Efficient ML-Assisted Particle Track Reconstruction Designs

Nadezhda Dobreva

nadezhda.dobreva@ru.nl

CHEP 24/10/2024

Radboud University

https://arxiv.org/abs/2407.07179

Team

- **The project collaborators:**
  - Radboud University (Nadezhda Dobreva)
  - Nikhef (Sascha Caron, Zef Wolffs, Uraz Odyurt)
  - SURF (Yue Zhao)
  - University of Valencia (Antonio Ferrer Sanchez, Roberto Ruiz de Austri Bazan, Jose D. Martin-Guerrero)

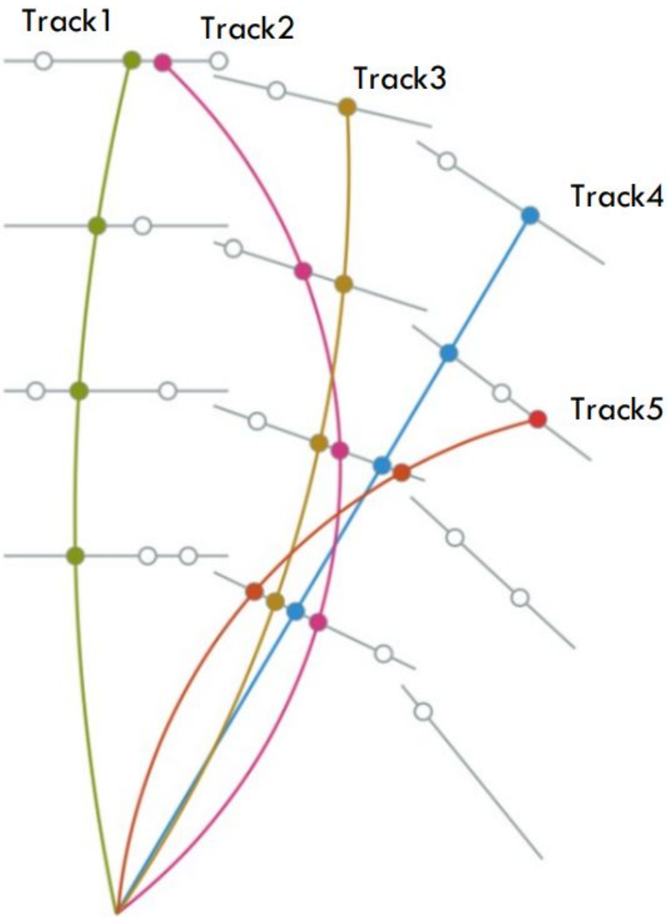
Transformers

U-nets

Problem Definition

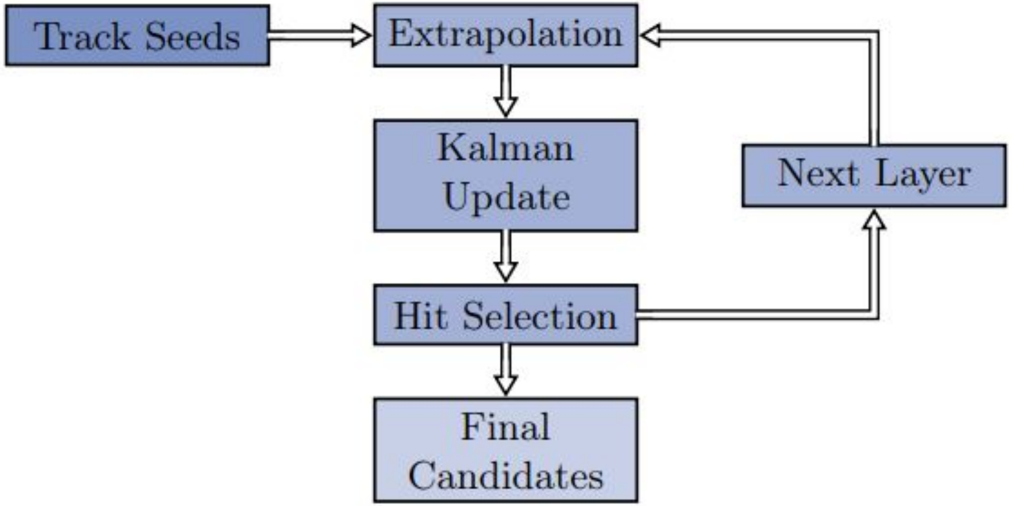
Track Reconstruction

- **Track finding**
  - Grouping hits that likely originate from the same particle
- **Track fitting**
  - The derivation of track parameter of a group of hits
- **Track parameters**
  - Describe the particle trajectory



Kalman Filters (KF)

- **Traditional algorithm for the task, used in LHC**
- **Track finding needs a combinatorial KF**



Scalability Issue

- **KF and CKF scale poorly, inherently sequential [1]**
- **High Luminosity LHC**
  - Number of generated particles and recorded hits to increase manyfold [3]
- **12s per event [2] ***
- **Fast KF: 1.8s per event [2] ***

*** Used CPU: Intel Xeon E5-2620v2**

Active Field of Research

- **Graph neural networks**
  - Goal is to identify connections between the hits that represent actual physical trajectories
  - 2.2s* per event [5]
- **U-nets**
  - A convolutional neural network for image segmentation
  - Investigated within our team: pixel segmentation

*** GPU used: Nvidia A100 Tensor Core**

The Transformer

What is a Transformer?

- Deep learning architecture
- Success in NLP (and many other fields)

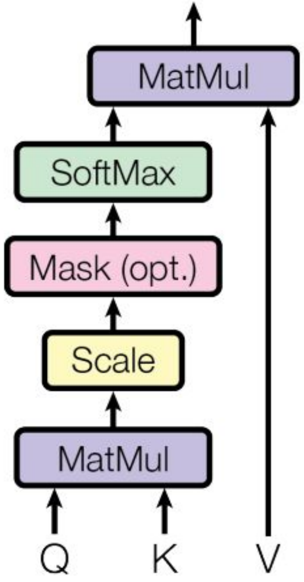
detector recordings → **transformer** → **physics information**

Why Use a Transformer?

