TrackFormers Part 2: Enhanced Transformer-Based Models for High-Energy Physics Track Reconstruction

Sascha Caron[1,2], Nadezhda Dobreva[1,2], Maarten Kimpel[3], Uraz Odyurt[4], Slav Pshenov[2,5], Roberto Ruiz de Austri Bazan[6], Eugene Shalugin[1,2], Zef Wolffs[2,5] and Yue Zhao[7⋆]

1 High-Energy Physics, Radboud University, Nijmegen, The Netherlands
2 National Institute for Subatomic Physics (Nikhef), Amsterdam, The Netherlands
3 Institute for Computing and Information Sciences, Radboud University, Nijmegen, The Netherlands
4 Faculty of Engineering Technology, University of Twente, Enschede, The Netherlands
5 Institute of Physics, University of Amsterdam, Amsterdam, The Netherlands
6 Instituto de Física Corpuscular, IFIC-UV/CSIC, Valencia, Spain
7 High Performance Machine Learning, SURF, Amsterdam, The Netherlands

⋆ yue.zhao@surf.nl

EuCAIF

The 2nd European AI for Fundamental Physics Conference (EuCAIFCon2025) Cagliari, Sardinia, 16-20 June 2025

Abstract

High-Energy Physics experiments are rapidly escalating in generated data volume, a trend that will intensify with the upcoming High-Luminosity LHC upgrade. This surge in data necessitates critical revisions across the data processing pipeline, with particle track reconstruction being a prime candidate for improvement. In our previous work, we introduced "TrackFormers", a collection of Transformer-based one-shot encoder-only models that effectively associate hits with expected tracks. In this study, we extend our earlier efforts by incorporating loss functions that account for inter-hit correlations, conducting detailed investigations into (various) Transformer attention mechanisms, and a study on the reconstruction of higher-level objects. Furthermore we discuss new datasets that allow the training on hit level for a range of physics processes. These developments collectively aim to boost both the accuracy, and potentially the efficiency of our tracking models, offering a robust solution to meet the demands of next-generation high-energy physics experiments.

Received Date
Accepted Date
Published Date

1 Introduction

The High-Luminosity LHC (HL-LHC) will generate unprecedented volumes of collision data, creating significant challenges for particle track reconstruction, where tens of thousands of detector hits must be accurately associated with their originating particles. Traditional reconstruction methods, while precise, struggle to scale efficiently to these data rates. Transformer-based machine learning models offer a promising alternative: in prior work, we introduced "TrackFormers", encoder-only, one-shot transformers that map hits directly to particle tracks. In this study, we extend this approach by exploring a new design combining geometric projection and lightweight clustering, a joint model conditioning classification on a regressor's predictions, and FlexAttention. To support model training and evaluation, we provide a fully reproducible ACTS-based hit-level dataset spanning signal and background processes across multiple pileup levels.

2 Improved methods

2.1 New datasets

We created a new hit-level dataset with a fully reproducible ACTS-based pipeline combining Monte-Carlo event simulation, detector response, and TrackML-style postprocessing.

The signal process is $pp \to t\bar{t}H, H \to b\bar{b}$ and the main background is inclusive $pp \to t\bar{t}$. Both are generated with `Pythia8`, producing stable truth-level particles as a starting point. Events are subsequently transported through a TrackML detector using the ACTS fast simulation (`Fatras`) and digitized into realistic channel-level measurements. This provides low-level hit data for machine learning models. From these digitized hits we further derive TrackML-style per-event triplets (`hits.csv`, `particles.csv`, `truth.csv`) with global coordinates and physics-motivated per hit weights.

We generate datasets at pileup levels 0, 5, 20, 50, and 200, each with 40k events (20k signal, 20k background).

2.2 Improved model design

2.2.1 Masking and projection

The quadratic scaling of attention with hit count renders naive transformers impractical for full pixel-detector HL-LHC events (sequence lengths $> 6 \times 10^4$). We address this with a hybrid design that combines geometric projection, clustering, and FlexAttention to exploit tracking locality, reducing attention cost by up to 400×.

