Transformers for Charged Particle Track Reconstruction in High-Energy Physics

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, Department of Physics and Astronomy, University College London, London, United Kingdom*

[2]*CERN, European Organization for Nuclear Research, Geneva, Switzerland*

(Received 28 November 2024; revised 21 July 2025; accepted 23 October 2025; published 9 December 2025)

Charged particle reconstruction, the identification and characterization of particles from collision data, is fundamental to nearly all research at particle colliders like the Large Hadron Collider (LHC). With the High-Luminosity upgrade (HL-LHC), particle multiplicities will increase substantially, overwhelming traditional track reconstruction algorithms and presenting computational bottlenecks. Here, we introduce a proof of concept for a powerful new method for charged particle reconstruction inspired by state-of-the-art machine learning (ML) approaches in computer vision. Our model leverages transformer neural networks to efficiently filter relevant signals and fully reconstruct particle trajectories, directly tackling the computational complexity that traditional methods face. Evaluated on the widely used TrackML dataset, our approach achieves state-of-the-art tracking efficiency (97%) and a low fake rate (0.7%), requiring just 97 ms to reconstruct on average 1300 particle trajectories from 55,000 detector hits for particles with transverse momentum above 750 MeV. These results represent a significant milestone in both performance and speed, demonstrating a shift toward unified, scalable ML solutions that offer substantial improvements for collider experiments.

DOI: 10.1103/md46-yqgd

Subject Areas: Particles and Fields

I. INTRODUCTION

Modern particle colliders, such as the Large Hadron Collider (LHC) [1] at CERN, explore fundamental physics by producing vast numbers of collision events, each containing thousands of particles. Analyzing these collisions is essential to addressing open questions in physics, including the properties of the Higgs boson [2,3], the nature of dark matter, and the fundamental structure of the Universe. A core task in analyzing collision data is reconstructing the trajectories (*tracks*) of charged particles produced in these collisions. Track reconstruction provides particle momenta and charge measurements which are critical for nearly all downstream tasks, including lepton reconstruction [4–7], hadronic τ reconstruction [8,9], particle flow algorithms [10,11], and jet flavor identification [12–14].

The upcoming High-Luminosity upgrade of the LHC (HL-LHC) will dramatically increase event complexity. HL-LHC beam crossings occur every 25 ns, corresponding to a crossing rate of 40 MHz. With each beam crossing producing up to 200 proton-proton collisions, the total collision rate will surpass 8×10^9 collisions per second, roughly a factor of 4 increase from current LHC operations. The average number of particles produced in each event will increase accordingly to $\mathcal{O}(10^4)$. This surge in particle multiplicity poses a formidable computational challenge for track reconstruction, as traditional algorithms typically scale worse than quadratically with the number of particle-detector interactions (*hits*) per event. Consequently, they rapidly become computationally prohibitive at HL-LHC multiplicities [15,16]. Addressing this computational bottleneck is a critical challenge, as efficient and accurate particle tracking directly impacts the physics discovery potential at the HL-LHC.

In this work, we address these computational challenges by developing a new charged particle tracking method built upon recent innovations in machine learning. Our approach introduces two novel ideas: a transformer-based [17] *hit filtering* stage and a MaskFormer-inspired [18] *track reconstruction* stage. The hit filtering stage uses windowed

*Contact author: sam.vanstroud.18@ucl.ac.uk
†Contact author: philippa.duckett.21@ucl.ac.uk
‡Contact author: max.hart.22@ucl.ac.uk
§Contact author: nikita.pond.18@ucl.ac.uk
‖Contact author: sebastien.rettie@cern.ch
¶Contact author: g.facini@ucl.ac.uk
**Contact author: timothy.scanlon@ucl.ac.uk

self-attention (SA), originally developed for language processing tasks, to efficiently reduce the initial hit multiplicity. This step avoids the costly combinatorial processing and complex graph construction inherent in prior methods. The subsequent track reconstruction stage leverages a MaskFormer architecture, a type of transformer originally designed for image segmentation, to simultaneously identify tracks, assign hits, and estimate their physical properties, removing the need for separate off-model postprocessing steps. Recent success applying transformer-based models to vertex reconstruction [19] further demonstrates the architecture's potential for general particle physics reconstruction tasks.

Evaluated on the widely used TrackML dataset (see Sec. IV A), our method achieves state-of-the-art tracking efficiency (97%) with low fake rates (0.6%), and inference times of approximately 100 ms per event. This represents a substantial improvement compared to typical inference times of seconds to tens of seconds per event observed with traditional CPU-based tracking algorithms [20]. However, further development is required before operational deployment. These steps include validation under realistic detector conditions, integration within experimental software frameworks, and evaluation of computational resources in production scenarios.

Beyond general-purpose collider experiments, specialized LHC experiments like LHCb [21], ALICE [22], and experiments beyond the LHC such as Mu3e [23] and LUXE [24] also rely on precise and scalable tracking solutions. Our fully learned approach, inherently flexible and adaptable, has potential as a general solution across diverse experimental setups. Integrating it into experiment-independent track reconstruction toolkits, such as A Common Tracking Software (ACTS) [25], would significantly streamline efforts to develop efficient tracking for new particle detectors.

The structure of this article is as follows. In Sec. II, we present an overview of related research. Section III introduces our transformer-based model architecture, detailing both the hit filtering and MaskFormer track reconstruction components. Section IV includes a description of the TrackML dataset, training procedures, and experiments used. The results are discussed in Sec. V, demonstrating the scalability and effectiveness of our approach compared to existing methods. Finally, Sec. VI includes a summary of our contributions and discusses future directions and broader implications of this work for high-energy physics.

II. RELATED WORK

Trajectories of charged particles are observed through their nondestructive interaction with sensitive detector elements in specialized silicon tracking detectors. In general-purpose collider experiments, these detectors are typically arranged around the collision region in concentric cylindrical (barrel) layers, with disks (end caps) at each end and a magnetic field oriented along the beam line. Because of the magnetic field, charged particles follow helical paths, curving in the transverse plane resulting in characteristic patterns of discrete three-dimensional position measurements, referred to as hits, across detector layers. Building upon this fundamental information, a variety of approaches to track reconstruction have been developed to infer the original particle trajectories from hit data. This section will review relevant examples of these methods.

A. Traditional track reconstruction methods

Charged particle tracking has historically relied on computational methods designed to handle detector data efficiently. While these algorithms have been successful in existing collider experiments, they typically scale poorly (worse than quadratically) with increasing particle multiplicities. This scaling behavior presents critical computational challenges as collision densities dramatically increase with the HL-LHC upgrade [15,16]. A summary of traditional solutions follows.

1. Combinatorial Kalman filter (CKF)

Combinatorial Kalman filter (CKF) tracking algorithms have been the standard approach for particle track reconstruction since their introduction to high-energy physics [26] and their widespread adoption in experiments at the Large Electron-Positron (LEP) collider [27,28]. These algorithms remain the dominant tracking method used by current collider experiments, including those at the LHC [29–32]. Both ATLAS and CMS employ CKFs or CKF techniques, which begin with computationally intensive track seeding [29,30]. Track seeding typically involves combinational algorithms that identify initial track candidates (seeds) from small groups of detector hits chosen based on geometric and kinematic constraints. Following seeding, tracks are built by iteratively adding hits and refining estimates of track parameters. Resolving competition between multiple track candidates for shared hits often requires further processing. While CKF tracking achieves high accuracy and robustness under realistic detector conditions, it becomes prohibitively resource intensive for HL-LHC environments [15,16]. Despite algorithmic optimizations [33], significant computational challenges remain: At HL-LHC multiplicities CKFs can take $\mathcal{O}(5\ \mathrm{s})$ per event on a single CPU core to reconstruct charged particles with a transverse momentum (p_{T}) greater than 1 GeV [20,34,35]. More recent developments by ATLAS demonstrated a sub-2 s configuration using ACTS in a "fast" mode, optimized for the data selection system (trigger) [36].

2. Cellular automata (CA)

Cellular automata (CA) [37] approaches mitigate combinatorial complexity by incrementally linking nearby hits into track segments based on local geometric criteria. It has been used in a variety of completed experiments [38–41]. CA-based approaches offer strong scalability and are

already deployed in production GPU-accelerated tracking pipelines, such as CMS's Patatrack system in the trigger [42,43]. The CMS experiment has also integrated CA-based track seeding into its phase-II pixel detector upgrade plans [44]. ALICE employs a GPU-accelerated CA method to reconstruct tracks in heavy-ion collisions, leveraging CA's ability to be parallelized and computational efficiency [45]. However, cellular automata require careful tuning of geometric rules and thresholds, and their performance can degrade significantly in dense particle environments where track overlap and ambiguous hit association rates increase.

3. Hough transform

Hough transform methods [34,46] reconstruct tracks by transforming hit positions into parameter space (e.g., curvature and angle), where collinear or curved track hits appear as peaks. Tracks are then identified by detecting these peaks. Hough transforms can efficiently manage combinational complexity and identify tracks in noisy environments with robustness against missing hits and noise from its global pattern-recognition approach. However, within higher track multiplicity settings, the computational demands and susceptibility to spurious peak formation from coincidental hit alignments require complex multistage solutions.

