Point Cloud Transformer for elementary particle signals segmentation

Anastasiia Nikolskaia, Pavel Goncharov, Gennady Ososkov, Egor Shchavelev

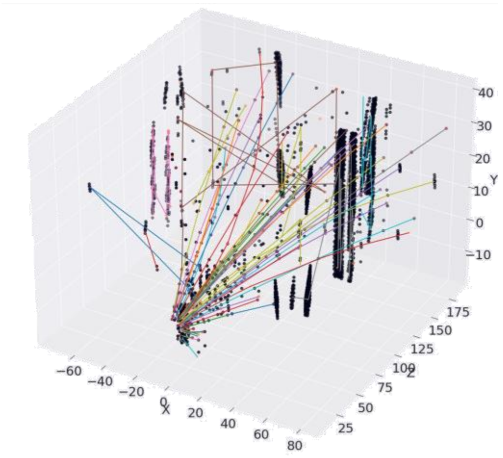
GRID-2023

Motivation

- Particle track reconstruction in dense environments such as the Run-4 detectors of the High Luminosity Large Hadron Collider (HL-LHC) and of MPD NICA is a challenging pattern recognition problem.
- Modern experiments produce data with high frequency.
- Well-known tracking algorithms based on the Kalman filter are not scaling well with such amounts of data.

Neural networks

- Deep learning algorithms bring a lot of potential to the tracking problem, due to their capability to model complex non-linear data dependencies, learn effective representations of high-dimensional data through training parallelize easily on high-throughput architectures such as GPUs
- One architecture may be adapted for different task or solve them simultaneously



Neural Network (online) → High recall, **low precision** Very fast → **Reconstructed event**

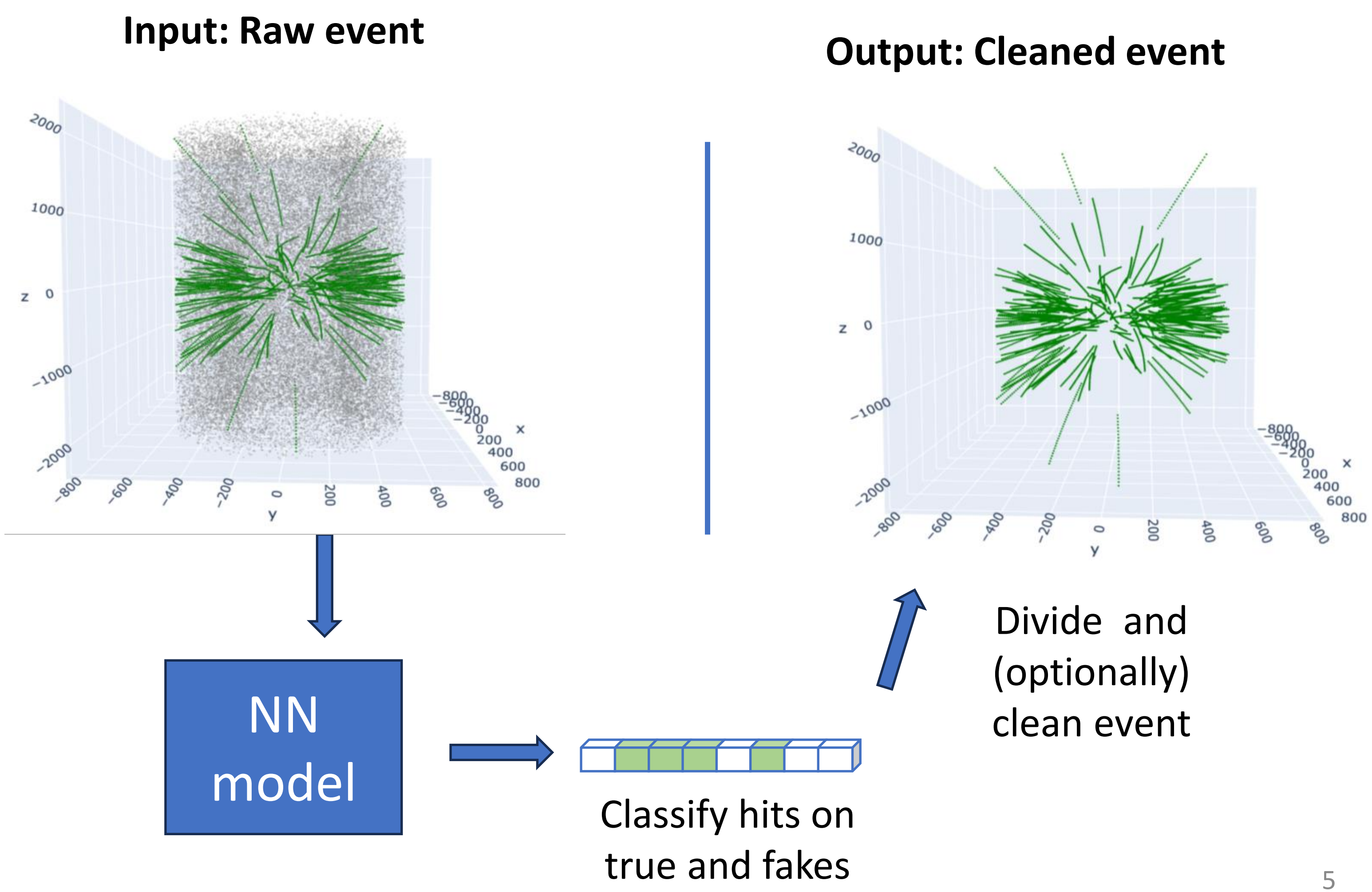
Kalman Filter (offline) → High recall, high precision **Too slow** → **Reconstructed event**

