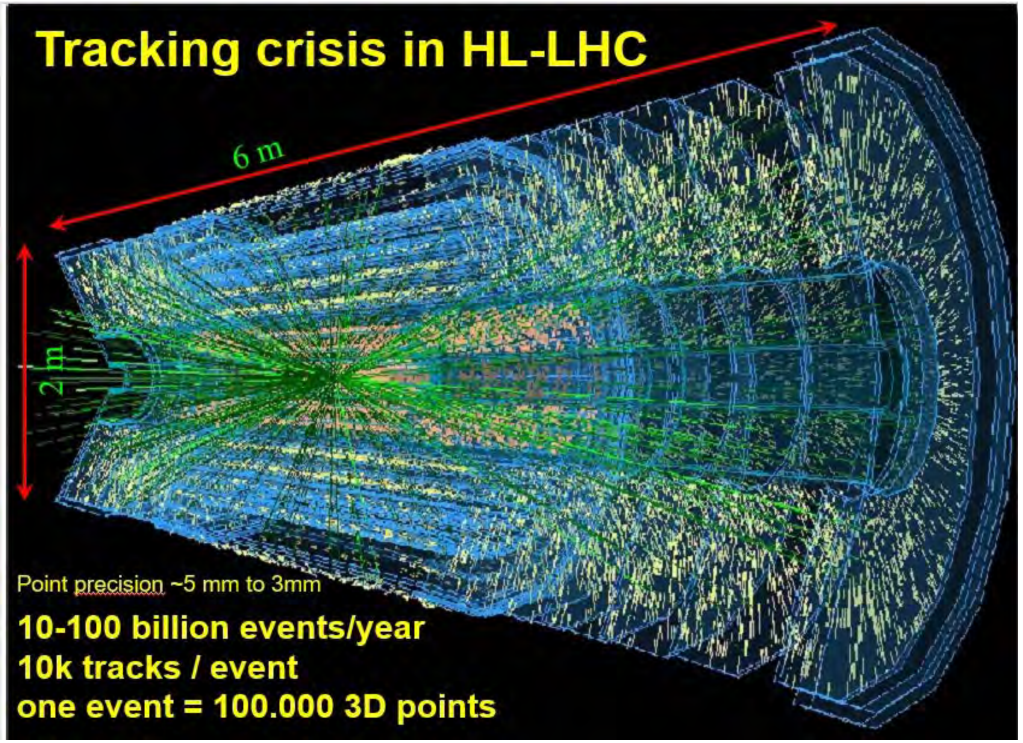
Deep neural network applications for particle tracking at the BM@N and SPD experiments

Goncharov Pavel, Rusov Daniil, Lubchenkov Leonid, Shchavelev Egor,
Ososkov Gennady

Problem Statement

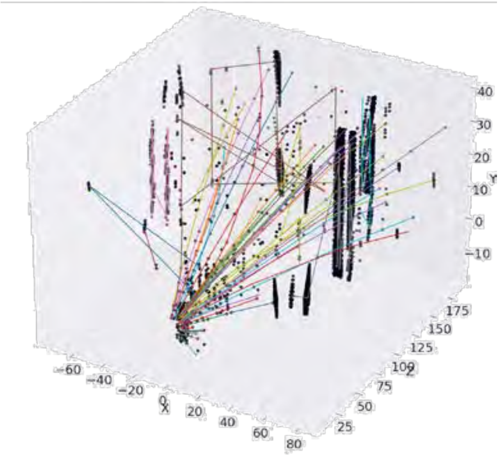
- **Particle track reconstruction in dense environments such** as the Run-4 detectors of the High Luminosity Large Hadron Collider (HL-LHC) and of MPD NICA is a **challenging pattern recognition problem**.
- Expected that SPD experiment will produce events with **frequency of 3 MHz**
- **Well-known tracking algorithms** based on the Kalman filter **are not scaling well** with the amounts of data being produced in modern experiments



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

What we propose for tracking?