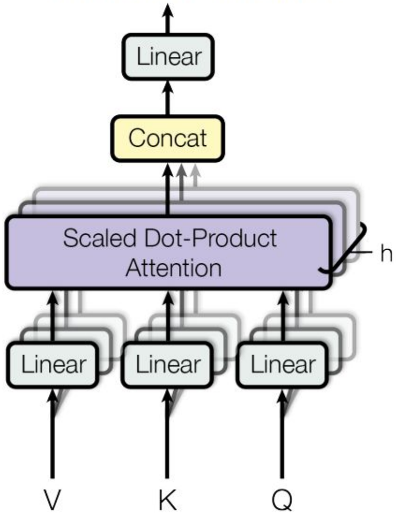
- **Can be parallelized**
- **Can handle variable length input**
- **Equivariant to input order**
- **Captures complex non-linear dynamics in data**

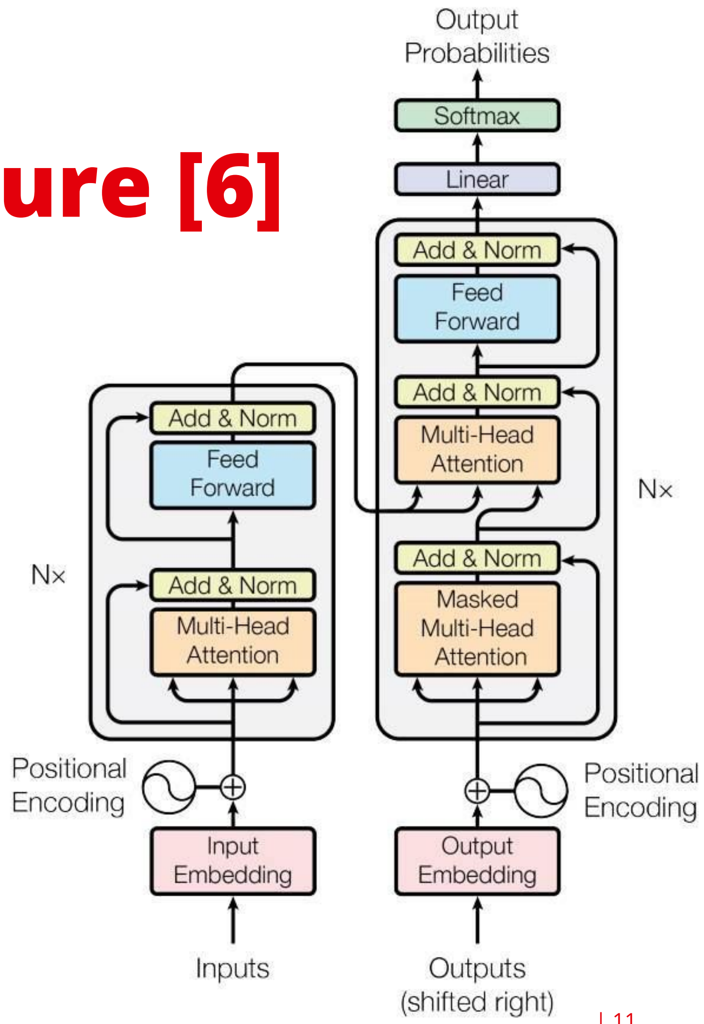
Transformer Architecture [6]

Scaled Dot-Product Attention



Multi-Head Attention



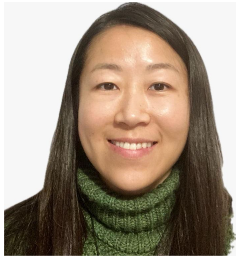


Proposed Approaches

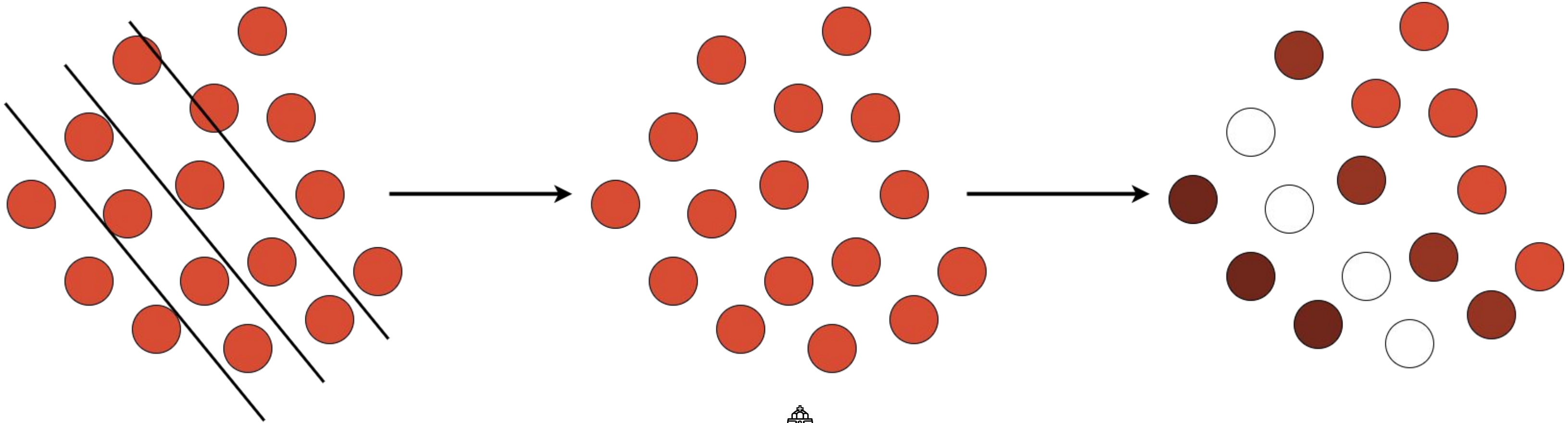
Four Pipelines [11]

- **U-Net:** Segments digital image representation of event into segments representing the different tracks
- **Encoder-Decoder Model (EncDec):** Autoregressively builds the full track, starting from a given seed
- **Encoder-only Classifier (EncCla):** Based on distribution of track parameters among classes, predict the class of each hit
- **Encoder-only Regressor (EncReg):** Regress track parameters of each hit and cluster together based on proximity