SPD Experiment

- SPD (Spin Physics Detector) is a future experiment at the NICA facility in Dubna. The main goal of this experiment is to test the foundations of quantum chromodynamics (QCD).
- It is expected that the SPD experiment will produce events with a frequency of 3 MHz (20 Gb/s).
- The events are received in time slices, making the development of an effective online solution crucial.
- There is a need to develop a modern system for fast event reconstruction in the straw detector.

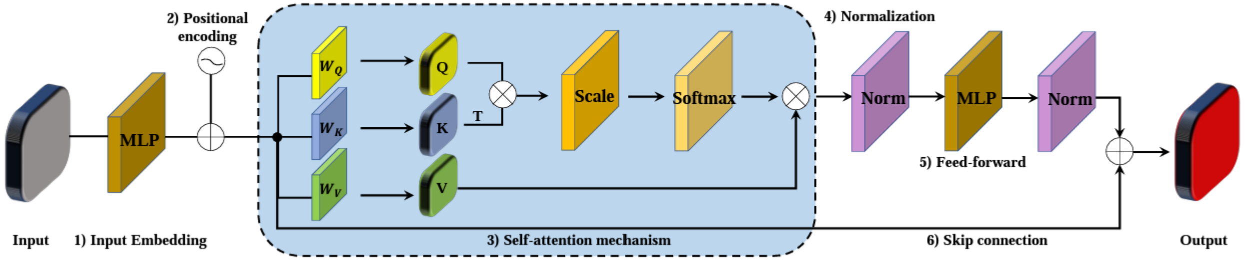
Event segmentation pipeline

- Event segmentation is an important stage in event recognition.
- It enhances further analysis and processing.
- Filtering out fake hits reduces the computational requirements and memory consumption for track reconstructions.
- It indicates promise for the model to tackle more complex tasks in the future.



