Deep learning methods in high luminosity track reconstruction scenario: applying TrackNET to TrackML challenge

Daniil Rusov (rusovd@jinr.ru)

Pavel Goncharov, Gennady Ososkov, Alexey Zhemchugov

Mathematical Modeling and Computational Physics, 2024
24 October, 2024

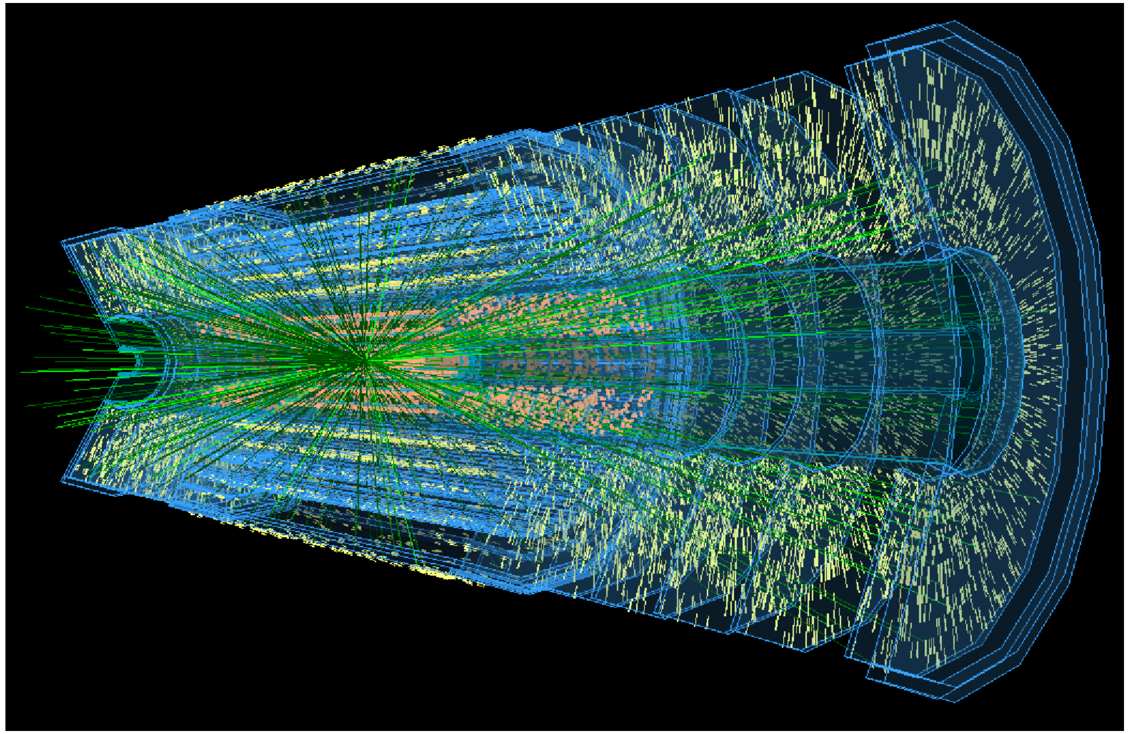
Problem Statement: The Need for Advanced Tracking Methods

Unprecedent scale of modern experiments:

- Up to 200 simultaneous proton-proton interactions is expected at High Luminosity Large Hadron Collider, 200 particle tracks on average, 40K of tracks considering pile-up
- The expected event rate of the SPD experiment is 3 MHz
- Event pileup makes track reconstruction more complicated
- Traditional tracking methods struggle with dense, overlapping particle tracks due to computational complexity and time constraints

Deep Learning for Efficient Track Reconstruction:

- DL models can handle high-dimensional data and complex spatial correlations between tracks
- Multiple scattering and inhomogeneous magnetic field effects could be learned from training data
- Effective parallelization using GPUs out of the box
- TrackML Challenge was launched to explore new scalable approaches for particles tracking



https://webific.ific.uv.es/web/en/content/taking-lhc-higher-luminosity

Deep learning-based methods have a potential to cope with immense data volumes in modern experiments

Classification of Track Reconstruction Methods

Local Tracking

Works with parts of event data (hits, track segments, detector parts).