Encoder-only Classifier: EncCla

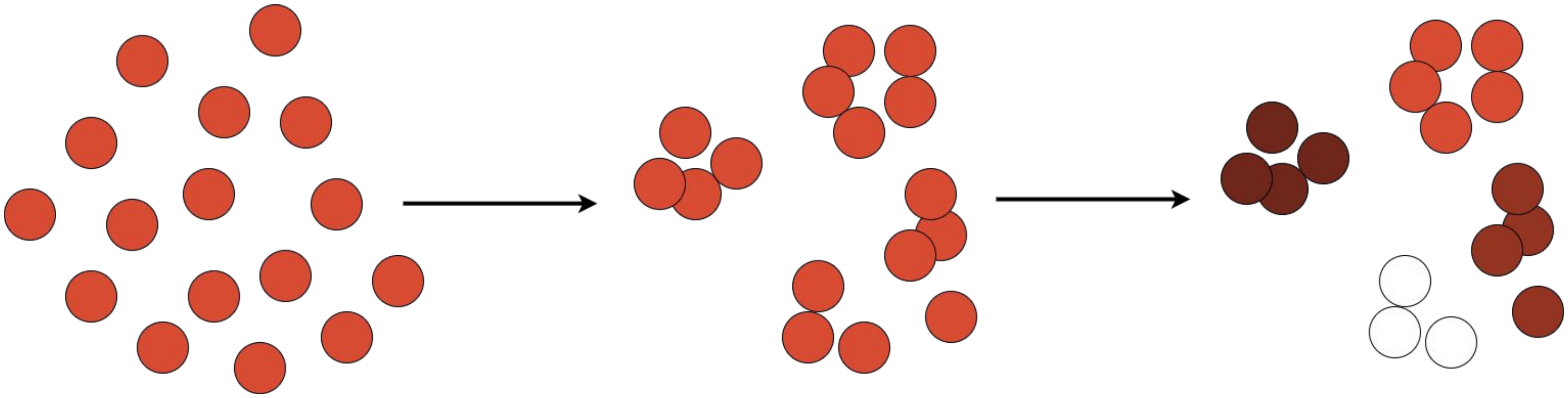


- Track defining parameters placed in balanced bins (i.e. classes)
- Transformer predicts the class of each hit



Encoder-only Regressor: EncReg

- **Used for regressing track-defining parameters**
- **Clustering hits based on regressed parameters**

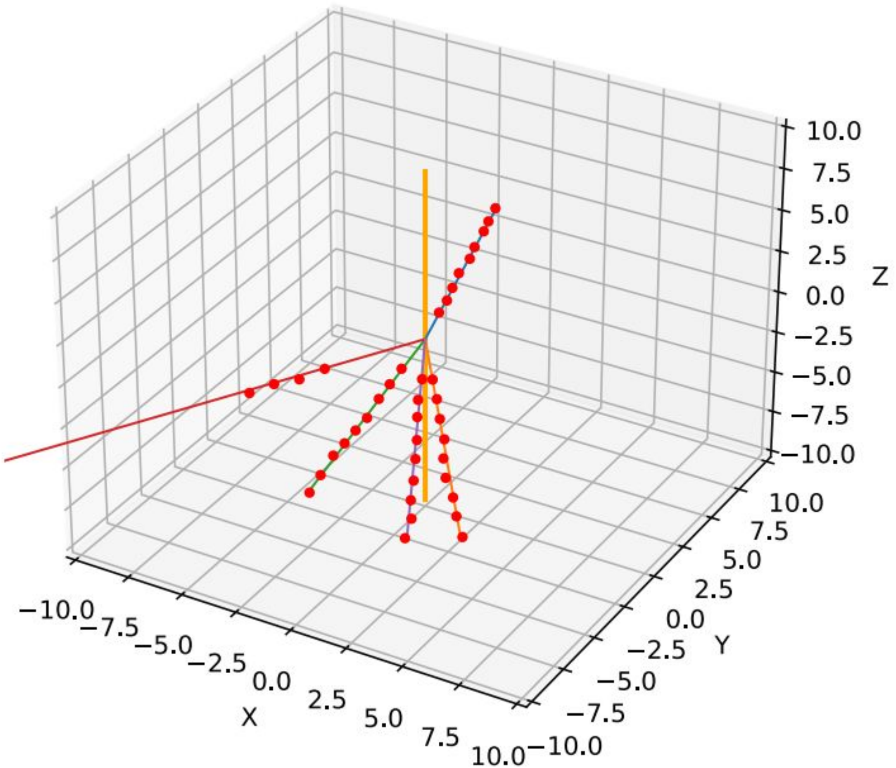


Simulations

Complexity-Reduced Approach

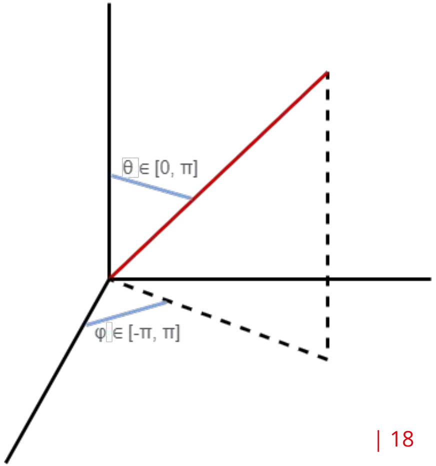
- **Iterative increase of complexity**
- **REDuced VIrtual Detector (REDVID) [9]**
  - https://virtualdetector.com/redvid/
  - https://indico.cern.ch/event/1338689/contributions/6015906/
- **TrackML-derived subsets [4]**