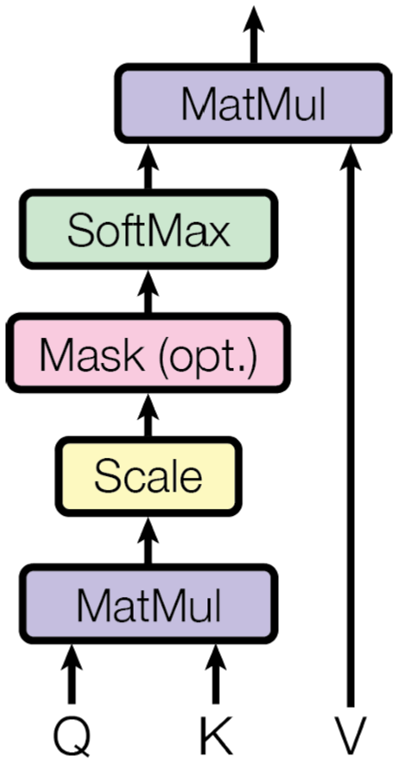
Transformers

- Transformers is the dominant architecture family in different fields: NLP, CV, autonomous driving etc. due to the ability to model long-range dependencies
- Transformers are invariant to permutation and cardinality of the input elements, which is highly suitable for point cloud processing
- The main feature of transformer is the attention mechanism, which aims to “understand” relation between input elements
- By assigning weights to the elements based on their relevance and importance, the attention mechanism enables the model to focus on relevant information during processing.



Attention mechanism

- Given embedded feature map, it is projected to three different feature spaces using three learnable weight matrices $W_Q \in R^{,C\times C_Q}, W_k \in R^{C\times C_K}, W_V \in R^{C\times C}$,where typically $C_K = C_Q$.We can formulate then
$$Query = XW_Q, Key = XW_K, Value = XW_V$$
- Given the Query, Key, and Value matrices, an attention map is formulated as:
$$Attention\_map = Softmax(Q_K^T/\sqrt{d_k})* \mathrm{V}$$
- The attention map of size of N×N measures the similarity of any two input points
- Attention map is permutation equivariant and dynamic during inference
- **Computational complexity of self-attention is N^2**



PCT

- PCT aims to transform(encode) the input points into a new higher dimensional feature space, which can characterize semantic affinities between points
- Encoder is based on transformer architecture using attention layers
- Using max-pooling, global representation of event is constructed.
- This representation is added to each point embedding to preserve context
- Decoder returns probability of being real track hit for each point

