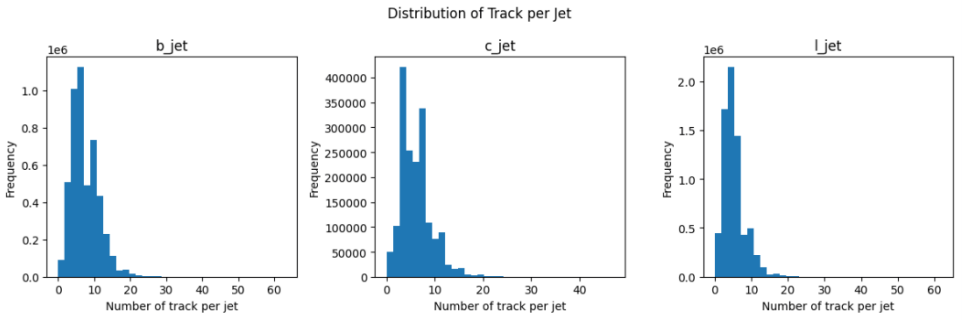
Log Summary: Analysis of the 'uproot' library revealed that its default data structure, Awkward Arrays, required inefficient, loop-intensive manipulation. It was determined that an optimization step to convert this data into a more performant format was necessary before proceeding.

- **Activities:**
- Developed a function to convert data from ‘.root‘ files into NumPy arrays for more efficient processing.
- Explored the new dataset and generated distribution plots for each data field.

Date: 26 June 2025

Log Summary: The initial strategy, agreed upon with supervisor Jonathan, is to train the new model using the same feature set as the reference paper to allow for a direct performance comparison.

- **Activities:**
- Performed a comparative analysis of the new dataset and the original reference dataset to identify any anomalies or discrepancies.
- Documented the findings of the comparison in preparation for a meeting with the dataset creator, Paul.




```
##### b_jet #####
min: 0.0 max: 63.0
mean: 7.428649798846175
--------------------
##### c_jet #####
min: 0.0 max: 47.0
mean: 6.196842314863086
--------------------
##### l_jet #####
min: 0.0 max: 62.0
mean: 5.2317473250670465
--------------------
```

Date: 27 June 2025

Log Summary: The dataset comparison revealed several feature mismatches and missing data fields. A meeting was held with Paul to discuss these issues and determine appropriate alternative features.

- **Activities:**
- Met with Paul to present the anomalies and missing fields found in the new dataset.
- As requested by Paul, generated plots of Tracks per Jet, Vertices per Jet, and Jets per Event to help validate his simulation algorithm.

Week 4: Dataloader Implementation and Debugging

Date: 30 June 2025

Log Summary: Analysis of the plots for Paul indicated an unexpectedly low number of valid jets per event. To further investigate, Paul requested a plot of the distances between secondary vertices to aid in debugging his simulation.

- **Activities:**
- Corrected the binning intervals of the plots from the previous week and submitted them to Paul.
- Generated plots showing the pairwise distances between all secondary vertices in the events.

Date: 1 July 2025

Log Summary: Resumed development of the dataloader pipeline while awaiting a resolution for the new dataset issues from Paul and Doga. The Summer Student Lecture Program commenced.

- **Activities:**
- Rewrote the code that produces the Track-Jet-Event comparison plots and distance plots for Doga to assist in her validation of the dataset.
- Began work to reproduce the reference paper's data structure from the new '.root' file.
- Successfully created the jet dataset component, converting from '.root' to '.h5'.

Date: 2 July 2025

Log Summary: The primary task of developing the track-to-jet assignment algorithm commenced. The objective is to produce a final data structure with the shape (number of jets, number of tracks, number of features).

- **Activities:**
- Developed an initial brute-force algorithm to assign tracks to their corresponding valid jets.

Date: 3 July 2025

Log Summary: An investigation of the initial data conversion results showed that while the jet data was correct, the track assignment algorithm was flawed, primarily due to test cases having only one valid jet per event.

- **Activities:**
- Discovered a logical bug in the assignment algorithm stemming from the simplified structure of the test case (1 valid jet/event).
- Redefined the problem statement and began modifying the algorithm to handle the general case.

Date: 4 July 2025

Log Summary: Successfully debugged the track assignment algorithm and developed a suite of validation tests to ensure the correctness of the entire data conversion pipeline.

- **Activities:**
- Completed the first version of the data conversion algorithm.
- Developed dataset validation logic, including tests for Jet and Track masks and padding of 'NaN' values in the track feature fields.