REDVID Linear Datasets

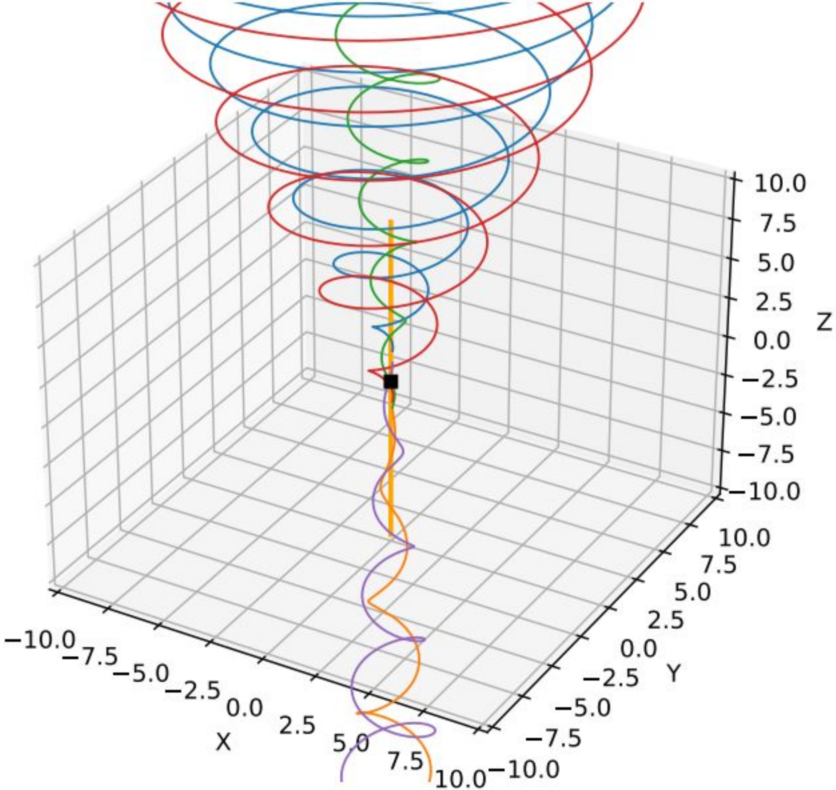


	3D-lin:10-50
Max nr. hits per event	450

EncCla: phi, theta, p
EncReg: sin(phi), cos(phi), theta

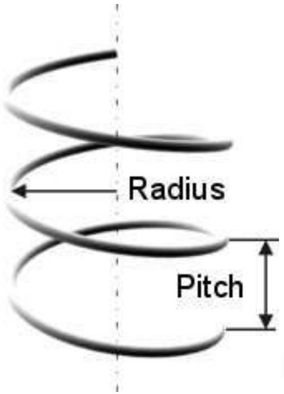


REDVID Helical Datasets

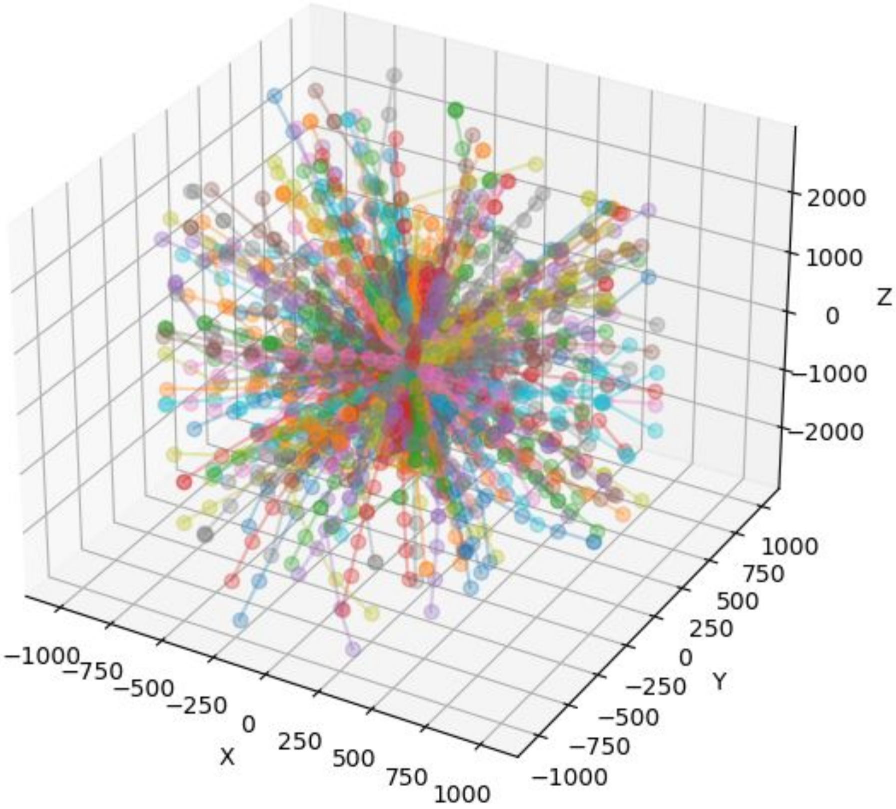


	3D-hel: 10-50	3D-hel: 50-100
Max nr. hits per event	450	900

EncCla, EncReg, U-net:
radial coefficient
pitch coefficient
azimuthal coefficient

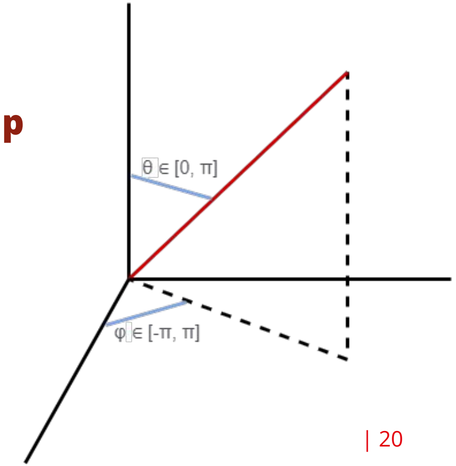


TrackML-derived Datasets [4]

