

# Transformer-based Track Reconstruction with Low- $p_T$ Filtering Enhancement

Ivan Tang

Department of Physics, Nanjing University `yiwen.tang@smail.nju.edu.cn`

July 1, 2025

## Abstract

Accurate reconstruction of charged particle trajectories is a key component in high-luminosity collider experiments such as the HL-LHC. We present **UniTrackFormer**, a Transformer-based model that addresses the TrackML challenge: grouping spatial detector hits into individual particle tracks. Our pipeline integrates learnable hit filtering and track reconstruction using object queries and mask attention. We further explore a hit-level filtering enhancement to improve recovery of low-transverse-momentum ( $p_T$ ) tracks, which are typically under-reconstructed due to sparse signatures. Preliminary results demonstrate that our model captures spatial coherence among hits, offering a modular framework for further development.

## 1 Introduction

The reconstruction of charged particle trajectories from detector hits underpins much of the physics analysis at collider experiments. Traditional tracking algorithms have relied heavily on combinatorial pattern recognition or graph-based heuristics, which face significant scalability challenges in the presence of increasing event complexity and pileup, as anticipated at the High-Luminosity Large Hadron Collider (HL-LHC). The TrackML challenge provides a realistic simulated dataset that emulates the conditions of a general-purpose LHC detector, offering a standardized benchmark to develop and evaluate machine learning approaches for scalable and accurate track reconstruction.

In this work, we propose **UniTrackFormer**, an end-to-end Transformer-based pipeline that jointly performs hit filtering and track reconstruction. The goal is to assign each detector hit to its corresponding particle trajectory and regress the particle’s momentum. Our design emphasizes modularity and scalability, and targets improved low- $p_T$  reconstruction by integrating a learnable filtering stage upstream of a Transformer decoder.

## 2 Related Work

Traditional tracking algorithms in high-energy physics typically involve combinatorial pattern recognition or graph-based heuristics. However, the increasing complexity and event pileup at colliders such as the HL-LHC motivate machine learning approaches. The TrackML Challenge [2] formalized this problem in a public benchmark setting, while recent work like TrackFormer [1] has introduced Transformer-based architectures that integrate hit assignment and parameter regression in a unified framework. Our work builds upon these foundations by implementing a lightweight but modular architecture, with an emphasis on data-driven hit filtering.

## 3 Methodology

### 3.1 Dataset and Preprocessing

We utilize the TrackML dataset, which simulates proton-proton collision events at the LHC. Each event comprises four CSV files:

- `hits.csv`: containing  $(x, y, z)$  spatial coordinates and module identifiers;
- `truth.csv`: providing hit-to-particle associations and true track parameters;
- `particles.csv`: detailing initial vertex positions, momentum vectors  $\vec{p}$ , and particle charges;
- `cells.csv`: offering fine-grained detector readout such as strip or pixel charge.

Our data pipeline converts these inputs into PyTorch tensors. Hit features are encoded as 27-dimensional vectors, incorporating spatial coordinates, layer and module encodings, cylindrical coordinate transformations, and optional detector or cell information. Labels include binary masks indicating hit-to-track assignments, track validity flags, and 6-dimensional regression targets corresponding to  $(p_x, p_y, p_z, v_x, v_y, v_z)$  parameters derived from the truth data. We partition the dataset into training, validation, and test splits on an event-wise basis (e.g., 80%/10%/10%) to prevent data leakage.

### 3.2 Model Architecture

UniTrackFormer employs a two-stage architecture comprising hit filtering followed by track reconstruction.

#### 3.2.1 Hit Filtering Module

The hit filtering module is implemented as a lightweight Transformer encoder with embedding dimension  $d = 128$  and  $L = 4$  layers. We adopt windowed self-attention to efficiently capture local spatial correlations. Positional encodings are incorporated using cylindrical coordinates, including cyclic encoding of the azimuthal angle  $\phi$ . The encoder outputs embeddings that feed into a two-layer feed-forward classifier, which assigns a learned score to each hit. Hits are then filtered by retaining the top- $K$  scored candidates for downstream processing.

#### 3.2.2 Track Reconstruction Module

Inspired by MaskFormer and DETR architectures, the track reconstruction module utilizes  $N = 64$  learnable object queries. These queries attend over the filtered hit embeddings via multi-head attention within a Transformer decoder. Each query predicts:

- A binary classification score indicating the presence of a valid track;
- A 6-dimensional regression vector encoding track parameters (momentum and vertex);
- A mask over the  $M$  filtered hits, computed as a dot product between the query embedding and hit features, representing hit-to-track assignments.