Week 5: Dataloader Verification and Optimization

Date: 7 July 2025

Log Summary: Continued the systematic verification of the data conversion algorithm's logic to ensure its robustness.

- **Activities:**
- Developed further dataset validation logic, including tests for Track-to-Jet position and feature value consistency.
- The converted dataset successfully passed all developed validation tests.

Date: 8 July 2025

Log Summary: With the new dataset from Doga and Paul still pending, the focus shifted to optimizing the performance of the data conversion pipeline, specifically targeting time and memory efficiency.

- **Activities:**
- Refactored several Python loops within the Jet and Track conversion algorithms into vectorized NumPy operations to reduce time complexity.
- Created a ‘JetConversion‘ class to encapsulate the logic, improve ease of use, and ensure memory efficiency.

Date: 9 July 2025

Log Summary: Following the completion of the ‘JetConversion‘ class, development began on the corresponding ‘TrackConversion‘ class.

- **Activities:**
- Created a ‘TrackConversion‘ class for streamlined and memory-efficient processing.
- Validated both the ‘JetConversion‘ and ‘TrackConversion‘ classes using the established testing suite.
- Conducted initial performance testing of the new class-based pipeline.

Date: 10 July 2025

Log Summary: Performance testing revealed inefficient Python loops remaining in the ‘TrackConversion‘ class. These were optimized to improve performance.

- **Activities:**
- Optimized the track assignment algorithm by replacing iterative loops with pre-allocated NumPy arrays and vectorized assignments.
- Conducted a code review with supervisor Jonathan and received constructive feedback.

Date: 11 July 2025

Log Summary: Attended a meeting with a Thai diplomat in the morning. The afternoon was dedicated to a feedback session with the supervisor to incorporate recent suggestions.

- **Activities:**
- Met with a diplomat alongside other Thai summer students.
- Added detailed comments and documentation to the code based on supervisor feedback.
- Developed functions for track data transformation (e.g., calculating ‘phi_{rel}‘ from ‘phi‘ and ‘pt_{frac}‘ from ‘pt‘).

Week 6: Finalizing Conversion Module

Date: 14 July 2025

Log Summary: The main focus was finalizing the track conversion algorithm. A brief discussion was held regarding an upcoming presentation at an ATLAS meeting.

- **Activities:**
- Discussed the upcoming ATLAS meeting scheduled for August 12th.
- Completed the development of the track conversion algorithm.
- Performed a simple test to ensure basic functionality.

Date: 15 July 2025

Log Summary: With the track conversion algorithm complete, its performance was evaluated. Work on the final component of the dataloader, the secondary vertex conversion, commenced.

- **Activities:**
- Conducted a performance evaluation of the track conversion algorithm.
- Began development of the secondary vertex (SV) conversion algorithm.

Date: 16 July 2025

Log Summary: Met with all supervisors to review the results from the partially completed conversion module. Anomalies in the output plots were identified and subsequently reported to Paul.

- **Activities:**
- Discussed results from the Jet and Track conversion modules.
- Identified an anomalously large number of secondary vertices per jet in the plots and reported the finding to Paul for investigation.

Date: 17 July 2025

Log Summary: Continued working on the SV conversion algorithm, encountering a challenge related to creating the necessary training labels. Discussed a potential solution with supervisor Jonathan.

- **Activities:**
- Continued development of the SV conversion algorithm.
- Discussed implementation difficulties with Jonathan regarding the Hadron Index and Truth Hadron Index required for mapping and mask creation.

Date: 18 July 2025

Log Summary: Work continued on generating the necessary labels for the SV mask creation, building upon yesterday's tasks.

- **Activities:**
- Continued the implementation of the label generation for SV mask creation.

Week 7: Completing Conversion and Initial Training (July 21 - July 25)

Date: 21 July 2025

Log Summary: Successfully completed the generation of mask prediction and classification labels for the secondary vertices. The next step is to generate the regression labels.

- **Activities:**
- Completed the implementation for SV mask prediction and classification labels.
- Began the implementation for the transformation of SV regression labels.

Date: 22 July 2025

Log Summary: The secondary vertex conversion module was finished today. An anomaly in an initial integrity test was discussed with the supervisor, who clarified that these labels would be provided directly in a future version of the '.root' file, obviating the need for manual calculation.

- **Activities:**
- Finished the SV conversion module.
- Conducted a small integrity test and identified an anomaly.
- Concluded with the supervisor that the regression task labels would be provided in the next data release.