As shown in Figure 1, hits are projected onto simplified detector surfaces to minimize track spread: a cylinder ($R = 91$ mm) for the barrel (using R–ϕ, z coordinates) and two planes ($z = \pm 920$ mm) for the endcaps (using x, y). For barrel regions, this makes tracks compact in (R–ϕ, z). For endcaps, cluster alignment is refined by reprojecting clusters over candidate z-vertex positions and selecting the z that maximizes alignment.

Light weight clustering (an iterative windowing algorithm developed by the authors) or DBSCAN is then applied on these projected surfaces to form local neighborhoods. Clusters on the cylindrical surface define sparse block masks for FlexAttention, ensuring that only physically plausible hit pairs attend and reducing the effective attention matrix by up to $\sim 400\times$. Endcap clusters use the vertex-z scan described above to sharpen alignment in the longitudinal direction. Clustering hyperparameters are tuned to maximize the reconstructible ratio, defined as tracks with $p_T > 0.9$ GeV, having ≥ 3 hits, and $\geq 50\%$ of those hits in a single cluster. Block masks are precomputed and cached for efficient reuse during training.

The encoder is a PyTorch Transformer with FlexAttention (12 layers, 4 heads, hidden dimension 192, feed-forward dimension 384). Input (x, y, z) hits are projected, clustered, and normalized. Training used AdamW with `bfloat16` mixed precision on NVIDIA H100s, gradi-

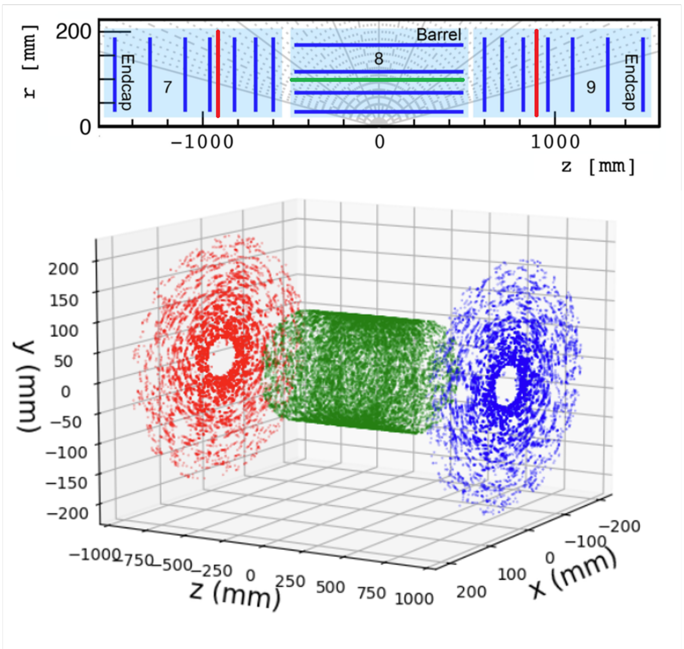

Figure 1: Top: the projection surfaces used for hit mapping, with the cylindrical barrel in green, two planar endcaps in red, and pixel detector layers in blue [1]. Bottom: pixel detector hits of one event projected onto these three surfaces.

ent clipping, and an adaptive learning rate schedule. The model was trained on 8 658 TrackML events with 96 validation and 96 test events.

Rather than regressing track parameters, the model maps each hit to a 32-dimensional embedding and is trained with a multi-positive InfoNCE contrastive loss: for each hit, all hits from the same track ($p_T > 0.9\,\text{GeV}$) are positives, while all others are negatives. At inference, the model produces an $N \times N$ cosine-similarity matrix, from which tracks are assembled by selecting high-similarity neighbors, eliminating the need for a separate clustering stage.

2.2.2 Joining regression and classification

In this experiment, we combine our strongest prior architectures into a unified two-stage model. Stage 1 is an EncReg-style encoder-only Transformer [2] that regresses track parameters ($\theta, \sin\phi, \cos\phi, q \in \{-1, 1\}$) together with four learned free latent variables. Here θ and ϕ are spherical momentum angles ($p = \sqrt{p_x^2 + p_y^2 + p_z^2}$, $\theta = \arccos(p_z/p)$, $\phi = \arctan 2(p_y, p_x)$).