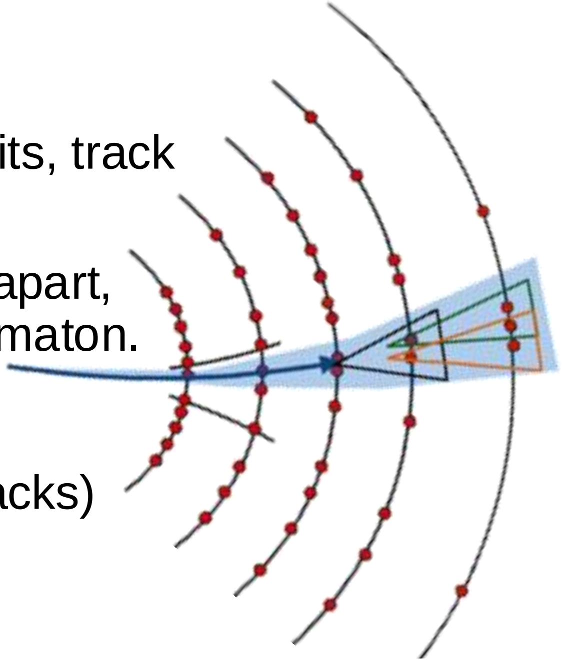
Examples: Kalman Filter (stand apart, bunch of methods), Cellular Automaton.

Pros:

- High parallelism (individual tracks)
- Lightweight and fast
- Low memory use

Cons:

- Requires post-processing for full event reconstruction
- Prone to false positives (due to lack of full event view)



CNNs on FPGAs for Track Reconstruction

Global Tracking

Uses full event data for track reconstruction.

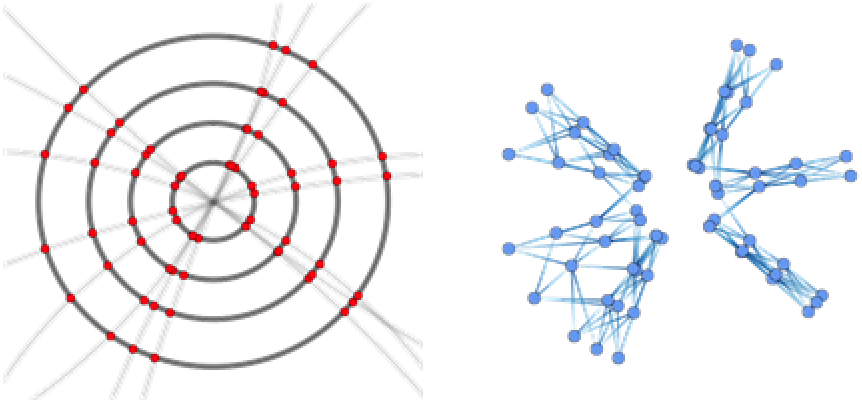
Examples: Graph Neural Networks, Hopfield network, Point Cloud Processing.

Pros:

- Higher quality metrics, fewer false positives
- Event-level parallelism possible

Cons:

- High memory requirements (entire event as input)



Graph Neural Networks in Particle Physics

Hybrid Tracking Methods

Hybrid Tracking

Combines local and global tracking methods.

Stages

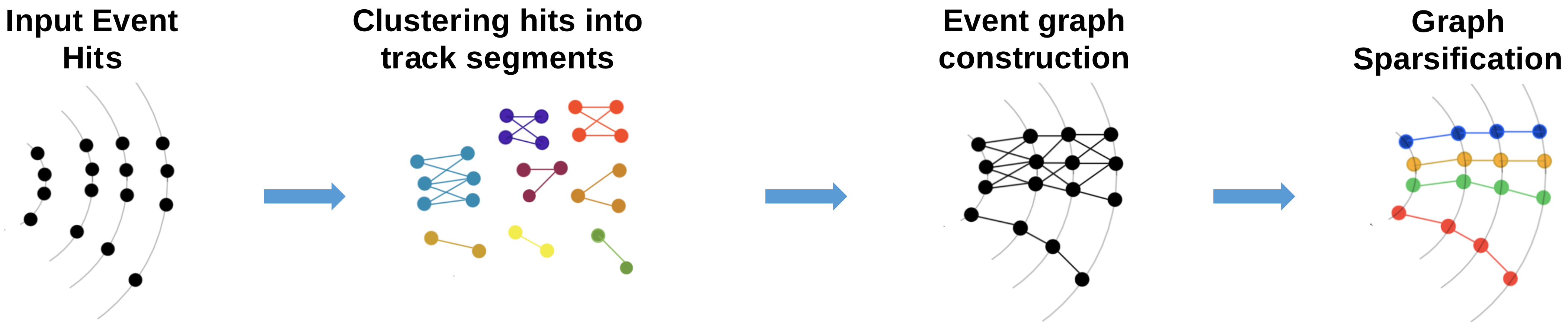
1. Track seeding, track candidates or event graph building. Main goal: high recall while reducing the number of false positives as much as possible.
2. Tracks selection either by various fitting criteria or ranking candidates using machine learning methods (e.g. graph sparsification, candidates’ classification). Main goal: increase precision without recall dropping.