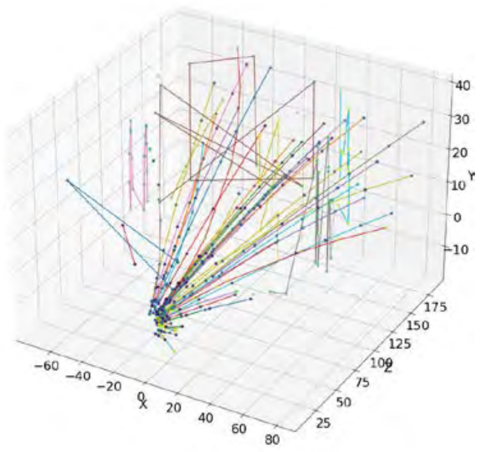


Neural network (online)

High recall, **low precision**
Very fast

Kalman filter (offline)

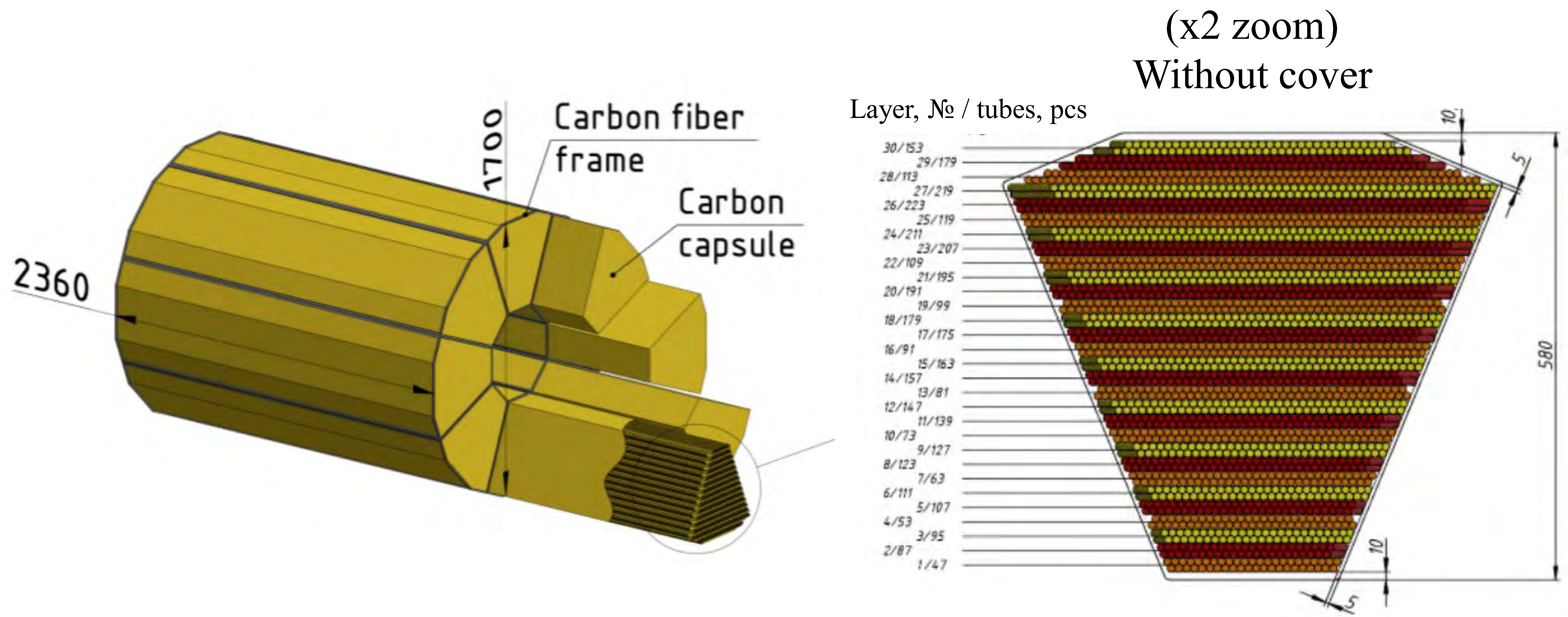
High recall, high precision
Too slow



Reconstructed event

SPD Experiment

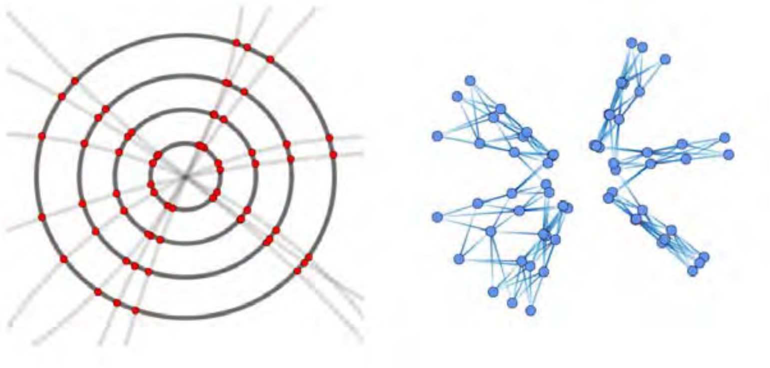
- SPD (Spin Physics Detector) – is a future experiment of NICA facility in Dubna. The main goal of the experiment is to test the foundations of quantum chromodynamics (QCD).
- Expected that SPD experiment will produce events with **frequency of 3 MHz (20 Gb/s)**
- **Only 2-5% of all data is interesting for physicists!**
- It is required to develop modern system for fast event reconstruction in straw detector



