Transformers for Track Reconstruction and Hit Clustering in High Energy Physics

Track reconstruction is a crucial part of High Energy Physics (HEP) experiments. Traditional methods for the task scale poorly, making deep learning an appealing alternative. Following the success of Transformer models in the field of natural language processing, we investigate the feasibility of training a Transformer to translate detector signals into track parameters.

We study and compare different architectures, which we benchmark on simplified datasets generated by the recently developed simulation framework REDuced VIrtual Detector (REDVID).

Authors

Nadezhda Dobreva (1) nadezhda.dobreva@ru.nl
Yue Zhao (2) yue.zhao@surf.nl
Zef Wolffs (3,4)
Uraz Odyurt (4,5)
Sascha Caron (4,5)

Affiliations

(1) Institute of Computing and Information Sciences, Radboud University, Nijmegen, the Netherlands
(2) High Performance Machine Learning group, SURF, Amsterdam
(3) Institute of Physics, University of Amsterdam
(4) Nikhef (Dutch National Institute for Subatomic Physics), Amsterdam
(5) High Energy Physics Department, Faculty of Science, Radboud University, Nijmegen, the Netherlands

01 The Problem

The **traditional method** for the task of reconstructing tracks is using **Kalman Filters** (KF). This algorithm finds all possible combinations of hits and makes candidate clusters from those that form a sufficiently long track. However, it **scales poorly with detector occupancy and is inherently sequential** [1]. Especially with the introduction of the High-Luminosity LHC, there will be a higher frequency of events, pile-up, and an increased number of interactions and generated secondary particles per collision. The hardware is being upgraded to capture these changes, and a software upgrade is also in order. This motivates work on a **solution that can run faster and in parallel**, e.g. using deep learning.

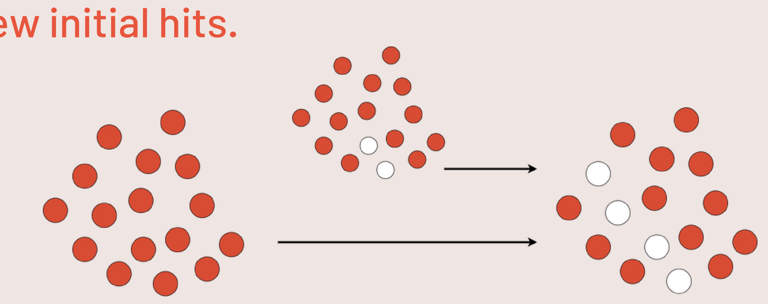
02 Transformers

Transformers are models that find patterns and meaning by tracking relationships in **sequential data**, using the attention mechanism which enables them to focus on certain parts of the data and ignore the irrelevant ones [2]. We believe they are a possible solution to the identified problem because they can:

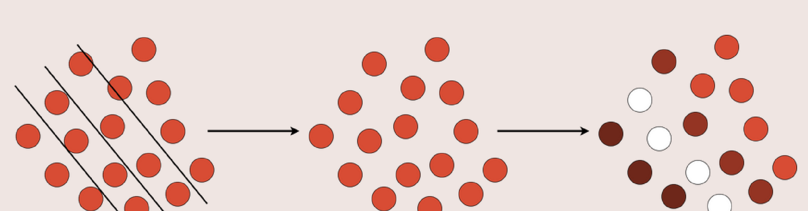
- Be operated in **parallel**,
- **Be made permutation equivariant,**
- **Work well with variable length inputs**,
- Be brought to **sub-quadratic time complexity in the number of tracks** during inference time.

We propose 3 architectures:

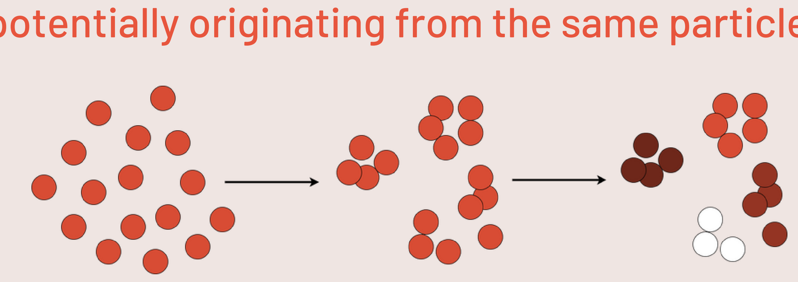
A1 **Encoder-decoder** which **autoregressively reconstructs a trajectory one hit at a time**, given a few initial hits.



A2 **Encoder-only** classifier that **produces a class label for each hit in an event**, given pre-defined bins (classes) within the track parameter space.

