ML4DQM HCAL Run Classification using Vision Transformers

Daksh Mor

April 4, 2025

Abstract

This paper documents the development and evaluation of Vision Transformer (ViT) models for classifying Large Hadron Collider (LHC) Hadronic Calorimeter (HCAL) DigiOccupancy data. The objective is to distinguish between two specific data-taking periods, Run 355456 and Run 357479, based on 2D detector hit maps. We explore the characteristics of the HCAL data, detail the preprocessing steps including log-transformation, and present the architectures of a standard ViT and a Mixture-of-Experts ViT (MoE-ViT). Both models are trained and evaluated on accuracy and Area Under the ROC Curve (AUC). The standard ViT achieved perfect classification (1.0 accuracy, 1.0 AUC) on the validation set within few training epochs, while the MoE-ViT reached near-perfect performance (0.9985 accuracy, 1.0 AUC). Attention map analysis reveals that both models focus on physically relevant detector regions, with the MoE-ViT exhibiting potentially more specialized attention patterns.

1 Introduction

The Large Hadron Collider (LHC) at CERN produces vast amounts of complex data. Ensuring the quality of this data is paramount for physics analysis. Data Quality Monitoring (DQM) involves continuously checking detector performance and data integrity. Machine learning techniques offer promising avenues for automating and enhancing DQM tasks.

1.1 Project Goal

The primary goal of this project is to develop and evaluate machine learning models, specifically Vision Transformer (ViT) architectures, capable of classifying 2D hit maps (DigiOccupancy) from the Hadronic Calorimeter (HCAL) detector based on their source run. The task focuses on distinguishing between data originating from Run 355456 and Run 357479.

1.2 Task Summary

- **Input:** 2D detector hit maps representing HCAL DigiOccupancy (counts per channel).
- **Output:** Binary classification indicating the source run (Run 355456 or Run 357479).
- **Models:** Standard Vision Transformer (ViT) and Mixture-of-Experts ViT (MoE-ViT).
- **Evaluation Metrics:** Classification Accuracy, Receiver Operating Characteristic (ROC) Curve, and Area Under the Curve (AUC).

2 Data Description and Exploratory Data Analysis

The dataset consists of DigiOccupancy maps from two distinct LHC runs.

2.1 Dataset Description

Two primary datasets were used, corresponding to the two source runs:

- **Dataset 1:** Run355456
- **Dataset 2:** Run357479

Key parameters for each dataset are summarized in Table 1. The data exhibits significant sparsity, with approximately 80% zero-value entries.

Table 1: Dataset Parameters

Parameter	Value	Description
Shape per Dataset	(10000, 64, 72)	(Samples, iEta, iPhi)
Value Range (Run 1)	0.0 – ≈ 1565	Hit Multiplicity
Value Range (Run 2)	0.0 – ≈ 1092	Hit Multiplicity
Sparsity (% Zeros)	≈ 79.77%	Zero-value cells

2.2 HCAL Coordinate System

The DigiOccupancy maps are represented in detector coordinates:

- **iEta:** Pseudorapidity index (64 bins), related to the polar angle along the beam axis.
- **iPhi:** Azimuthal angle index (72 bins), representing the angle around the beam axis.

Figure 1 illustrates the unrolled detector plane, highlighting the active regions typically observed in the data.

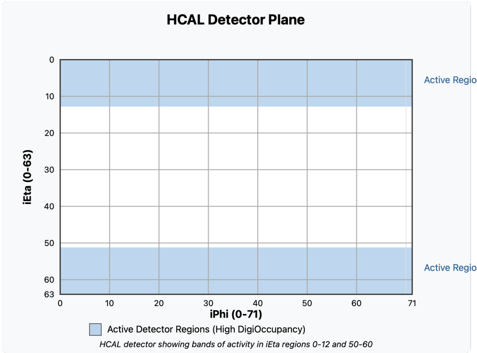

HCAL Detector Plane
iEta (0-63)
0
10
20
30
40
50
60
63
Active Region
Active Region
0
10
20
30
40
50
60
71
iPhi (0-71)
Active Detector Regions (High DigiOccupancy)
HCAL detector showing bands of activity in iEta regions 0-12 and 50-60

Figure 1: Schematic of the unrolled HCAL detector plane (iPhi vs. iEta). Shaded areas indicate the approximate location of active bands.