B. Emerging solutions

The computational demands of real-time track reconstruction at facilities like the HL-LHC have spurred exploration into hardware accelerators. Field programmable gate arrays and GPUs are prominent candidates due to their inherent parallel-processing capabilities. Both CMS and ATLAS are actively developing field programmable gate array-based track finders for their HL-LHC trigger upgrades, with studies indicating potential for submicrosecond latencies [34,44]. GPUs have also proven effective, accelerating software-level tracking for experiments like ALICE and LHCb by leveraging massive parallelism for tasks such as cellular automata and Kalman filtering [45,47].

Beyond traditional algorithms, GPUs are increasingly facilitating advanced machine-learning- (ML) based tracking methods. This section reviews several modern approaches to track reconstruction, many of which target GPU deployment.

1. TrackML challenge

The TrackML challenge [20,48] catalyzed community engagement in developing efficient and scalable track reconstruction algorithms. The dataset (described in Sec. IV) comprises simulated HL-LHC collision events under idealized detector conditions, with solutions ranked by accuracy (weighted fraction of correctly assigned hits) and speed (time per event on two CPU cores). The winning submission of the throughput phase, "Mikado" [48], employed 60 passes of combinational searches with localized helix fitting and optimized data structures to achieve 94.4% accuracy at 0.56 s/event. While innovative, Mikado's reliance on complex, handcrafted logic and an extensive, largely *ad hoc* optimization process for its ∼20,000 tunable hyperparameters limits its wider applicability.

2. GNN4ITk

Graph neural networks (GNNs) [49] have emerged as a leading approach to address the computational challenges of track reconstruction at the HL-LHC. The ATLAS Collaboration's GNN4ITk pipeline [50,51] (based on Refs. [52,53]) involves graph construction from hits, GNN-based edge scoring, edge filtering, and iterative graph segmentation for track candidate extraction. While achieving high tracking efficiency (over 90%–99% depending on purity criteria, with 0.1% fake rate [54]), the initial graph construction remains a bottleneck; even with recent optimizations, processing remains in the 500–700 ms range for events with $\mathcal{O}(10^5)$ hits [55]. Furthermore, the reliance on custom preprocessing (graph construction) and postprocessing (graph segmentation) marries a learned GNN-based edge filtering step with nonlearned components, which may not be optimal compared to a fully end-to-end learned approach.

3. Hierarchical graph neural network (HGNN)

Hierarchical graph neural networks (HGNN) [56] offer an alternative to GNN4ITk's iterative segmentation by grouping nodes into *supernodes*, enlarging the GNN's receptive field. Though HGNN can improve reconstruction efficiency, its reported high fake rate (over 50%) and increased computational cost necessitate further postprocessing, limiting its feasibility for HL-LHC deployment.

4. Object condensation (OC)

Object condensation (OC) [57] reconstructs tracks by clustering hits in a learned latent space. Representative hits are selected to characterize tracks, with remaining hits assigned via an off-model clustering algorithm. OC also employs an edge classification step to prune hit connections. A potential limitation is its reliance on single representative hits, which couples hit feature extraction and object identification. Furthermore, the current clustering-based hit assignment is not learned end to end and does not natively support assigning hits to multiple tracks, a scenario commonly encountered in complex collision events with closely spaced or overlapping particle trajectories.

5. Experimental transformer approaches

Early explorations of transformers for track reconstruction [58–61] have typically focused on direct

hit classification or sequential modeling of tracks with autoregressive models. Detailed comparisons are often challenging as these works may not have been validated on the full-scale TrackML dataset or may not fully characterize performance across tracking efficiency, fake rate, and timing.

6. MaskFormers

Recent computer vision advancements, including bounding box prediction [62,63] and sophisticated instance segmentation [18,64], offer promising alternatives to geometric deep learning for ML-based track reconstruction. When analyzing objects depicted within images, the MaskFormer architecture [18,65] effectively identifies multiple instances of the same object class. It achieves this by assigning pixels to distinct object instances, each represented by a unique segmentation mask. By replacing the image pixels with an unordered set of hits and the image-depicted objects with charged particle tracks, we leverage MaskFormers to address the challenges of track reconstruction. The utility of MaskFormers in high energy physics was recently demonstrated in Ref. [19], which successfully adapted the MaskFormer architecture to reconstruct displaced secondary decay vertices. In this work, we apply 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 and less task-specific approaches in particle physics reconstruction.

Table I summarizes 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 ($p_{\mathrm{T}}^{\mathrm{min}}$), maximum pseudorapidity (η) used to define target particles (for more information see Sec. IV).

III. MODEL ARCHITECTURE

A. Leveraging transformers

Transformers [17] have revolutionized natural language processing and computer vision tasks, offering a scalable and efficient architecture for processing sequential data. The attention mechanism within enables the network to determine the relevance between different input features. Each feature (e.g., a detector hit or track) is projected into three learned representations: a *key* which defines what each element offers for comparison, a *value* containing the information to be aggregated, and a *query* which initiates the search for relevant information. Attention is computed by taking the dot product of each query with all keys, producing similarity scores that determine how much information to gather from each value. In our model, queries correspond to latent (i.e., learned) track representations, and the keys and values are derived from the embedded features of detector hits. The output of this attention process is a mask for each query, representing the model's confidence that individual inputs (e.g., hits) belong to that query (e.g., a reconstructed track). In *self-attention* the queries, keys, and values all come from the same input set, allowing elements (e.g., hits or latent tracks) to attend to each other. In *cross-attention* (CA) queries come from one source (e.g., latent track representations) and keys/values come from another (e.g., detector hits), enabling the model to associate tracks with relevant hits. However, due to transformer's quadratic complexity in the number of input tokens, custom GNN-based architectures have so far been preferred for particle tracking tasks [51,57].

To neutralize the quadratic complexity of transformers, we hypothesize 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 [67] with a window size w. This results in $\mathcal{O}(M \times w)$ complexity scaling, which is linear in the

TABLE I. Comparison of ML-based approaches to charged particle track reconstruction. Preprocessing refers to initial hit selection or graph construction steps. Postprocessing refers to track grouping or filtering stages. Tracking detectors at general-purpose collider experiments are composed of pixel (P) and strip (S) layers of silicon-based nondestructive charged particle detector elements; see Ref. [66] for more information. More information about the TrackML detector and dataset can be found in Sec. IV A. Our approach is tested with two different limits on $|\eta|^{\mathrm{max}} - 2.5$ for the 600-MeV and 750-MeV trainings and 4.0 for the 1-GeV training (see Sec. IV B for more information).

	$p_{\mathrm{T}}^{\mathrm{min}}$ (GeV)	$\lvert\eta\rvert^{\mathrm{max}}$	Layers	Preprocessing	Postprocessing
GNN4ITk	1.0	4.0	P + S	Edge classification	Graph traversal
HGNN	1.0	4.0	P + S	Edge classification	GMPool
OC	0.9	4.0	P	Edge classification	Clustering
This work	0.6	2.5	P	Hit filter	None
	0.75	2.5	P	Hit filter	None
	1.0	4.0	P	Hit filter	None

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` [68] for efficient computation and `SwiGLU` activation [69] for improved performance.

This approach allows us to efficiently encode 60,000 pixel hits in 25 ms on a single GPU, making it suitable for low-latency trigger-level track reconstruction at the HL-LHC. Scaling to higher hit multiplicities could be achieved through algorithmic optimizations, or simply scaling to multiple GPUs. A common encoder architecture 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 Sec. V C.

B. 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 Sec. II, 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 remaining hits are labeled signal.

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 M hits are first passed through an embedding layer to produce a $d = 256$-dimensional representation for each hit. Positional encodings are applied [17] in the cylindrical hit coordinates (see Sec. IV A) to allow the SA mechanism to readily identify nearby hits. In particular, a cyclic positional encoding [70] is used for ϕ. The initial hit embeddings are then passed into an efficient SA transformer encoder as described in Sec. III A. The encoder has 12 layers (eight for the 1-GeV model; see Sec. IV), model dimension d, and feedforward 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 preprocessing step into other traditional and ML-based tracking pipelines, or repurposed for other tasks such as pileup mitigation.

C. 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 [18,65,71] which forms an explicit latent representation for each track candidate, allowing for joint optimization of hit assignments and track parameters. Similar to Ref. [19], we include modifications to handle sparse inputs, and to allow additional regression tasks for the reconstructed tracks.

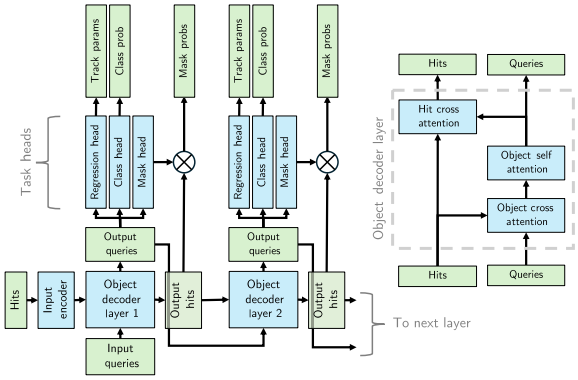

FIG. 1. Overview of the track reconstruction model, with data and operations being shown in green and blue, respectively. M input tokens representing the hits are fed into an initial transformer encoder. The object decoder then takes a set of N object queries, which represent tracks, and iteratively updates them with information from the input elements and other object queries. Finally, three task heads are used to categorize each object as being from one of $C + 1$ object classes (including a 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-hand side, a detailed view of an object decoder layer is shown.