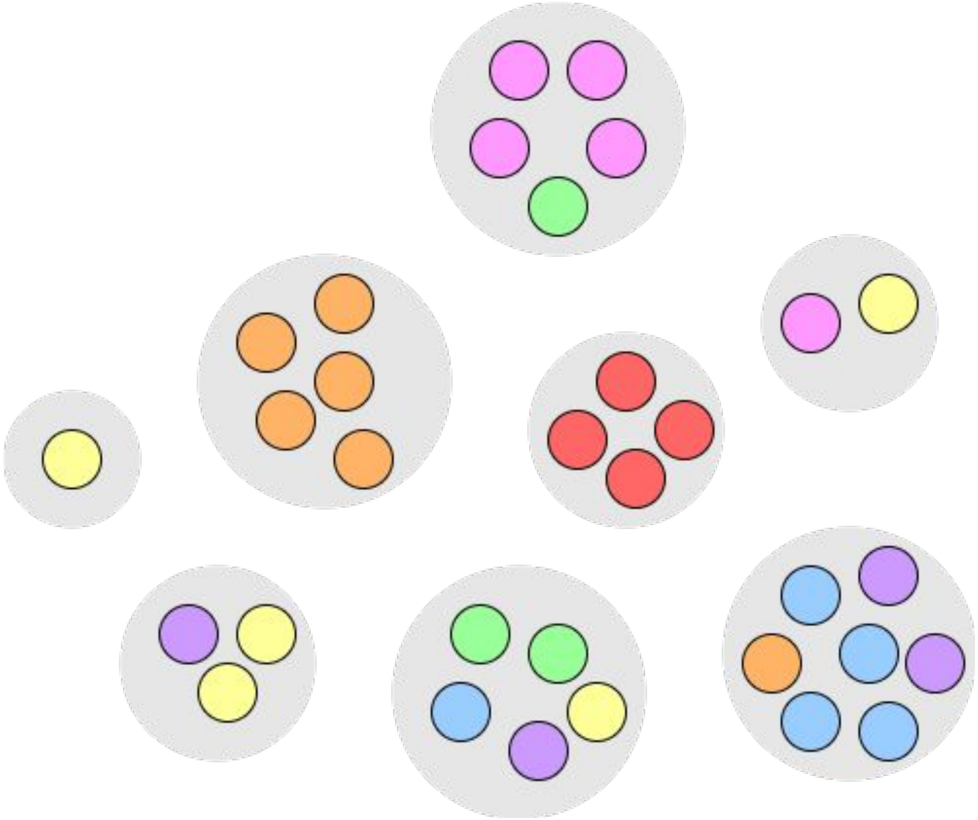


	TML:10-50	TML:200-500
Max nr. hits per event	700	5000

EncCla: phi, theta, q, p
EncReg: sin(phi), q, cos(phi), theta



Results

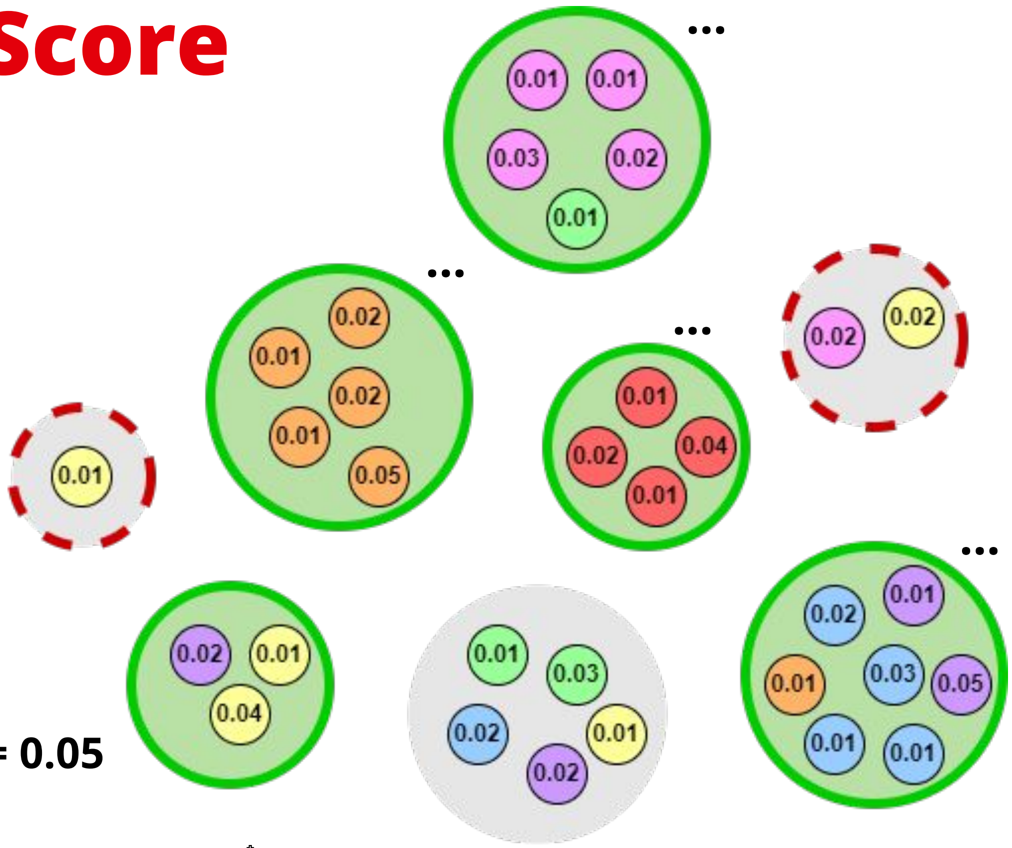


FitAccuracy Score

TrackML [4]

sum of weights of majority particle (>50% hits in cluster come from it)

0.04 + 0.01 = 0.05



FitAccuracy Scores [11]

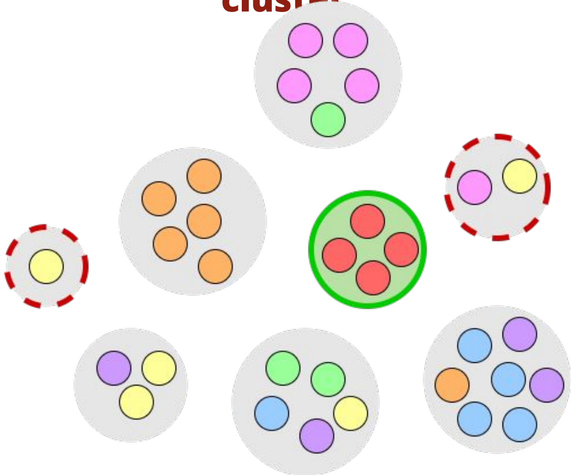
Data set	FitAccuracy score				
	EncDec	EncCla	EncReg	EncReg-FA	U-Net
REDVID - 10-50 linear tracks	93%	93%	97%	-	68%
REDVID - 10-50 helical tracks	85%	93%	92%	-	62%
REDVID - 50-100 helical tracks	85%	88%	85%	-	57%
TrackML - 10-50 tracks	26%	94%	93%	-	-
TrackML - 200-500 tracks	-	78%	70%	67%	-

* FA = Flash attention

Other Efficiency Scores [5]