2.3 Basic Statistics

A comparison of basic statistical measures between the two runs is shown in Table 2. Notably, while the sparsity structure is identical, Run 357479 exhibits a higher mean hit value, suggesting intensity differences are key discriminators.

Table 2: Basic Statistical Comparison

Statistic	Run 355456	Run 357479
Min Value	0.0	0.0
Max Value	≈ 1565	≈ 1092
Mean Value	≈ 157.14	≈ 181.08
Median Value	0.0	0.0
Std Deviation	≈ 364.31	≈ 362.53
Zero % (Overall)	$\approx 79.77\%$	$\approx 79.77\%$

2.4 Key Visualizations & Observations

Exploratory visualizations provide insights into the data structure and differences between runs.

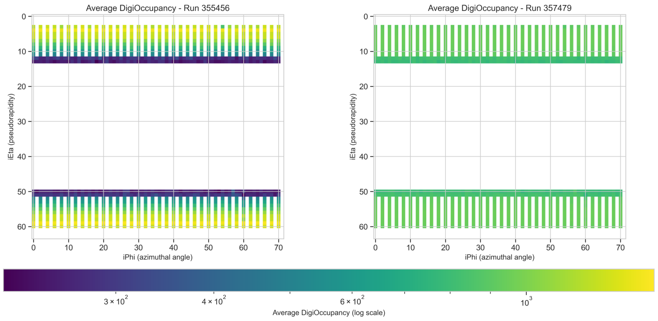


Figure 2: Average DigiOccupancy (log scale). Activity is concentrated in two bands (iEta $\approx 0 - 12and \approx 50 - 60).Run357479showshigheraverageintensity.$

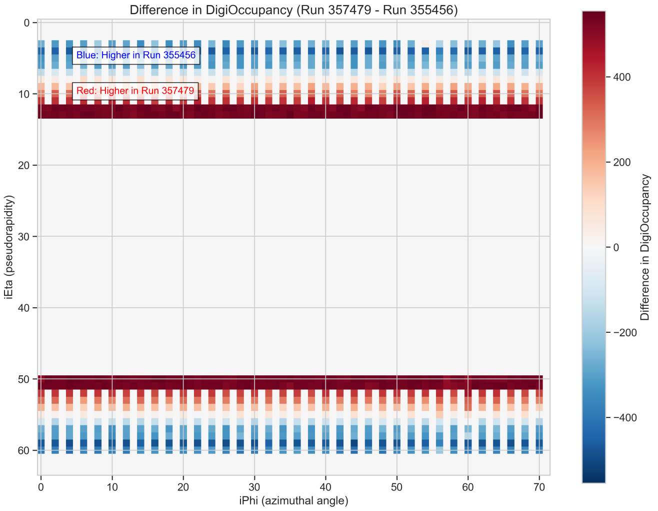


Figure 3: Difference map (Run 357479 - Run 355456). Red indicates higher intensity in Run 2, Blue higher in Run 1, mainly within active bands.