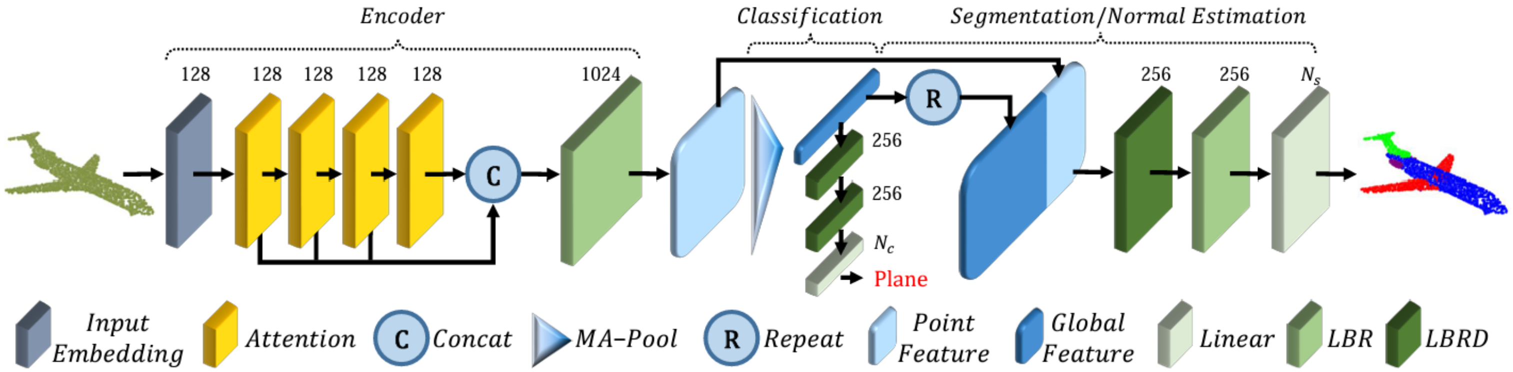
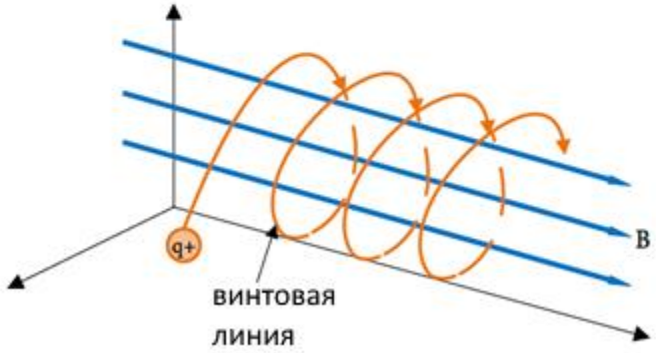
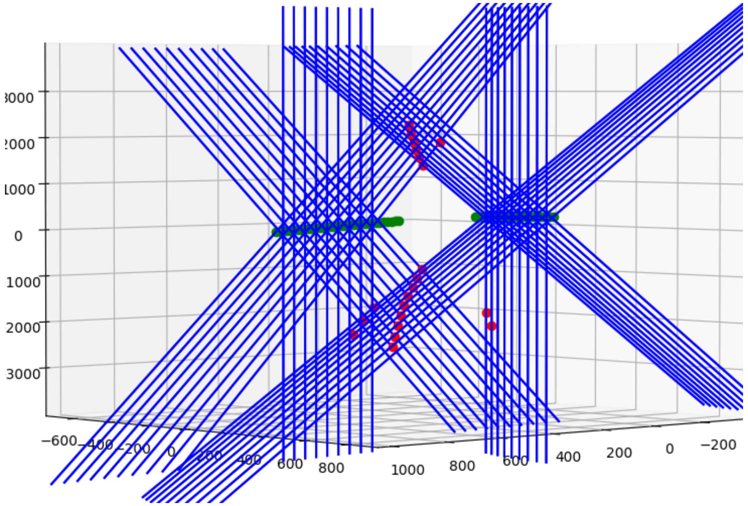


Figure from "PCT: Point cloud transformer"

Data generation

- Generator is written as a Python program.
- Multiplicity in each event is given by a random number from 1 to 10 tracks per event.
- The transverse momentum of a particle is a random number with a uniform distribution in the range of values from 100 to 1000 MeV/s.
- Vertex coordinates are also random in the volume corresponding to the area of particle collisions.
- The particle trajectory is represented by a selection of points on a segment of a helix with a helix pitch h=$2\pi/B\ |m/q|vcos\alpha$ and radius R=$1/B|m/q|vsin\alpha$.
- Detector configuration with 35 stations is considered.





Experiments

- Data for training: 5000 generated events
- Data for testing: 100 generated events
- Used metrics:
  - $recall = \frac{N_{true}^{pred}}{N_{in}}$
  - $precision = \frac{N_{true}^{pred}}{N^{pred}}$