Stage 2 is an EncCla-style encoder-only Transformer for per-hit classification. For each hit we concatenate the raw coordinates with the regressor outputs,

$$(x, y, z, \theta, \sin\phi, \cos\phi, q, \text{latent}_1, \ldots, \text{latent}_4),$$

project to an embedding, and pass through encoder blocks. A final linear head produces a categorical distribution over quantile-binned (ϕ, θ, p, q) classes; the predicted class is the maximum-probability bin. Although regressed parameters are already predictive, they further enrich the classifier's input features.

The model is trained end-to-end with a joint loss

$$\mathcal{L} = \alpha\,\mathcal{L}_{\text{reg}} + \beta\,\mathcal{L}_{\text{cla}}, \qquad \alpha = 1,\ \beta = 0.3,$$

where $\mathcal{L}_{\text{reg}}$ is per-hit MSE on $(\theta, \sin\phi, \cos\phi, q)$ and $\mathcal{L}_{\text{cla}}$ is cross-entropy over class labels.

The joint model, denoted JM $X{:}Y$ with X regressor layers and Y classifier layers, retains the one-pass property of both components: a single forward pass produces track parameters and per-hit classes, enabling downstream use without extra clustering stages.

2.2.3 FlexAttention

In our previous work [2] we experimented with FlashAttention [3]; here we instead adopt FlexAttention [4]. This change is driven by a key limitation of FlashAttention: its lack of flexible, batch-wise pad masking for variable sequence lengths. As a result, training was restricted to a batch size of one to avoid padding. FlexAttention overcomes this constraint through its BlockMask mechanism, which enables processing of heterogeneous events within the same batch while maintaining near state-of-the-art kernel performance [4]. With FlexAttention, we preserve the GPU inference speedups previously observed, now without the batch size restriction. Equally importantly, its memory efficiency allowed us to co-train both the regressor and classifier on a single NVIDIA A100 GPU (40 GiB HBM2), whereas with FlashAttention only one of these models could fit on the same hardware during training.

3 Results

3.1 Masking and projection

A breakdown of inference latency is as follows:

- Clustering: projected best-case 6 ms with parallel DBSCAN,
- Block-mask creation: 2 ms per event,
- Transformer encoder: 20 ms per event,
- Track–hit assignment: 47 ms per event.

The resulting end-to-end runtime is on the order of tens of milliseconds per event, significantly faster than existing GNN pipelines (0.5–1 s) [5] and comparable to the state of the art ($\sim$100 ms) [6].

In terms of physics performance, our model achieves $\sim$90% track double-majority efficiency in the barrel and 91% in the endcaps after vertex-z refinement. Efficiencies in the barrel are uniform across p_T, with expected drops at low $|\eta|$ and near $|\eta| \approx 2.0$ due to detector geometry.

Relative to the EncReg and EncCla models from our previous iteration, which achieved $\sim$70% efficiency on reduced TrackML datasets ($\sim$5% HL-LHC density), the present design scales to tens of thousands of hits per event while maintaining sub-200 ms inference latency and improved efficiency. These results establish projection-based clustering with FlexAttention and contrastive similarity learning as a practical solution for HL-LHC scale tracking.

3.2 Joining regression and classification

The accuracy and TrackML score [1] for JM and EncCla models with FlexAttention are shown in Table 1. Deeper models (more encoder layers) consistently improve both metrics. Adding EncReg and passing its regressed parameters to EncCla yields an additional $\sim$2.4% accuracy and $\sim$2% TrackML score gain. Unlike EncCla, the regressor showed little benefit from greater depth, so EncReg was kept shallow.

Model	EncCla 6	JM 6:6	EncCla 7	JM 7:7	EncCla 9	JM 7:9	EncCla 15	JM 9:15
Accuracy	69.7%	72.8%	74.2%	76.6%	76.2%	78.4%	78.5%	**80.5%**
TrackML score	79.9%	82.3%	84.2%	86.4%	87.3%	89.0%	89.8%	**91.4%**

Table 1: The accuracy and TrackML scores of different models across Joined Model (JM) and EncCla configurations. The best scores are print in bold and the second best are underlined. The numbers after the model name signal the layer depth. For JM configurations the first number denotes the layer depth of the regressor and the second number denotes the layer depth of the classifier.