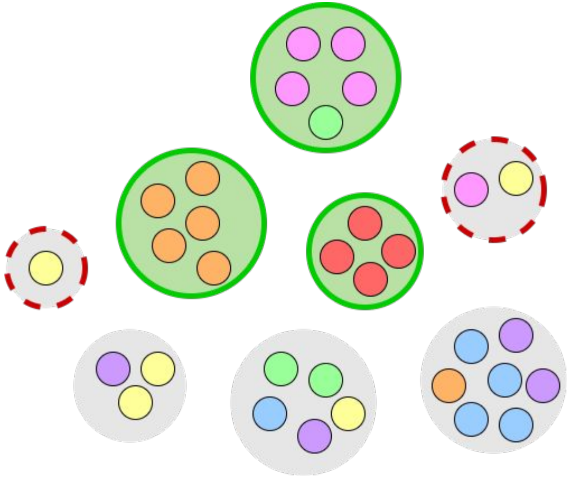
Perfect

only hits from 1 particle
no hits of it outside of cluster



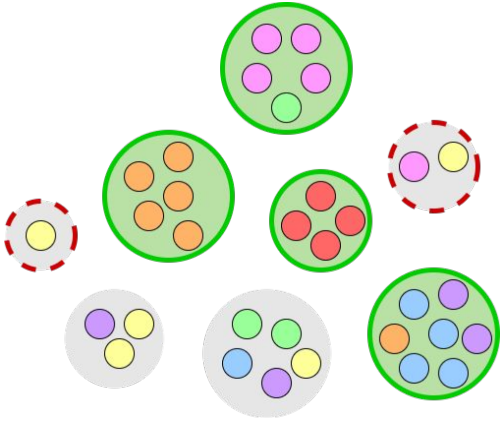
LHC

>=75% hits are from 1 particle



Double Majority

>=50% hits from 1 particle and <50% of its hits outside cluster



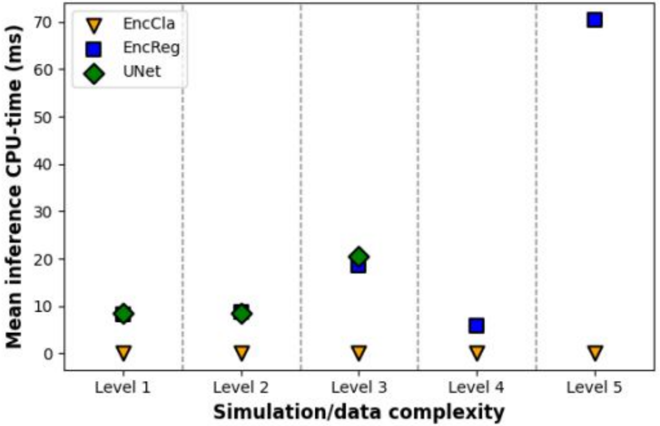
Physics Performance of EncReg

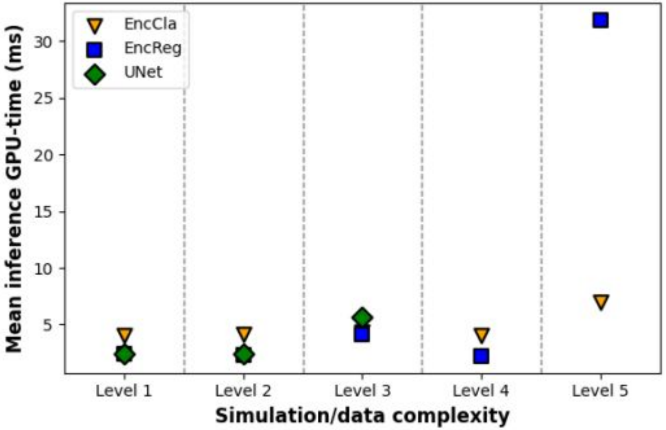
	3D-lin:10-50	3D-hel:10-50	3D-hel:50-100	TML:10-50	TML:200-500
TrackML score	0.97	0.92	0.85	0.932	0.7 (0.67)
ϵ^{perf}	0.94	0.78	0.6	0.78	0.4 (0.36)
ϵ^{DM}	0.97	0.94	0.89	0.91	0.75 (0.72)
ϵ^{LHC}	0.98	0.96	0.92	0.97	0.82 (0.79)

Table 6.2: Summary of the ϵ^{perf}, ϵ^{DM}, ϵ^{LHC} and TrackML scores obtained by the Transformer Regressor models for the 5 used datasets. For the models trained on TML:200-500, the result with Flash attention is in parentheses.

Run on Nvidia A100 GPU

Computational Performance





Data set	Model	Inference (mean) CPU side	Inference (mean) GPU side	Inference (mean) Wall-clock
REDVID - 10-50 linear tracks	EncDec	n/a	n/a	41 s
	EncCla	0.1 ms	4.0 ms	-
	EncReg	8.3 ms	2.4 ms	-
	U-Net	8.5 ms	2.4 ms	-
REDVID - 10-50 helical tracks	EncDec	n/a	n/a	19 s
	EncCla	0.1 ms	4.1 ms	-
	EncReg	8.7 ms	2.3 ms	-
	U-Net	8.6 ms	2.4 ms	-
REDVID - 50-100 helical tracks	EncDec	n/a	n/a	27 s
	EncCla	0.1 ms	4.3 ms	-
	EncReg	18.6 ms	4.1 ms	-
	U-Net	20.4 ms	5.6 ms	-
TrackML - 10-50 tracks	EncDec	n/a	n/a	16 s
	EncCla	0.1 ms	4.0 ms	-
	EncReg	5.8 ms	2.2 ms	-
	U-Net	n/a	n/a	-
TrackML - 200-500 tracks	EncDec	n/a	n/a	-
	EncCla	0.1 ms	7.0 ms	-
	EncReg	70.5 ms	31.9 ms	-
	EncReg-FA	72.2 ms	3.6 ms	-
	U-Net	n/a	n/a	-

Transformers for tracking: promising and worth further research!

https://arxiv.org/abs/2407.07179

Thank you.
Questions?

References

[1] Braun, N. “Combinatorial Kalman filter and high level trigger reconstruction for the Belle II experiment.” Springer, 2019.

[2] ATLAS Collaboration. “Fast track reconstruction for HL-LHC.” Tech. Rep. ATL-PHYS-PUB-2019-041, CERN, Geneva, 2019.

[3] Apollinari, Giorgio, Lucio Rossi, and Oliver Brüning. “High luminosity LHC project description.” No. CERN-ACC-2014-0321. 2014.

[4] Amrouche, Sabrina, et al. "The tracking machine learning challenge: throughput phase." Computing and Software for Big Science 7.1 (2023): 1.

[5] Ju, Xiangyang, et al. "Performance of a geometric deep learning pipeline for HL-LHC particle tracking." The European Physical Journal C 81 (2021): 1-14.

[6] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems, 30 (2017).

[7] Dao, Tri, et al. "Flashattention: Fast and memory-efficient exact attention with io-awareness." Advances in Neural Information Processing Systems 35 (2022): 16344-16359.