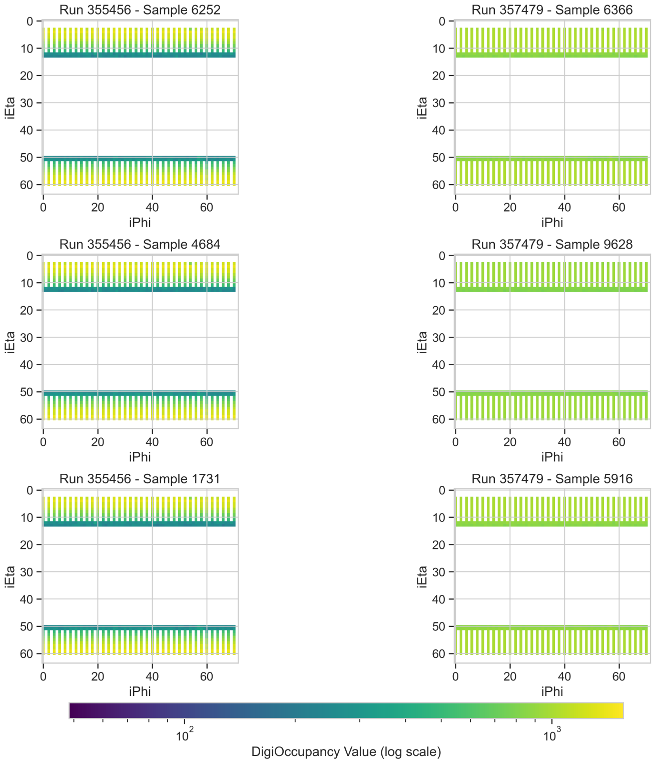


Figure 4: Random sample images confirm the band structure and sparsity. Intensity varies across individual luminosity sections.

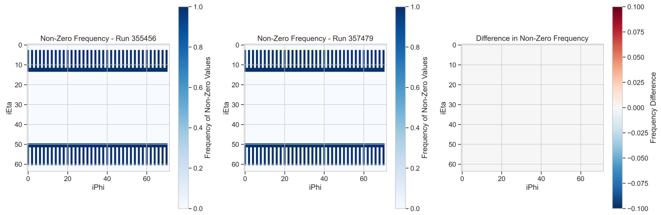


Figure 5: Frequency of non-zero hits. Active regions (Left/Middle) are nearly identical. Minimal difference in hit locations (Right) confirms intensity as the key differentiator.

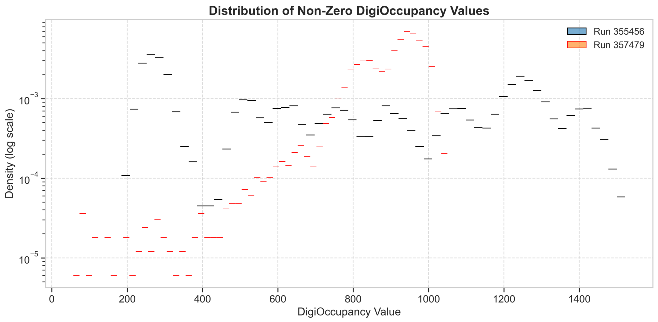


Figure 6: Distribution of non-zero DigiOccupancy values (log density scale). Similar shapes with subtle differences between runs.

2.5 EDA Conclusions

- Data is highly structured, sparse, with activity concentrated in specific iEta bands.
- Active regions are consistent across runs.
- The primary distinguishing feature between runs is the hit intensity within these active bands.
- The wide dynamic range necessitates normalization.
- Symmetry exists between the top and bottom active bands.
- Standard image augmentations like rotations or flips are likely inappropriate due to the physical meaning of the axes.

3 Data Preprocessing

A straightforward preprocessing pipeline was implemented to prepare the data for model training (Figure 7).

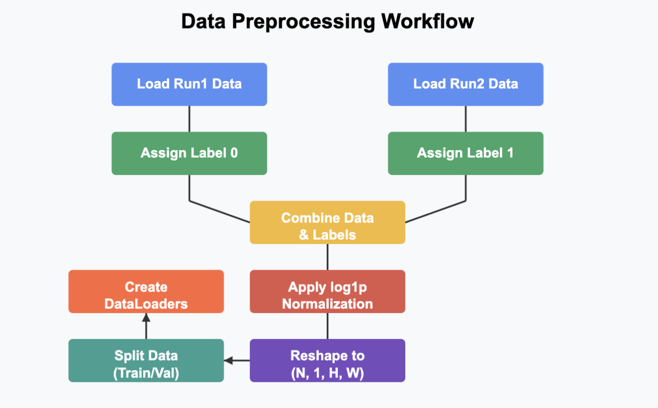


Figure 7: Data preprocessing workflow.