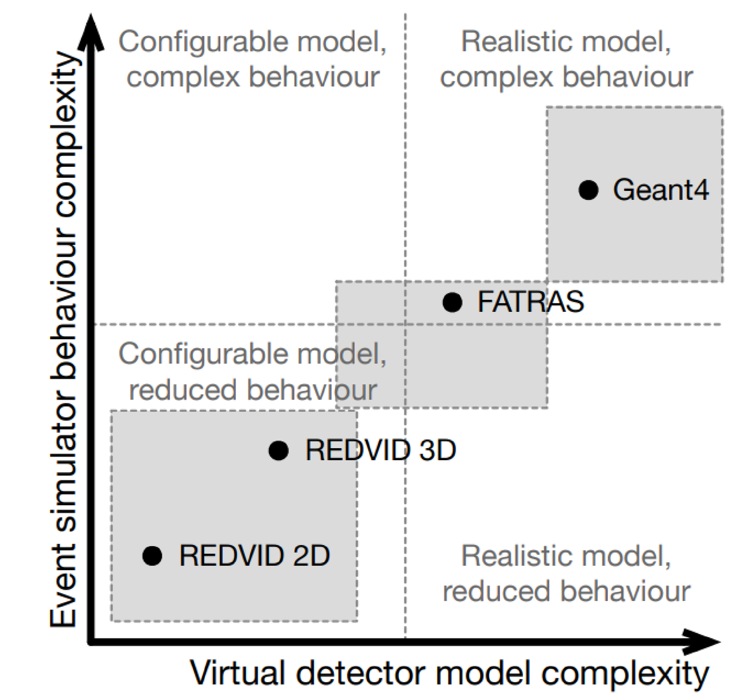


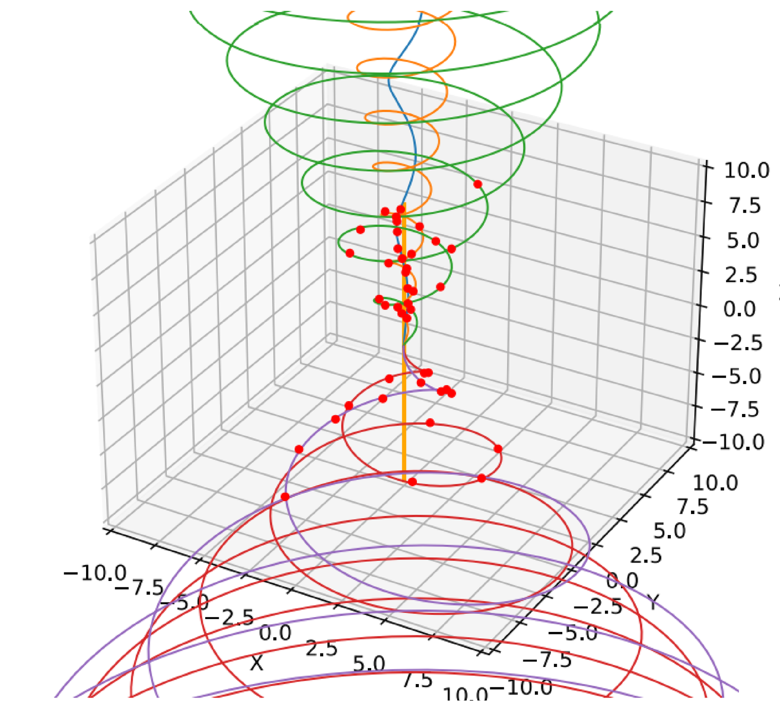
A3 **Encoder-only** regressor model that **regresses track parameter values for each hit in an event**, followed by clustering to identify groups of hits potentially originating from the same particle.



03 Data

The **REDVID simulator** [3, 4] is a customizable tool for creating events and deriving from them "recorded" hit point coordinates and trajectory function parameters. We use it for the generation of datasets at different levels of complexity, which facilitates the search for suitable architecure by narrowing down the initial search space. We report on **three noisy 3D datasets with helical particle trajectories and varying amount of tracks per event**: 3, 1-20 and 10-50. The tracks are given in the spherical coordinates and their uniquely defining parameters are **pitch and radial coefficient**.





Visualization of 5 helical tracks from the REDVID data.

