Transformers for Charged Particle Track Reconstruction in High Energy Physics

A Preprint

Samuel Van Stroud[1], Philippa Duckett[1], Max Hart[1], Nikita Pond[1],
Sébastien Rettie[1,2], Gabriel Facini[1], and Tim Scanlon[1]
[1]*Centre for Data Intensive Science and Industry, University College London*, [2]*CERN*

November 12, 2024

Abstract

Reconstructing charged particle tracks is a fundamental task in modern collider experiments. The unprecedented particle multiplicities expected at the High-Luminosity Large Hadron Collider pose significant challenges for track reconstruction, where traditional algorithms become computationally infeasible. To address this challenge, we present a novel learned approach to track reconstruction that adapts recent advances in computer vision and object detection. Our architecture combines a Transformer hit filtering network with a MaskFormer reconstruction model that jointly optimises hit assignments and the estimation of the charged particles' properties. Evaluated on the TrackML dataset, our best performing model achieves state-of-the-art tracking performance with 97% efficiency for a fake rate of 0.6%, and inference times of 100 ms. Our tunable approach enables specialisation for specific applications like triggering systems, while its underlying principles can be extended to other reconstruction challenges in high energy physics. This work demonstrates the potential of modern deep learning architectures to address emerging computational challenges in particle physics while maintaining the precision required for groundbreaking physics analysis.

Keywords Particle Physics · Track Reconstruction · TrackML · Machine Learning · Object Detection · Transformers

1 Introduction

Particle colliders such as the Large Hadron Collider (LHC) [1] at CERN have deepened our understanding of fundamental physics through the analysis of high-energy proton-proton (pp) collisions. These interactions generate a multitude of subatomic particles that propagate outward from the collision point through surrounding detectors. The trajectories of charged particles (*tracks*) are an essential observable in general purpose collider experiments, as they reveal the presence of the charged particles produced in a collision and provide information about their properties, such as momenta and charge [2, 3]. The accurate measurement of charged particles is crucial for many downstream tasks, such as lepton reconstruction [4–7], hadronic τ reconstruction [8, 9], particle flow jet algorithms [10, 11], and jet flavour identification [12, 13]. Collectively, these processes underpin nearly all analyses conducted at LHC experiments, for example searches for and precision measurements of rare processes such as Higgs boson pair production [14, 15].

Particle trajectories are observed via the interaction of charged particles with sensitive detector elements in specialised silicon tracking detectors. In general purpose collider experiments, these detectors are arranged around the collision region in concentric cylindrical (barrel) layers, with disks (endcaps) at each open end. The resulting discrete 3-dimensional position measurements are referred to as *hits*. Tracks are formed by grouping compatible hits and extracting the track parameters via fitting a trajectory to the assigned hits. The complexity of track reconstruction increases with the particle and hit multiplicity, both of which will increase dramatically with the planned High-Luminosity LHC (HL-LHC) upgrade [16]. The HL-LHC is designed to increase the LHC's collision rate by up to a factor of seven, enabling up to 200 simultaneous pp interactions and generating $\mathcal{O}(10\text{k})$ particles with $\mathcal{O}(100\text{k})$ hits per event. This dramatic increase in hit multiplicity presents significant challenges for conventional approaches to track reconstruction, which exhibit scaling behaviour that is worse than quadratic in the number of hits and leads to prohibitive computing resource requirements [17, 18]. Under HL-LHC conditions, track reconstruction is expected to dominate the CPU budget for offline track reconstruction at ATLAS [19, 20], making the development of fast and scalable tracking algorithms critical to ensuring feasible computing demands. These new approaches must not only be computationally efficient but also maintain or improve physics performance in this demanding new environment.

Aside from general purpose collider experiments, specialised LHC detectors like LHCb [21], dedicated to heavy flavour physics, and ALICE [22], which focuses on heavy-ion collisions, as well as experiments beyond the LHC such as Mu3e [23], also depend on precise tracking capabilities to resolve particle trajectories in high-density and low-momentum environments. These experiments, each with unique detector configurations and tracking requirements, highlight the need for flexible community-wide solutions that can easily be adapted to a range of particle densities, momenta, and detector geometries.

In this work, we introduce a novel method for fast and tunable track reconstruction using MaskFormer [24], a Transformer-based architecture [25] originally developed for image object detection. Our approach processes a set of input hits and reconstructs trajectories by simultaneously assigning hits to tracks and estimating track parameters, removing the need for dedicated off-model pre- and post-processing steps. By leveraging the efficiency and scalability of Transformers, we address the computational bottlenecks of traditional algorithms and meet the stringent demands of the HL-LHC environment. Our model achieves state-of-the-art performance in both accuracy and speed, successfully reconstructing particles with $p_{\mathrm{T}} > 600$ MeV – a notable improvement over existing machine learning (ML) approaches. It also naturally allows for multiple assignment of hits to tracks. Previous success in applying Transformer-based models to vertex reconstruction [26] further demonstrates the potential of this architecture to generalize across varied particle physics reconstruction tasks.

Our approach relies on a stand-alone hit filtering stage to reduce the input hit multiplicity before passing them to the track reconstruction MaskFormer. This filtering step is conceptually simple and computationally efficient, and is able to effectively replace the graph construction stage used in the previous approaches described in Section 2. Given the strong performance of the Transformer-based hit filtering model, we anticipate that this approach may benefit other approaches to track reconstruction, including traditional algorithms.

The structure of this article is as follows. In Section 2, we present an overview of related research. Section 3 details the model architecture, and Section 4 outlines the simulated samples and training procedures used. The results are discussed in Section 5, followed by the conclusions in Section 6.

2 Related Work

GNN4ITK Graph Neural Networks (GNNs) [27] have emerged as a leading approach to address the computational challenges of track reconstruction at the HL-LHC. The Exa.TrkX collaboration has developed a multi-stage GNN pipeline for processing detector hits called GNN4ITK [28, 29]. The method begins with a graph construction preprocessing step to define initial node connectivity. A GNN then scores the edges in this graph, and edge filtering is applied to low-probability connections. Finally, track candidates are extracted in a postprocessing step through an iterative graph segmentation algorithm. While this pipeline demonstrates promising performance and scalability, the initial graph construction stage remains a computational bottleneck.

HGNN Hierarchical Graph Neural Networks (HGNN) [30] have been proposed as an alternative to the iterative graph segmentation approach used by GNN4ITK. The method groups nodes into *super-nodes*, enlarging the receptive field of the graph convolutions and allowing message passing across disconnected graph components. While HGNN improves track reconstruction efficiency, it has a high fake rate that would have to be reduced via additional post-processing and an increased computational cost, both of which raise concerns about its viability for HL-LHC deployment.

Object Condensation Object condensation (OC) [31] is an approach to track reconstruction in which hits are clustered in a learned latent space. The method selects representative hits to characterise entire tracks, with remaining hits assigned through an off-model clustering algorithm. Similar to other approaches, OC relies on an edge classification step to reduce hit connections. We hypothesise that OC's reliance on single representative hits to characterise entire tracks may limit its effectiveness, as it couples hit feature extraction and track reconstruction stages. Additionally, the assignment of hits to tracks is not explicitly provided by the model, and the current clustering-based assignment does not support the assignment of hits to multiple tracks.

Experimental Transformer approaches Transformers have been explored for track reconstruction in previous works [32–34], which either directly classify hits or rely on modelling tracks as a sequence of hits and training autoregressive models. Detailed comparisons with these approaches are omitted as they have not yet been validated on the full TrackML dataset or do not fully characterise tracking performance in terms of efficiency, fake-rate and timing.

Computer Vision Recent advancements in computer vision, particularly in object detection and instance segmentation, offer a promising alternative to the geometric deep learning techniques used by existing ML-based track reconstruction efforts. In computer vision, object detection has evolved from simple bounding box prediction [35, 36] to sophisticated pixel-level instance segmentation [24, 37]. The MaskFormer architecture [24, 38], built on Transformer networks [25], has been particularly successful in simultaneously handling semantic and instance segmentation tasks. This approach distinguishes multiple instances of the same object class through unique instance masks while maintaining shared class labels —- a capability directly relevant to particle track reconstruction. By replacing the image pixels with an unordered set of hits and the image-depicted objects with charged particle tracks, we can leverage the MaskFormer architecture to address the challenges of track reconstruction.