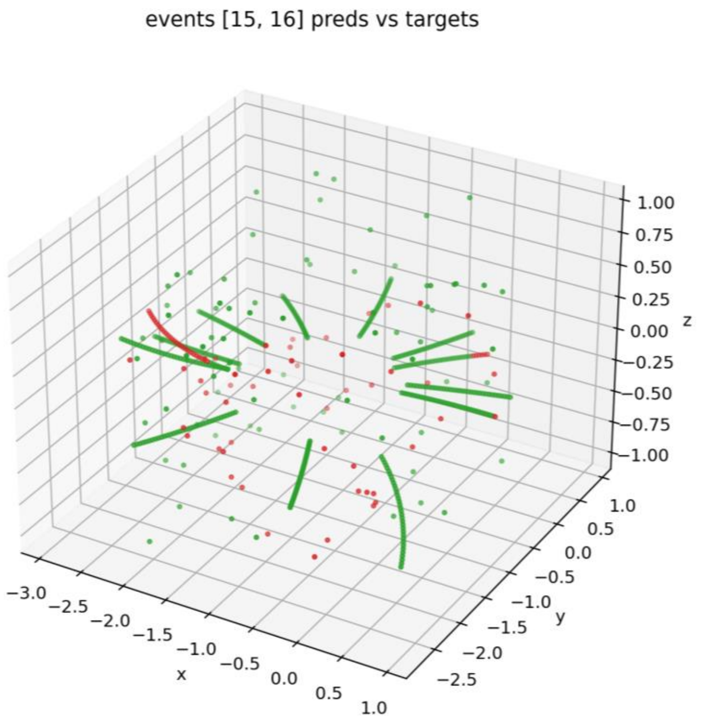
- N_{true}^{pred} - no. real hits that the network found
- N_{in} - no. all real hits known from Monte-Carlo
- N^{pred} - no. all predicted hits

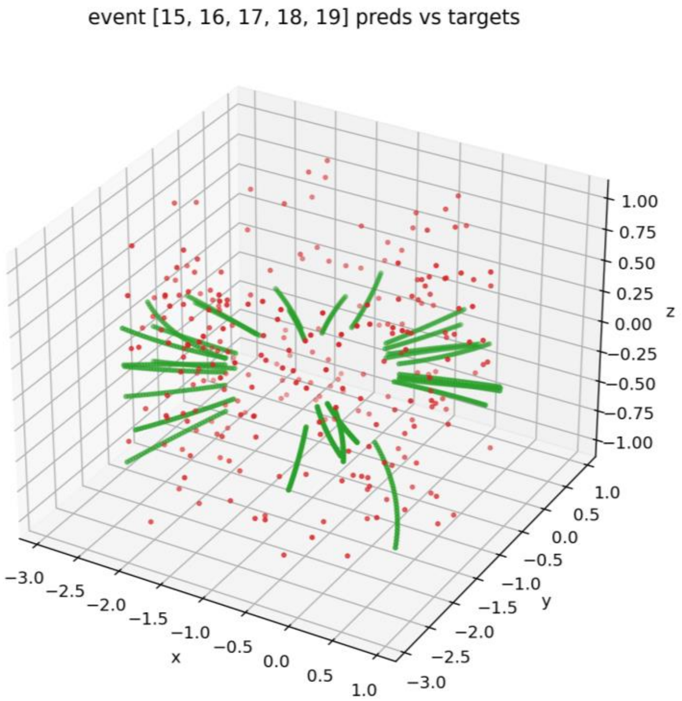
Number of events in time slice	Precision	Recall	Speed (sec/event)
1	0.86	0.97	0.07
2	0.81	0.98	0.125 (0.23)
5	0.77	1.0	0.21 (1.08)
10	0.88	0.92	0.73 (7.25)