The object decoder initializes 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 Sec. IV A). The object queries are then passed through eight decoder layers. In each layer, the object queries aggregate information from relevant hits via bidirectional CA, and from other object queries via SA. The `MaskAttention` operator [65] 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.

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 binarized, and a value of 1 indicates assignment of the hit to the track, while 0 indicates no assignment. Using an elementwise sigmoid operation as opposed to applying a softmax over the hits or tracks allows the model to assign a single hit to multiple tracks, which can occur due to random track crossings or from collimated particles in the core of high-p_T jets [72]. 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 [73] $L_{Regression}$ is used. The mask loss L_{Mask} is a combination of a Dice loss L_{Dice} [74] and focal loss L_{Focal} [75]. The total loss is then the weighted sum,

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

where the weights are coarsely optimized 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 [76]. Regression targets are scaled to be of order ~ 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 [65].

IV. DATASET AND EXPERIMENTAL SETUP

A. Dataset

The TrackML challenge [20,48,66] was established to promote the development of novel techniques for particle track reconstruction in the high pileup environments anticipated at the HL-LHC. It has since become a standard benchmark for evaluating track reconstruction methods. The dataset simulates a generalized LHC-like detector, inspired by planned ATLAS and CMS upgrades for the HL-LHC, and is composed of approximately 9000 simulated events. Each event features one hard top quark-antiquark pair ($t\bar{t}$) interaction, overlaid with an additional 200 soft QCD interactions, simulating the expected high pileup conditions at the HL-LHC.

An x-y and r-z view of a sample event is depicted in Figs. 2 and 3, respectively. As shown in Fig. 4, $\mathcal{O}(10^4)$ particles and $\mathcal{O}(10^5)$ hits are simulated per event, resulting in $\mathcal{O}(10^8)$ tracks to be identified from approximately $\mathcal{O}(10^9)$ 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 end cap disks on each side. The outer tracking system comprises strip detectors with six barrel layers and six end cap disks per side. In this work, we focus exclusively on the innermost pixel detector, including the central barrel and end cap 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.

Hits are three-dimensional space points formed from pixel elements which have been clustered together.

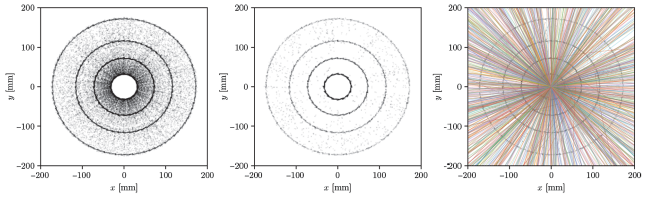

FIG. 2. Event display in the transverse x-y plane, showing a view down the beamline. This projection captures the cylindrical detector cross section perpendicular to the beam axis (z), with the origin corresponding to the nominal interaction point. Hits and tracks are shown in the transverse plane, where particle trajectories typically curve due to the magnetic field oriented along the z axis. Left: the positions of pixel hits for a single event. Middle: hits passing the 750-MeV filter at a cut of 0.1. Right: filtered hits along with the trajectories of reconstructible particles (assuming a homogeneous 2-T magnetic field) that satisfy $p_T > 1$ GeV and $|\eta| < 2.5$.

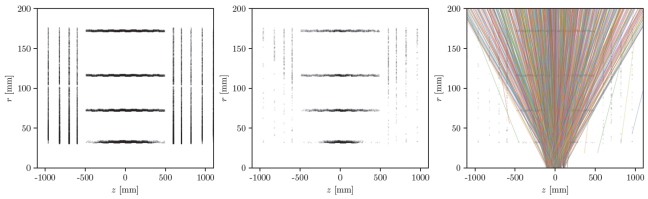

FIG. 3. Event display in the r-z projection of a cylindrical detector. The horizontal axis corresponds to the beamline direction (z), and the vertical axis is the radial distance from the beamline (r). The azimuthal coordinate (ϕ) is suppressed in this view, such that tracks and hits are projected onto the r-z plane irrespective of their azimuthal angle. Left: the positions of pixel hits for a single event. Middle: hits passing the 750-MeV filter at a cut of 0.1. Right: filtered hits along with the trajectories of reconstructible particles (assuming a homogeneous 2-T magnetic field) that satisfy $p_{\mathrm{T}} > 1$ GeV and $|\eta| < 2.5$.

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 center 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 [77]. 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 Ref. [78].

B. Experiments and training setup

In this work, we focus on the task of reconstructing charged particles from hits in the pixel layers of the TrackML detector. Using only pixel layer information presents a challenging reconstruction task with limited data, requiring the model to efficiently leverage limited information about each particle. We perform three different training experiments, each with different selection criteria for the reconstructed particles. In all cases, particles must satisfy the base criterion of having at least three hits left in the pixel layers. All hits from particles not matching this criteria are labeled as noise and targeted for removal by the hit filtering model. The transverse momentum and pseudorapidity selections vary across the three experimental configurations:

(1) $p_{\mathrm{T}}^{\mathrm{min}} = 600$ MeV: Pseudorapidity selection of $|\eta|^{\mathrm{max}} = 2.5$
(2) $p_{\mathrm{T}}^{\mathrm{min}} = 750$ MeV: Pseudorapidity selection of $|\eta|^{\mathrm{max}} = 2.5$
(3) $p_{\mathrm{T}}^{\mathrm{min}} = 1$ GeV: Wider pseudorapidity selection of $|\eta|^{\mathrm{max}} = 4.0$, demonstrating the model's capability to reconstruct tracks efficiently across the full detector acceptance.

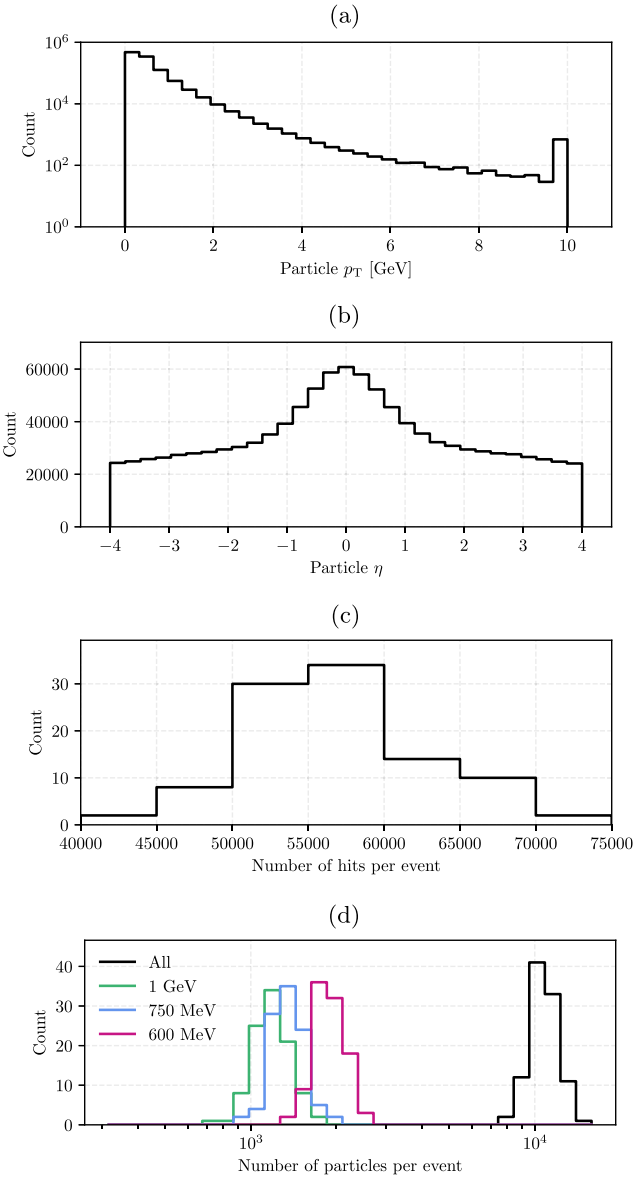

FIG. 4. Histograms summarizing the kinematics and object multiplicities of the TrackML dataset restricted to the inner pixel layers. (a) Transverse momentum (p_{T}) distribution of all particles. (b) Pseudorapidity (η) distribution of all particles. (c) Number of pixel layer hits per event before filtering. (d) Number of total particles per event, showing all particles (black) and those passing the three p_{T} thresholds used in this work.

For each experiment, a hit filtering model is first trained to select hits from reconstructable particles. A tracking model is trained to reconstruct tracks based on the hits that pass the filtering model's selection. The filtering models each have approximately 8×10^6 trainable parameters, while the tracking models have approximately 22×10^6 trainable parameters. A summary of the different experiments is provided in Table II. The different choices of $p_{\mathrm{T}}^{\mathrm{min}}$ and $|\eta|^{\mathrm{max}}$ allow us to explore the trade-offs between model complexity, inference time, and performance. They also

TABLE II. Summary of the models trained with different p_T^{min} and $|\eta|^{max}$ selections. For each selection, the number of hits prefiltering and postfiltering 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.

p_T^{min}	$\|\eta\|^{max}$	Hits (prefiltering)	Hits (postfiltering)	Particles	Object queries
600 MeV	2.5	57,000	12,000	1800	2100
750 MeV	2.5	57,000	8,000	1300	1800
1 GeV	4.0	57,000	11,000	1100	1500

provide an insight into the impact of reducing the effective training statistics (i.e., the number of target particles), as discussed in Sec. V.