Pros:

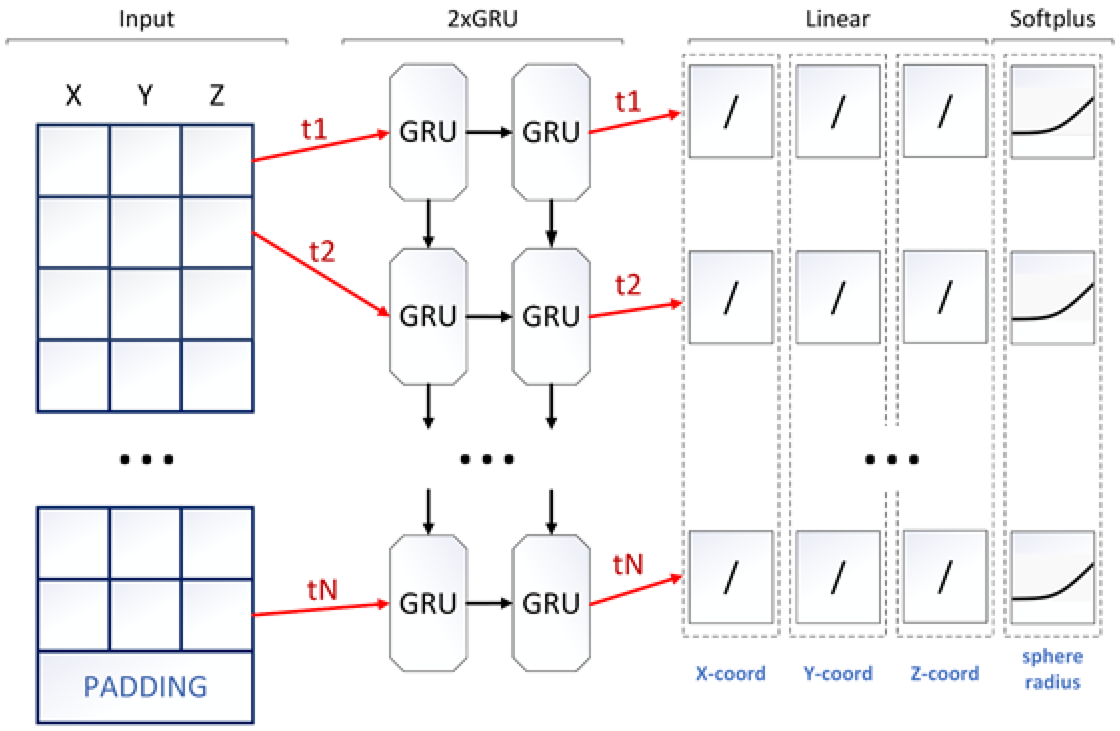
- Achieves both high performance and efficiency.

Cons:

- Errors depend on multiple models.



TrackNET as Local Tracking Method



Model Architecture

How the model works?

- Locality – one track-candidate during the inference
- The model predicts center and radius of the sphere where to search for the next hit
- All event hits are placed in the spatial search index
- Only K nearest to the center of sphere hits are checked (setting K=1 leads to linear computational complexity)
- Candidate tracks are extended by hits that fall into sphere.
- Extended track-candidates are fed back to the model input.