### 3.3 Loss Function

The training objective is a multi-task loss combining classification, mask prediction, and parameter regression:

$$\mathcal{L} = \alpha \cdot \mathcal{L}_{\text{cls}} + \beta \cdot \mathcal{L}_{\text{mask}} + \gamma \cdot \mathcal{L}_{\text{param}}$$

where

- $\mathcal{L}_{\text{cls}}$  is the binary cross-entropy loss on track validity classification;
- $\mathcal{L}_{\text{mask}}$  combines Dice loss and binary cross-entropy to supervise hit assignment masks;
- $\mathcal{L}_{\text{param}}$  is the mean squared error loss on the 6D track parameter regression.

The hyperparameters  $\alpha, \beta, \gamma$  balance the contributions of each term.

## 4 Experiments and Results

### 4.1 Training Protocol

We conduct training over 5 epochs on a subset of TrackML events, employing the Adam optimizer with a learning rate of  $10^{-4}$ . The model converges quickly under limited epochs, indicating effective representation learning even in low-resource settings.

### 4.2 Qualitative Evaluation

Figures 1 and 2 illustrate the spatial distribution of hits and their predicted track assignments compared to ground truth labels. Despite limited hyperparameter tuning, the model effectively captures spatial structures and produces coherent clusters of hits corresponding to particle trajectories.

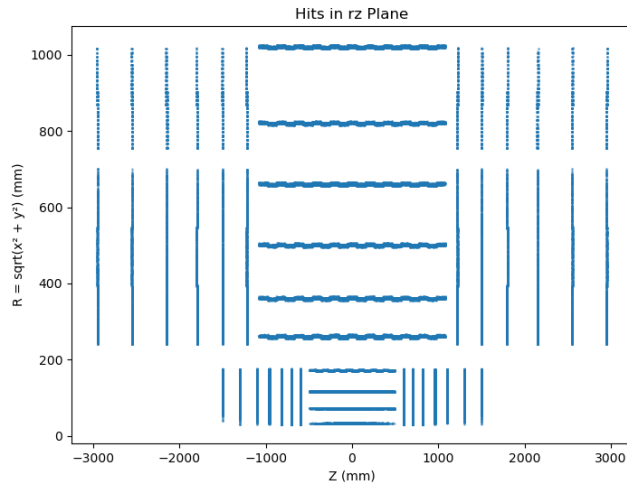


Figure 1: RZ projection of detector hits colored by predicted track assignments.

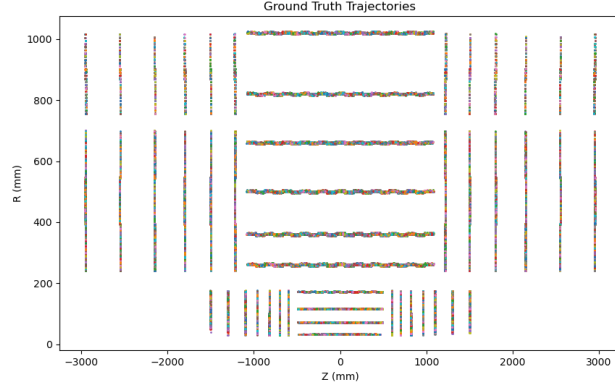


Figure 2: Visualization of predicted track assignments at the hit level. Points with the same color correspond to hits produced by the same reconstructed track.

### 4.3 Additional Hit Visualizations

To better understand the detector hit distributions and spatial patterns, we include several projections:

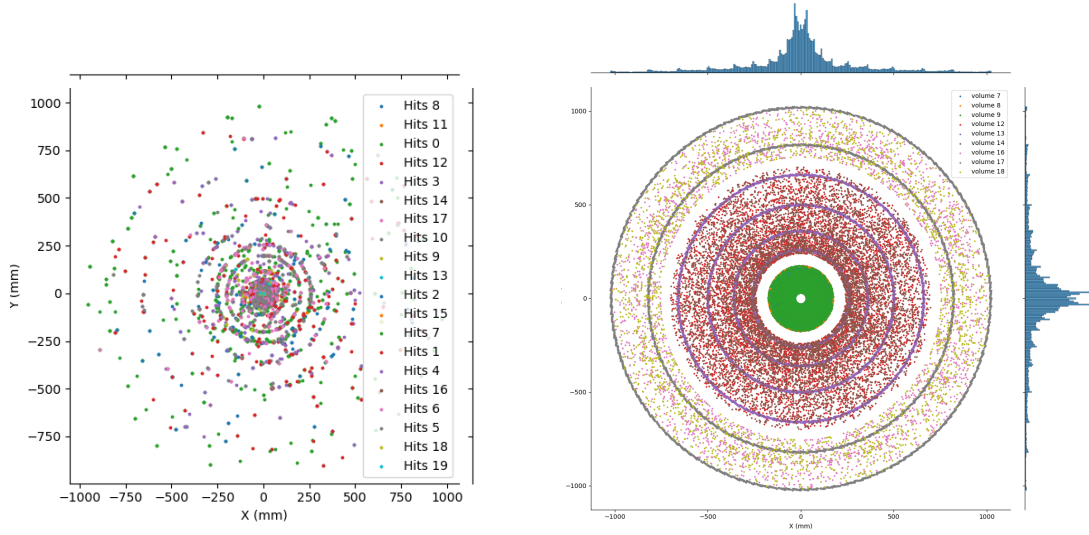


Figure 3: XY projections of hits before and after filtering.