External view of the SPD straw detector (left) and its module in section (right)

Graph Neural Network (GNN) for Particle Tracking

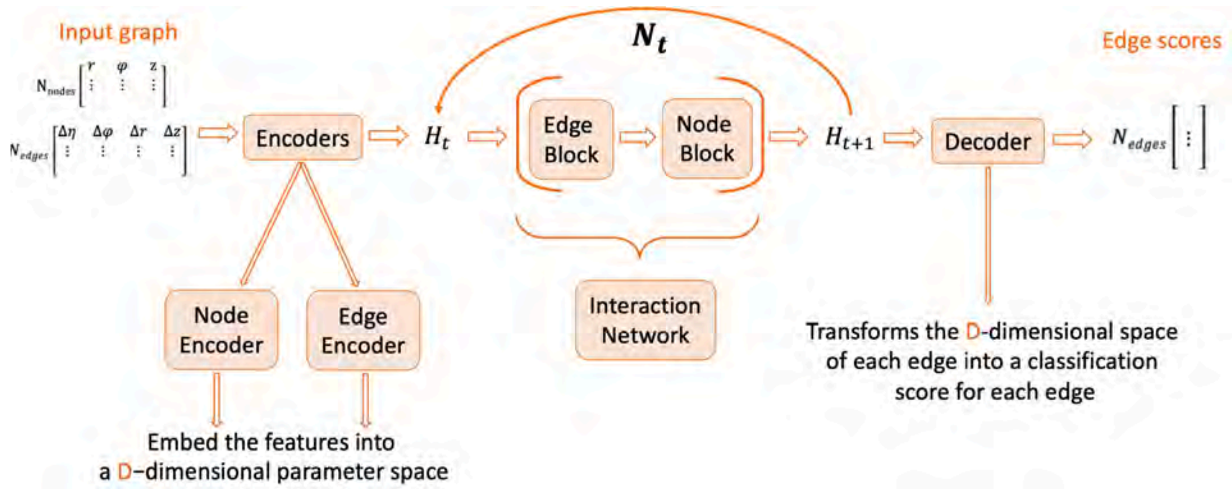
Event as a graph conception



https://iris-hep.org/projects/accel-gnn-tracking.html

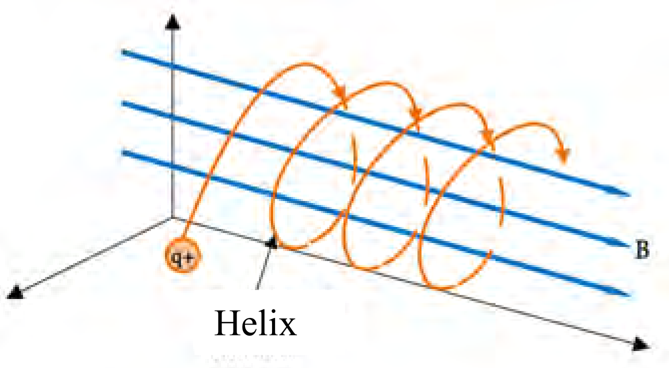
- Nodes are hits. The nodes between the adjacent stations may be connected by edges that are the possible segments of tracks
- Nodes are not connected within one layer
- Most of the edges will be pruned during preprocessing according to spatial features $(d\varphi, dz, w)$
- Another type of input is **reversed graph** – edges become nodes, and pairs of nodes become edges (segment of three hits will be one edge with two nodes in the new graph representation)