MaskFormers in High Energy Physics The utility of computer vision approaches in particle physics was recently demonstrated in Ref. [26], which successfully adapted the MaskFormer architecture to reconstruct displaced secondary decay vertices. While particle physics tasks have traditionally relied on specialised architectures designed for specific problems, this work showed that a single versatile architecture can effectively handle unordered sets of particle physics data, successfully identifying and characterising multiple vertices within jets. We extend this by applying the same architectural foundation to the distinct challenge of track reconstruction. The successful application of a common architecture to two different reconstruction tasks -— vertex finding and track reconstruction —- suggests a promising path toward more unified approaches in particle physics reconstruction, potentially reducing the field's reliance on task-specific solutions.

Table 1 summarises the key aspects of the various ML-based approaches to charged particle track reconstruction discussed in this section, including a comparison of the minimum transverse momentum ($p_{\mathrm{T}}^{\mathrm{min}}$) and maximum pseudorapidity (η) used to define target particles (for more information see Section 4).

	$p_{\mathrm{T}}^{\mathrm{min}}$	max $\lvert\eta\rvert$	Layers Used	Preprocessing	Postprocessing
GNN4ITK [29]	1 GeV	4.0	Pixel + strip	Edge classification	Graph traversal
HGNN [30]	1 GeV	4.0	Pixel + strip	Edge classification	GMPool [30]
OC [31]	900 MeV	4.0	Pixel	Edge classification	Clustering
This Work	600 MeV	2.5	Pixel	Hit filtering	None

Table 1: Comparison of ML-based approaches to charged particle track reconstruction. Tracking detectors at general purpose collider experiments are composed of pixel and strip layers, see Ref. [39] for more information. More information about the dataset can be found in Section 4.1.

3 Model Architecture

3.1 Leveraging Transformers

Transformers [25] have revolutionised natural language processing and computer vision tasks, offering a scalable and efficient architecture for processing sequential data. However, due to their quadratic complexity in the number of input tokens, custom GNN-based architectures have so far been preferred for particle tracking tasks [29, 31].

To neutralise the quadratic complexity of Transformers, we hypothesise that hits only need to attend to nearby hits in the azimuthal angle ϕ, and apply this powerful prior of ϕ-locality by ordering hits in ϕ and applying sliding window attention [40] with a window size w. This results in $\mathcal{O}(M \times w)$ complexity scaling, which is linear in the number of input hits M. To allow hits to communicate around the $\pm\pi$ boundary, the first $w/2$ hits are appended to the end of the sequence and vice versa. We further benefit from the fused attention kernels provided by `FlashAttention2` [41] for efficient computation, and `SwiGLU` activation [42] for improved performance.

This approach allows us to efficiently encode 60k pixel hits in $25\,\mathrm{ms}$ on a single GPU, making it suitable for low-latency trigger-level track reconstruction at the HL-LHC. The encoder is used in both the hit filtering and track reconstruction stages of our model, as described in the following sections. Detailed timing results can be found in Section 5.3.

3.2 Hit Filtering Model

The large number of hits present in each event makes it computationally infeasible to feed all hits directly into the tracking model. As described in Section 2, previous approaches to ML-based track reconstruction have required a graph construction stage based on geometric constraints and/or edge classification to reduce the input hit multiplicities. In contrast, we introduce a Transformer-based hit filtering model to reduce the input hit multiplicities by directly predicting whether each hit is noise or signal. Noise hits are defined as hits belonging to particles that we do not wish to reconstruct, for example, particles below a p_{T} threshold or outside a specified pseudorapidity range, in addition to the intrinsic noise hits which do not belong to any simulation-level particle.

The hit filtering model is composed of an embedding layer, a Transformer encoder, and a dense hit-level classifier. In the model, the input features of each hit are first passed through an embedding layer to produce a $d = 256$-dimensional representation for each of the M hits. Positional encodings are applied [25] in the cylindrical hit coordinates (see Section 4.1) to allow the self-attention (SA) mechanism to readily identify nearby hits. In particular, a cyclic positional encoding [43] is used for ϕ. The initial hit embeddings are then passed into an efficient SA Transformer encoder as described in Section 3.1. The encoder has twelve layers (eight for the $1\,\mathrm{GeV}$ model, see Section 4), model dimension d, and feed-forward dimension $2d$. A window size of $w = 1024$ is used for the sliding window attention. The output embeddings are classified using a dense network with three hidden layers.

The hit stand-alone filtering step offers broad applicability beyond our MaskFormer-based approach to track reconstruction. It could be integrated as a pre-processing step into other traditional and ML-based tracking pipelines, or repurposed for other tasks such as pileup mitigation.

3.3 Track Reconstruction Model

The tracking model follows an encoder-decoder architecture as shown in Fig. 1, with the encoder processing the input hits and the decoder reconstructing the tracks. The encoder model is the same as the hit filtering model, except for the window size which is decreased to 512 to compensate for the reduced hit multiplicities after filtering. The object decoder is a MaskFormer-based model [24, 38, 44] which forms an explicit latent representation for each track candidate, allowing for joint optimisation of hit assignments and track parameters. Similar to Ref. [26], we include modifications to handle sparse inputs, and to allow additional regression tasks for the reconstructed tracks.

The object decoder initialises a set of N object queries as learned vectors of dimension d. Each object query represents a possible output track. The number of object queries N sets the maximum number of tracks that can be reconstructed per event, and is chosen as the maximum number of tracks over the events in the training sample (described in Section 4.1). The object queries are then passed through eight decoder layers. In each layer, the object queries aggregate information from relevant hits via bi-directional cross-attention (CA), and from other object queries via SA. The `MaskAttention` operator [38] is used to generate attention masks from the intermediate mask proposals produced by the preceding decoder layer, encouraging object queries to attend only to relevant hits. The resulting object queries from the object decoder are processed by three task heads, which predict the track class, track-to-hit assignments, and track properties. The number of decoder layers and model dimension d was chosen to achieve good performance with reasonable training and inference times..

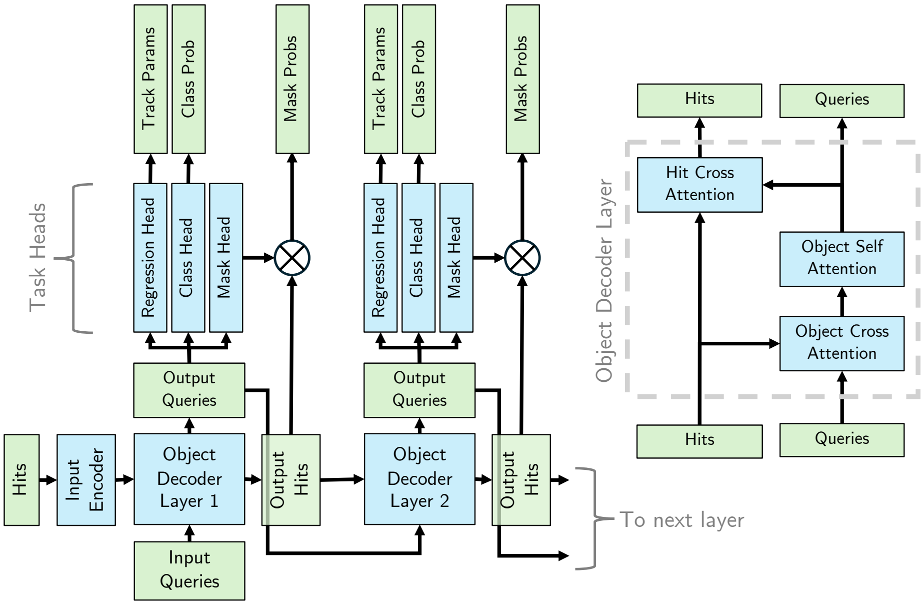

Figure 1: Overview of the track reconstruction model, with data and operations being shown in green and blue respectively. M input tokens representing the hits are fed into an initial Transformer encoder. The object decoder then takes a set of N object queries, which represent tracks, and iteratively updates them with information from the input elements and other object queries. Finally, three task heads are used to: categorise each object as being from one of $C + 1$ object classes (including a NULL class), estimate R regression targets, and predict $N \times M$ binary masks which provide the assignment of input hits to output tracks. The resulting embedded queries and hits are then fed into another decoder layer to refine the predictions and produce an intermediate auxiliary loss for each layer. These decoder layers can be stacked repeatedly to increase accuracy at the expense of computational cost. On the right, a detailed view of an object decoder layer is shown.