Date: 23 July 2025

Log Summary: A comprehensive evaluation of the entire conversion module was conducted, focusing on time, memory, and data integrity. A performance bottleneck with the 'uproot' library was identified, with file reading taking up to 10 minutes for a 4GB file.

- **Activities:**
- Tested the full conversion module, measuring performance and verifying integrity. Noted a significant delay in the initial file reading stage with 'uproot'.
- Prepared testing results for discussion with the supervisor.

Date: 24 July 2025

Log Summary: Discussed the ‘uproot‘ library performance issue with Jonathan to brainstorm solutions. Began developing an evaluation script for the module and registered for an upcoming student presentation session.

- **Activities:**
- Discussed potential solutions to the ‘uproot‘ problem with Jonathan.
- Developed an evaluation script for the data conversion module.
- Registered for the student session presentation.

Date: 25 July 2025

Log Summary: To address the performance issue, a deeper study of the ‘uproot‘ library was undertaken. Based on this, a re-architected data conversion module was designed to improve I/O efficiency.

- **Activities:**
- Studied advanced features within the ‘uproot‘ library for efficient data access.
- Designed a new data conversion module to solve the performance bottleneck, focusing on optimized file I/O while retaining the core conversion logic.

Week 8: New Conversion Module and First Full Training

Date: 28 July 2025

Log Summary: The focus shifted to the practical aspects of model training, including setting up configuration files for the new dataset and debugging the associated changes.

- **Activities:**
- Set up the ‘.YAML‘ configuration file for model training runs.
- Fixed bugs in the training pipeline resulting from changes in the ‘.YAML‘ configuration.

Date: 29 July 2025

Log Summary: A significant milestone was reached with the first successful training run using the newly processed dataset. Initial analysis of the loss curves indicated that the model was learning as expected.

- **Activities:**
- Performed the first model training with the new dataset.
- Analyzed the loss curves to confirm that the model was learning correctly.

Date: 30 July 2025

Log Summary: Began development of the re-architected data conversion module, starting with testing necessary functions and defining the overall class structure.

- **Activities:**
- Tested all functions required for the new data conversion module.
- Started development by defining the class structure and function signatures.

Date: 31 July 2025

Log Summary: Continued the development of the new data conversion module, with a specific focus on implementing chunk-based processing to improve efficiency.

- **Activities:**
- Continued developing the data conversion module and implemented chunk processing for I/O operations.

Date: 1 August 2025

Log Summary: The optimized data conversion module was completed. The day was spent evaluating its integrity and performance, confirming that it met the project's requirements, reducing the conversion time for a 4GB file to approximately 6 minutes.

- **Activities:**
- Finished the new data conversion module.
- Evaluated the integrity and performance of the new module.

Week 9: Presentations and Hyperparameter Tuning (August 4 - August 8)

Date: 4 August 2025

Log Summary: To verify the correctness of the optimized conversion module, a new model training was initiated using the data it produced. Preparation for the student session presentation also began.

- **Activities:**
- Trained the model with data from the new conversion module to validate its output.
- Began preparing the presentation for the student session.

Date: 5 August 2025

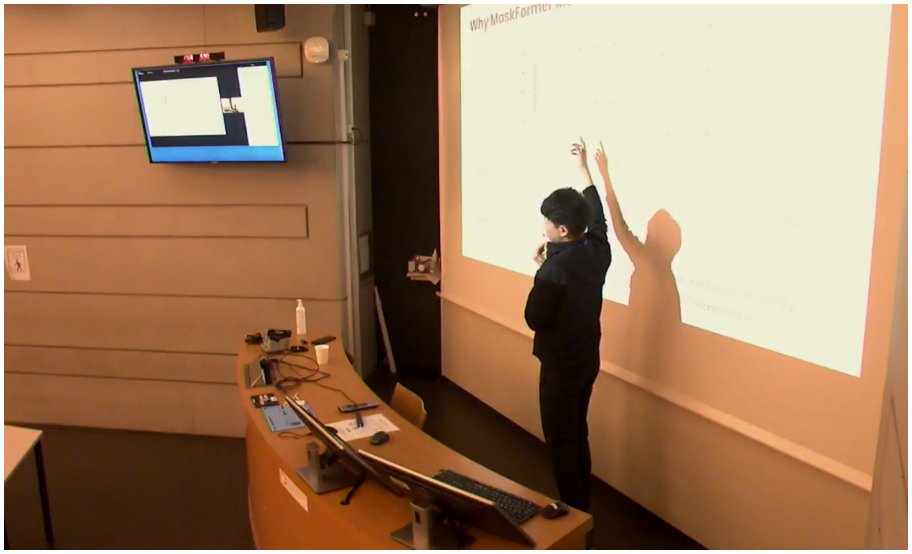
Log Summary: The entire day was dedicated to preparing for tomorrow's student session presentation.