The goal of the GNN is to classify the edges of the graph and throw out the non-existing hit connections



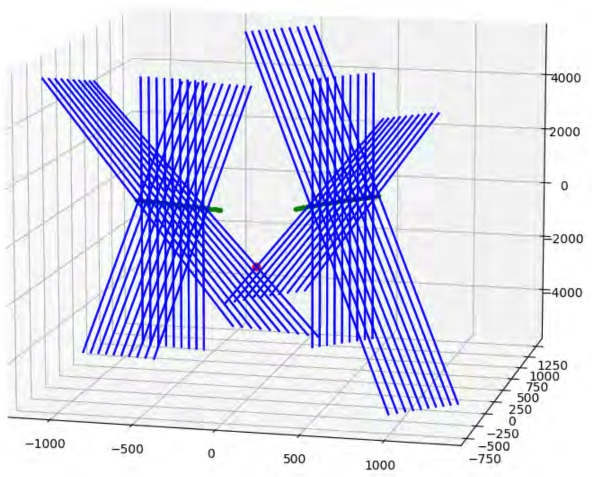
https://indico.ijclab.in2p3.fr/event/5999/contributions/25965/attachments/18789/25271/ltd_representation_summary.pdf

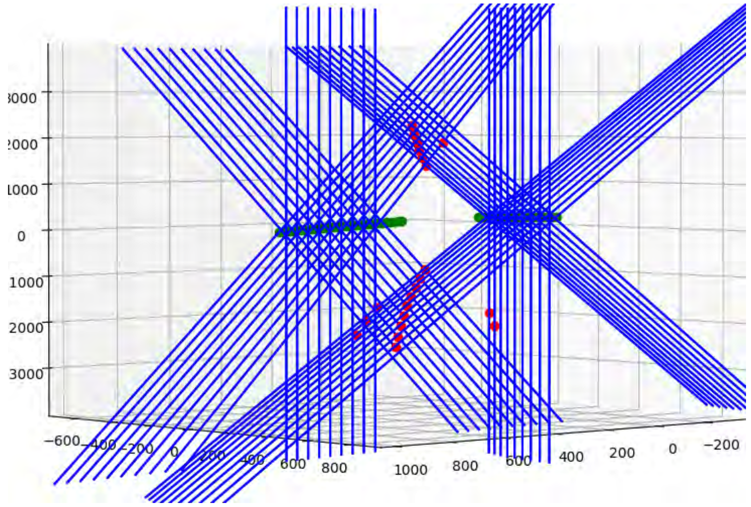
- Generator is a simple Python program
- Multiplicity in each event is given by a random number from 1 to 10
- The transverse momentum of a particle is a random number with a uniform distribution in the range of values from 100 to 1000 MeV/s
- Vertex coordinates are also random
- The particle trajectory is represented by a selection of points on a segment of a helix with a helix pitch $h = \frac{2\pi}{B}|\frac{m}{q}|vcos\alpha$ and radius $R = \frac{1}{B}|\frac{m}{q}|vsin\alpha$
- Detector inefficiency is also modelled (some hits may be missed)



Process of fakes generation

(two tracks in the event)





Dealing with Detector Inefficiency

- GNN makes prediction for the edges
- Apply classification threshold to drop out not interesting segments
- Collect all continuous track segments (full tracks and parts of tracks)
- Save track segments as pairs of nodes – node without input edge, node without output edge
- If
    - the station number of the 1st element of the j-th pair is 2 more than the station number of the 2nd element of the i-th pair
    - AND the corresponding coordinate differences do not exceed $2d\varphi_{max}$, $2dz_{max}$ and exceed $2d\varphi_{min}$, $2dz_{min}$ (criterion from the graph pruning)
- Connect such nodes and traverse the new graph

Testing Results (SPD)

Data for testing:

- 5000 generated events
- Detector inefficiency – 1%

Used metrics:

- $recall = \frac{N_{true}^{rec}}{N_{in}}$
- $precision = \frac{N_{true}^{rec}}{N^{rec}}$

N_{true}^{rec} - no. real tracks that the network found
N_{in} - no. all real tracks known from Monte-Carlo
N^{rec} - no. all reconstructed tracks