Since the number of object queries is fixed, but the number of tracks in each event is variable, a dense binary classifier with a single hidden layer of size d is used to predict whether each object query corresponds to a track or not. If not, other outputs for that object query are ignored. The assignment of each of the hits to tracks is given by an M-dimensional mask over the input hits for each of the N object queries. Mask tokens are formed from the query embeddings via a dense network with a single hidden layer of size $2d$. The mask for each hit is computed by taking the dot product between the updated hit embeddings from the object decoder and the query mask token for each object query. After applying a sigmoid activation, the mask is binarised, and a value of 1 indicates assignment of the hit to the track, while 0 indicates no assignment. The choice of sigmoid activation allows the model to assign a single hit to multiple tracks. Track parameters are regressed using a dense network with four output nodes, with each node corresponding to an output target. We regress the particle 3-momenta in Cartesian space (p_x, p_y, p_z), along with the z position of the production vertex v_z. The total momentum is not regressed directly, but rather calculated from the component predictions and added to the loss to enforce consistency. Other track parameters, such as the track transverse momentum p_T, angle ϕ and pseudorapidity η, are derived from the regressed Cartesian components of the momentum.

The training loss is the sum of the loss terms from each of the tasks. For the object class, a categorical cross-entropy loss L_CE is used. For the regression targets, a SmoothL1 loss [45] $L_\text{Regression}$ is used. The mask loss L_Mask is a combination of a Dice loss L_Dice [46] and focal loss L_Focal [47]. The total loss is then the weighted sum

$$L = 0.1L_\text{CE} + \underbrace{2L_\text{Dice} + 50L_\text{Focal}}_{L_\text{Mask}} + 0.1L_\text{Regression}, \tag{1}$$

where the weights are coarsely optimised to provide an good trade-off between the performance of the different tasks. To ensure that the loss is invariant over permutations of the object queries, the loss is defined using the optimal bipartite matching between the predicted and target objects, as computed with an efficient linear assignment problem solver [48]. Regression targets are scaled to be of order $\sim$1 during the loss computation so that they do not dominate the loss or interfere with the matching process. Finally, the intermediate outputs from each decoder layer are used to compute auxiliary loss terms which are included in the total loss [38].

4 Dataset & Experimental Setup

4.1 Dataset

The TrackML challenge [49–51] was established to promote the development of novel techniques for particle track reconstruction in the dense environments anticipated at the HL-LHC. The TrackML dataset has since become a standard benchmark for evaluating track reconstruction methods. The dataset simulates a generalised LHC-like detector, inspired by planned ATLAS and CMS upgrades for the HL-LHC, and is composed of approximately nine thousand simulated events, each containing an average of 200 simultaneous pp interactions. A sample event is depicted in Fig. 2. As shown in Fig. 3, $\mathcal{O}(10\text{k})$ particles and $\mathcal{O}(100\text{k})$ hits are simulated per event, resulting in $\mathcal{O}(100\text{M})$ tracks to be identified from approximately $\mathcal{O}(1\text{B})$ hits across the entire dataset.

For each event, TrackML provides information about simulated hits, particles, and their associations. The detector geometry follows a typical collider layout, with distinct tracking subsystems arranged in concentric layers around the beam axis. The innermost tracking system consists of silicon pixel sensors arranged in four cylindrical barrel layers, complemented by seven pixel endcap disks on each side. The outer tracking system comprises strip detectors with six barrel layers and six endcap disks per side. In this work, we focus exclusively on the innermost pixel detector, including the central barrel and endcap layers. This restricted geometry may present a more difficult challenge than using all layers, as on average only 4-6 hits are available per track. However the computational complexity is reduced, making it suitable for our initial studies.

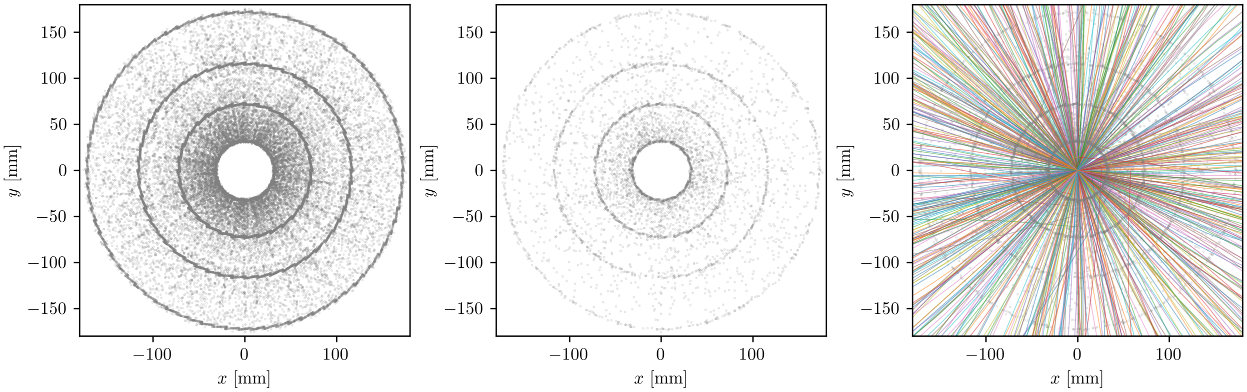

Figure 2: (Left) The positions of pixel hits in the x-y plane, facing down the beam-line, for a single event. (Middle) Hits passing the MF-1 GeV filter at a cut of 0.1. (Right) Filtered hits along with the trajectories of reconstructable particles that satisfy $p_\text{T} > 1\,\text{GeV}$ and $|\eta| < 2.5$.

Hits are 3-dimensional spacepoints formed from pixel elements which have been clustered together. The model is given information about the position of each hit and the shape and orientation of the corresponding clusters. For the hit position, the superset of the Cartesian and cylindrical coordinates, each defined with the origin at the geometric centre of the detector, is used as this was found to improve convergence during training. Additional position information is provided in the form of the hit pseudorapidity η, and the conformal tracking coordinates [52]. Finally, information about the associated cluster is provided in the form of the charge fraction (the sum of the charge in the cluster divided by the number of activated channels), the cluster size, and the cluster shape, following the approach in [53].

4.2 Experiments & Training Setup

We aim to reconstruct particles that satisfy the following criteria: at least three hits in the pixel layers, pseudorapidity $|\eta| < 2.5$, and transverse momentum exceeding a variable threshold p_T^min, which is set to $600\,\text{MeV}$, $750\,\text{MeV}$, and $1\,\text{GeV}$ in different experiments.