- **Activities:**
- Finalized preparation for the student session presentation.

Date: 6 August 2025

Log Summary: Successfully delivered the presentation at the Student Session. Afterwards, a discussion was held with the supervisor about the project's next steps while awaiting the next version of the dataset.

- **Activities:**
- Presented project progress at the Student Session.
- Discussed next steps with the supervisor.



Date: 7 August 2025

Log Summary: An analysis of the model's performance on the reference dataset revealed a bias in the regression task, with results underperforming those reported in the reference paper. Began experimenting with learning rates to address this.

- **Activities:**
- Observed model performance on the reference dataset and identified a performance deficit in the regression task.
- Initiated new training runs with different hyperparameter setups, focusing on the learning rate.

Date: 8 August 2025

Log Summary: Continued hyperparameter tuning by experimenting with different weights for the multi-task loss function. The results from all recent training sessions were then analyzed and summarized.

- **Activities:**
- Performed training runs with different hyperparameter setups, focusing on the loss weights.
- Concluded and analyzed the results from the recent hyperparameter tuning experiments.

Week 10: ATLAS Meeting and Pre-Training Development

Date: 11 August 2025

Log Summary: The day's work involved preparing and updating the presentation for the ATLAS Lab meeting scheduled for the next day.

- **Activities:**
- Prepared and updated the presentation for the ATLAS Lab meeting.

Date: 12 August 2025

Log Summary: Delivered a presentation of the project's progress and findings at the ATLAS experiment lab meeting.

- **Activities:**
- Presented at the ATLAS experiment lab meeting.

Date: 13 August 2025

Log Summary: A discussion with supervisor Jonathan focused on potential methods to improve the ML pipeline and model performance. Pre-training was identified as a promising solution, and work on the necessary components began.

- **Activities:**
- Studied the configuration of the PyTorch Lightning framework's command-line interface.
- Began creating the dataloader for the self-supervised pre-training task.

Date: 14 August 2025

Log Summary: Work on the pre-training pipeline continued, with a focus on implementing the specific components required for the model's new training objective.

- **Activities:**
- Implemented the Hungarian matching algorithm and a new task head for the pre-training task.

Date: 15 August 2025

Log Summary: The pre-training pipeline was successfully executed for the first time. The initial results showed only minimal signs of convergence, indicating that hyperparameter adjustment would be crucial.

- **Activities:**
- Created the configuration file for the pre-training jobs.

Week 11: Final Dataset Validation and Bug Fixing

Date: 18 August 2025

Log Summary: An investigation of a new zero-pileup HL-LHC '.root' file revealed an invalid branch with misaligned ground truth matching. These findings were documented for supervisors to aid future dataset creation. Due to time constraints, a workaround was necessary.

- **Activities:**
- Obtained a new HL-LHC simulation dataset with zero pileup.

- Performed dataset validation and identified a ground truth misalignment.
- Reported findings to supervisors.

Date: 19 August 2025

Log Summary: Successfully developed a workaround to create the correct ground truth labels for the model despite the issues in the source file. The method involves combining two different ground truth sources and constraining the statistics.

- **Activities:**
- Implemented a modification to the data conversion module to generate correct ground truth labels.

Date: 20 August 2025

Log Summary: Received and began validating a new fifty-pileup HL-LHC simulation dataset, a crucial step for testing the model's robustness under more complex conditions.

- **Activities:**
- Obtained the new fifty-pileup HL-LHC simulation dataset.
- Performed initial dataset validation.
- Reported findings to supervisors for approval for further experiments.

Date: 21 August 2025

Log Summary: Identified a structural mismatch between the output of the conversion module and the reference dataset. Developed testing code to isolate and resolve the issue.

- **Activities:**
- Implemented modifications to the conversion module to correct the data structure.
- Addressed and fixed bugs within the module.

Date: 22 August 2025

Log Summary: Continued debugging the conversion module. After implementing the fixes, a full training process was initiated to validate that both the conversion module and the training pipeline are now functioning correctly.

- **Activities:**
- Finalized bug fixes for the data conversion module.
- Validated the end-to-end process by initiating a training run to ensure the functionality of the conversion module and training pipeline.

