Locality-Sensitive Hashing-Based Efficient Point Transformer for Charged Particle Reconstruction

Shitij Govil[1,*] **Jack P. Rodgers**[2,*] **Yuan-Tang Chou**[3,†] **Siqi Miao**[1,†] **Amit Saha**[1]

Advaith Anand[3] **Kilian Lieret**[4] **Gage DeZoort**[4] **Mia Liu**[2]

Javier Duarte[5] **Pan Li**[1] **Shih-Chieh Hsu**[3]

[1]Georgia Institute of Technology [2]Purdue University [3]University of Washington
[4]Princeton University [5]University of California San Diego

Abstract

Charged particle track reconstruction is a foundational task in collider experiments and the main computational bottleneck in particle reconstruction. Graph neural networks (GNNs) have shown strong performance for this problem, but costly graph construction, irregular computations, and random memory access patterns substantially limit their throughput. The recently proposed Hashing-based Efficient Point Transformer (HEPT) offers a theoretically guaranteed near-linear complexity for large point cloud processing via locality-sensitive hashing (LSH) in attention computations; however, its evaluations have largely focused on embedding quality, and the object condensation pipeline on which HEPT relies requires a post-hoc clustering step (e.g., DBScan) that can dominate runtime. In this work, we make two contributions. First, we present a unified, fair evaluation of physics tracking performance for HEPT and a representative GNN-based pipeline under the same dataset and metrics. Second, we introduce HEPTv2 by extending HEPT with a lightweight decoder that eliminates the clustering stage and directly predicts track assignments. This modification preserves HEPT's regular, hardware-friendly computations while enabling ultra-fast end-to-end inference. On the TrackML dataset, optimized HEPTv2 achieves approximately 28 ms per event on an A100 while maintaining competitive tracking efficiency. These results position HEPTv2 as a practical, scalable alternative to GNN-based pipelines for fast tracking.

1 Introduction

With many fundamental questions unanswered in particle physics, the Large Hadron Collider (LHC) [Evans and Bryant, 2008] at CERN remains the flagship machine for understanding unsolved questions at the energy frontier. The LHC is scheduled for significant upgrades in its accelerator complex in the coming year, substantially increasing the number of simultaneous proton-proton interactions in each collision record (pile-up) for experiments such as ATLAS [Aad et al., 2024] and CMS [Chatrchyan et al., 2008], from <μ> = 60 to <μ> = 200. Consequently, a large number of particles need to be reconstructed and processed from the detector measurements. Particle trajectory reconstruction is the most computationally expensive, but key component of particle reconstruction and identification. Traditional rule-based algorithms such as the Combinatorial Kalman Filter (KF)

[*] Equal contribution. [†] Corresponding authors: `ytchou@uw.edu`, `siqi.miao@gatech.edu`.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Machine Learning and the Physical Sciences.

have been used primarily for this task [Aad et al., 2025]. However, the unprecedented data rates and complexity of High-Luminosity LHC (HL-LHC) conditions demand innovative approaches with lower inference latency.

The development of machine learning (ML)-based algorithms has emerged as an alternative solution to tackle these challenges. GNNs have been widely adopted, primarily because they exploit the local inductive bias inherent in particle interactions. However, their expensive graph construction and irregular neighborhood aggregation lead to hardware inefficiencies [Stark et al., 2022, Lieret et al., 2023b, Jang et al., 2010, Hashemi et al., 2018, Abadal et al., 2021], making GNNs less suitable for the data throughput needed for HL-LHC [CMS Collaboration, 2022]. In another line of work, multiple efficient point transformers have been proposed for large point cloud processing [Zhao et al., 2021, Liu et al., 2023, Wang et al., 2023, Wu et al., 2024], but their reliance on voxelization or fixed serialization patterns can poorly capture the irregular, complex geometry of tracking. HEPT [Miao et al., 2024] mitigates these issues by employing randomized serialization via LSH [Datar et al., 2004], preserving locality while reducing complexity to linear. However, its reliance on costly post-hoc clustering and the lack of physics-based evaluation limit its utility in real-time tracking.