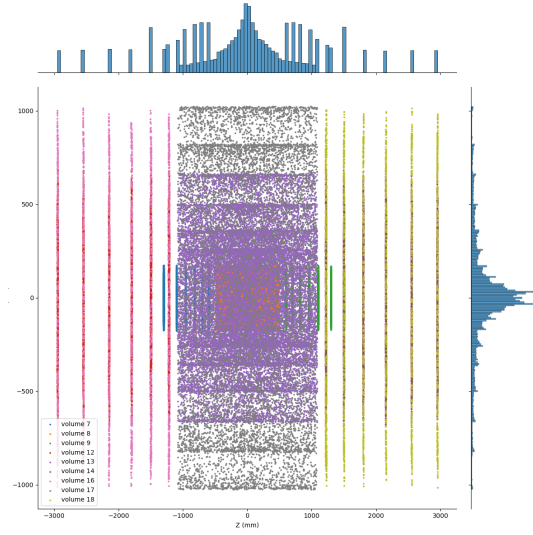


Figure 4: YZ projection showing vertical detector geometry.

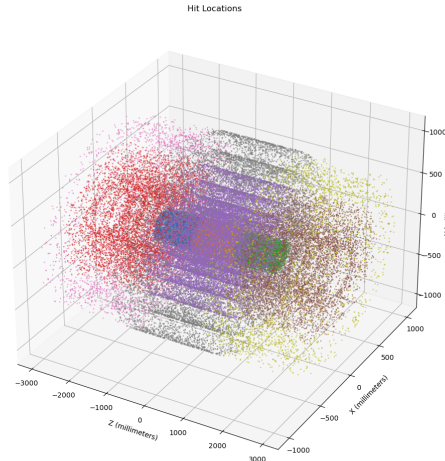


Figure 5: 3D visualization of hits in cylindrical detector geometry.

#### 4.4 Loss Curve

The training and validation loss curves depicted in Figure 6 demonstrate stable convergence behavior throughout the training process.

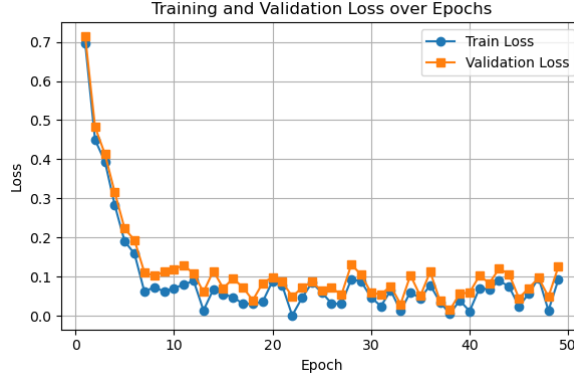


Figure 6: Training loss progression over epochs.

## 5 Discussion

While our results demonstrate that UniTrackFormer successfully groups hits into physically plausible tracks, there remain several avenues for improvement. One prominent challenge is the reconstruction of low- $p_T$  tracks, which often leave sparse and noisy signatures across the detector. Naively applying a top- $K$  filter based on classification score may disproportionately exclude these trajectories. We propose mitigating this via a separate Transformer-based filtering stage trained with physics-informed objectives to better distinguish signal hits from background.

In addition, further improvements may stem from incorporating temporal or sequential features, for example by leveraging the radial ordering of hits or augmenting positional encodings with additional geometric or temporal information. A comprehensive evaluation employing physics-aware metrics such as tracking efficiency, purity, and fake rate is essential to benchmark performance relative to existing algorithms and to guide future model refinements.

## 6 Conclusion

We introduced UniTrackFormer, a Transformer-based architecture for end-to-end particle track reconstruction under realistic collider conditions. Our modular architecture integrates learnable hit filtering with object query-based track decoding, demonstrating promising qualitative results on the TrackML dataset. Future work will focus on enhancing low- $p_T$  track recovery, scaling to full datasets, and performing rigorous quantitative evaluations using physics-motivated metrics. These efforts aim to advance the applicability of deep learning methods for real-time and high-precision tracking in next-generation collider experiments.

## Acknowledgements

The author gratefully acknowledges the guidance and support of Prof. Ligang Xia from the School of Physics, Nanjing University.

## References

- [1] S. Van Stroud et al., *TrackFormer: End-to-End Reconstruction with Transformers*, arXiv:2411.07149 (2024).
- [2] TrackML Challenge Organizers, *The TrackML Particle Tracking Challenge*, EPJ Web Conf. 150, 00037 (2018).