For each experiment, a hit filtering model and a tracking model are trained. The hit filtering models are referred to HF-p_T^min. The HF-$600\,\text{MeV}$, HF-$750\,\text{MeV}$ models each have approximately 8M trainable parameters, while the HF-$1\,\text{GeV}$ model, which was optimised to minimise inference times, has approximately 5M parameters. A track reconstruction model is then trained on each set of filtered hits to reconstruct tracks with the same p_T^min thresholds, which are similarly referred to as MF-p_T^min. Each tracking model has approximately 22M trainable parameters. A summary of the different experiments is provided in Table 2. The different choices of p_T^min allow us to explore the

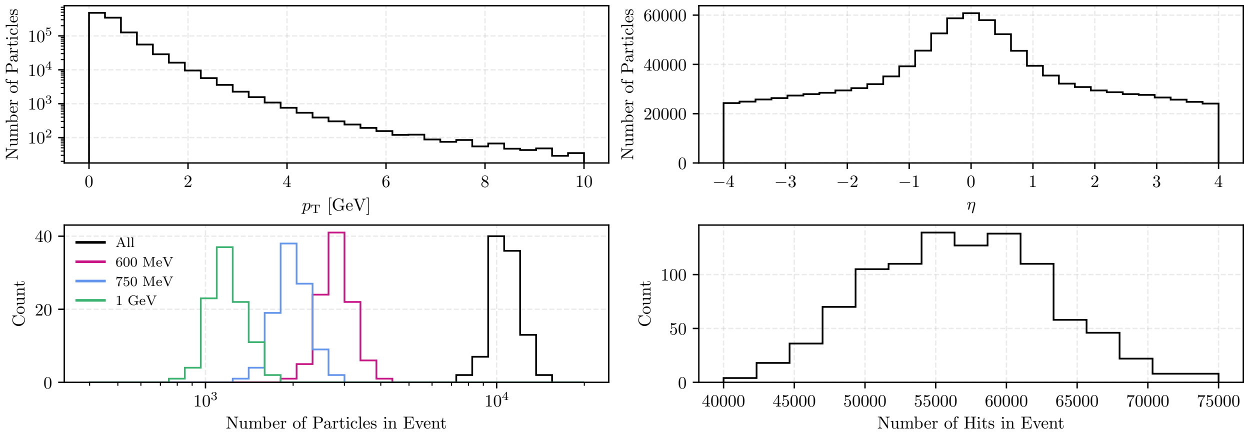

Figure 3: Histograms summarising the kinematics and object multiplicities of the TrackML dataset restricted to the inner pixel layers. (Top left) The p_T distribution of all particles. (Top right) A histogram showing the distribution of the pseudorapidity η of the particles. (Bottom left) The distribution of the number of total particles in each event. The distribution for all particles is shown in black. Also shown are the number of particles left after applying the three p_T cuts used in this work. (Bottom right) The distribution of the number of hits in the events before filtering.

trade-offs between model complexity, inference time, and performance. They also provide an insight into the impact of reducing the effective training statistics (i.e. the number of target particles), as discussed in Section 5.

For each of the validation and testing sets, 100 events are reserved with the remaining events used for training. Models are trained on a single NVIDIA A100 GPU for 30 epochs. The hit filtering models trained in approximately 10 hours, while the tracking models trained in 20-60 hours, depending on the p_T^{min} threshold. A batch size of a single event is used. Inference times can be found in Section 5.3.

p_T^{min}	Hits (Pre)	Hits (Post)	Particles	Object Queries
1 GeV	57k	6k	800	1100
750 MeV	57k	8k	1300	1800
600 MeV	57k	12k	1800	2100

Table 2: Summary of the models trained with different minimum particle thresholds p_T^{min}. For each threshold, the number of hits pre- and post-filtering is shown, along with the average number of target particles in the event, and the configured number of object queries for the associated track reconstruction model. Hit and particle counts are averaged over the test set.

5 Results

As described in Section 4.1, we consider a particle to be reconstructable if it leaves at least three hits in the pixel detector, has an absolute pseudorapidity of $|\eta| < 2.5$, and a transverse momentum p_T above a variable threshold p_T^{min}. The hit filtering performance is discussed in Section 5.1, while the tracking performance is discussed in Section 5.2 and inference times are presented in Section 5.3. All results shown are obtained using the test set of 100 events.

5.1 Hit Filtering

In the hit filtering task, hits are labelled as *signal* if they belong to a reconstructable particle and *noise* otherwise. The filter performance is evaluated using binary efficiency and purity. The hit efficiency is defined as the fraction of signal hits retained after filtering, while the purity represents the fraction of retained hits that are signal hits.

Fig. 4 demonstrates how these metrics vary with the probability threshold used to classify hits. At our chosen operating threshold of 0.1, the models achieve impressive performance while significantly reducing the input multiplicity for downstream tracking. The dramatic reduction in hit multiplicity achieved - from approximately 57k to 6k-12k hits depending on the model - are detailed in Table 2 and also visualised in Fig. 2.

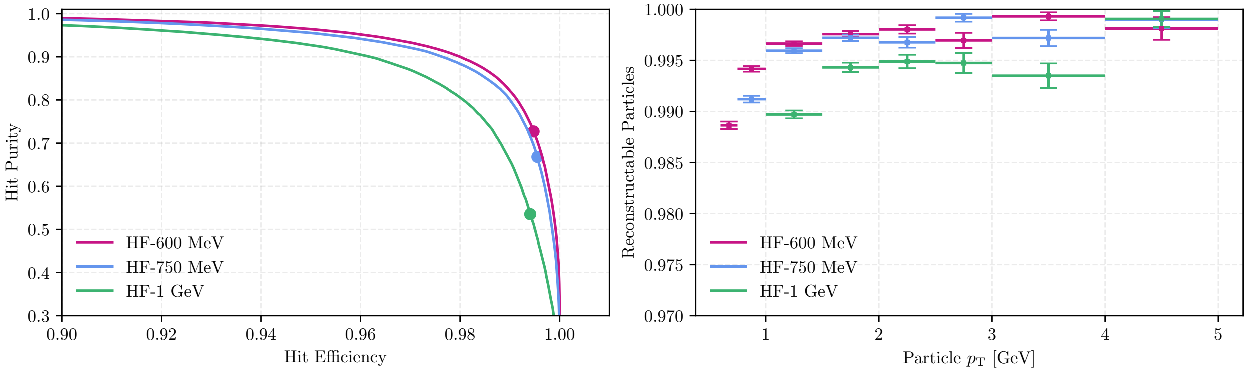

Figure 4: (Left) Signal hit purity as a function of the signal hit efficiency for the three different hit filtering models. The markers show the efficiency and purity at the chosen threshold of 0.1 and each model achieves an area under the curve of 0.998. (Right) The fraction of particles that remain reconstructable as a function of simulated particle p_{T} after filtering hits that fall below the 0.1 threshold. Binominal errors are indicated by the vertical spans.

As summarised in Table 3, the HF-600 MeV model maintains a hit efficiency of 99.5% while improving purity from 15.6% pre-filter to 72.7% post-filter. Similarly, the HF-750 MeV and HF-1 GeV models achieve efficiencies of 99.6% and 99.4% respectively, with corresponding purities of 66.8% and 53.6%. This represents dramatic purity improvements from their pre-filter values of 11.1% and 6.8%. When considering only hits from particles in the central detector region ($|\eta|$ < 2.5), the initial purities are higher at 30.6%, 21.9%, and 13.3% respectively, still demonstrating a significant improvement.

Model	Initial Purity ($|\eta| < 2.5$)	Filter Efficiency	Filter Purity	Reconstructable
HF-1 GeV	6.8% (13.3%)	99.4%	53.6%	99.1%
HF-750 MeV	11.1% (21.9%)	99.6%	66.8%	99.4%
HF-600 MeV	15.6% (30.6%)	99.5%	72.7%	99.3%

Table 3: Performance of the hit filtering models on the test set. From left to right the columns show the initial hit purity (and the purity considering only hits from particles with $|\eta| > 2.5$), the post-filter efficiency and purity, and the percentage of particles which remain reconstructable (i.e. retain three or more hits in the pixel layers) after the application of the filter. Statistical uncertainties are negligible.

We also examine the fraction of particles that remain reconstructable after filtering (i.e. those that retain at least three pixel hits). The results are shown as a function of the particle p_{T} in Fig. 4, and integrated values are shown in Table 3. For particles with $p_{\mathrm{T}} > 1\,\mathrm{GeV}$, both the HF-600 MeV and HF-750 MeV models maintain excellent performance, preserving more than 99.5% of particles. This performance degrades as particle p_{T} approaches $p_{\mathrm{T}}^{\mathrm{min}}$. The HF-1 GeV model's reduced performance across all p_{T} ranges (99.1% reconstructable fraction compared to 99.4% for HF-750 MeV) demonstrates the advantage of training with lower p_{T} particles (increasing the training statistics and avoiding boundary effects), and the benefit of using a larger model.