	No. generated fakes	Preprocessing time (event/sec.)	GNN inference time (event/sec.)	Track bulding procedure (event/sec.)
Ordinary graph	100	427.59	312.99	319.33
	1000	46.26	299.52	309.83
	10000	10.5	250.69	184.71
Reversed graph	100	261.52	208.94	248.86
	1000	32.61	211.57	214.43
	10000	12.51	166.64	134.59

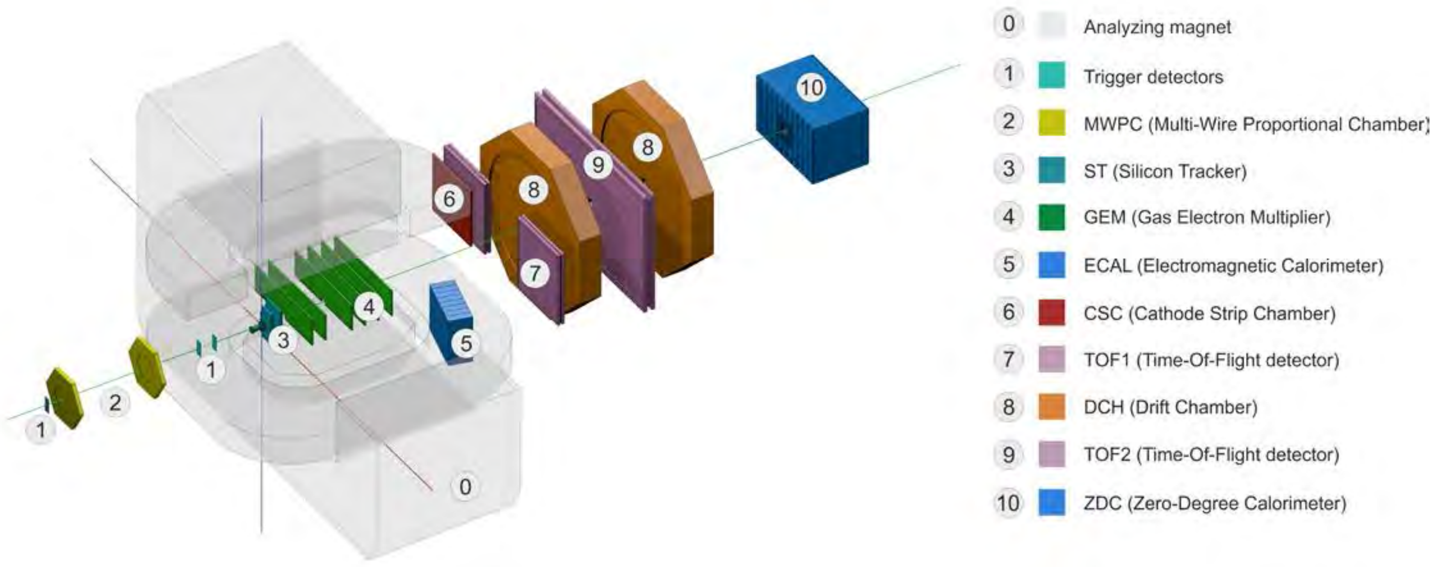
	Standard algorithm (no handling of missed hits)		Improved algorithm (dealing with inefficiency)	
	Ordinary graph	Reversed graph	Ordinary graph	Reversed graph
Recall	0.6767	0.6887	0.9104	**0.9113**
Precision	0.8582	**0.9543**	0.8324	0.9364

BM@N Experiment Run 7

Dataset

- Run 7 has a configuration - 6 GEM stations and 3 silicon stations before the GEM ones
- Monte-Carlo simulation using LAQGSM
- 1 000 000 events for training and validation
- Beam energy equal to 3,2 GeV
- Interactions with the argon beam and plumbum target (Ar+Pb)
- Magnet amperage was set to 1250 A
- Target distribution: X (mean: 0,7 cm, std: 0,33 cm), Y (mean: -3,7 cm, std: 0,33 cm), Z (center: -1,1 cm, thickness: 0,25 cm)
- Multiplicity of events varies up to 100 tracks per event, and 37 tracks per event on the average
- Number of hits can be more than 500 per station
- About 68% of all hits are fakes
- **Tracks containing less than 4 hits, spinning tracks and tracks with holes are removed from the dataset**