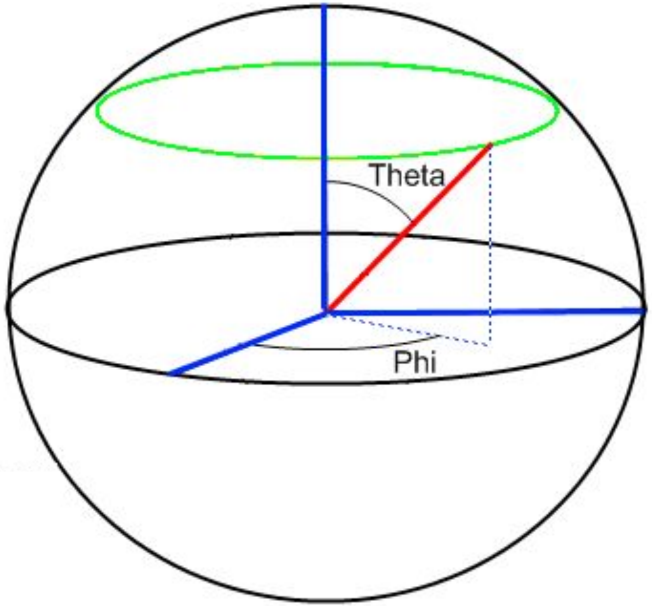
[8] Stewart, Geoffrey, and Mahmood Al-Khassaweneh. "An implementation of the HDBSCAN* clustering algorithm." Applied Sciences 12.5 (2022): 2405.

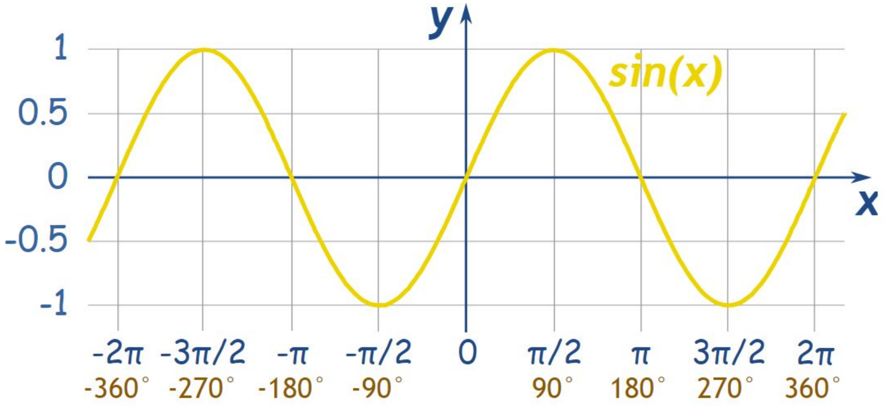
[9] Reduced Simulations for High-Energy Physics, a Middle Ground for Data-Driven Physics Research, Uraz Odyurt, Stephen Nicholas Swatman, Ana-Lucia Varbanescu, Sascha Caron, 2023

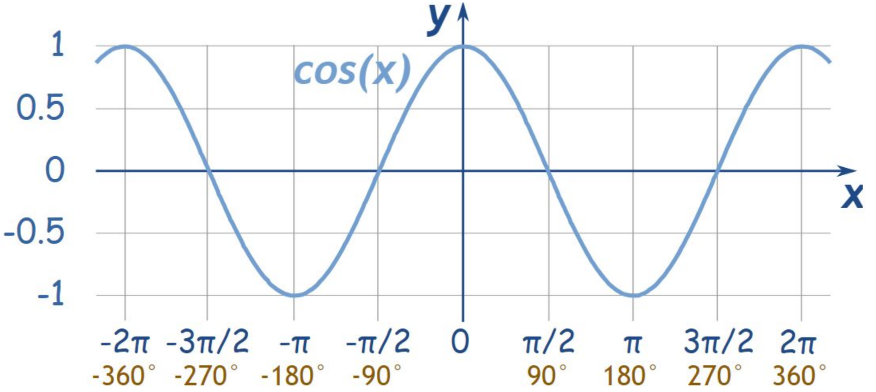
[10] Lieret, Kilian, et al. "High Pileup Particle Tracking with Object Condensation." arXiv preprint arXiv:2312.03823 (2023).

[11] Caron, Sascha, et al. “TrackFormers: In Search of Transformer-Based Particle Tracking for the High-Luminosity LHC Era.” (2024) https://arxiv.org/abs/2407.07179