The steps involved are:

1. **Load:** Load run data from `.npy` files.
2. **Label:** Assign labels: 0 for Run 355456, 1 for Run 357479.
3. **Combine:** Concatenate data arrays and labels.
4. **Normalize:** Apply a logarithmic transformation, $X_{norm} = \log(1 + X)$, using `np.log1p`. This handles the large value range and data skewness while preserving zero entries.
5. **Reshape:** Add a channel dimension to conform to image input formats: (N, 64, 72) → (N, 1, 64, 72).
6. **Split:** Partition the data into training (80%) and validation (20%) sets (Table 3).
7. **DataLoaders:** Create PyTorch `DataLoader` instances for efficient batching during training.

Table 3: Data Split

Set	Percentage	# Samples
Training	80%	16000
Validation	20%	4000
Total	100%	20000

4 Model Architectures

Two Vision Transformer variants were employed for the classification task.

4.1 Standard Vision Transformer (ViT)

The standard ViT model [?] processes images by dividing them into patches, linearly embedding these patches, adding positional information, and feeding the resulting sequence of tokens through a series of Transformer blocks (Figure 8). A special classification ([CLS]) token is prepended to the sequence, and its final output state is used for classification.

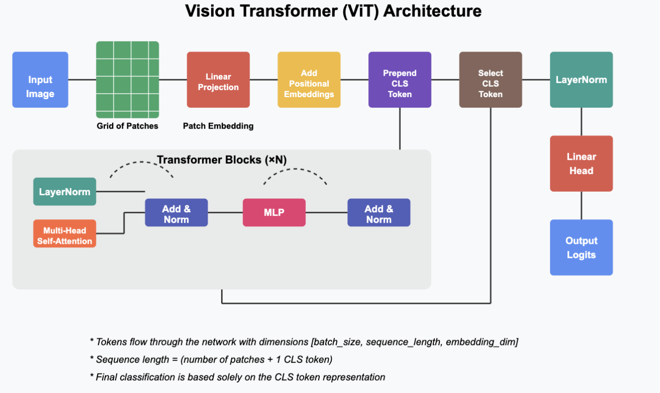

Vision Transformer (ViT) Architecture
Input Image
Grid of Patches
Linear Projection
Patch Embedding
Add Positional Embeddings
Prepend CLS Token
Select CLS Token
LayerNorm
Linear Head
Output Logits
Transformer Blocks (×N)
LayerNorm
Multi-Head Self-Attention
Add & Norm
MLP
Add & Norm
* Tokens flow through the network with dimensions [batch_size, sequence_length, embedding_dim]
* Sequence length = (number of patches + 1 CLS token)
* Final classification is based solely on the CLS token representation

Figure 8: Standard Vision Transformer Architecture.

Key components include: Patch Embedding, Positional Embedding, [CLS] Token, and Transformer Blocks containing Multi-Head Self-Attention (MSA) and Multi-Layer Perceptron (MLP) layers. Hyperparameters are listed in Table 4. Note the relatively large, non-square patch size chosen to suit the input dimensions and potential feature scale.