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 h, while the tracking models trained in 20–60 h, depending on the p_T^{min} and $|\eta|^{max}$ selections applied. A batch size of a single event is used. Inference times are provided in Sec. V C.

V. RESULTS

As described in Sec. IV A, we consider a particle to be reconstructible if it leaves at least three hits in the pixel detector, has an absolute pseudorapidity less than $|\eta|^{max}$, and a transverse momentum $p_T > p_T^{min}$. The hit filtering performance is discussed in Sec. V A, while the tracking performance is discussed in Sec. V B and inference times are presented in Sec. V C. All results shown are obtained using the test set of 100 events.

A. Hit filtering

In the hit filtering task, hits are labeled as signal if they belong to a reconstructible 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.

Figure 5 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 II and also visualized in Fig. 2.

As summarized in Table III, the 600-MeV model maintains a hit efficiency of 99.5% while improving purity from 15.6% prefilter to 72.7% postfilter. Similarly, the 750-MeV and 1-GeV models achieve efficiencies of 99.6% and 98.8%, respectively, with corresponding purities of 67.2% and 64.5%. This represents dramatic purity improvements from their prefilter values of 11.2% and 13.1%. When considering only hits from particles passing the requirement on $|\eta|^{max}$ (and excluding noise particles), the initial purities are higher at 34.0%, 24.3%, and 14.8%, respectively; however, the postfilter purity still represents a significant improvement.

We also examine the fraction of particles that remain reconstructible after filtering (i.e., those that retain at least three pixel hits). The results are shown as a function of the

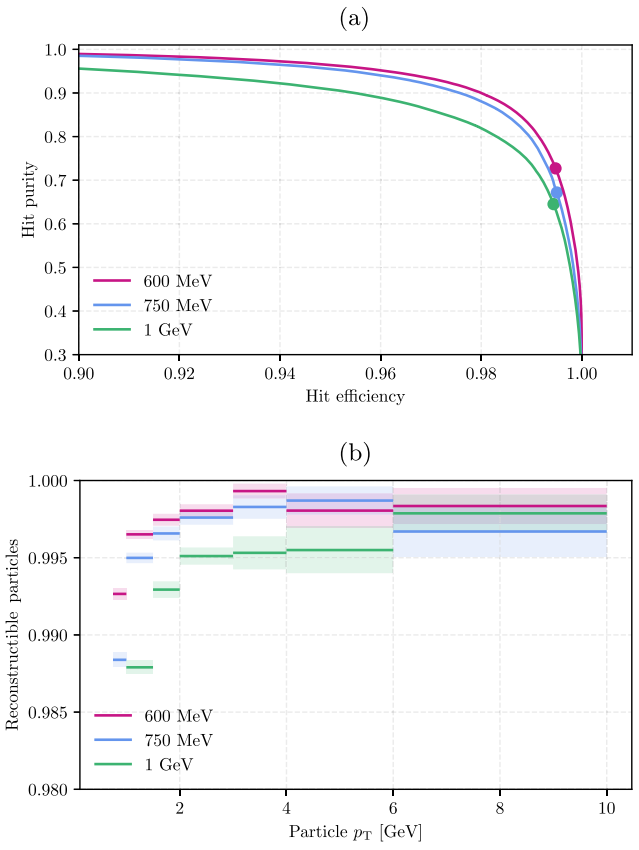

FIG. 5. Hit filtering performance for each of the three models. (a) Signal hit purity as a function of the signal hit efficiency. The markers show the efficiency and purity at the chosen threshold of 0.1 and each model achieves an area under the curve of 0.998. (b) The fraction of particles that remain reconstructible as a function of simulated particle p_T after filtering hits that fall below the 0.1 threshold. Binomial errors are indicated by the shaded regions, and the final bin includes overflow.

TABLE III. Performance of the hit filtering models on the test set. From left to right the columns show the initial hit purity (and the purity calculated using only hits from particles that fall within the specified $|\eta|^{\max}$ excluding detector noise hits not associated with any particle), the postfilter efficiency and purity, and the percentage of particles which remain reconstructible (i.e., retain three or more hits in the pixel layers) after the application of the filter. $|\eta|^{\max}$ is 2.5 for the 600- and 750-MeV models and 4 for the 1-GeV model. Statistical uncertainties are negligible.

Model	Initial purity ($\lvert\eta\rvert < \lvert\eta\rvert^{\max}$)	Filter efficiency	Filter purity	Reconstructible
600 MeV	15.6% (34.0%)	99.5%	72.7%	99.7%
750 MeV	11.2% (24.3%)	99.5%	67.2%	99.6%
1 GeV	13.1% (14.8%)	99.4%	64.5%	98.8%

particle p_T in Fig. 5, and integrated values are shown in Table III. For particles with $p_T > 1$ GeV, each model achieves excellent performance, preserving more than 99.5% of particles. The performance for each model degrades as the particle p_T approaches the $p_T^{\min}$ used to train the model, indicating that the $p_T^{\min}$ threshold needs to be set below the desired minimum particle p_T for reconstruction. Given the strong performance of the hit filtering model, we anticipate that this approach may find use as a preburner to other track reconstruction algorithms, including traditional approaches.

B. Tracking

Track reconstruction performance is evaluated in terms of the efficiency and fake rate. Tracking efficiency is defined as the fraction of reconstructible 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 reconstructible particle. These tracks either result from accidental combinations of unrelated hits or fail to meet the matching threshold due to relevant hits not being included in the predicted track mask or insufficient detector activity from the corresponding particle to have enough hits to satisfy the matching criteria. A lower fake rate indicates better suppression of such spurious predictions. We consider two different criteria to define whether a track is matched to a particle or not: *double majority* (DM) and *perfect* matching. Under the double majority 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. The efficiencies using the DM and perfect match criteria are noted as $\varepsilon^{\mathrm{DM}}_{p_T>1\ \mathrm{GeV}}$ and $\varepsilon^{\mathrm{perfect}}_{p_T>1\ \mathrm{GeV}}$, 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 matched simulated p_T. To enable comparison with other methods, we also define a fake rate $f^{\mathrm{DM}}_{p_T>0.9\ \mathrm{GeV}}$, which considers only tracks matched to target particles with $p_T > 0.9$ GeV. The particle-level inefficiencies resulting from the filtering step are included in the tracking efficiencies discussed here. The duplicate rate d is defined as the fraction of reconstructed tracks that have identical hit assignments.

The tracking performance of the different models depends on the values of $p_T^{\min}$ and $|\eta|^{\max}$ used to define the target particles during training. As shown in Fig. 6, all three models achieve a plateau efficiency of approximately 97.5% for particles satisfying $2 < p_T < 6$ GeV. Below 2 GeV, the efficiency drops significantly as the particle p_T approaches the $p_T^{\min}$ used to train the model, and falls to nearly zero below this threshold. The reconstruction efficiency for particles with $p_T > 6$ GeV drops slightly from the plateau to around 95%–96% depending on the model; however, the statistical uncertainties are large in this region due to the sharp decrease in particle multiplicity with increasing p_T. Notably, the models are able to perfectly reconstruct particles at high rates (above 90% in most cases). However, a stronger decreasing trend with p_T is observed for the perfect match, reflecting the increased difficulty in assigning hits at high p_T. The overall high performance underscores the model's 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).

Figure 6 also shows the fake and duplicate rates as a function of the reconstructed track p_T. In general, the fake rate remains low but shows an increasing trend with p_T. The 600-MeV model has a significantly higher fake rate at low p_T compared to the 750-MeV model (1.5% compared to 0.7%), possibly due to the increased complexity associated with reconstructing low p_T particles. The rate of duplicate tracks (those with identical hit assignments) is also examined, and is comfortably below 0.5% for tracks below 6 GeV for all models. Above 6 GeV, the duplicate rate increases to 0.5%–1.5% depending on the model, although statistical uncertainties are large.

The efficiencies, fake rates, and duplicate rates are shown as a function of η in Fig. 7. The efficiency remains

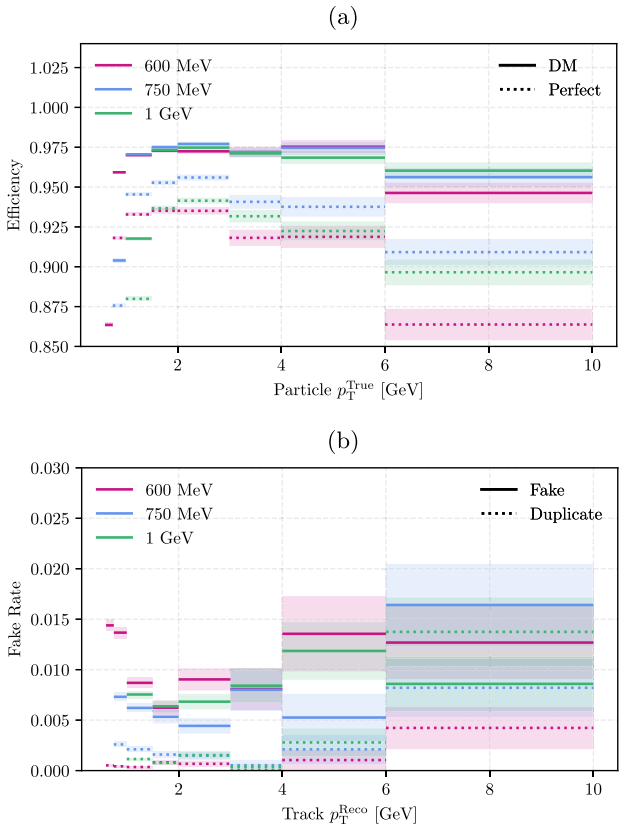

FIG. 6. (a) Track reconstruction efficiency as a function of the simulated particle p_T for the DM (solid lines) and perfect (dashed lines) matching criteria. (b) Fake rate using the DM matching criteria (solid lines) and duplicate rate (dashed lines) as a function of the reconstructed track p_T. The shaded regions indicate the binomial errors, and the final bin includes overflow.