Inference times are modest: CPU latency is stable at 0.1 ms across all models, while GPU latency scales linearly with depth, adding ∼2.4 ms per encoder layer on an NVIDIA A100 (40 GiB).

Model	EncCla 6	JM 6:6	EncCla 7	JM 7:7	EncCla 9	JM 7:9	EncCla 15	JM 9:15
CPU inf. time (ms)	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
GPU inf. time (ms)	16.1	31.6	18.5	38.2	24.0	42.8	39.0	61.6

Table 2: The CPU inference time and GPU inference time per event in milliseconds of different models across Joined Model (JM) and EncCla configurations. The numbers after the model name signal the layer depth. For JM configurations the first number denotes the layer depth of the regressor and the second number denotes the layer depth of the classifier.

Overall, scaling encoder-only Transformer trackers and coupling regression with classification in a single forward pass substantially improves performance over previous work. EncCla models show monotonic gains in TrackML score with depth, reaching 89% (vs. 78% previously). Injecting physics-based features from the regressor into the classifier provides a further ∼2% absolute uplift. These improvements are enabled by FlexAttention, which allows deeper architectures to train on the same hardware, albeit with roughly doubled GPU inference time.

4 Conclusion and future work

We release a fully reproducible ACTS-based hit-level dataset of $pp \to t\bar{t}H, H \to b\bar{b}$ signal and $pp \to t\bar{t}$ background events, providing TrackML-style formats across multiple pileup conditions (0–200) to enable realistic large-scale tracking benchmarks for machine learning models. In our experiments with new model architectures, we have shown that projection-based clustering combined with FlexAttention block masking provides an efficient way to scale transformer-based trackers to HL-LHC hit densities, cutting attention cost by up to 400× while retaining end-to-end inference times in the $\mathcal{O}(10^2)$ ms range. In addition, deeper encoder-only architectures continue to deliver strong performance, while fusing regressed parameters into the classifier provides modest but consistent improvements. All of these gains are achieved within a single end-to-end inference call, preserving the simplicity that makes encoder-only designs appealing for HL-LHC deployment.

Acknowledgements

This work used the Dutch national e-infrastructure with the support of the SURF Cooperative using grant no. EINF-11730. The work of R. RdA was supported by PID2020-113644GB-I00 from the Spanish Ministerio de Ciencia e Innovación and by the PROMETEO/2022/69 from the Spanish GVA. The author(s) gratefully acknowledges the computer resources at Artemisa, funded by the European Union ERDF and Comunitat Valenciana as well as the technical support provided by the Instituto de Fisica Corpuscular, IFIC (CSIC-UV).

References

[1] Kiehn, Moritz, Amrouche, Sabrina, Calafiura, Paolo, Estrade, Victor, Farrell, Steven, Germain, Cécile, Gligorov, Vava, Golling, Tobias, Gray, Heather, Guyon, Isabelle, Hushchyn, Mikhail, Innocente, Vincenzo *et al.*, *The trackml high-energy physics tracking challenge on kaggle*, EPJ Web Conf. **214**, 06037 (2019), doi:10.1051/epjconf/201921406037.

[2] 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*, The European Physical Journal C **85**(4) (2025), doi:10.1140/epjc/s10052-025-14156-3.

[3] T. Dao, *FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning*, doi:10.48550/arXiv.2307.08691, ArXiv:2307.08691 [cs] (2023).

[4] J. Dong, B. Feng, D. Guessous, Y. Liang and H. He, *Flex Attention: A Programming Model for Generating Optimized Attention Kernels*, doi:10.48550/arXiv.2412.05496, ArXiv:2412.05496 [cs] (2024).

[5] ATLAS Collaboration, *Computational performance of the atlas itk gnn track reconstruction pipeline*, Tech. rep., CERN, Geneva, [Online]. Available: https://cds.cern.ch/record/2914282 (2024).

[6] S. V. Stroud *et al.*, *Transformers for charged particle track reconstruction in high energy physics*, arXiv preprint (2024), [Online]. Available: http://arxiv.org/abs/2411.07149.