Table 4: Standard ViT Hyperparameters

Parameter	Value	Description
img_size	(64, 72)	Input dimensions (H, W)
patch_size	32 (8×4 patch grid)	Patch dimensions
in_channels	1	Grayscale input
num_classes	2	Binary classification
embed_dim	64	Embedding dimension
depth	4	# Transformer blocks
num_heads	2	# Attention heads
mlp_ratio	4.0	MLP hidden dim ratio
drop_rate	0.1	Dropout rate (MLP/Embed)
attn_drop_rate	0.0	Dropout rate (Attention)

4.2 Mixture-of-Experts ViT (MoE-ViT)

The MoE-ViT [?] modifies the standard ViT by replacing the MLP layer within each Transformer block with a Mixture-of-Experts (MoE) layer (Figure 9).

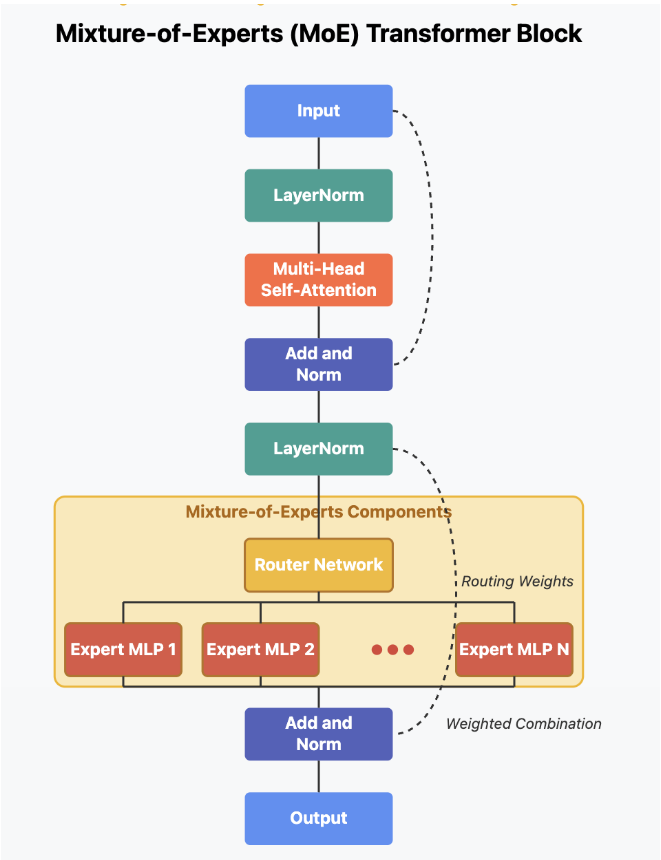


Figure 9: MoE Transformer Block Schematic.

The MoE layer consists of multiple independent 'expert' MLPs and a 'router' network. For each input token (patch), the router dynamically selects a sparse combination of experts to process it. This allows for model capacity to increase significantly while keeping computational cost manageable, potentially enabling expert specialization. The hyperparameters used for the MoE-ViT are shown in Table 5. Note the smaller embedding dimension, depth, and head count compared to the standard ViT configuration tested, as specified in the source implementation.

Table 5: MoE-ViT Hyperparameters

Parameter	Value	Description
img_size	(64, 72)	Input dimensions (H, W)
patch_size	32 (8×4 patch grid)	Patch dimensions
in_channels	1	Grayscale input
num_classes	2	Binary classification
embed_dim	32	Embedding dimension
depth	4	# MoE Transformer blocks
num_heads	2	# Attention heads
num_experts	2	# Experts per MoE block
mlp_ratio	4.0	MLP hidden dim ratio (per expert)
drop_rate	0.1	Dropout rate
attn_drop_rate	0.0	Dropout rate (Attention)

5 Training Methodology

The models were trained using a standard supervised learning procedure outlined in Figure 10.