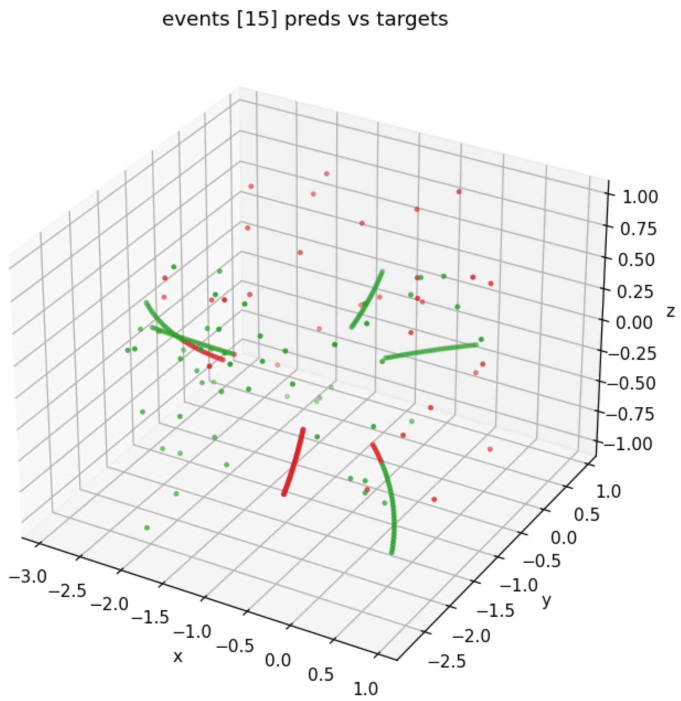
Speed value in brackets illustrates speed for whole time slice

Experiment results

- Predicted and real labels of hits for portions of 1, 2, 5 events
- Red color – predicted and real label are different, green – same







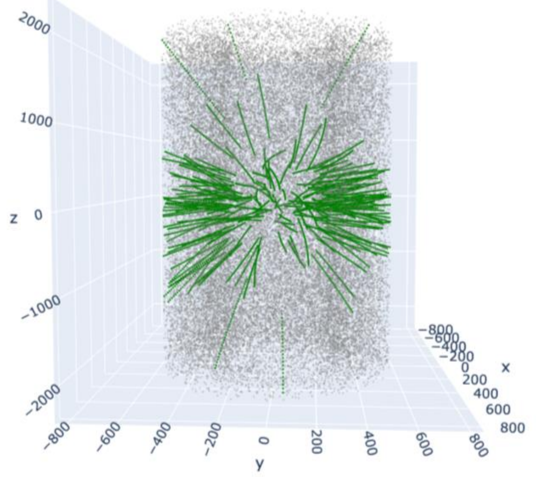
Note: Seems that the model overfitted to predict true hits

Event partition

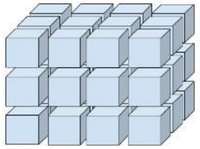
- Given time slice of multiple events, we get huge numbers of hits (60 events will produce > 32000 of hits)
- Complexity of attention block is O(N^2), which leads to insufficient time and memory consumption
- The time slice may be divided into smaller regions and analyzed locally. Hence complexity of model tends to O(N)