5.2 Tracking

Track reconstruction performance is evaluated in terms of the efficiency and fake rate. Tracking efficiency is defined as the fraction of reconstructable particles that are correctly matched to a reconstructed track. The fake rate is defined as the fraction of reconstructed tracks that are not well-matched to any reconstructable particle. We consider two different criteria to define whether a track is matched to a particle or not: *double majority* and *perfect* matching. Under the double majority (DM) criteria, a match occurs if $> 50\%$ of the particle's hits are assigned to the track and $> 50\%$ of the hits on the track are from that particle. Under the perfect criteria, a match occurs if all of the particle's hits are assigned to the track, and no other hits are assigned. Each track is matched to the simulated particle that contributes the largest number of hits to the track. If two particles have the same number of hits on a given track, one is chosen at random. $\varepsilon^{\mathrm{DM}}_{p_{\mathrm{T}}>1\,\mathrm{GeV}}$ and $\varepsilon^{\mathrm{perfect}}_{p_{\mathrm{T}}>1\,\mathrm{GeV}}$ are the efficiencies using the DM and perfect match criteria respectively. Subscripts are used to indicate the minimum p_{T} of the matched simulated particles included in the calculation. The fake rate f^{DM} is defined using the DM matching and without any selections on p_{T}. To enable comparison with other methods, we also define a fake rate $f^{\mathrm{DM}}_{p_{\mathrm{T}}>0.9\,\mathrm{GeV}}$, which considers only tracks matched to target particles with $p_{\mathrm{T}} > 0.9\,\mathrm{GeV}$. The inefficiencies resulting from the filtering step are included in the tracking efficiencies discussed here.

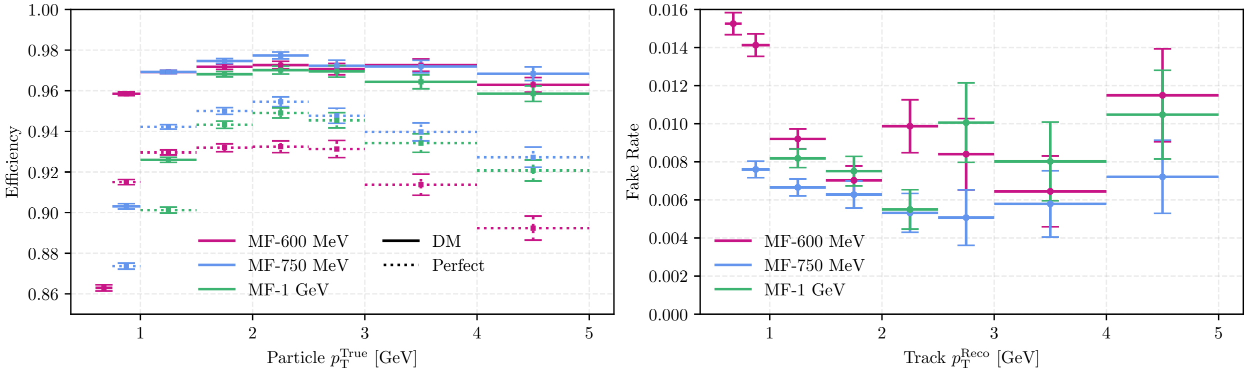

Figure 5: Tracking efficiency as a function of simulated particle p_{T} (left) and fake rate as a function of reconstructed track p_{T} (right). For the efficiency plot (left) the solid lines show the efficiency under the double match criteria, while the dashed lines show the efficiency under the perfect match criteria. The vertical span of the markers indicate the binomial error about the mean while the horizontal span shows the bin extent.

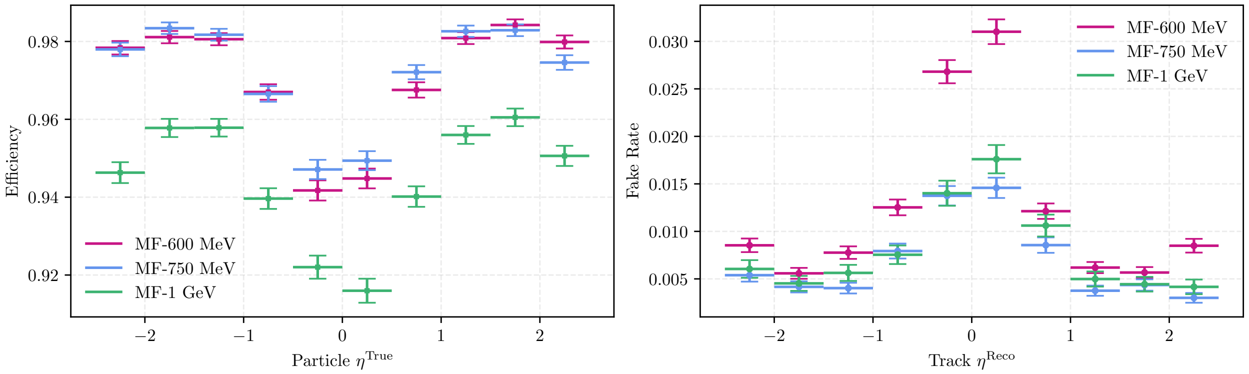

Figure 6: Tracking efficiency as a function of simulated particle η (left) and fake rate as a function of reconstructed track η (right). In both cases, the double majority matching is used. The vertical span of the markers indicate the binomial error about the mean while the horizontal span shows the bin extent. Only tracks with $p_{\mathrm{T}} > 1\,\mathrm{GeV}$ are shown.

The tracking performance of the different models depends on the value of $p_{\mathrm{T}}^{\mathrm{min}}$ used to define the target particles during training. As shown in Fig. 5, all three models achieve a plateau efficiency of approximately 97% for higher-p_{T} tracks, with the efficiency dropping steadily as particle p_{T} approaches $p_{\mathrm{T}}^{\mathrm{min}}$ and falling to nearly zero below this threshold. The MF-600 MeV and MF-750 MeV models effectively reconstruct tracks down to their respective $p_{\mathrm{T}}^{\mathrm{min}}$ thresholds with efficiencies of 86% and 90% respectively at these points. Above this threshold, the efficiencies quickly plateau, reaching approximately 96% and 97% for $p_{\mathrm{T}} > 750\,\mathrm{MeV}$ and $1\,\mathrm{GeV}$ respectively. In contrast, the MF-1 GeV model only reaches its performance plateau for particles above approximately $1.5\,\mathrm{GeV}$. The improved performance of the MF-600 MeV and MF-750 MeV models versus the the MF-1 GeV model demonstrates the advantage of using a lower value of $p_{\mathrm{T}}^{\mathrm{min}}$ for training. This provides increased statistics and a broader coverage in p_{T}, outweighing the additional complexity of reconstructing lower-p_{T} particles. In terms of fake rates, the MF-750 MeV configuration performs best with approximately 0.5%, while the MF-600 MeV and MF-600 MeV models show slightly higher rates, particularly in the challenging low-p_{T} region for the MF-600 MeV model.

The models also achieve high perfect track reconstruction rates, with the MF-600 MeV and MF-750 MeV models perfectly reconstructing approximately 94% of tracks, and the MF-1 GeV model achieving a perfect reconstruction rate of approximately 92%. This high performance underscores the models ability to handle the combinatorial complexity of the TrackML dataset, with only a low number of output tracks not perfectly reconstructed (around 2-4% depending on the model and p_T).

Fig. 6 shows the efficiency and fake rate as a function of the particle and reconstructed track η, respectively, demonstrating good performance across the full η region studied. The drop in efficiency and increase in fake rate in the central η region is consistent with the increased track density in this region.

Performance comparisons with existing methods are summarised in Table 4, though several important methodological differences should be noted. For example, while OC and HGNN include particles in the forward region ($|\eta| < 4$), our approach focuses on the more challenging central region ($|\eta| < 2.5$) where track density is highest. Despite this more demanding task, the MF-750 MeV model achieves a higher efficiency ($\varepsilon^{\mathrm{DM}}_{p_\mathrm{T}>0.9\ \mathrm{GeV}} = 97.1\%$) than OC while reducing the fake rate by a factor of 3 to $f^{\mathrm{DM}}_{p_\mathrm{T}>0.9\ \mathrm{GeV}} = 0.2\%$. In addition, our approach demonstrates a significantly improved perfect match efficiency. The HGNN method includes hits on the strip layers and requires more than 5 hits for a particle to be reconstructable, and would require additional post-processing steps after the model to reduce the high fake rate. While these differences complicate direct comparisons, our results achieve slightly reduced efficiency with significantly better fake rate, whilst using less information. Furthermore, as shown in Table 5, our method achieves this while significantly improving inference times.