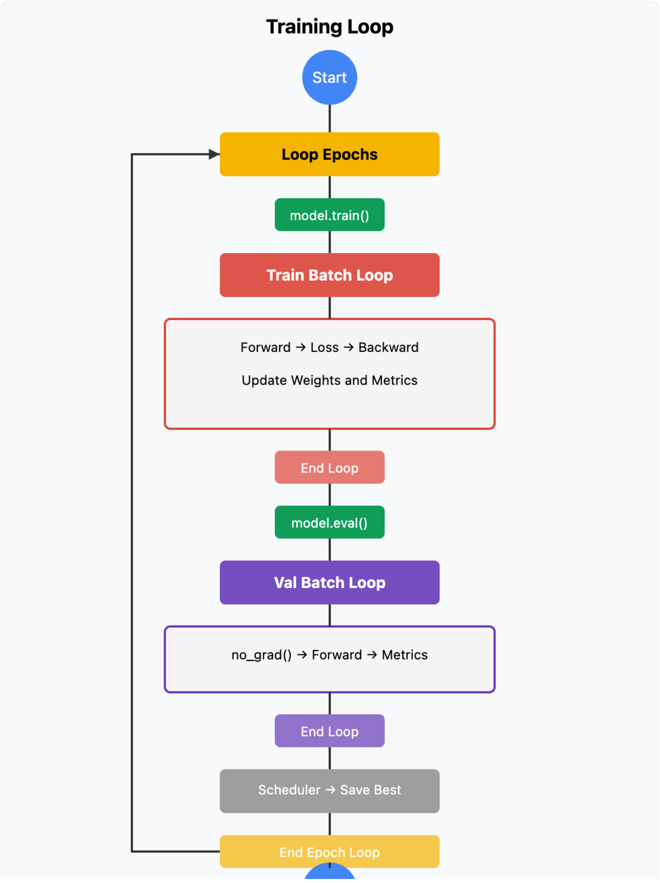


Figure 10: Training and Validation Loop.

5.1 Setup

- **Device:** CUDA-enabled GPU if available, otherwise CPU.
- **Loss Function:** `nn.CrossEntropyLoss`, suitable for multi-class classification (combines LogSoftmax and NLLLoss).
- **Optimizer:** `AdamW` [?], an Adam variant with improved weight decay handling.
- **Learning Rate Scheduler:** `ReduceLROnPlateau`. Reduces the learning rate when validation loss plateaus, aiding convergence.
- **Epochs:** Models were trained for 3 epochs based on initial observations of rapid convergence.

5.2 Training Curves

Figure 11 shows the training and validation loss and accuracy for the standard ViT. Figure 12 compares the training dynamics of both models. The standard ViT demonstrates faster initial convergence on the validation set within the short training period. The MoE-ViT, while starting slower, shows strong improvement and achieves high accuracy by the third epoch.

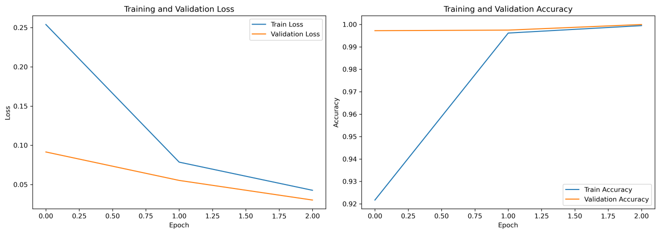


Figure 11: Standard ViT training and validation loss and accuracy over 3 epochs.

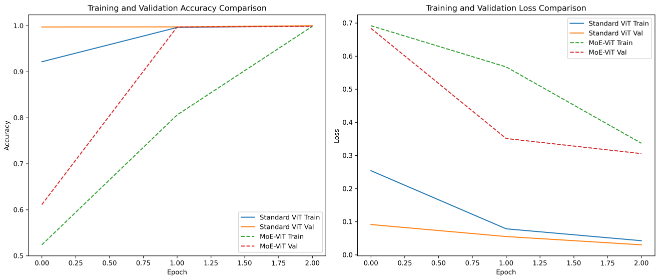


Figure 12: Comparison of training/validation loss and accuracy for Standard ViT vs. MoE-ViT.

6 Evaluation and Results

Model performance was assessed on the held-out validation set (4000 samples, 20% of total data).