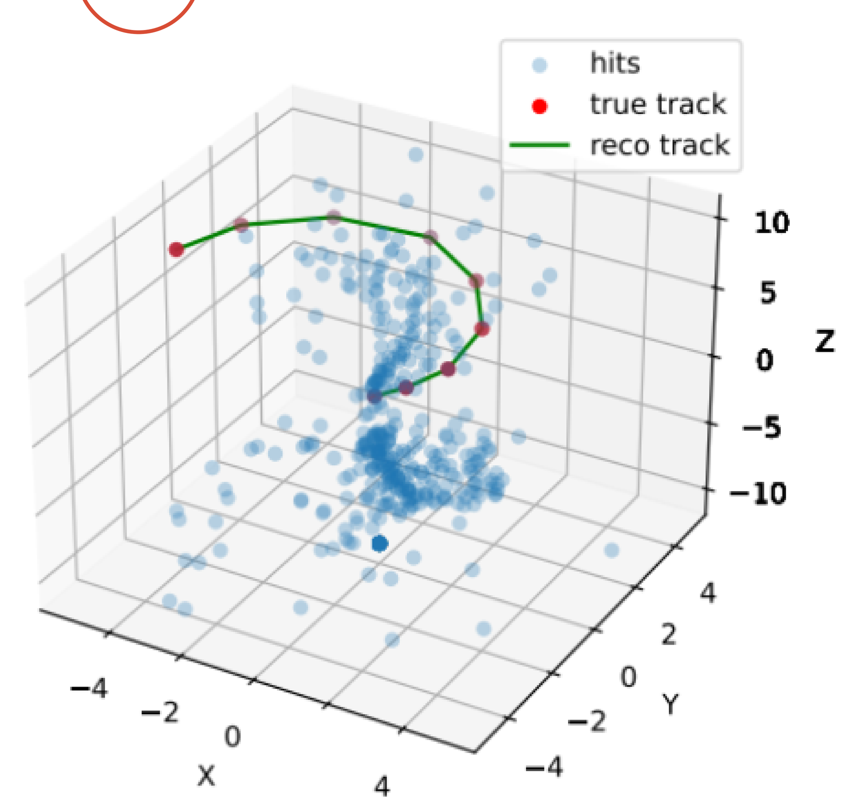
04 Results

The **TrackML score** is a custom metric, designed for the TrackML challenge [5]. For a reconstructed track to be considered for the scoring, it must have four or more hits, and at least 50% of the hits in the track must be correctly predicted. The score of a track is the sum of correctly assigned hit weights.