	$\varepsilon^{\mathrm{perfect}}_{p_\mathrm{T}>1\ \mathrm{GeV}}$	$\varepsilon^{\mathrm{perfect}}_{p_\mathrm{T}>0.9\ \mathrm{GeV}}$	$\varepsilon^{\mathrm{DM}}_{p_\mathrm{T}>1\ \mathrm{GeV}}$	$\varepsilon^{\mathrm{DM}}_{p_\mathrm{T}>0.9\ \mathrm{GeV}}$	f^{DM}	$f^{\mathrm{DM}}_{p_\mathrm{T}>0.9\ \mathrm{GeV}}$
MF-1 GeV	91.9%	-	94.4%	-	0.8%	-
MF-750 MeV	94.5%	94.2%	97.1%	97.0%	0.6%	0.2%
MF-600 MeV	92.9%	92.7%	97.0%	96.9%	0.9%	0.4%
OC [31]	-	85.8%	-	96.4%	-	0.9%
HGNN [54]	-	-	97.9%	-	36.7%	-

Table 4: Comparison of the results for the various models using the TrackML dataset on the test set. For the efficiencies, we use two different p_T thresholds to facilitate comparisons with previous approaches. OC [31] uses $p_\mathrm{T}^{\mathrm{min}} = 0.9\,\mathrm{GeV}$, while HGNN [54] uses $p_\mathrm{T}^{\mathrm{min}} = 1\,\mathrm{GeV}$ and also includes hits from the strip layers. Both OC and HGNN attempt to reconstruct particles satisfying $|\eta| < 4$, whereas we target $|\eta| < 2.5$ in this work. Statistical uncertainties are negligible.

Analysis of model performance versus training set size for the MF-1 GeV model reveals that the model has not yet reached performance saturation when using the complete TrackML dataset. This suggests significant potential for improved performance through additional training data. Furthermore, in applications where inference speed is not critical, enhanced performance could be achieved by increasing the model size.

Fig. 7 shows the residuals between reconstructed and target track parameters, where the target parameters are derived from DM matched particles. While v_z is regressed directly, the p_T, η, and ϕ are constructed from the track's Cartesian momentum. The residuals demonstrate minimal bias, however the current regression performance does not yet match the precision achieved by established experiments like ATLAS and CMS, which employ highly tuned parametric fitting techniques [2, 3]. Regardless, these results serve as a promising proof-of-principle for our novel approach of jointly performing hit-to-track assignment and track parameter fitting in a single model. Several avenues exist for improving the parameter estimation. First, incorporating hits from the outer strip layers would significantly enhance momentum resolution. Secondly, directly regressing the parameters of interest, rather than constructing them from the Cartesian momentum, would likely improve performance. Finally, the model could be extended to estimate track parameter uncertainties and their correlations, bringing the approach closer to the comprehensive track fitting methods used in production environments.

5.3 Inference Time

Inference time is a critical metric for track reconstruction at particle colliders. Our Transformer-based models demonstrate competitive performance in this key metric, with the hit filtering forward pass requiring on average approximately 23 ms to process a single event comprising $\mathcal{O}(60\mathrm{k})$ hits when using an NVIDIA A100 GPU. The tracking inference time depends on the choice of $p_\mathrm{T}^{\mathrm{min}}$, and ranges from 51 ms for the 1 GeV model to 101 ms for the 600 MeV model, as detailed in Table 5. For the 750 MeV model, which represents a good balance between tracking performance and inference time, the total time for hit filtering and tracking is 100 ms.

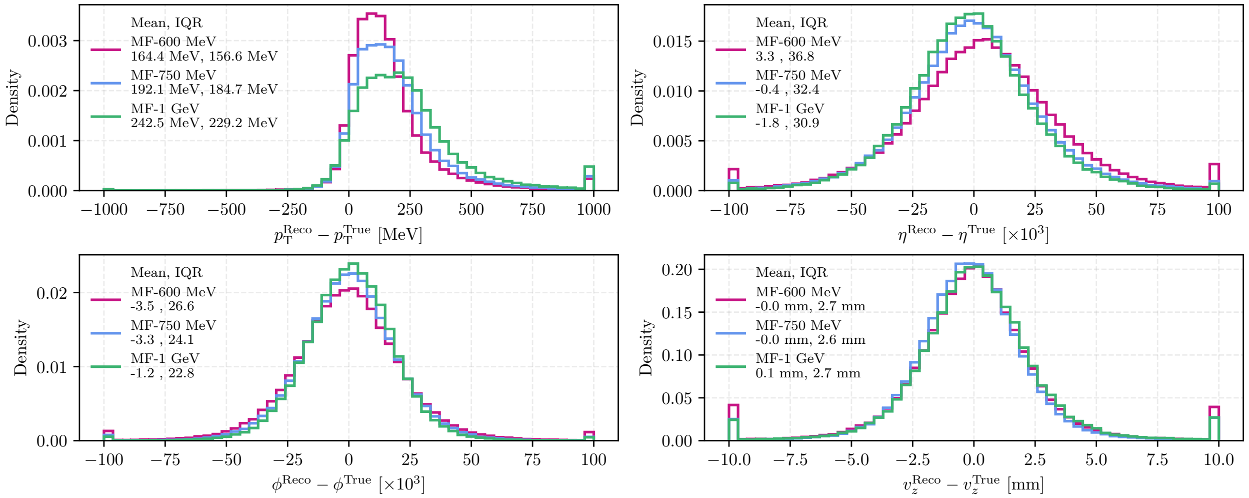

Figure 7: Residuals between the regressed parameters of reconstructed tracks and the targets from the DM matched tracks. Tracks matched to simulated particles with $p_T > p_T^{\text{min}}$ are used for each model. The mean μ and inter-quartile range (IQR) of the residuals for each model is shown in the legend. Residuals that lie outside the plot range are placed into extremal bins. Note that residuals for the angle ϕ and pseudorapidity η are shown in thousandths in both the plot and the legend values.

	Tracking time [ms]	Filter + tracking time [ms]
MF-1 GeV	50 ± 7	73 ± 8
MF-750 MeV	77 ± 9	100 ± 11
MF-600 MeV	101 ± 12	124 ± 14

Table 5: Mean inference times and standard deviations for the tracking model forward pass, and combined filtering and tracking forward pass, evaluated on an NVIDIA A100 GPU with a batch size of 1.

To understand scaling behaviour, we analysed inference times as a function of hit multiplicity in Fig. 8. Both the filtering and tracking models exhibit linear scaling with the number of input hits, a critical result achieved through the optimisations described in Section 3.1. This linear scaling allows us to extrapolate performance to more realistic detectors. With the $\mathcal{O}(300\text{k})$ hits expected in the ITk at the start of Run 4 [55], we project a combined filter and tracking time of 390 ms, assuming the linear scaling holds and the hit filter retains 25% of all hits. This projection is competitive with recent results from the GNN4ITk project [56] and timing studies in Ref. [54], and is comparable to the timings achieved by an optimised non-ML approach to trigger-level tracking [57].

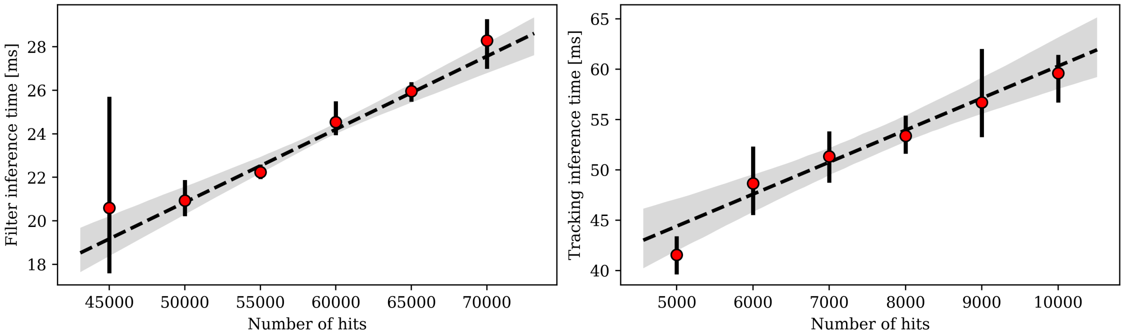

Figure 8: Inference times as a function of the input hit multiplicity for the hit filtering model (left) and 1 GeV track reconstruction model (right), evaluated on an NVIDIA A100 GPU with a batch size of 1. The average time in each bin and the 95% confidence intervals are shown by the markers and vertical error bars. A linear fit is shown in the dashed line, along with a 95% confidence interval shown by the shaded region.