To overcome these limitations, we present the first unified evaluation of HEPT against a strong, representative GNN pipeline [Caillou et al., 2022] on realistic HL-LHC datasets with physics-grounded metrics. In addition, we introduce HEPTv2, an end-to-end efficient tracking architecture that augments the standard HEPT encoder with a lightweight transformer decoder for direct track assignment, achieving an average of 28 ms per event on the TrackML dataset [Amrouche et al., 2023, 2019], using an NVIDIA A100. This design preserves HEPT's efficiency while improving scalability, establishing HEPTv2 as a competitive alternative to GNN-based pipelines.

2 Related Work

Traditional Methods and GNNs for Tracking. Traditional particle tracking methods, such as the KF and Hough Transforms, have been used extensively in high-energy physics experiments, but face performance limitations under high pile-up conditions due to their combinatorial complexity. Recent advancements in deep learning have driven the adoption of GNNs for track reconstruction, particularly through edge classification approaches [Ju et al., 2021, Lazar et al., 2023, DeZoort et al., 2021]. Additionally, object condensation techniques [Lieret et al., 2023a] have been explored.

Efficient Point Transformers. To avoid the quadratic complexity of standard self-attention [Vaswani et al., 2017] when processing large point clouds, Point Transformer [Zhao et al., 2021] introduces a local attention mechanism that couples positional encoding with neighborhood selection. More recently, studies have centered on point cloud serialization, which projects high-dimensional point-cloud data into one-dimensional sequences while preserving locality. Recent works such as FlatFormer [Liu et al., 2023], DSVT [Wang et al., 2023], and PointTransformerV3 [Wu et al., 2024] adopt fixed serialization patterns, which improve tractability but struggle with irregular detector geometry. Among these, HEPT [Miao et al., 2024] stands out by employing randomized serialization via LSH, providing a theoretical guarantee of near-linear complexity and offering strong potential for tracking.

3 Methodology

3.1 Exa.TrkX

We select Exa.TrkX [Caillou et al., 2022] as the representative GNN-based pipeline for tracking. Exa.TrkX consists of three stages, in which two different neural networks are employed. The first stage is the metric learning stage, where the features are pushed into an embedding space via multilayer perceptron (MLP), which then outputs higher-dimensional features, with the n-dimensional distance determining whether two hits are neighboring. A GNN is then trained to prune out false positive edges. Finally, the track building stage reconstructs the particle tracks from the pruned graphs, with track efficiency determined by comparing to truth. The Exa.TrkX pipeline is retrained and optimized to maintain high pixel tracking efficiency using the acorn framework [Atkinson et al.].

3.2 Locality-Sensitive Hashing-Based Efficient Point Transformer (HEPT)

HEPT extends the transformer by incorporating LSH into the attention mechanism to achieve near-linear complexity while encoding a local inductive bias. Concretely, E2LSH is used with OR & AND constructions to map points to one-dimensional buckets and apply block-diagonal attention within buckets. Each point is hashed into m_1 tables (OR-LSH), where each table concatenates m_2 independent hash functions (AND-LSH): $g(x) = [h_1(x), \ldots, h_{m_2}(x), \quad h_j(x) = \lfloor \frac{a_j \cdot x + b_j}{r} \rfloor$. After mapping the OR & AND hash into one single hash code, points are sorted to yield a 1D serialization, which allows the employment of block-diagonal (windowed) attention, converting irregular all-to-all interactions into regular, hardware-friendly operations.

3.3 HEPTv2

We extend HEPT as an end-to-end particle tracker, mapping raw detector hits directly to track identities. The standard HEPT module is used to create encoded embeddings for each particle hit, which is then passed into a point-wise classifier that determines which points are likely to be part of a track (i.e. not background noise). On top of the encoded point features, a query-based instance decoder inspired by Mask3D [Cheng et al., 2022, Schult et al., 2022], iteratively refines a fixed set of track hypotheses. Each learnable query attends to the selected hits, alternating self-attention, cross-attention, and feed-forward layers, outputting a per-query confidence and a dense mask indicating which hits belong to that hypothesized track. Points that were previously rejected by the classifier skip the decoding and instead have per-query mask logits computed in one shot via the learned queries.