6.1 Standard ViT Results

The standard ViT achieved perfect classification performance on the validation set after 3 epochs.

- **Accuracy:** 1.0000
- **AUC:** 1.0000

The confusion matrix (Figure 13) and ROC curve (Figure 14) confirm this perfect separation.

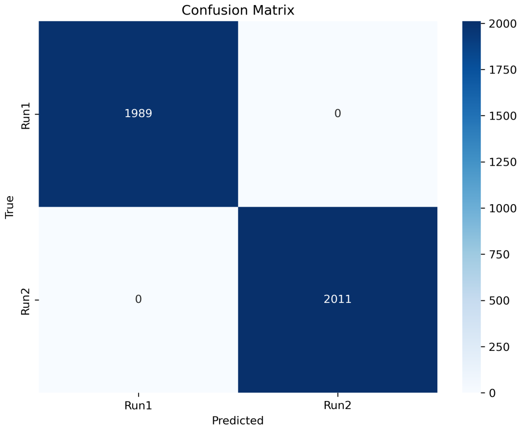


Figure 13: Standard ViT Confusion Matrix on Validation Set.

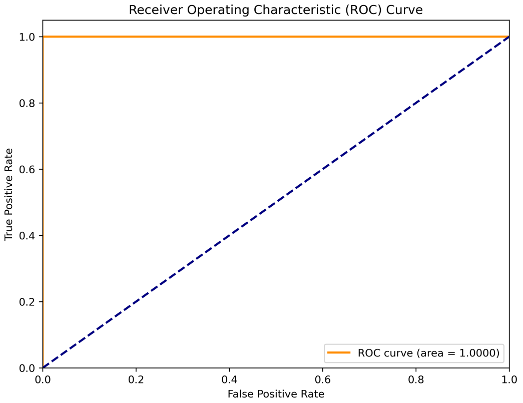


Figure 14: Standard ViT ROC Curve on Validation Set (AUC = 1.0).

6.2 MoE-ViT Results

The MoE-ViT also performed exceptionally well, achieving near-perfect scores.

- **Accuracy:** 0.9985 (6 misclassifications out of 4000)
- **AUC:** 1.0000

While slightly below the standard ViT's perfect score within 3 epochs, the MoE-ViT demonstrated strong learning capabilities, achieving perfect AUC and very high accuracy.

7 Attention Mechanism Analysis

To understand how the models make predictions, we visualize the attention patterns, specifically the attention weights applied by the [CLS] token to the input patch tokens in the final Transformer block. This indicates which parts of the input image are most influential for the classification decision.

7.1 Standard ViT Attention

Figure 15 shows attention maps for representative samples. The standard ViT consistently focuses its attention on the active horizontal bands (high iEta and low iEta regions) where the primary intensity differences between runs exist. The exact focus areas within these bands vary slightly depending on the specific input sample.

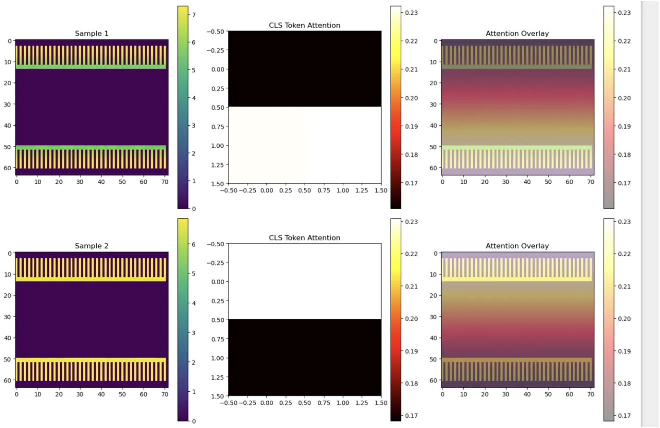
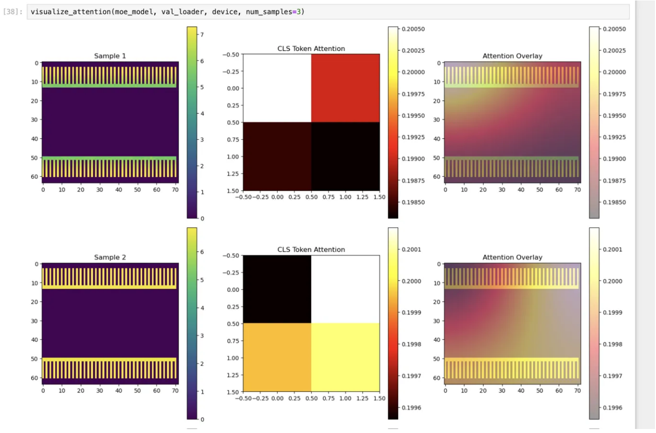