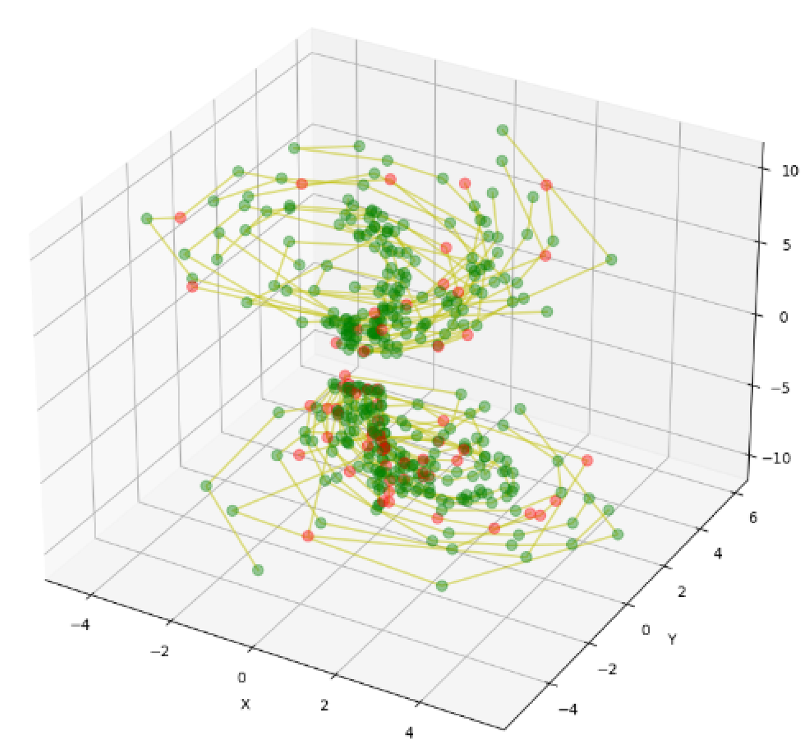
A1 >85% accuracy

A2 ~88% accuracy, ~98% TrackML score

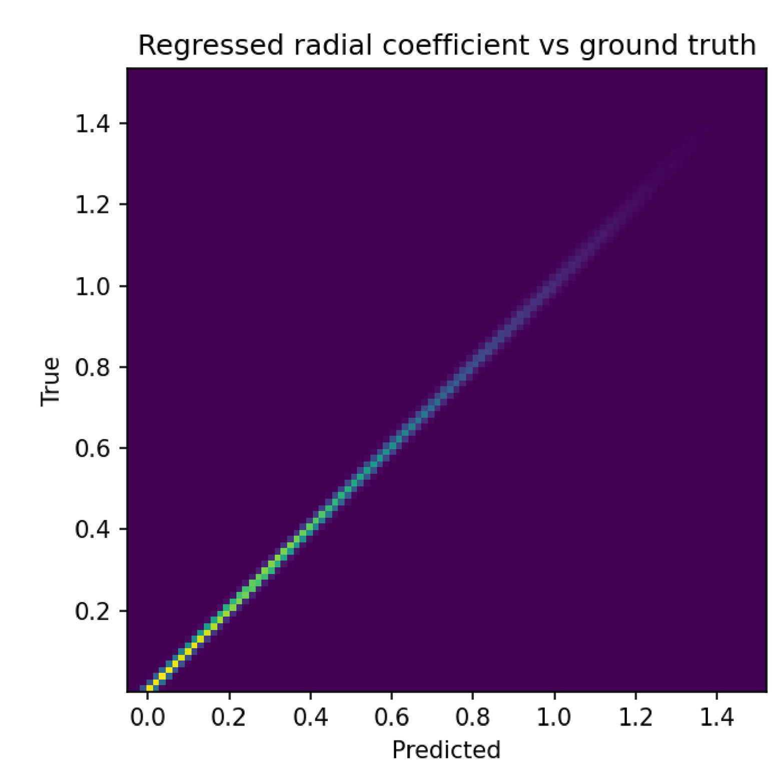
A3 >87% TrackML score

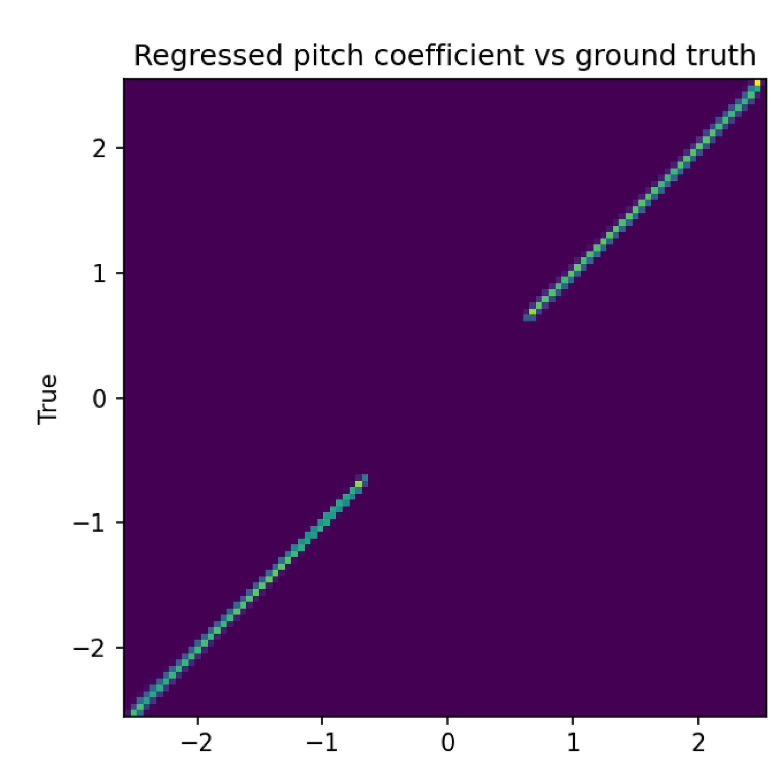


An example of a reconstructed track by A1.



An example of (in)correctly classified hits by A2.





Examples of regressed track parameters by A3.

05 Next Steps

- Training and evaluating on the TrackML dataset. This will require optimization techniques, such as **flash attention**.
- **Optimizing** the model and data structure.
- Assembling a time- and space-efficient pipeline for the task.

References

[1] Tsaris, Aristeidis, et al. "The HEP. TrkX project: deep learning for particle tracking." Journal of Physics: Conference Series. Vol. 1085. IOP Publishing, 2018.
[2] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems (2017).
[3] Odyurt, Uraz, et al. "Reduced Simulations for High-Energy Physics, a Middle Ground for Data-Driven Physics Research", 2023.
[4] https://virtualdetector.com/redvid/
[5] Kiehn, Moritz, et al. "The TrackML high-energy physics tracking challenge on Kaggle." EPJ Web of Conferences. Vol. 214. EDP Sciences, 2019.