At inference, the model converts the predicted masks into final track assignments by combining per-query confidence with per-hit mask activations and assigning each hit to its most plausible track (or to background). Small-fragment suppression removes spurious instances, yielding clean, per-event track IDs for each hit. Training is fully end-to-end with a composite loss:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{nce}}\,\mathcal{L}_{\text{NCE}} + \lambda_{\text{clf}}\,\mathcal{L}_{\text{CLF}} + \lambda_{\text{ce}}\,\mathcal{L}_{\text{CE}} + \lambda_{\text{mask}}\,\mathcal{L}_{\text{BCE}} + \lambda_{\text{dice}}\,\mathcal{L}_{\text{Dice}}$$

Here $\mathcal{L}_{NCE}$ is the InfoNCE [van den Oord et al., 2018] loss used in standard HEPT, which enforces embedding consistency across hits for the same particle and separation from hits of different particles. $\mathcal{L}_{CLF}$ is for the per-hit binary classifier to classify points likely to be part of a track. $\mathcal{L}_{CE}$, $\mathcal{L}_{BCE}$, and $\mathcal{L}_{Dice}$ are those losses used in object segmentation in computer vision, where $\mathcal{L}_{CE}$ supervises query classification (object vs background), $\mathcal{L}_{BCE}$ provides point-wise mask accuracy, and $\mathcal{L}_{Dice}$ enforces overlap between true and predicted track-masks. To improve robustness to noise, we introduce a learning curriculum that progressively admits harder, lower-momentum hits over epochs. This allows the model to first master clean, reconstructible tracks, and then adapt to realistic backgrounds.

4 Experiments

4.1 Dataset

The TrackML dataset [Amrouche et al., 2023, 2019] contains approximately 10,000 simulated proton-proton collision events in the LHC, generated with Pythia 8. Each event includes a signal originating from hard QCD interactions in a top–antitop quark decay, as well as a background from 200 soft QCD interactions to realistically model pileup expected at the HL-LHC. A transverse momentum cut of $p_T > 150$ MeV is applied for event selection. The simulated detector consists of three types of subdetector layers: pixel, short-Strip, and long-Strip. In this paper, we focus on measurements from the Pixel detector. In our study, we split 900/500/500 for train/val/test.

4.2 Model Settings

Exa.TrkX. The GNN model used has approximately 854K parameters. Both the edge encoder network and node network have 8 layers with a hidden dimensionality of 256. The edge cut threshold is tuned to 0.1.

HEPT/HEPTv2. The HEPT encoder uses 3 OR hash tables, 50 regions with 2 AND hash tables, and 8 attention heads. HEPTv2 has additional 4 transformer decoder layers and 3000 instance queries that cross-attend to each hit, resulting a total of 1.1M parameters. The number of queries is set larger than the maximum track multiplicity observed across all events, ensuring full coverage.

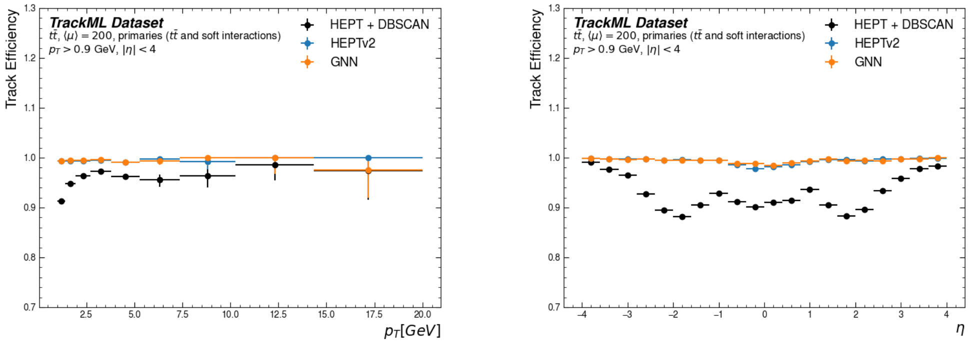

Figure 1: Tracking efficiency as a function of (a) transverse momentum, p_{T}, and pseudorapidity, η.

4.3 Pixel Tracking Performance

The tracking performance of the pipeline is evaluated by the efficiency and the fake rate. In this study, charged particles are required to be within the region covered by the Pixel detectors, $|\eta| < 4.0$ and transverse momentum $p_{\mathrm{T}} > 900$ MeV. The matching between particles and track candidates is defined as follows: A particle is considered matched to a track candidate if at least half of the particle hits are within the candidate. A track candidate is matched to a particle if purity is higher than 50%. For the efficiency evaluation, only particles with a minimum of three pixel hits are considered reconstructable. The tracking efficiency is defined as the fraction of matched reconstructable particles. Figure 1 displays the tracking efficiency as a function of particles p_{T} and η distributions. HEPTv2 maintains competitive tracking performance compared with the Exa.TrkX pipeline across all p_{T} ranges. The η dependence is mainly due to fewer measurements in the central barrel region, which is especially impactful for the HEPT+DBScan pipeline.