Pros:

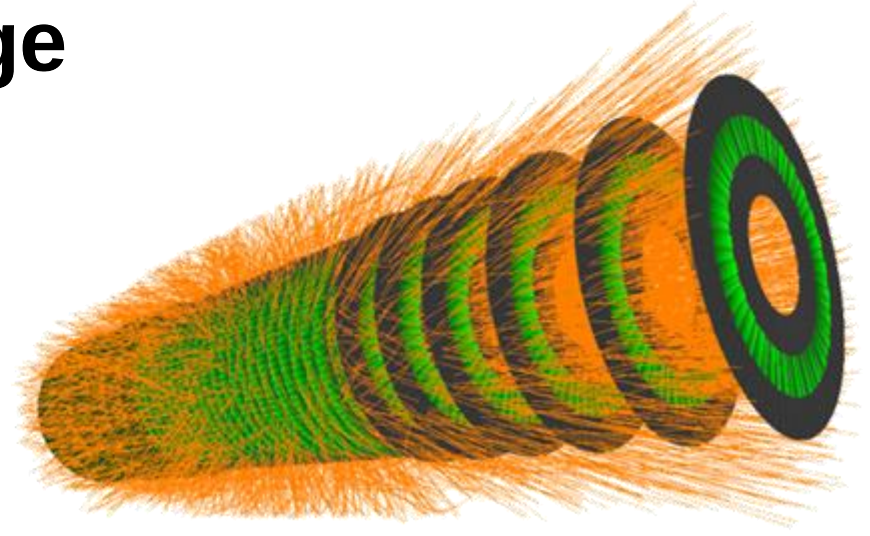
- Fast and Extremely Lightweight
- Few hyperparameters to tune – loss weights and K
- No need for seeding – prediction starts from single hit

Cons:

- lot of false positives or so-called ghosts, because of its local nature of prediction

TrackML Particle Tracking Challenge

- 100 GB of simulated data encompassing around 10,000 events
- 10000 tracks per event on average
- Each track has about 10 hits – 100000 signals in one event
- Straight-line tracks (high momentum) are rare and have more weight in competition scoring function



https://www.kaggle.com/c/trackml-particle-identification/overview

Noticeable participants:

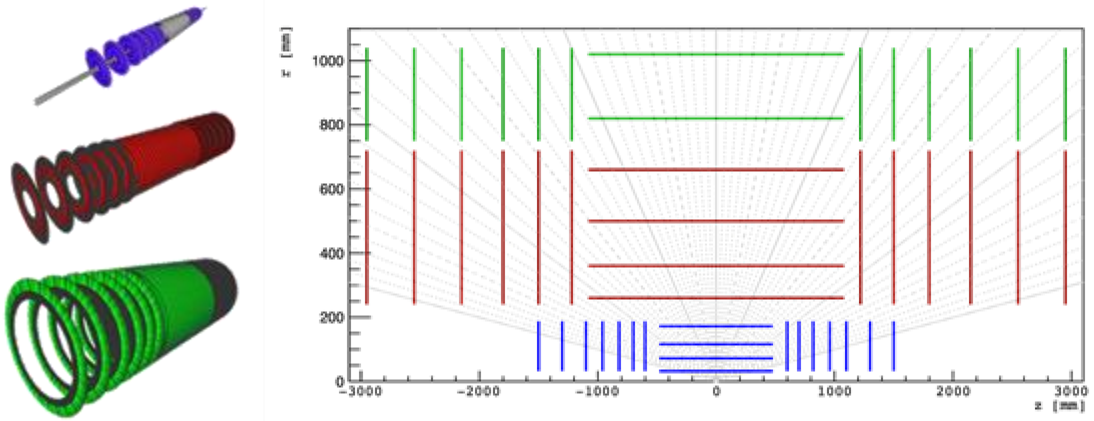
- 1st: top-quarks – Logistic regression for pairs and triplets, helix extrapolation (8 min/event).
- 2nd: outrunner – Dense NN for pair prediction, circle fitting (3+ hrs/event).
- 3rd: Sergey Gorbunov – Triplet seeds, helix fit with magnetic field estimation (0.56 sec/event).
- 9th: CPMP – DBSCAN clustering, filtered by module frequency (10 hrs/event, 30,000+ DBSCAN runs).
- 12th: Finnies – DBSCAN seeding, **LSTM for predicting next 5 hits** (slow, no speed given).