Rotational Invariance of Phi







Rotational Invariance of Phi

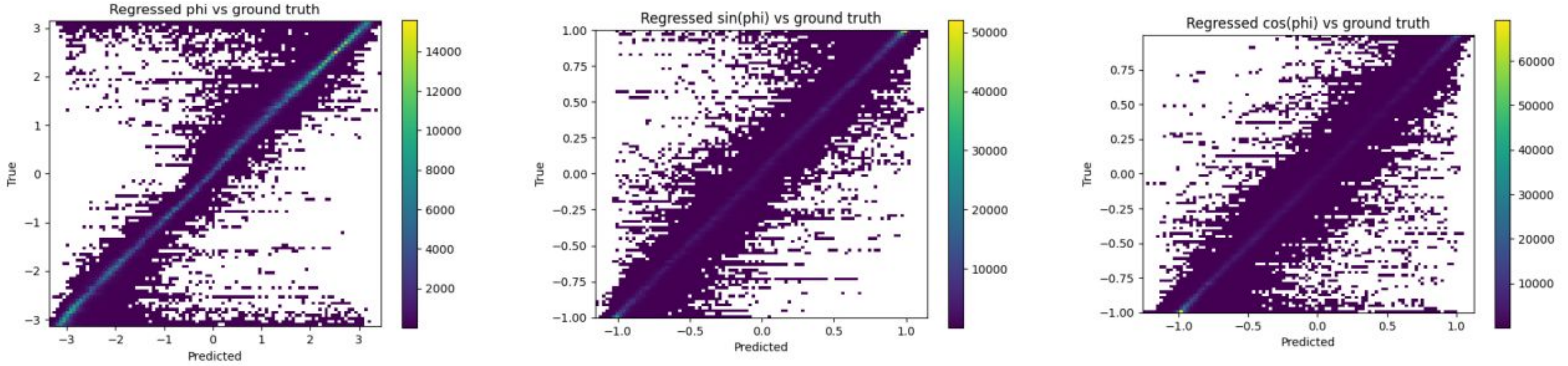
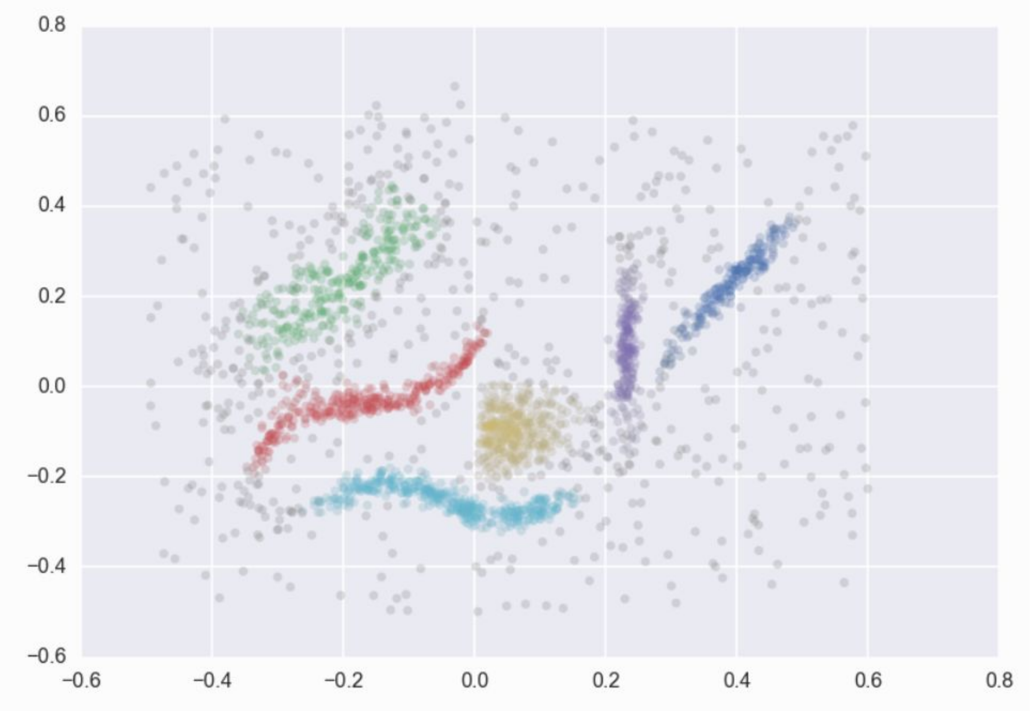


Figure B.1: Visualization of the regressed ϕ plotted against the actual ϕ for a model trained to regress θ, ϕ, q (left), and of the regressed $cos(\phi), sin(\phi)$ plotted against the actual values for a model trained to regress $\theta, sin(\phi), cos(\phi), q$ (middle, right). The models were trained on the TML:10-50 dataset.

Image: https://hdbscan.readthedocs.io/en/latest/advanced_hdbscan.html

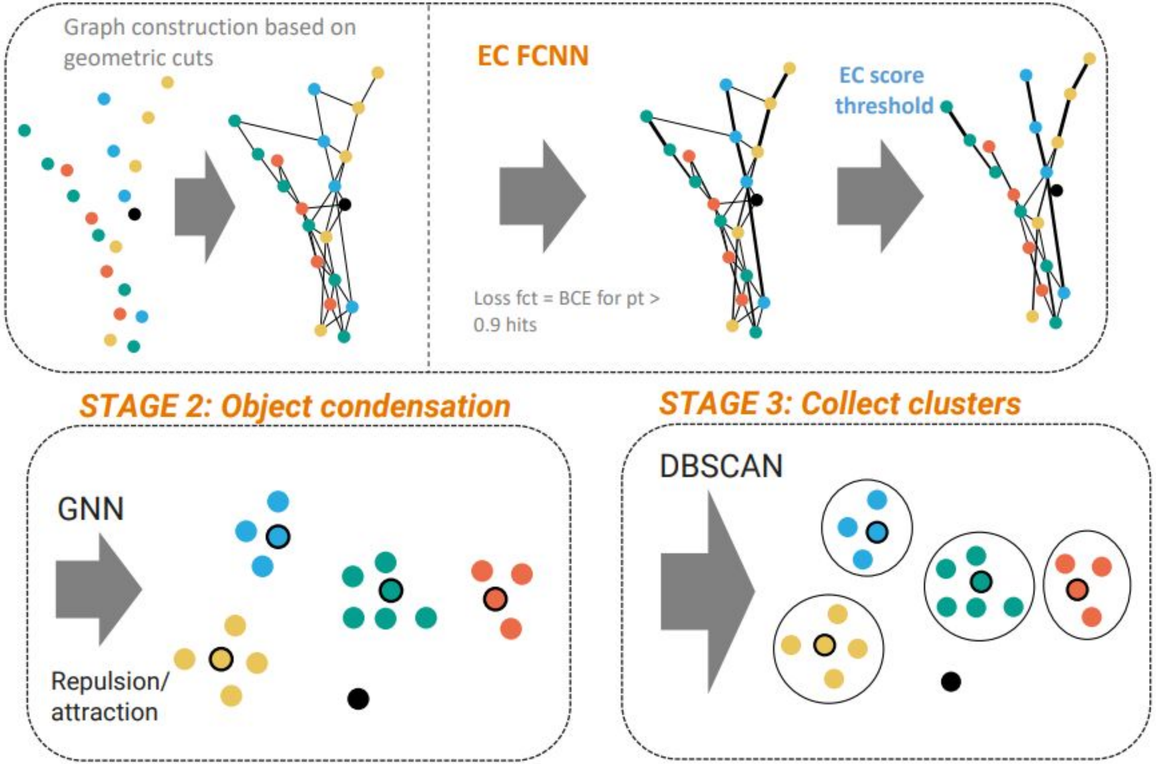
HDBSCAN [8]

- **No pre-specified number of clusters**
- **No assumptions about the data and cluster distribution**
- **Time complexity in O(n^2)**



Graph Neural Networks

Lieret et al. [10]



Memory Bottleneck

- **MHA – memory intensive**
  - L x H matrices of S x S floating point values
  - L - nr. layers, H - nr. heads, S - sequence length
- **Flash attention [7]**
  - Splitting matrix in blocks and doing calculations separately, then combining results
  - 3x faster and 20x more memory efficient

Refiner Network

- **Autoregressive model**
  - Adds hits to reconstructed tracks that were missed by the pipeline
- **Binary classifier network**
  - Determines whether each hit of a cluster truly belongs there
- **Regressor network**
  - Transformer Regressor, but per-cluster not per-event
  - Hits with track parameters too different from the rest of the group get removed