BM@N Run 7 Front View

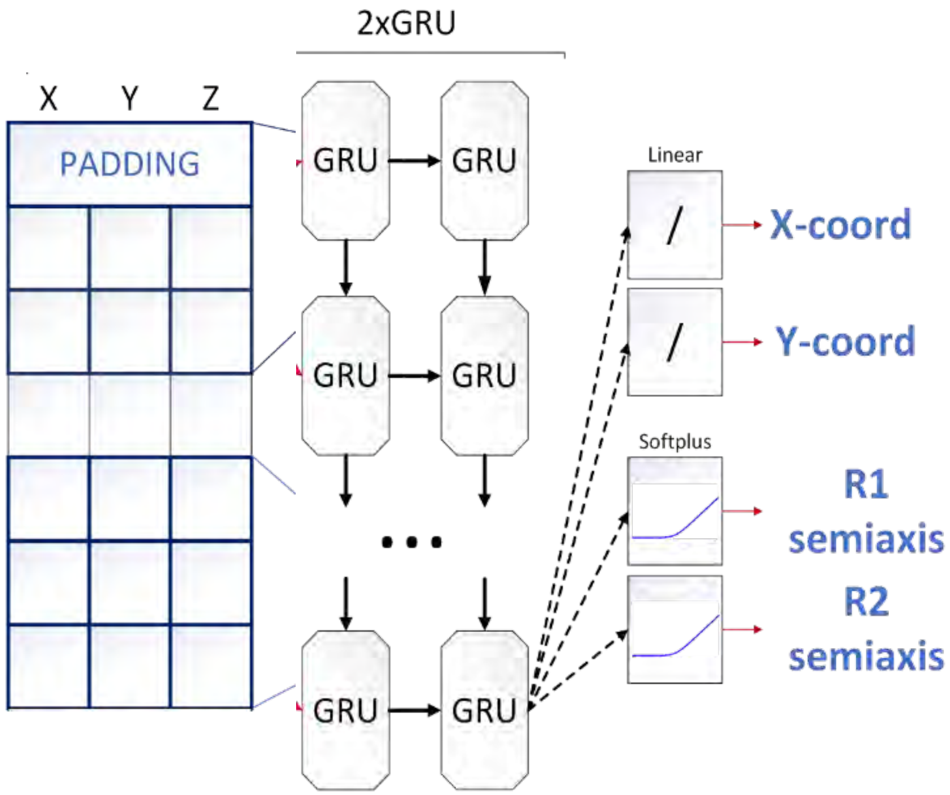


https://bmn.jinr.ru/gallery/

With such amount of hits and fakes ratio it is not very easy to fit into memory with the event graph!

Local Tracking with TrackNETv3

TrackNETv3 model



How the model works?

- TrackNETv3 is a model for local track reconstruction
- Locality – one particular track-candidate during the prediction phase
- The model predicts the center and semi-axes of the ellipse on the next coordinate plane where to search for the next hit
- All hits are placed in the spatial search index (Faiss)
- Only K nearest to the center of ellipse hits are checked (we set K=5)
- Candidate tracks are extended by hits that fall into ellipse
- Extended track-candidates are fed back to the model input

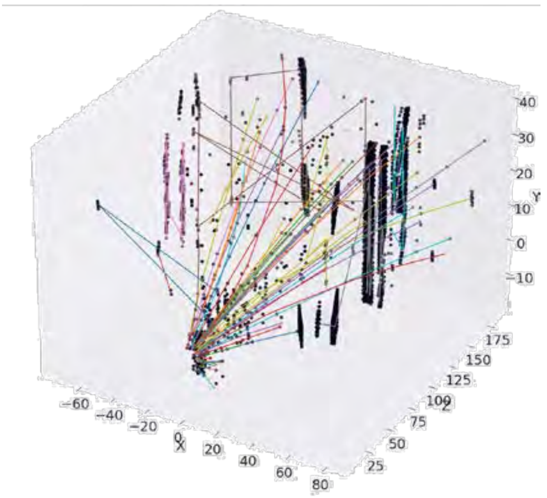
Pros:

- Fast
- Lightweight
- No problems with memory consumption
- Each track can be processed separately in parallel

Cons:

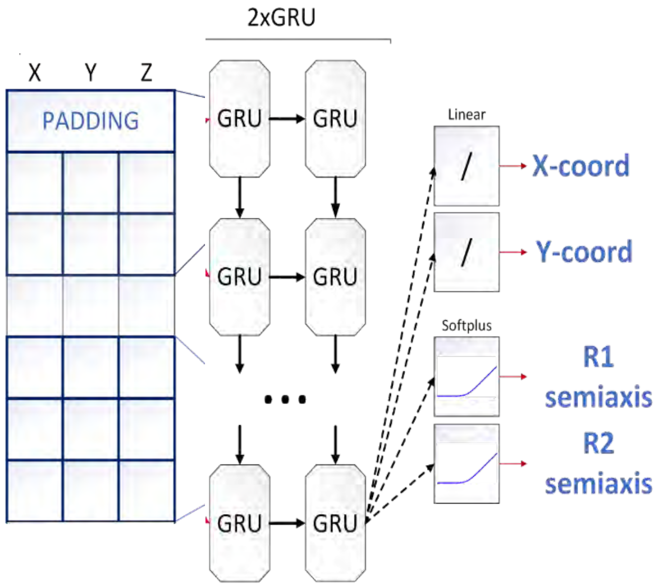
- lot of false positives or so-called ghosts, because of its local nature of prediction

Graph Neural Network as a Global Classifier



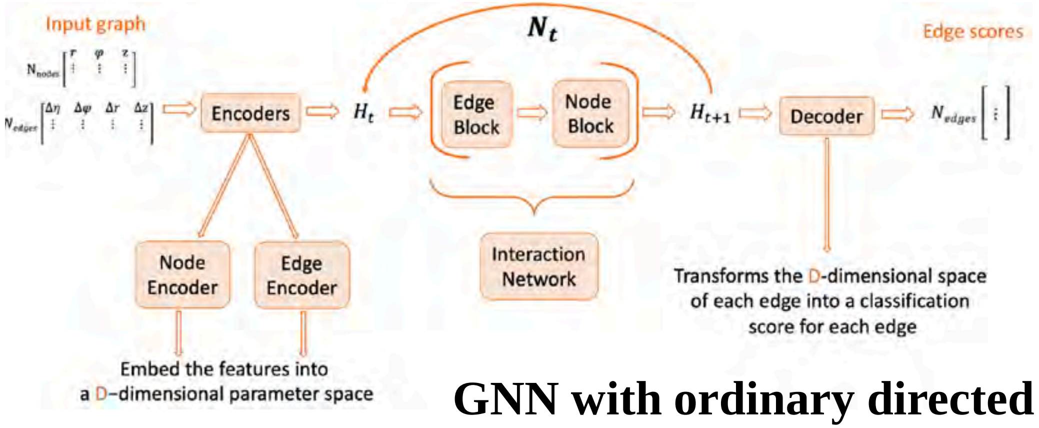
1) Raw event hits

TrackNETv3



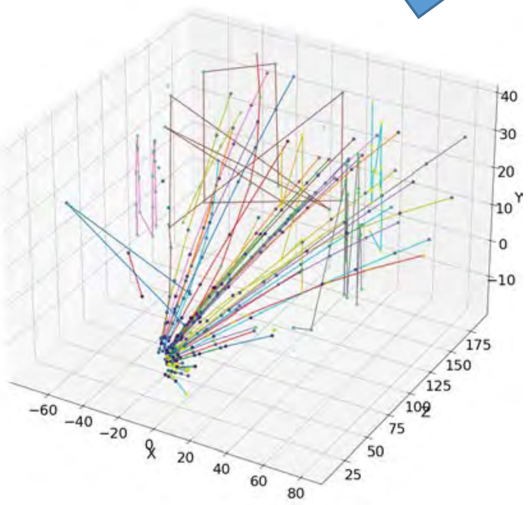
Reconstructed event

2) Graph from TrackNET predictions



GNN with ordinary directed graph

3) Pruned graph from GNN



4) Track building

Testing Results (BM@N Run 7)

Testing setup:
- 25 000 generated events
- Xeon(R) Gold 6148 CPU @ 2.40GHz
- Two types of GNN with different weights for positive and negative edges
- **No spinning tracks, tracks with less than 4 hits, tracks with missing hits**