The current inference times, while already competitive, has been achieved without extensive optimisation. Our approach could likely achieve substantial speed improvements through established techniques such as model pruning and quantisation. Additional gains could be realised by training smaller models on expanded datasets, while inference throughput could be enhanced by batching multiple events during inference.

6 Conclusion

We present a novel application of the Transformer architecture to charged particle track reconstruction, achieving state-of-the-art results on the full TrackML dataset. Our approach demonstrates exceptional performance, with an efficiency of approximately 97.0% and a fake rate of 0.6% for the best performing, and can be used to reconstruct particles with transverse momenta as low as $600\,\mathrm{MeV}$. The model's inference time of approximately $100\,\mathrm{ms}$ makes it highly competitive for real-world applications.

This success stems from two key innovations: a Transformer-based hit filtering stage that effectively reduces input multiplicities without requiring complex graph construction, and a MaskFormer track reconstruction stage that leverages cross-attention to build an explicit latent representation of each track. This unified approach enables simultaneous track finding and parameter estimation while naturally handling hit sharing between tracks. The model's linear scaling with hit multiplicity, enabled by efficient attention kernels and demonstrated in our timing studies, suggests promising scalability for future detector environments. By aligning with recent advancements in the field of machine learning, our approach stands to benefit from future developments, offering scalable, adaptable, and high-performance solutions to meet the increasing computational demands of high-energy physics.

Future work could involve the reconstruction of particles in the forward region and at lower transverse momenta. Combining the filtering and reconstruction steps into a single model could streamline training and reduce inference times. Finally, incorporating additional information from other detector systems, such as calorimeters, could further enhance the reconstruction of charged particles and enable the simultaneous reconstruction of neutral particles.

Beyond its immediate application, our work offers several broader contributions. The hit filtering component can be readily integrated into existing approaches to tracking. The model's tunable nature, allowing trade-offs between efficiency and inference times and the targeting of certain particle kinematics, makes it adaptable for various high energy physics experiments, and requirements from triggering systems to offline reconstruction. Most significantly, the success of Transformer architectures in both tracking and vertexing demonstrates the potential for a unified approach to particle physics reconstruction, moving away from specialised solutions and toward generalised learned models that could significantly impact how we process and analyse particle physics data.

Acknowledgments

We gratefully acknowledge the support of the UK's Science and Technology Facilities Council (STFC). We also extend our thanks to UCL for the use of their high-performance computing facilities, with special thanks to Edward Edmondson for his expert management and technical support.

References

[1] Lyndon Evans and Philip Bryant. "LHC Machine". In: *JINST* 3 (2008), S08001. DOI: 10.1088/1748-0221/3/08/S08001.

[2] ATLAS Collaboration. "Software Performance of the ATLAS Track Reconstruction for LHC Run 3". In: *Comput. Softw. Big Sci.* 8 (2023), p. 9. DOI: 10.1007/s41781-023-00111-y. arXiv: 2308.09471 [hep-ex].

[3] CMS Collaboration. "Description and performance of track and primary-vertex reconstruction with the CMS tracker". In: *JINST* 9 (2014), P10009. DOI: 10.1088/1748-0221/9/10/P10009. arXiv: 1405.6569 [hep-ex].

[4] ATLAS Collaboration. "Electron and photon efficiencies in LHC Run 2 with the ATLAS experiment". In: *JHEP* 05 (2024), p. 162. DOI: 10.1007/JHEP05(2024)162. arXiv: 2308.13362 [hep-ex].

[5] ATLAS Collaboration. "Muon reconstruction and identification efficiency in ATLAS using the full Run 2 pp collision data set at $\sqrt{s} = 13\,\mathrm{TeV}$". In: *Eur. Phys. J. C* 81 (2021), p. 578. DOI: 10.1140/epjc/s10052-021-09233-2. arXiv: 2012.00578 [hep-ex].

[6] CMS Collaboration. "Electron and photon reconstruction and identification with the CMS experiment at the CERN LHC". In: *JINST* 16 (2021), P05014. DOI: 10.1088/1748-0221/16/05/P05014. arXiv: 2012.06888 [hep-ex].

[7] CMS Collaboration. “Performance of the CMS muon detector and muon reconstruction with proton–proton collisions at $\sqrt{s} = 13$ TeV”. In: *JINST* 13 (2018), P06015. DOI: 10.1088/1748-0221/13/06/P06015. arXiv: 1804.04528 [hep-ex].

[8] ATLAS Collaboration. *Reconstruction, Identification, and Calibration of hadronically decaying tau leptons with the ATLAS detector for the LHC Run 3 and reprocessed Run 2 data*. Tech. rep. Geneva: CERN, 2022. URL: https://cds.cern.ch/record/2827111.

[9] CMS Collaboration. “Performance of reconstruction and identification of τ leptons decaying to hadrons and ν_τ in pp collisions at $\sqrt{s} = 13$ TeV”. In: *JINST* 13.10 (2018), P10005. DOI: 10.1088/1748-0221/13/10/P10005. arXiv: 1809.02816 [hep-ex].

[10] ATLAS Collaboration. “Jet reconstruction and performance using particle flow with the ATLAS Detector”. In: *Eur. Phys. J. C* 77 (2017), p. 466. DOI: 10.1140/epjc/s10052-017-5031-2. arXiv: 1703.10485 [hep-ex].

[11] CMS Collaboration. “Particle-flow reconstruction and global event description with the CMS detector”. In: *JINST* 12 (2017), P10003. DOI: 10.1088/1748-0221/12/10/P10003. arXiv: 1706.04965 [hep-ex].

[12] ATLAS Collaboration. “ATLAS flavour-tagging algorithms for the LHC Run 2 *pp* collision dataset”. In: *Eur. Phys. J. C* 83 (2023), p. 681. DOI: 10.1140/epjc/s10052-023-11699-1. arXiv: 2211.16345 [physics.data-an].

[13] CMS Collaboration. “Identification of heavy-flavour jets with the CMS detector in *pp* collisions at 13 TeV”. In: *JINST* 13 (2018), P05011. DOI: 10.1088/1748-0221/13/05/P05011. arXiv: 1712.07158 [hep-ex].

[14] ATLAS Collaboration. “Combination of searches for Higgs boson pairs in *pp* collisions at $\sqrt{s} = 13$ TeV with the ATLAS detector”. In: *Phys. Lett. B* 800 (2020), p. 135103. DOI: 10.1016/j.physletb.2019.135103. arXiv: 1906.02025 [hep-ex].

[15] CMS Collaboration. “Constraints on the Higgs boson self-coupling from the combination of single and double Higgs boson production in proton–proton collisions at $\sqrt{s} = 13$ TeV” (2024). arXiv: 2407.13554 [hep-ex].

[16] I. Zurbano Fernandez et al. “High-Luminosity Large Hadron Collider (HL-LHC): Technical design report”. 10/2020 (Dec. 2020). Ed. by I. Béjar Alonso et al. DOI: 10.23731/CYRM-2020-0010.

[17] ATLAS Collaboration. *ATLAS HL-LHC Computing Conceptual Design Report*. Tech. rep. 2020. URL: https://cds.cern.ch/record/2729668.

[18] CMS Offline Software and Computing. *CMS Phase-2 Computing Model: Update Document*. Tech. rep. Geneva: CERN, 2022. URL: https://cds.cern.ch/record/2815292.