Most of the solutions repeat the classical pipeline for tracking – seeding followed by trajectory fitting.

TrackNET Training Overview on TrackML Dataset

Space shrinking/compression:

- for external detectors, the distance between layers is ~2 times larger than for internal ones. Shrink the space of external detectors by factor two.



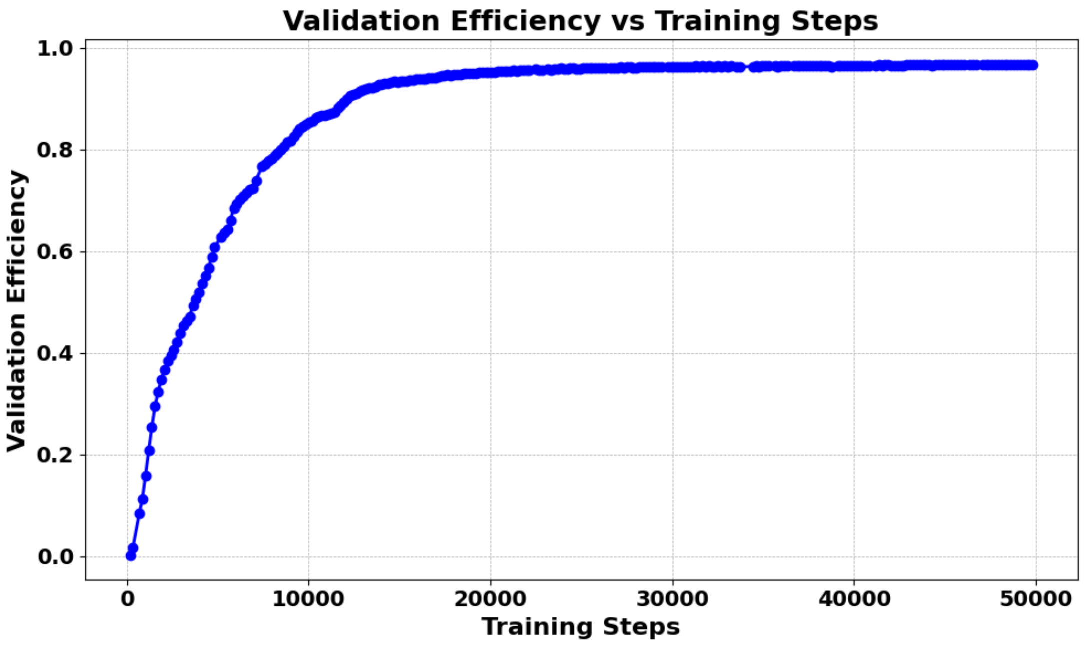
[The Tracking Machine Learning challenge: Accuracy phase]

Multiple hits on the same layer during training due to modules intersections:

- take the closest one (least r in cylindrical coordinates)

Picking seeds:

- taking all hits from the innermost layers



Training

- 10 mln tracks, 300 epochs
- 15 hours on single Nvidia V100 32GB
- weight in TrackNET loss alpha = 0.9999 because of unnormalized coords and large detector