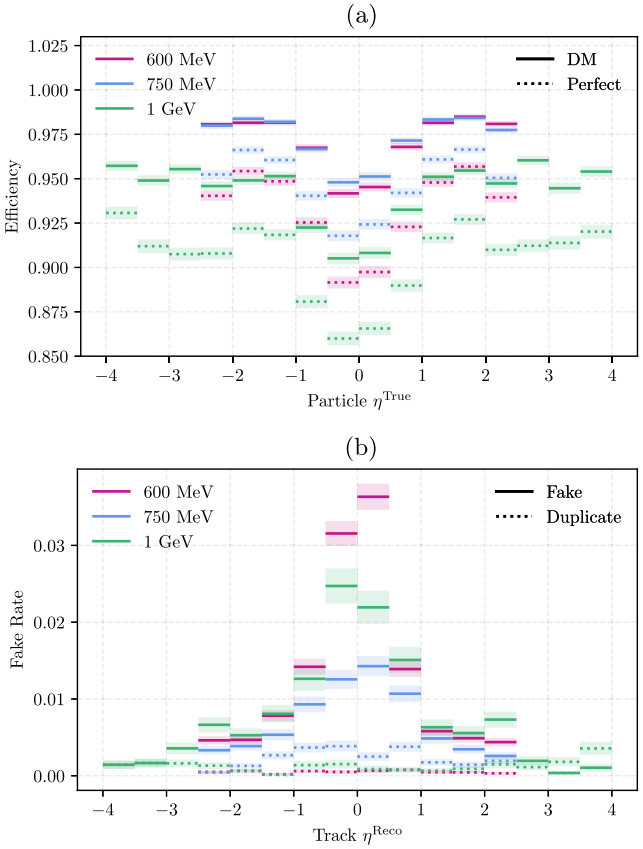

FIG. 7. (a) Track reconstruction efficiency as a function of the simulated particle η for the DM (solid lines) and perfect (dashed lines) matching criteria. For consistency, only particles with $p_T > 1$ GeV are considered. (b) Fake rate using the DM matching criteria (solid lines) and duplicate rate (dashed lines) as a function of the reconstructed track η. The shaded regions indicate the binomial errors.

high across the full η range, with a slight decrease of approximately 2.5% in the central region ($|\eta| < 1$) for all models. A corresponding increase in the fake rate is observed in this region, consistent with the increased particle density and p_T in this region. The efficiency of the 1-GeV model is slightly lower than the other two models due to the drop in reconstruction efficiency for particles with p_T approaching $p_T^{\min}$, as discussed above. The duplicate rate remains low across the full η range, with a slight increase in the central region.

Performance comparisons with existing methods are summarized in Tables IV and V, though several important methodological differences should be noted. While OC and HGNN include particles in the forward region ($|\eta| < 4$), our 600- and 750-MeV models focus on the more challenging central region ($|\eta| < 2.5$) where track density is highest, and our 1-GeV model uses $|\eta| < 4$. Given the reduced performance near the p_T threshold, the 750-MeV model provides the most meaningful comparison with OC. Despite the expectation that particles in the forward region are easier to reconstruct due to lower track density, the 750-MeV model achieves a slightly higher efficiency ($\varepsilon^{\mathrm{DM}}_{p_T>1\ \mathrm{GeV}} = 97.2\%$) than OC (96.4%) while reducing the fake rate by a factor of 3 to $f^{\mathrm{DM}}_{p_T>0.9\ \mathrm{GeV}} = 0.3\%$. In addition, our approach demonstrates improved perfect match efficiency, increasing the rate by 9 percentage points. The HGNN method includes hits on the strip layers and requires more than 5 hits for a particle to be reconstructible,

TABLE IV. Comparison of the results for the various models using the TrackML dataset on the test set. HGNN [79] includes hits from the pixel and strip layers. HGNN aims to reconstruct particles satisfying $|\eta| < 4$, whereas we target $|\eta| < 2.5$ for the 600- and 750-MeV models and $|\eta| < 4.0$ for the 1-GeV model. Statistical uncertainties are negligible.

Model	$\varepsilon^{\mathrm{perfect}}_{p_T>1\ \mathrm{GeV}}$	$\varepsilon^{\mathrm{DM}}_{p_T>1\ \mathrm{GeV}}$	f^{DM}	d
1 GeV	90.4%	94.1%	0.7%	0.1%
750 MeV	94.8%	97.2%	0.7%	0.2%
600 MeV	93.2%	97.1%	1.2%	0.1%
HGNN	⋯	97.9%	36.7%	⋯

TABLE V. Comparison of the results for the various models using the TrackML dataset on the test set. OC [57] uses $p_{\mathrm{T}}^{\mathrm{min}} = 0.9$ GeV and also includes hits from the pixel and strip layers. OC aims to reconstruct particles satisfying $|\eta| < 4$, whereas we target $|\eta| < 2.5$ for the 600- and 750-MeV models and $|\eta| < 4.0$ for the 1-GeV model. Statistical uncertainties are negligible.

Model	$\epsilon^{\mathrm{perfect}}_{p_{\mathrm{T}}>0.9\ \mathrm{GeV}}$	$\epsilon^{\mathrm{DM}}_{p_{\mathrm{T}}>0.9\ \mathrm{GeV}}$	$f^{\mathrm{DM}}_{p_{\mathrm{T}}>0.9\ \mathrm{GeV}}$	d
750 MeV	94.5%	97.1%	0.3%	0.2%
600 MeV	93.0%	97.0%	0.2%	0.1%
OC	85.8%	96.4%	0.9%	⋯

and would require additional postprocessing steps after the model to reduce the high fake rate. While these differences complicate direct comparisons, our results achieve comparable efficiencies with significantly better fake rate, all while using less information and being significantly faster, as discussed in Sec. V C.

Figure 8 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, with the exception of the p_{T} residuals which have a positive skew, possibly due to boundary effects or the relative abundance of low-p_{T} particles [80]. 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 [29,30]. 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, and may already be sufficiently precise for simple triggering applications. Several avenues exist for improving the parameter estimation. First, incorporating hits from the outer strip layers would significantly enhance momentum resolution. Second, directly regressing the parameters of interest, rather than constructing them from the Cartesian momentum, would likely further 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. Alternatively, a Kalman filter or least-squares fit could be applied to the set of hits assigned to a given track prediction, as is done in ATLAS and CMS, to extract high-precision track parameters.

Analysis of model performance versus training set size for the 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, our approach provides a convenient trade-off between model size, performance, and inference speed:

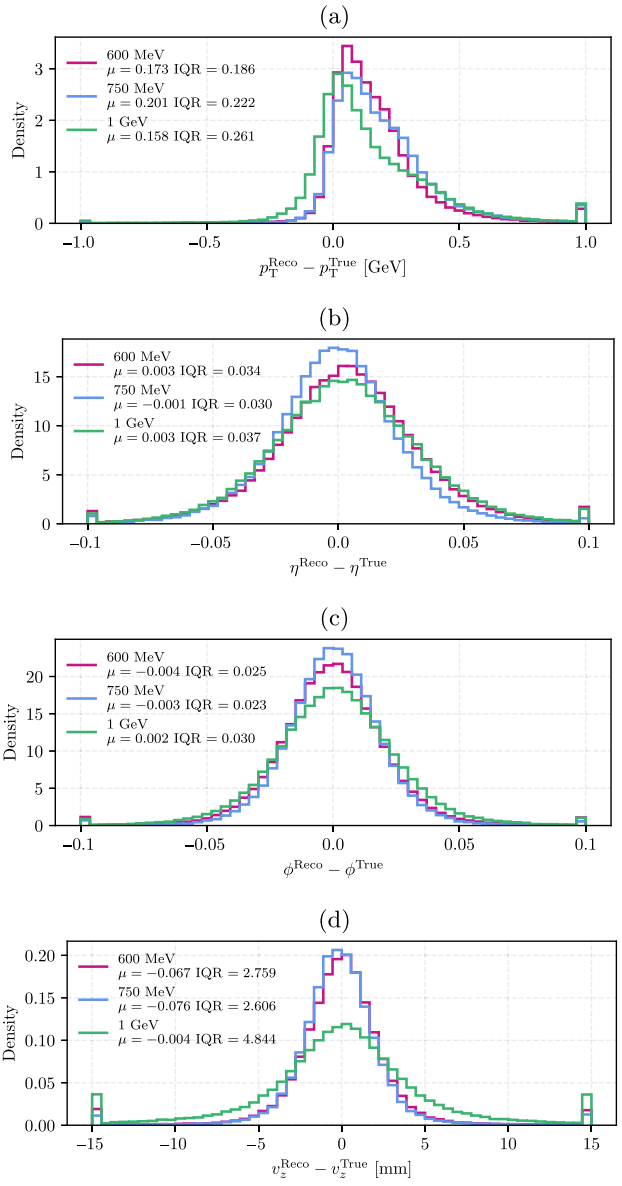

FIG. 8. Residuals between the regressed parameters of reconstructed tracks and the targets from the DM matched tracks. Tracks matched to simulated particles with $p_{\mathrm{T}} > p_{\mathrm{T}}^{\mathrm{min}}$ are used for each model. The mean μ and interquartile range (IQR) of the residuals for each model is shown in the legend. Residuals that lie outside the plot range are placed into extremal bins.