Week 12: Final Model Training and Reporting

Date: 25 August 2025

Log Summary: Due to persistent bugs in the source ‘.root‘ file that prevented the completion of the pre-training study, the final model was trained using a traditional, direct supervised learning approach.

- **Activities:**
- Trained the final model for the report.
- Modified the validation code for mask prediction to combine two model outputs for a more reliable validation score.

Date: 26 August 2025

Log Summary: An online internship visit was conducted by a professor from KMITL. Following this, the focus was on validating the latest model's results and exploring hyperparameter adjustments.

- **Activities:**
- Participated in an online internship visit with a KMITL professor.
- Validated the results of the final model and identified potential improvements through hyperparameter adjustments.

Date: 27 August 2025

Log Summary: The model was retrained with the adjusted hyperparameters identified yesterday. A project demonstration was given to supervisors, and the drafting of the final report began.

- **Activities:**
- Trained the model with adjusted hyperparameters.
- Provided a demonstration and project tutorial for supervisors.
- Drafted the outline for the final report.

Date: 28 August 2025

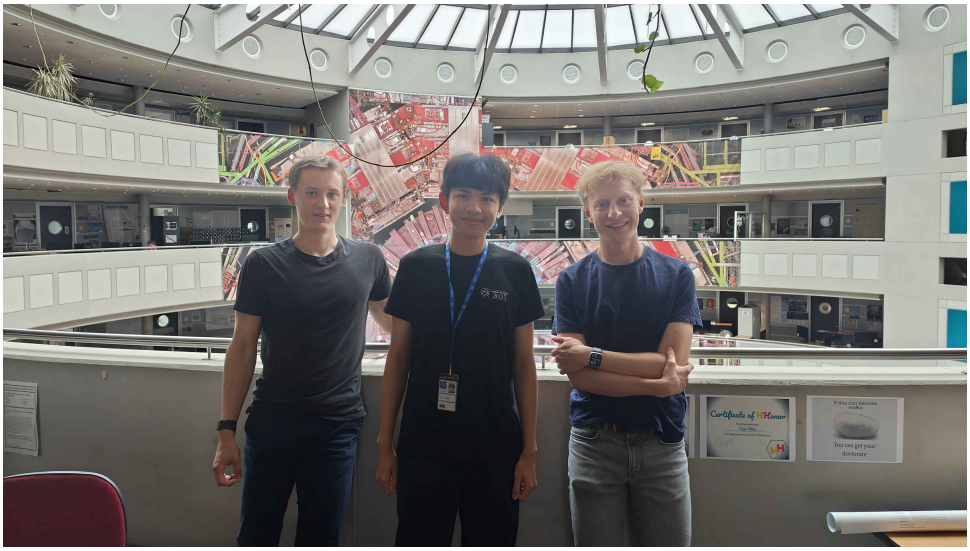
Log Summary: The primary focus was the completion of the final report. The latest training results were validated and included, and the full draft was submitted for a final review by supervisors.

- **Activities:**
- Validated the final set of training results.
- Completed the second half of the full report.
- Adjusted the first half of the report according to supervisor feedback.
- Submitted the full report draft to supervisors for final feedback.

Date: 29 August 2025

Log Summary: On the final day of the internship, the last revisions were made to the report based on supervisor feedback. A concluding discussion about machine learning research and future plans was held over lunch, followed by departure formalities.

- **Activities:**
- Performed the final revision of the report according to supervisor feedback.
- Held a discussion with supervisor Jonathan about recent research in machine learning and future plans.
- Completed the contact termination process and said farewells.



7 Biography

Education

- Final-Year Undergraduate Student, B.Eng. in Robotics and AI Engineering
  King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand
- Current GPAX: 3.89 / 4.00

Certifications

- Certified in Cybersecurity (CC), *ISC2*

Awards and Accomplishments

- Honorable Mention, Hackathon AI Cooking (2024)
- Merit Award, Asia Pacific ICT Alliance (APICTA) Awards (2021, 2022)
- Winner, Thailand ICT Award (TICTA) (2020, 2021)
- 1st Place, National Software Contest (NSC), Thailand (2019)

Research Interests

- Machine learning for high-dimensional data representation
- Alternative technique for multi-modal models

Future Plans

- To pursue a Ph.D. in Computer Science with a specialization in Machine Learning.

CERN SUMMER STUDENT PROGRAMME 2025

รายงานการเข้าร่วมโครงการนักศึกษาภาคฤดูร้อนเซิร์น