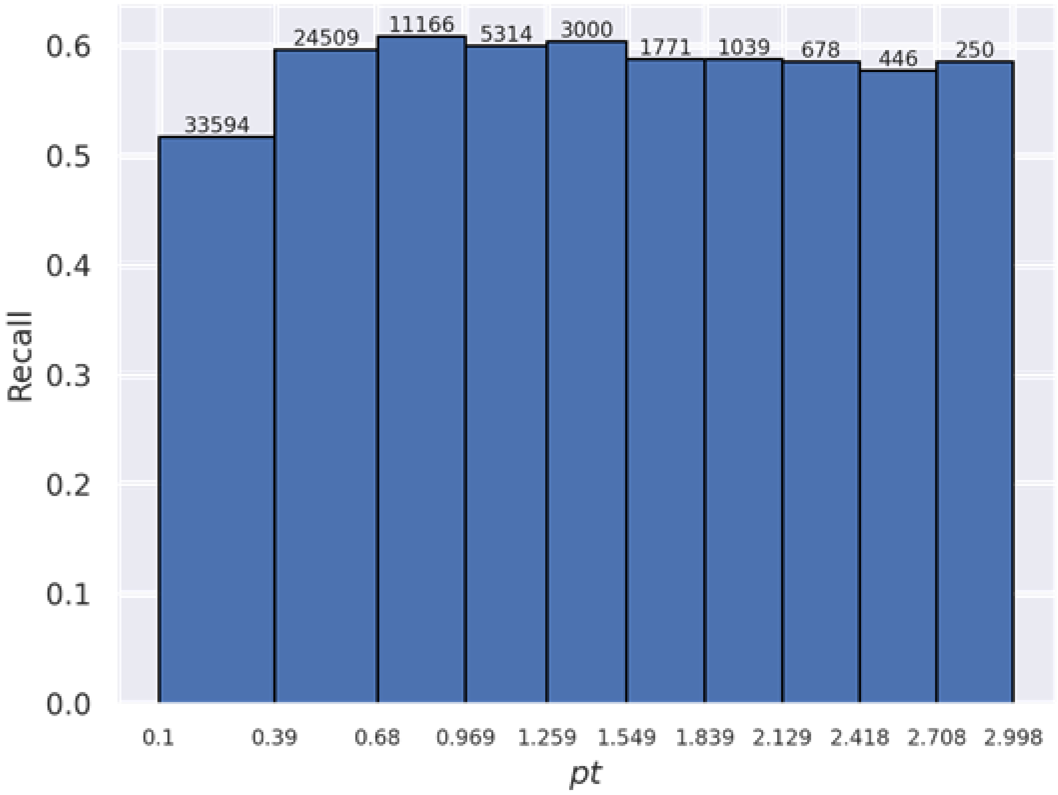
TrackML Evaluation Results (work in progress)

Testing setup:

- Xeon(R) Gold 6148 CPU @ 2.40GHz (only CPU, optimization of memory operations is required)
- **No data reduction** based either on particle momentum (no pt cut) or number of events (original pile-up)
- No specific tuning for the TrackML scoring metric – **considering all tracks with equal importance**
- Following TrackML metric, a track is considered fully reconstructed if >50% of hits were recognized correctly
- In case of duplicate track candidates, **only one is included in the final metrics**

K searched hits	1	2
Recall (%)	35,15	56,41
Precision (%)	33,47	1,01
Event processing time (sec)	6,4289	45,2144

Statistics on 10 events (100K tracks)



Note:

- Building precision vs pt plot requires momentum estimation for the track-candidates

Conclusion

- The TrackNET model demonstrates high performance even in challenging environments.
- Due to the model's local nature, a second stage of track-candidate selection is necessary to further improve precision.
- The model's first application to the TrackML dataset has been successfully conducted.
- Further work is needed to enhance performance: tuning to the TrackML scoring metric, applying data reductions, and balancing training samples based on relevance.
- Without any optimizations, the model processes an average TrackML event in approximately 10 seconds on a single CPU.
- The code of TrackNET application to TrackML data will be open-source soon.
- The results of TrackNET application to TrackML were the part of the talk presented at CHEP 2024 at Kraków this week.

Deep learning methods in high luminosity track reconstruction scenario: applying TrackNET to TrackML challenge

Daniil Rusov (rusovd@jinr.ru)

Pavel Goncharov, Gennady Ososkov, Alexey Zhemchugov

Mathematical Modeling and Computational Physics, 2024
24 October, 2024

StepAhead TrackNET: Dealing with Detector Inefficiency

- The network is designed to predict the continuation of a track even when multiple hits are missing.
- It predicts two steps ahead simultaneously (covering two spheres of potential hit locations).
- If no hit is found in the 1st sphere, the 2nd sphere is checked.
- When a hit is in the 2nd sphere, the track is extended using a **virtual point** at the center of the 1st sphere.
- While predictions based on the first sphere are less accurate (due to larger uncertainty), this broader search radius helps locate the next track hits.
- However, using a virtual point in place of a missing hit **for the very first prediction** (first hit and virtual point) can introduce confusion. To mitigate this, track candidates without hits in the first sphere are temporarily saved and extended later using both the virtual point and the hit from the second sphere.