When inference speed is critical, model size can be reduced to achieve faster processing times, while in applications where speed is not the primary concern, model size can be increased to enhance tracking performance. This flexibility allows the same fundamental approach to be adapted to different experimental requirements and computational constraints.

C. Inference time

Inference time is a critical metric for track reconstruction at particle colliders, directly impacting the feasibility of

TABLE VI. 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.

	Tracking time (ms)	Filter + tracking time (ms)
600 MeV	100 ± 12	123 ± 14
750 MeV	74 ± 9	97 ± 11
1 GeV	76 ± 9	99 ± 11

real-time event processing. Our transformer-based models demonstrate competitive performance in this key metric. The hit filtering forward pass requires on average approximately 23 ± 3 ms to process a single event comprising O (60k) hits when using an NVIDIA A100 GPU. The hit filtering model is just in time (JIT) compiled via `torch.compile`, which significantly accelerates the forward pass during inference.

The tracking inference time depends on the choice of $p_{\mathrm{T}}^{\mathrm{min}}$, and ranges from approximately 75 ms for the 750-MeV and 1-GeV models to 100 ms for the 600-MeV model, as detailed in Table VI. For the 750-MeV model, which represents a good balance between tracking performance and inference time, the total time for hit filtering and tracking combined is approximately 97 ms on average.

To understand scaling behavior in order to estimate the timing requirements to consider all detector layers, we analyzed inference times as a function of hit multiplicity in Fig. 9. Both the filtering and tracking models exhibit linear scaling with the number of input hits, a critical result achieved through the architectural choices described in Sec. III A. This linear scaling is crucial for adaptability to varying event complexities.

The linear scaling observed in Fig. 9 enables us to extrapolate performance to the higher hit multiplicities anticipated in the full detector of a typical HL-LHC event. For the $\mathcal{O}(350\ \mathrm{k})$ hits expected in the ITk detector at the start of run 4 [81], we project a combined filter and tracking time of approximately 800 ms to reconstruct tracks with $p_{\mathrm{T}}^{\mathrm{min}} = 1$ GeV and $|\eta|^{\mathrm{max}} = 4$, assuming this linear scaling holds.

Direct comparisons of inference times across different track reconstruction methods are inherently challenging due to variations in experimental setups and differences in reported metrics. These include differences in detector geometry, the total number of hits considered, the inclusion of various subdetectors, and the choices of target particle definitions.

In light of these caveats, our projected inference time of ~800 ms for the full ITk detector is comparable to recent, optimized results from the GNN4ITk project [55], which represents a highly customized approach developed through extensive, multiyear optimization efforts. Remarkably, our end-to-end learned approach achieves comparable timing performance while leveraging largely off-the-shelf ML

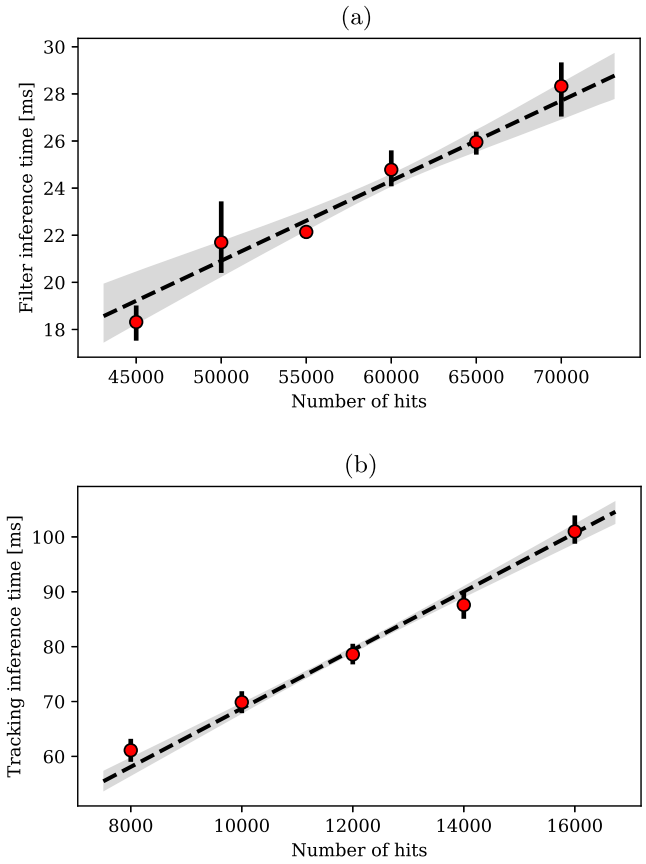

FIG. 9. Inference times as a function of the input hit multiplicity for the hit filtering model (a) and 1-GeV track reconstruction model (b), evaluated on an NVIDIA A100 GPU with a batch size of 1. The average time in each bin and the 95% confidence intervals are shown by the markers and vertical error bars. A linear fit is shown in the dashed line, along with a 95% confidence interval shown by the shaded region.

architectures demonstrating a strategy that offers potential advantages in flexibility and broader applicability, despite requiring significantly less development effort. Furthermore, our results improve upon the timing studies in Ref. [79] and are comparable with optimized non-ML approach to trigger-level tracking [34], as visualized in Fig. 10.

In this study, MaskFormer achieves a high efficiency and low fake rate using exclusively pixel detector hits. While incorporating strip hits could further improve performance, it would also increase hit multiplicity and thus processing time. A key finding is the strong tracking performance achieved without this additional information, suggesting a potential advantage: For certain applications, the computational cost of processing all strip hits might be avoidable if pixel-based performance is sufficient. Thus, the ≈ 800 ms projection for all $\mathcal{O}(350\ \mathrm{k})$ ITk hits might be an upper limit if strong performance is maintained by primarily using pixel data or a subset of strip hits, enabling faster inference.

The timing estimates assume continued linear scaling with hit multiplicity. In realistic detector environments

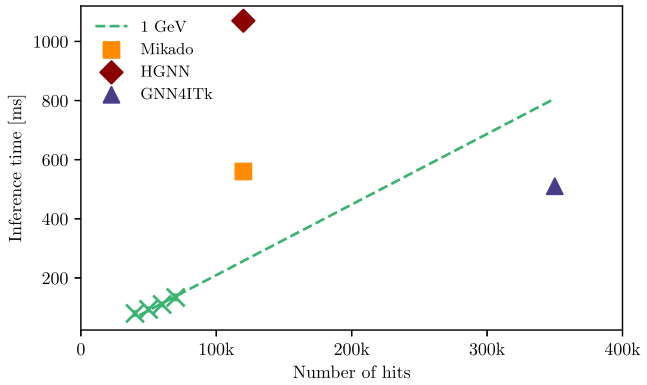

FIG. 10. Combined filtering and tracking inference times for the MaskFormer track reconstruction model (1 GeV, trained with $|\eta|^{\max} = 4$) as a function of input hit multiplicity. A linear fit to the MaskFormer data is shown and, assuming its validity, extrapolated to the O(350k) hits expected in the full ITk detector. This extrapolation allows for an approximate comparison with other approaches: Mikado [20], HGNN [79], and GNN4ITk [55]. Direct comparisons should be approached with caution due to significant methodological differences: MaskFormer, Mikado, and HGNN are evaluated on the TrackML dataset, while GNN4ITk uses a realistic ITk detector simulation. Furthermore, MaskFormer, HGNN, and GNN4ITk are benchmarked on NVIDIA A100 GPUs, whereas Mikado is evaluated on 2 CPU cores. Additional variations in detector geometries, event multiplicities, and target particle definitions may also affect comparisons. This figure shows that the model presented has great potential to be the state-of-the-art solution for a highly combinatoric process: fast and accurate charged particle reconstruction.

additional factors, such as geometric irregularities or nonuniform magnetic fields, may necessitate larger ϕ windows and introduce deviations from this behavior. Hardware constraints, including the inability to fit the full model and input data into device memory, could render training infeasible. Validating these extrapolations using full detector simulations remains an important future step.

Nevertheless, the current inference times have been achieved with minimal optimization, indicating substantial potential for future speed improvements. Our approach is poised for significant enhancements through architecture optimization, enabling JIT compilation for the full tracking model, and the application of established techniques such as model pruning and quantization. Additional gains are anticipated from training smaller, more efficient models on expanded datasets, while inference throughput can be further boosted by increasing the batch size. These clearly defined avenues for development are expected to markedly enhance the real-world applicability and performance of our method.

VI. 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 strong performance, with the 750-MeV model achieving an efficiency for particles with $p_{\mathrm{T}} > 1$ GeV of approximately 97.2% and a fake rate of 0.7%. We also demonstrate the model's capacity to reconstruct low-p_{T} tracks with the 600-MeV configuration, and tracks across the full $|\eta|$ range with the 1-GeV, $|\eta|^{\max} = 4$ model. The model's inference time of approximately 100 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.