Table 1 presents a summary of tracking performance and fake rate. Exa.TrkX achieves a track matching efficiency of 0.994 with a fake rate of 0.002. Compared to the HEPT + DBScan implementation, which exhibits reduced efficiency and a higher fake rate, the end-to-end HEPTv2 pipeline demonstrates substantial improvements in both tracking efficiency and latency (also see Sec. 4.4). Although its fake rate is higher than that of Exa.TrkX, this trade-off is offset by its enhanced efficiency and reduced latency, positioning HEPTv2 as a robust and competitive approach.

Implementation	Pixel Tracking Efficiency	Fake Rate
Exa.TrkX	**0.994**	**0.002**
HEPTv2	0.993	0.113
HEPT + DBScan	0.923	0.070

Table 1: Tracking eff. and fake rate on simulated TrackML ttbar events with an avg. pileup of 200.

4.4 Inference Latency

Previous literature has shown that HEPT encoder can be hundreds of times faster than graph neural networks (GNNs) on TrackML when run on GPUs [Miao et al., 2024]. Therefore, here we focus on comparing three HEPT configurations: the standard HEPT encoder (which outputs only hit embeddings without track assignment), HEPT+DBScan (implemented following [Lieret et al., 2023a] on CPUs), and HEPTv2. Table 2 shows that while the standard HEPT encoder is very fast, adding a clustering algorithm such as DBScan eliminates much of the advantage. By contrast, HEPTv2 adds only a small latency overhead while preserving HEPT's strong performance, highlighting its ability to deliver ultra-fast end-to-end track assignments without sacrificing tracking performance.

Implementation	Inference Time (ms)
HEPT Encoder (No track assignment)	23.69 ± 0.10
HEPTv2	27.73 ± 0.16
HEPT + DBScan	1425.23 ± 2.46

Table 2: Inference time on simulated TrackML ttbar events with an avg. pileup of 200.

5 Conclusion and Future Work

This work presents a comprehensive evaluation of HEPT and HEPTv2 for particle tracking against a strong GNN baseline, using only Pixel detector measurements as a demonstration on the TrackML dataset. Our results demonstrate that HEPTv2 achieves Pixel tracking efficiency comparable to the state-of-the-art Exa.TrkX pipeline, while offering significantly reduced latency.

The HEPTv2 framework provides a promising approach for accelerating charged particle tracking with lower latency and computational overhead, making it well-suited for online trigger environments. Future work will focus on extending this architecture to incorporate full detector systems, including the Strip detector, and evaluating its impact on overall tracking performance across diverse metrics and experimental conditions.

Acknowledgments

Y. Chou, J. Duarte, P. Li, M. Liu, S. Miao, J. P. Rodgers, and S. Hsu are supported by National Science Foundation (NSF) grant No. PHY-2117997. S. Miao and P. Li are also supported by NSF grant No. IIS-2239565. This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02-05CH11231. We also acknowledge support from the NVIDIA Academic Grant Program and the IDEaS Cyberinfrastructure Awards.

References

G. Aad et al. The ATLAS experiment at the CERN Large Hadron Collider: a description of the detector configuration for Run 3. *JINST*, 19(05):P05063, 2024. doi: 10.1088/1748-0221/19/05/P05063.

G. Aad et al. Expected tracking performance of the ATLAS Inner Tracker at the High-Luminosity LHC. *JINST*, 20(02):P02018, 2025. doi: 10.1088/1748-0221/20/02/P02018.

S. Abadal, B. Martinez, H. Zhou, and E. Alarcón. Graph neural network acceleration with in-memory computing. *Nature Electronics*, 4:681–690, 2021.

S. Amrouche et al. *The Tracking Machine Learning challenge : Accuracy phase*. 4 2019. doi: 10.1007/978-3-030-29135-8_9.

S. Amrouche et al. The Tracking Machine Learning Challenge: Throughput Phase. *Comput. Softw. Big Sci.*, 7(1):1, 2023. doi: 10.1007/s41781-023-00094-w.