Figure 15: Standard ViT Attention: Input (Left), CLS token attention heatmap (Middle), Overlay (Right). Attention is focused on active detector bands.

7.2 MoE-ViT Attention

The MoE-ViT exhibits more structured, sometimes quadrant-like, attention patterns compared to the standard ViT (Figure 16). Interestingly, the attended quadrants or regions differ significantly between samples. This suggests that the router network might be dynamically assigning tokens from different spatial regions to specialized experts, leading to distinct attention patterns based on the input features.



```
[38]: visualize_attention(moe_model, val_loader, device, num_samples=3)
```

Figure 16: MoE-ViT Attention: Shows more structured, sample-dependent attention patterns, potentially reflecting expert specialization.

7.3 MoE Early Stopping Insight

It has been observed in MoE literature that with prolonged training, experts can sometimes converge towards similar functions, diminishing the benefits of the MoE architecture. The distinct attention patterns observed here might be characteristic of early-stage training where expert specialization is more pronounced. The high performance achieved quickly suggests that stopping training relatively early might be beneficial for retaining this potentially useful expert diversity for this specific task.

8 Conclusion

We successfully developed and evaluated Vision Transformer models for classifying HCAL DigiOccupancy data based on its source run.

- Both the standard ViT and MoE-ViT achieved very high classification performance on the validation set, distinguishing between Run 355456 and Run 357479 effectively.
- Exploratory data analysis identified hit intensity differences within consistent active detector regions as the key distinguishing feature.
- Logarithmic normalization (`log1p`) proved effective for preprocessing the sparse data with a wide value range.
- The standard ViT converged rapidly, achieving perfect accuracy and AUC within 3 epochs.
- The MoE-ViT also reached near-perfect accuracy and perfect AUC, exhibiting distinct, structured attention patterns potentially indicative of expert specialization, especially in early training stages.
- Attention visualization confirmed that both models learn to focus on the physically relevant active bands of the HCAL detector.

These results demonstrate the potential of ViT-based models for data quality monitoring tasks in high-energy physics, leveraging their ability to capture global spatial patterns in detector data.

9 Reproducibility

To reproduce the results presented in this paper:

- **Environment:** Set up a Python environment and install the required packages: `numpy`, `torch`, `torchvision`, `scikit-learn`, `matplotlib`, `seaborn`, `tqdm`. A requirements file or environment specification should be available in the code repository.
- **Data:** Obtain the datasets (`Run355456.npy`, `Run357479.npy`) and place them in an accessible directory (e.g., `./data/`), as expected by the code.
- **Code Repository:** The complete source code, including Jupyter notebooks (`EDA.ipynb`, `Model_Training.ipynb`), helper scripts, and detailed setup/execution instructions, is publicly available on GitHub: `https://github.com/daksh-mor/ML4DQM/edit/main/`
- **Execution:** Clone the repository and follow the instructions provided in the repository's README file to run the analysis notebooks or scripts.