While our method demonstrates promising performance on the TrackML dataset, we note that real detector environments introduce additional complexities not included in this dataset. For example, including nonuniform geometries, material interactions, sensor inefficiencies, and magnetic field inhomogeneities can all affect performance and time-scaling behaviors. Evaluating the robustness of transformer-based tracking in such settings remains an important future direction. It will also be critical to assess generalizability and deployment feasibility by testing on simulated events in the planned ATLAS and CMS tracking detector upgrades.

There are several directions for future work to enhance the capabilities and applicability of the model. First, we plan to improve the algorithmic efficiency of the model, allowing it to handle the increased hit multiplicities that would stem from including the strip hits, and reconstructing additional particles with even lower transverse momenta. Second, combining the hit filtering and track reconstruction stages into a single model could streamline the training process and reduce inference times. Third, we plan to test the model and scaling properties on more realistic detector simulations, including irregularities in the detector layout, material effects, and magnetic field variations to validate generalization. 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 charged and neutral particles, paving the way toward full particle-flow-style reconstruction.

In the near term, the architecture is already suitable for hybrid use within existing reconstruction pipelines. The model's tunable nature, enabling trade-offs between performance and inference time and targeting of specific

particle kinematics, makes it adaptable to various high-energy physics experiments, from trigger-level systems to off-line reconstruction, without requiring changes to established reconstruction techniques. The hit filtering component can be readily integrated as a fast preprocessing step to reduce input multiplicity. Similarly, the tracking model could act as a learned track-seeding mechanism, replacing or augmenting standard pattern recognition algorithms and integrating seamlessly with current pipelines to deliver high-quality track parameter estimates.

Beyond these immediate uses, our work contributes to a broader shift in reconstruction strategy. An end-to-end optimizable framework for charged particle reconstruction, such as the one presented, could offer significant benefit to a diverse range of experimental setups beyond traditional collider experiments. Most significantly, the success of transformer-based architectures in both tracking and vertexing points toward a unified approach to particle physics reconstruction, moving away from specialized solutions and toward generalized learned models leveraging cross-domain advances in machine learning. Such models could significantly impact how we process and analyze particle collision data and meet the evolving challenges of high-energy physics.

ACKNOWLEDGMENTS

We gratefully acknowledge the support of the UK's Science and Technology Facilities Council (STFC). S. V. S is supported by STFC (Project No. ST/X005992/1). P. D., M. H., and N. P. are supported by the STFC UCL Centre for Doctoral Training in Data Intensive Science (Project No. ST/W00674X/1) and by departmental and industry contributions. G. F. and T. S. received support from the STFC (Project No. ST/W00058X/1) and T. S. is supported by the Royal Society (Project No. URF/R/180008). S. R. is supported by CERN, the Banting Postdoctoral Fellowship program, and the Natural Sciences and Engineering Research Council of Canada. 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. We thank Finnbar Wilson for assistance in the creation of the accompanying visual image.

[1] L. Evans and P. Bryant, *LHC machine*, J. Instrum. **3**, S08001 (2008).

[2] G. Aad *et al.* (ATLAS Collaboration), *Combination of searches for Higgs boson pairs in pp collisions at $\sqrt{s} =$ 13 TeV with the ATLAS detector*, Phys. Lett. B **800**, 135103 (2020).

[3] A. Hayrapetyan *et al.* (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*, Phys. Lett. B **861**, 139210 (2025).

[4] G. Aad *et al.* (ATLAS Collaboration), *Electron and photon efficiencies in LHC run 2 with the ATLAS experiment*, J. High Energy Phys. 05 (2024) 162.

[5] G. Aad *et al.* (ATLAS Collaboration), *Muon reconstruction and identification efficiency in ATLAS using the full run $2pp$ collision data set at $\sqrt{s} = 13$ TeV*, Eur. Phys. J. C **81**, 578 (2021).

[6] A. M. Sirunyan *et al.* (CMS Collaboration), *Electron and photon reconstruction and identification with the CMS experiment at the CERN LHC*, J. Instrum. **16**, P05014 (2021).

[7] A. M. Sirunyan *et al.* (CMS Collaboration), *Performance of the CMS muon detector and muon reconstruction with proton-proton collisions at $\sqrt{s} = 13$ TeV*, J. Instrum. **13**, P06015 (2018).

[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*, Technical Report, CERN, Geneva, 2022, https://cds.cern.ch/record/2827111/.

[9] A. M. Sirunyan *et al.* (CMS Collaboration), *Performance of reconstruction and identification of τ leptons decaying to hadrons and ν_τ in pp collisions at $\sqrt{s} = 13$ TeV*, J. Instrum. **13**, P10005 (2018).

[10] M. Aaboud *et al.* (ATLAS Collaboration), *Jet reconstruction and performance using particle flow with the ATLAS Detector*, Eur. Phys. J. C **77**, 466 (2017).

[11] A. M. Sirunyan *et al.* (CMS Collaboration), *Particle-flow reconstruction and global event description with the CMS detector*, J. Instrum. **12**, P10003 (2017).

[12] G. Aad *et al.* (ATLAS Collaboration), *ATLAS flavour-tagging algorithms for the LHC run $2pp$ collision dataset*, Eur. Phys. J. C **83**, 681 (2023).

[13] G. Aad *et al.* (ATLAS Collaboration), *Transforming jet flavour tagging at ATLAS*, arXiv:2505.19689.

[14] A. M. Sirunyan *et al.* (CMS Collaboration), *Identification of heavy-flavour jets with the CMS detector in pp collisions at 13 TeV*, J. Instrum. **13**, P05011 (2017).

[15] ATLAS Collaboration, *ATLAS HL-LHC computing conceptual design report*, Technical Report, CERN, 2020, https://cds.cern.ch/record/2729668.

[16] CMS Offline Software and Computing, *CMS Phase-2 computing model: Update document*, Technical Report, CERN, Geneva, 2022, https://cds.cern.ch/record/2815292.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, *Attention is all you need*, in *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, edited by I. Guyon, U. von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett (Curran Associates, Inc., Red Hook, NY, 2017), pp. 5998–6008, arXiv:1706.03762.

[18] B. Cheng, A. G. Schwing, and A. Kirillov, *Per-pixel classification is not all you need for semantic segmentation*, in *Proceedings of the 35th International Conference on Neural Information Processing Systems (NIPS '21)* (Neural Information Processing Systems, San Diego CA, 2021).