M. J. Atkinson, J. Burzynski, P. Butti, J. Burleson, S. Caillou, P. Calafiura, J. Chan, C. Collard, S. Dittmeier, S. A. Farrell, B. Huth, X. Ju, A. Lazar, R. Liu, T. Minh Pham, D. c. a. Murnane, M. Neubauer, A. Poreba, C. Rougier, J. Stark, H. Torres, and A. Vallier. Acorn - a charged object reconstruction network. `https://gitlab.cern.ch/gnn4itkteam/acorn/`. URL `https://github.com/GNN4ITkTeam/CommonFramework`.

S. Caillou, P. Calafiura, S. A. Farrell, X. Ju, D. T. Murnane, C. Rougier, J. Stark, and A. Vallier. ATLAS ITk Track Reconstruction with a GNN-based pipeline. Technical report, CERN, Geneva, 2022. URL `https://cds.cern.ch/record/2815578`.

S. Chatrchyan et al. The CMS Experiment at the CERN LHC. *JINST*, 3:S08004, 2008. doi: 10.1088/1748-0221/3/08/S08004.

B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022.

CMS Collaboration. Computing model and performance for the hl-lhc, 2022. Internal/whitepaper reference used by HEPT; match the exact citation you prefer.

M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 253–262, 2004.

G. DeZoort, S. Thais, J. Duarte, V. Razavimaleki, M. Atkinson, I. Ojalvo, M. Neubauer, and P. Elmer. Charged Particle Tracking via Edge-Classifying Interaction Networks. *Comput. Softw. Big Sci.*, 5 (1):26, 2021. doi: 10.1007/s41781-021-00073-z.

L. Evans and P. Bryant. Lhc machine. *Journal of Instrumentation*, 3(08):S08001, aug 2008. doi: 10.1088/1748-0221/3/08/S08001. URL `https://dx.doi.org/10.1088/1748-0221/3/08/S08001`.

M. Hashemi, S. Swanson, and S. W. Keckler. Understanding the impact of irregular memory access patterns on gpu performance. In *International Symposium on Workload Characterization (IISWC)*, pages 224–233, 2018.

B.-K. Jang, S. Venugopalakrishnan, S.-W. Han, and V. K. Prasanna. An empirical study of irregular code on gpus. In *International Symposium on Workload Characterization (IISWC)*, pages 116–125, 2010.

X. Ju et al. Performance of a geometric deep learning pipeline for HL-LHC particle tracking. *Eur. Phys. J. C*, 81(10):876, 2021. doi: 10.1140/epjc/s10052-021-09675-8.

A. Lazar et al. Accelerating the Inference of the Exa.TrkX Pipeline. *J. Phys. Conf. Ser.*, 2438(1): 012008, 2023. doi: 10.1088/1742-6596/2438/1/012008.

K. Lieret, G. DeZoort, D. Chatterjee, J. Park, S. Miao, and P. Li. High Pileup Particle Tracking with Object Condensation. 12 2023a.

M. Lieret, J. Duarte, V. Loncar, et al. Graph neural networks for charged particle tracking on fpgas. *Frontiers in Big Data*, 6, 2023b.

Y.-X. Liu et al. Flatformer: Flattened window attention for efficient point cloud transformers. In *NeurIPS*, 2023.

S. Miao, Z. Lu, M. Liu, J. Duarte, and P. Li. Locality-sensitive hashing-based efficient point transformer with applications in high-energy physics. In *ICML*, 2024.

J. Schult, F. Engelmann, A. Hermans, O. Litany, S. Tang, and B. Leibe. Mask3d: Mask transformer for 3d semantic instance segmentation. *arXiv preprint arXiv:2210.03105*, 2022.

G. Stark, T. Aarrestad, J. Duarte, et al. Scalable graph neural networks for particle track reconstruction. *Comput. Softw. Big Sci.*, 6(1), 2022.

A. van den Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. doi: 10.48550/arXiv.1807.03748.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Z. Wang et al. Dsvt: Dynamic sparse voxel transformer with rotated sets. In *CVPR*, 2023.

X. Wu, L. Jiang, P.-S. Wang, Z. Liu, X. Liu, Y. Qiao, W. Ouyang, T. He, and H. Zhao. Point transformer v3: Simpler faster stronger. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4840–4851, 2024.

H. Zhao, L. Jiang, et al. Point transformer. In *ICCV*, 2021.