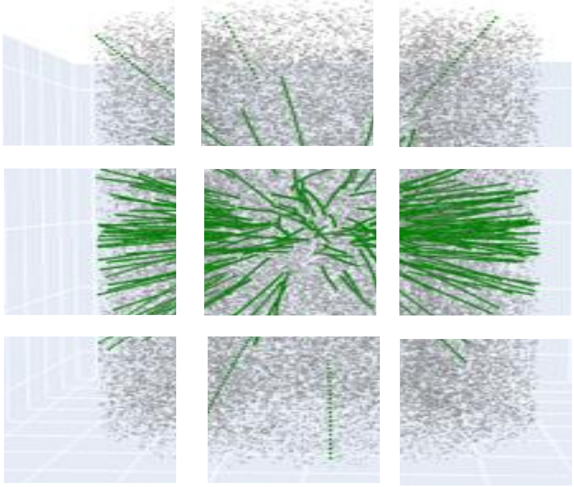
Event segmentation pipeline

Input: Raw event

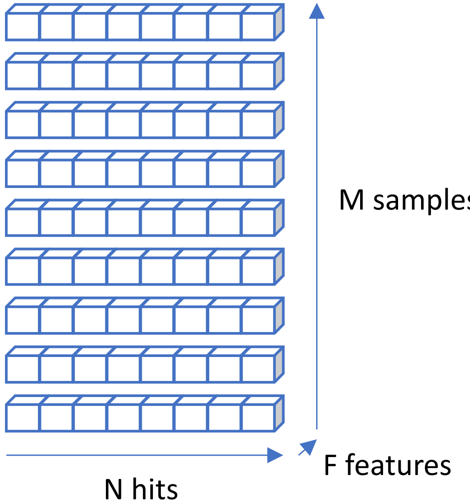


Divide the detector space into M voxels, i.e. smaller subspaces



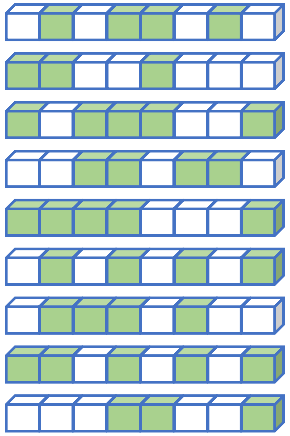


Take hits from each voxel and form a batch of M x N x F samples pretending that each sample is a mini-event