[19] S. Van Stroud, N. Pond, M. Hart, J. Barr, S. Rettie, G. Facini, and T. Scanlon, *Secondary vertex reconstruction with MaskFormers*, Eur. Phys. J. C **84**, 1020 (2024).
[20] S. Amrouche *et al.*, *The tracking machine learning challenge: Throughput phase*, Comput. Software Big Sci. **7**, 1 (2023).
[21] A. A. Alves, Jr. *et al.* (LHCb Collaboration), *The LHCb detector at the LHC*, J. Instrum. **3**, S08005 (2008).
[22] K. Aamodt *et al.* (ALICE Collaboration), *The ALICE experiment at the CERN LHC*, J. Instrum. **3**, S08002 (2008).
[23] L. Vigani and T. Rudzki (Mu3e Collaboration), *The Mu3e detector*, J. Instrum. **17**, C05024 (2022).
[24] H. Abramowicz *et al.* (LUXE Collaboration), *Technical design report for the LUXE experiment*, Eur. Phys. J. Special Topics **233**, 1709 (2024).
[25] X. Ai *et al.*, *A common tracking software project*, Comput. Software Big Sci. **6**, 8 (2022).
[26] R. Frühwirth, *Application of Kalman filtering to track and vertex fitting*, Nucl. Instrum. Methods Phys. Res., Sect. A **262**, 444 (1987).
[27] D. Decamp *et al.* (ALEPH Collaboration), *ALEPH: A detector for electron-positron annihilations at LEP*, Nucl. Instrum. Methods Phys. Res., Sect. A **294**, 121 (1990).
[28] P. Aarnio *et al.* (DELPHI Collaboration), *The DELPHI detector at LEP*, Nucl. Instrum. Methods Phys. Res., Sect. A **303**, 233 (1991).
[29] G. Aad *et al.* (ATLAS Collaboration), *Software performance of the ATLAS track reconstruction for LHC run 3*, Comput. Software Big Sci. **8**, 9 (2023).
[30] S. Chatrchyan *et al.* (CMS Collaboration), *Description and performance of track and primary-vertex reconstruction with the CMS tracker*, J. Instrum. **9**, P10009 (2014).
[31] R. Aaij *et al.* (LHCb Collaboration), *Design and performance of the LHCb trigger and full real-time reconstruction in run 2 of the LHC*, J. Instrum. **14**, P04013 (2019).
[32] Y. Belikov, M. Ivanov, K. Safarik, and J. Bracinik, *TPC tracking and particle identification in high density environment*, eConf C **0303241**, TULT011 (2003).
[33] ATLAS Collaboration, Fast track reconstruction for HL-LHC, Technical Report, CERN, Geneva, 2019, https://cds.cern.ch/record/2693670.
[34] ATLAS Collaboration, *Technical design report for the phase-II upgrade of the ATLAS trigger and data acquisition system—event filter tracking amendment*, Technical Report, CERN, Geneva, 2022, https://cds.cern.ch/record/2802799.
[35] CMS Collaboration, *The phase-2 upgrade of the CMS level-1 trigger*, Technical Report, CERN, Geneva, 2020, final version, https://cds.cern.ch/record/2714892.
[36] ATLAS Collaboration, *Improvements and current status of the ACTS-based ITk main pass reconstruction*, ATLAS Public Note IDTR-2025-04, 2025, https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PLOTS/IDTR-2025-04/.
[37] J. von Neumann, *Theory of Self-Reproducing Automata*, edited by A. W. Burks (University of Illinois Press, Urbana, IL, 1966).
[38] C. Verkerk and W. Wojcik*Proceedings of the 10th International Conference on Computing in High-Energy Physics (CHEP '92)*, edited by C. Verkerk and W. Wojcik (CERN, Annecy, France, 1992).
[39] A. Glazov, I. Kisel, E. Konotopskaya, and G. Ososkov, *Filtering tracks in discrete detectors using a cellular automaton*, Nucl. Instrum. Methods Phys. Res., Sect. A **329**, 262 (1993).
[40] M. P. Bussa, L. Fava, L. Ferrero, A. Grasso, and V. V. e. a. Ivanov, *On a possible second level trigger for the experiment disto*, Nuovo Cimento A **109**, 327 (1996).
[41] I. Kisel, V. Kovalenko, F. Laplanche, R. Arnold, and C. e. a. Augier, *Cellular automaton and elastic net for event reconstruction in the NEMO-2 experiment*, Nucl. Instrum. Methods Phys. Res., Sect. A **387**, 433 (1997).
[42] A. Bocci, V. Innocente, M. Kortelainen, F. Pantaleo, and M. Rovere, *Heterogeneous reconstruction of tracks and primary vertices with the CMS pixel tracker*, Front. Big Data **3**, 601728 (2020).
[43] CMS Collaboration, *Performance of track reconstruction at the CMS high-level trigger in 2022 data*, CMS detector performance summary CMS-DP-2023-028, 2023, https://cds.cern.ch/record/2860207.
[44] CMS Collaboration, *The phase-2 upgrade of the CMS tracker*, Technical Report, CERN, Geneva, 2017. https://cds.cern.ch/record/2272264.
[45] D. Rohr, S. Gorbunov, and V. Lindenstruth (ALICE Collaboration), *GPU-accelerated track reconstruction in the ALICE High Level Trigger*, J. Phys. Conf. Ser. **898**, 032030 (2017).
[46] P. V. C. Hough, *Method and means for recognizing complex patterns*, U.S. Patent No. 33,069,654 (1962).
[47] R. Aaij *et al.*, *Allen: A high level trigger on GPUs for LHCb*, Comput. Software Big Sci. **4**, 7 (2020).
[48] S. C. Amrouche *et al.*, *The tracking machine learning challenge: Accuracy phase*, *The NeurIPS '18 Competition* (Springer, Cham, 2019).
[49] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, *The graph neural network model*, IEEE Trans. Neural Networks **20**, 61 (2009).
[50] C. Rougier, A. Vallier, D. T. Murnane, J. Stark, P. Calafiura, S. Farrell, S. Caillou, and X. Ju, *CTD2022: ATLAS ITk Track Reconstruction with a GNN-based Pipeline*, *Connecting the Dots Workshop 2022 (CTD2022)* (Zenodo, Geneva Switzerland, 2023).
[51] S. Caillou, P. Calafiura, X. Ju, D. Murnane, T. Pham, C. Rougier, J. Stark, and A. Vallier, *Physics performance of the ATLAS GNN4ITk track reconstruction chain*, EPJ Web Conf. **295**, 03030 (2024).
[52] X. Ju *et al.*, *Performance of a geometric deep learning pipeline for HL-LHC particle tracking*, Eur. Phys. J. C **81**, 876 (2021).
[53] C. Biscarat, S. Caillou, C. Rougier, J. Stark, and J. Zahreddine, *Towards a realistic track reconstruction algorithm based on graph neural networks for the HL-LHC*, EPJ Web Conf. **251**, 03047 (2021).
[54] S. Caillou *et al.*, *Novel fully-heterogeneous GNN designs for track reconstruction at the HL-LHC*, in *Proceedings of the 26th International Conference on Computing in High Energy and Nuclear Physics (CHEP 2023)* (2023), Vol. 295, p. 09028, 10.1051/epjconf/202329509028.
[55] ATLAS Collaboration, *Computational performance of the ATLAS ITk GNN track reconstruction pipeline*,

Technical Report, CERN, Geneva, 2024, https://cds.cern.ch/record/2914282.

[56] R. Liu, P. Calafiura, S. Farrell, X. Ju, D. T. Murnane, and T. M. Pham, *Hierarchical graph neural networks for particle track reconstruction*, in *Proceedings of the 21st International Workshop on Advanced Computing and Analysis Techniques in Physics Research: AI meets Reality* (2023), arXiv:2303.01640.

[57] K. Lieret, G. DeZoort, D. Chatterjee, J. Park, S. Miao, and P. Li, *High pileup particle tracking with object condensation*, arXiv:2312.03823.

[58] S. Caron, N. Dobreva, A. Ferrer Sánchez, J. D. Martín-Guerrero, U. Odyurt, R. Ruiz de Austri Bazan, Z. Wolffs, and Y. Zhao, *TrackFormers: In search of transformer-based particle tracking for the high-luminosity LHC era*, arXiv:2407.07179.

[59] A. Huang, Y. Melkani, P. Calafiura, A. Lazar, D. T. Murnane, M.-T. Pham, and X. Ju, *A language model for particle tracking*, arXiv:2402.10239.

[60] Y. Melkani and X. Ju, *TrackSorter: A transformer-based sorting algorithm for track finding in high energy physics*, arXiv:2407.21290.

[61] S. Miao, Z. Lu, M. Liu, J. Duarte, and P. Li, *Locality-sensitive hashing-based efficient point transformer with applications in high-energy physics*, arXiv:2402.12535.

[62] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, *You only look once: Unified, real-time object detection*, arXiv:1506.02640.

[63] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, *End-to-end object detection with transformers*, arXiv:2005.12872.

[64] K. He, G. Gkioxari, P. Dollár, and R. Girshick, *Mask R-CNN*, arXiv:1703.06870.

[65] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, *Masked-attention mask transformer for universal image segmentation*, in *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Computer Vision and Pattern Recognition, 2021), pp. 1280–1289.

[66] M. Kiehn, S. Amrouche, P. Calafiura, V. Estrade, S. Farrell, C. Germain, V. Gligorov, T. Golling, H. Gray, I. Guyon, M. Hushchyn, V. Innocente, E. Moyse, D. Rousseau, A. Salzburger, A. Ustyuzhanin, J.-R. Vlimant, and Y. Yilnaz, *The TrackML high-energy physics tracking challenge on Kaggle*, Eur. Phys. J. Web Conf. **214**, 06037 (2019).

[67] I. Beltagy, M. E. Peters, and A. Cohan, *Longformer: The long-document transformer*, arXiv:2004.05150.

[68] T. Dao, *FlashAttention-2: Faster attention with better parallelism and work partitioning*, arXiv:2307.08691.

[69] N. Shazeer, *GLU variants improve transformer*, arXiv:2002.05202.

[70] H. Lee, C. Homeyer, R. Herzog, J. Rexilius, and C. Rother, *Spatio-temporal outdoor lighting aggregation on image sequences using transformer networks*, Int. J. Comput. Vis. **131**, 1060 (2022).

[71] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. B. Girshick, *Segment anything*, in *Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV)* (Curran Associates, New York, 2023), pp. 3992–4003.

[72] M. Aaboud *et al.* (ATLAS Collaboration), *Performance of the ATLAS track reconstruction algorithms in dense environments in LHC run 2*, Eur. Phys. J. C **77**, 673 (2017).

[73] R. Girshick, *Fast R-CNN*, arXiv:1504.08083.

[74] C. H. Sudre, W. Li, T. K. M. Vercauteren, S. Ourselin, and M. J. Cardoso, *Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations*, in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, Held in Conjunction with MICCAI 2017, Quebec City, QC* (Springer, Cham, 2017), p. 240.

[75] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, *Focal loss for dense object detection*, IEEE Trans. Pattern Anal. Mach. Intell. **42**, 318 (2020).

[76] S. Guthe and D. Thuerck, *Algorithm 1015: A fast scalable solver for the dense linear (sum) assignment problem*, ACM Trans. Math. Softw. **47**, 1 (2021).

[77] M. Hansroul, H. Jeremie, and D. Savard, *Fast circle fit with the conformal mapping method*, Nucl. Instrum. Methods Phys. Res., Sect. A **270**, 498 (1988).

[78] P. J. Fox, S. Huang, J. Isaacson, X. Ju, and B. Nachman, *Beyond 4D tracking: Using cluster shapes for track seeding*, J. Instrum. **16**, P05001 (2021).

[79] R. Liu, P. Calafiura, S. Farrell, X. Ju, D. T. Murnane, and T. M. Pham, *Hierarchical graph neural networks for particle track reconstruction*, in *Proceedings of the 21st International Workshop on Advanced Computing and Analysis Techniques in Physics Research: AI meets Reality* (2023), arXiv:2303.01640.

[80] Z. Cao, Z. Liu, D. Luo, H. Zhou, Z. Bu, T. Zhang, T. Zhao, M. R. Lyu, Z. Lin, B. Li *et al.*, *Systematic bias of machine learning regression models and correction*, arXiv:2405.15950.

[81] J. D. Bossio Sola *et al.* (ATLAS Collaboration), *Technical design report for the ATLAS inner tracker strip detector*, Technical Report, CERN, Geneva, 2017.