[19] ATLAS Collaboration. “The ATLAS Experiment at the CERN Large Hadron Collider”. In: *JINST* 3 (2008), S08003. DOI: 10.1088/1748-0221/3/08/S08003.

[20] ATLAS Collaboration. *Fast Track Reconstruction for HL-LHC*. Tech. rep. Geneva: CERN, 2019. URL: https://cds.cern.ch/record/2693670.

[21] A. Augusto Alves Jr. et al. “The LHCb Detector at the LHC”. In: *JINST* 3 (2008), S08005. DOI: 10.1088/1748-0221/3/08/S08005.

[22] K. Aamodt et al. “The ALICE experiment at the CERN LHC”. In: *JINST* 3 (2008), S08002. DOI: 10.1088/1748-0221/3/08/S08002.

[23] L. Vigani and T. Rudzki. “The Mu3e detector”. In: *JINST* 17.05 (2022), p. C05024. DOI: 10.1088/1748-0221/17/05/C05024.

[24] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. “Per-Pixel Classification is Not All You Need for Semantic Segmentation” (2021). arXiv: 2107.06278 [cs.CV].

[25] Ashish Vaswani et al. “Attention Is All You Need” (2023). arXiv: 1706.03762 [cs.CL].

[26] Samuel Van Stroud et al. “Secondary vertex reconstruction with MaskFormers”. In: *Eur. Phys. J. C* 84.10 (2024), p. 1020. DOI: 10.1140/epjc/s10052-024-13374-5. arXiv: 2312.12272 [hep-ex].

[27] Franco Scarselli et al. “The Graph Neural Network Model”. In: *IEEE Transactions on Neural Networks* 20.1 (2009), pp. 61–80. DOI: 10.1109/TNN.2008.2005605.

[28] Sylvain Caillou et al. *ATLAS ITk Track Reconstruction with a GNN-based pipeline*. Tech. rep. 2022. URL: https://cds.cern.ch/record/2815578.

[29] Caillou, Sylvain et al. “Physics Performance of the ATLAS GNN4ITk Track Reconstruction Chain”. In: vol. 295. 2024, p. 03030. DOI: 10.1051/epjconf/202429503030.

[30] Ryan Liu et al. “Hierarchical Graph Neural Networks for Particle Track Reconstruction”. *21th International Workshop on Advanced Computing and Analysis Techniques in Physics Research: AI meets Reality*. Mar. 2023. arXiv: 2303.01640 [hep-ex].

[31] Kilian Lieret et al. “High Pileup Particle Tracking with Object Condensation”. Dec. 2023. arXiv: 2312.03823 [physics.data-an].

[32] Sascha Caron et al. "TrackFormers: In Search of Transformer-Based Particle Tracking for the High-Luminosity LHC Era". In: *arXiv e-prints*, arXiv:2407.07179 (July 2024), arXiv:2407.07179. DOI: 10.48550/arXiv.2407.07179. arXiv: 2407.07179 [hep-ex].

[33] Andris Huang et al. "A Language Model for Particle Tracking". In: *arXiv e-prints*, arXiv:2402.10239 (Feb. 2024), arXiv:2402.10239. DOI: 10.48550/arXiv.2402.10239. arXiv: 2402.10239 [hep-ph].

[34] Yash Melkani and Xiangyang Ju. "TrackSorter: A Transformer-based sorting algorithm for track finding in High Energy Physics". In: *arXiv e-prints*, arXiv:2407.21290 (July 2024), arXiv:2407.21290. DOI: 10.48550/arXiv.2407.21290. arXiv: 2407.21290 [cs.LG].

[35] Joseph Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection". In: *CoRR* abs/1506.02640 (2015). arXiv: 1506.02640.

[36] Nicolas Carion et al. "End-to-End Object Detection with Transformers" (2020). arXiv: 2005.12872 [cs.CV].

[37] Kaiming He et al. "Mask R-CNN" (2018). arXiv: 1703.06870 [cs.CV].

[38] Bowen Cheng et al. "Masked-attention Mask Transformer for Universal Image Segmentation" (2022). arXiv: 2112.01527 [cs.CV].

[39] Moritz Kiehn et al. "The TrackML high-energy physics tracking challenge on Kaggle". *European Physical Journal Web of Conferences*. Vol. 214. European Physical Journal Web of Conferences. July 2019, 06037, p. 06037. DOI: 10.1051/epjconf/201921406037.

[40] Iz Beltagy, Matthew E. Peters, and Arman Cohan. "Longformer: The Long-Document Transformer" (Apr. 2020). DOI: 10.48550/arXiv.2004.05150.

[41] Tri Dao. "FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning" (July 2023). DOI: 10.48550/arXiv.2307.08691.

[42] Noam Shazeer. "GLU Variants Improve Transformer" (Feb. 2020). DOI: 10.48550/arXiv.2002.05202.

[43] Haebom Lee et al. "Spatio-Temporal Outdoor Lighting Aggregation on Image Sequences using Transformer Networks" (Feb. 2022). DOI: 10.48550/arXiv.2202.09206.

[44] Alexander Kirillov et al. "Segment Anything" (2023). arXiv: 2304.02643 [cs.CV].

[45] Ross Girshick. "Fast R-CNN" (Apr. 2015). DOI: 10.48550/arXiv.1504.08083.

[46] Carole H. Sudre et al. "Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations" (2017), pp. 240–248. ISSN: 1611-3349. DOI: 10.1007/978-3-319-67558-9_28.

[47] Tsung-Yi Lin et al. "Focal Loss for Dense Object Detection" (Aug. 2017). DOI: 10.48550/arXiv.1708.02002. arXiv: 1708.02002 [cs.CV].

[48] Stefan Guthe and Daniel Thuerck. "Algorithm 1015: A Fast Scalable Solver for the Dense Linear (Sum) Assignment Problem". In: *ACM Trans. Math. Softw.* 47.2 (Apr. 2021). ISSN: 0098-3500. DOI: 10.1145/3442348.

[49] Moritz Kiehn et al. "The TrackML high-energy physics tracking challenge on Kaggle". In: vol. 214. 2019, p. 06037. DOI: 10.1051/epjconf/201921406037.

[50] Sabrina Amrouche et al. "The Tracking Machine Learning challenge : Accuracy phase". *The NeurIPS '18 Competition: From Machine Learning to Intelligent Conversations*. Apr. 2019. DOI: 10.1007/978-3-030-29135-8_9. arXiv: 1904.06778 [hep-ex].

[51] Sabrina Amrouche et al. "The Tracking Machine Learning Challenge: Throughput Phase". In: *Comput. Softw. Big Sci.* 7.1 (2023), p. 1. DOI: 10.1007/s41781-023-00094-w. arXiv: 2105.01160 [cs.LG].

[52] M. Hansroul, H. Jeremie, and D. Savard. "Fast circle fit with the conformal mapping method". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 270.2 (1988), pp. 498–501. DOI: https://doi.org/10.1016/0168-9002(88)90722-X.

[53] Patrick J. Fox et al. "Beyond 4D Tracking: Using Cluster Shapes for Track Seeding". In: *JINST* 16.05 (2021), P05001. DOI: 10.1088/1748-0221/16/05/P05001. arXiv: 2012.04533 [physics.ins-det].

[54] Ryan Liu et al. "Hierarchical Graph Neural Networks for Particle Track Reconstruction". *21st International Workshop on Advanced Computing and Analysis Techniques in Physics Research: AI meets Reality*. Mar. 2023. arXiv: 2303.01640 [hep-ex].

[55] ATLAS Collaboration. *Technical Design Report for the ATLAS Inner Tracker Strip Detector*. Tech. rep. Geneva: CERN, 2017. URL: https://cds.cern.ch/record/2257755.

[56] ATLAS Collaboration. *Computational Performance of the ATLAS ITk GNN Track Reconstruction Pipeline*. Tech. rep. Geneva: CERN, 2024. URL: https://cds.cern.ch/record/2914282.

[57] ATLAS Collaboration. *Technical Design Report for the Phase-II Upgrade of the ATLAS Trigger and Data Acquisition System - Event Filter Tracking Amendment*. Tech. rep. 2022. DOI: 10.17181/CERN.ZK85.5TDL.