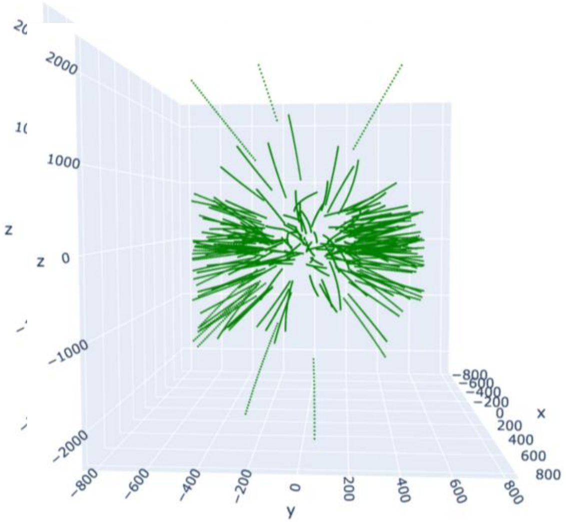
PCT model

Classify hits on true and fakes



Combine into one event

Output: Cleaned event

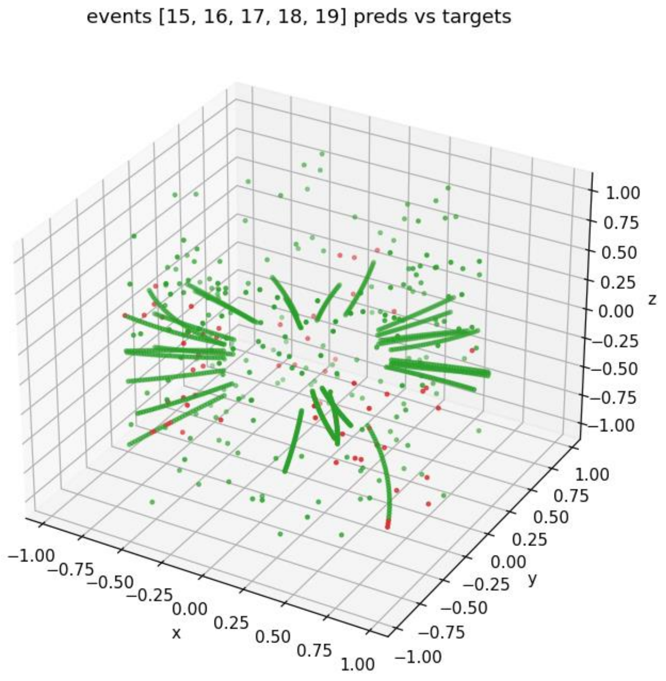


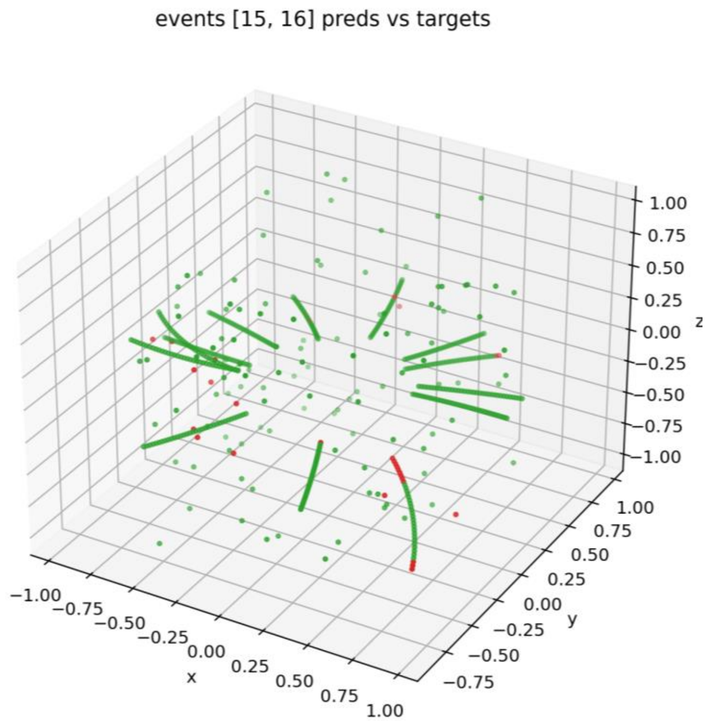
Experiments

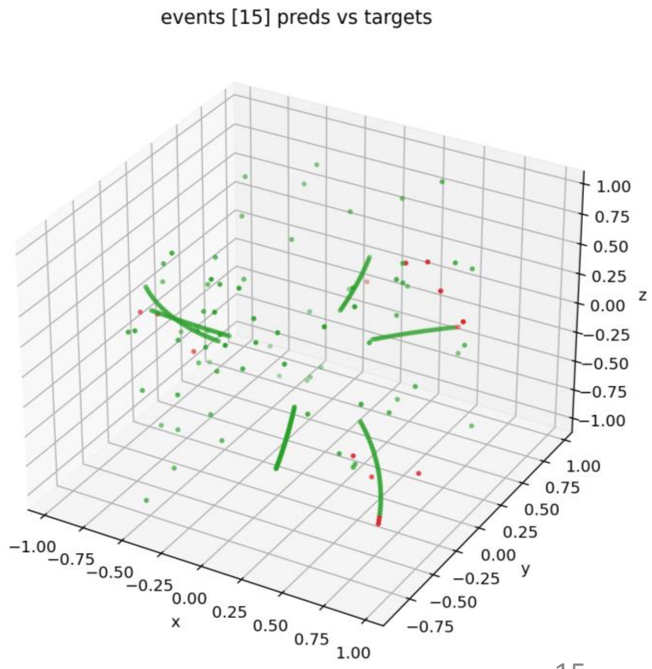
Events per time slice	Number of parts	Precision	Recall	Speed (events/sec)
1	16	0.96	0.98	0.06
1	32	0.97	0.97	0.7
1	64	0.96	0.99	0.07
2	64	0.97	0.95	0.17 (0.34)
5	64	0.93	0.97	0.27 (1.37)
10	64	0.90	0.97	0.59 (5.96)

Experimental results

- Predicted and real labels of hits for portions of 1, 2, 5 events
- Red color – predicted and real label are different, green – same







Future work

- Replace the attention block with a linear version.
- Develop a hierarchical segmentation model that utilizes both global and local voxel-wise features.
- Develop a dynamic event partitioning technique that uses anchor points for more efficient data processing.
- Adapt the model to reconstruct hits from raw data obtained from straw signals.
- Modify the model for the specific task of track recognition.
- Test the architecture on BM@N, MPD, and BESIII datasets to evaluate its performance and effectiveness.