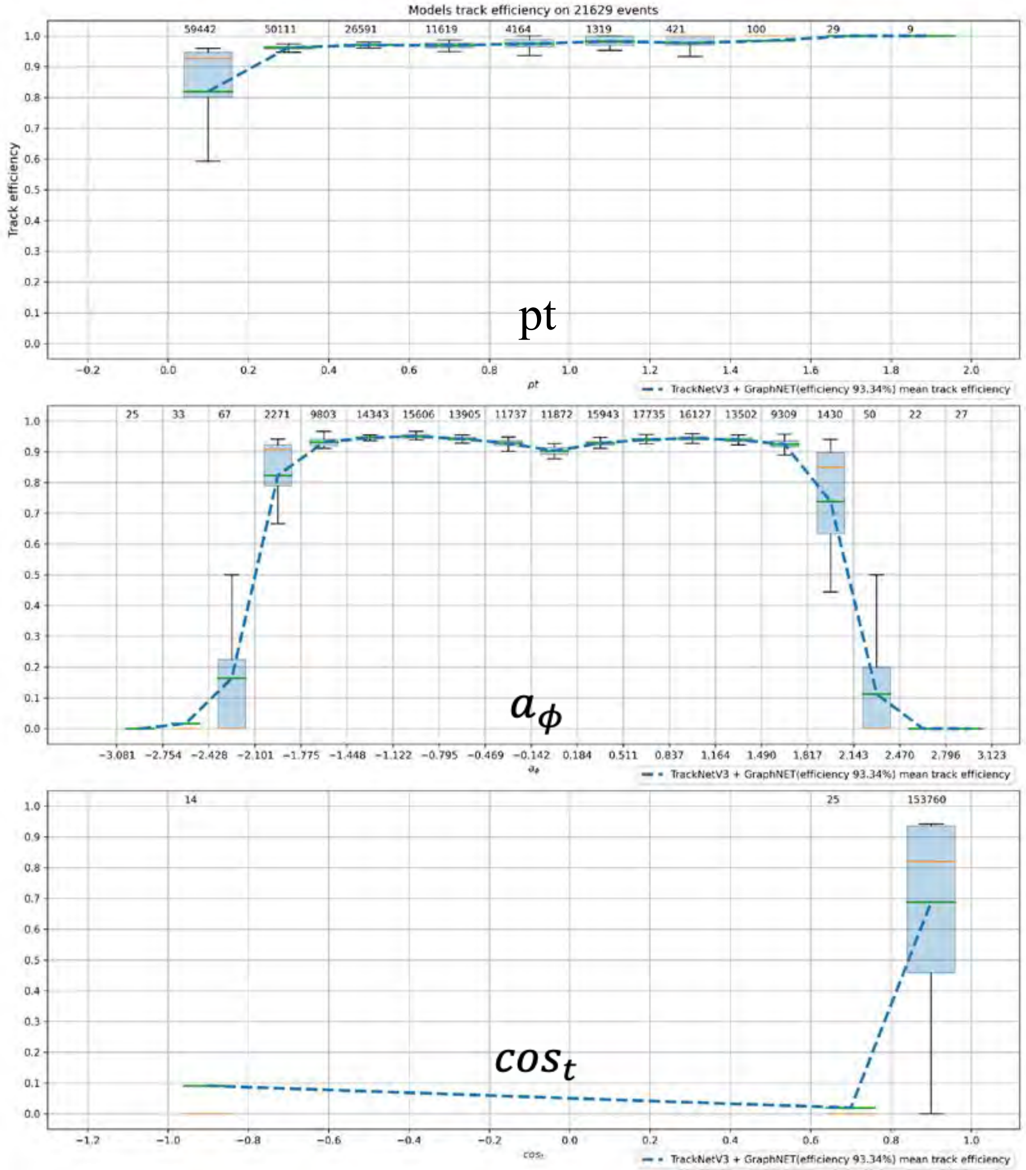
Used metrics:
- $recall = \frac{N_{true}^{rec}}{N_{in}}$
- $precision = \frac{N_{true}^{rec}}{N^{rec}}$

- N_{true}^{rec} - no. real tracks that the network found
- N_{in} - no. all real tracks known from Monte-Carlo
- N^{rec} - no. all reconstructed tracks
- Track is reconstructed if >=70% of hits are reconstructed

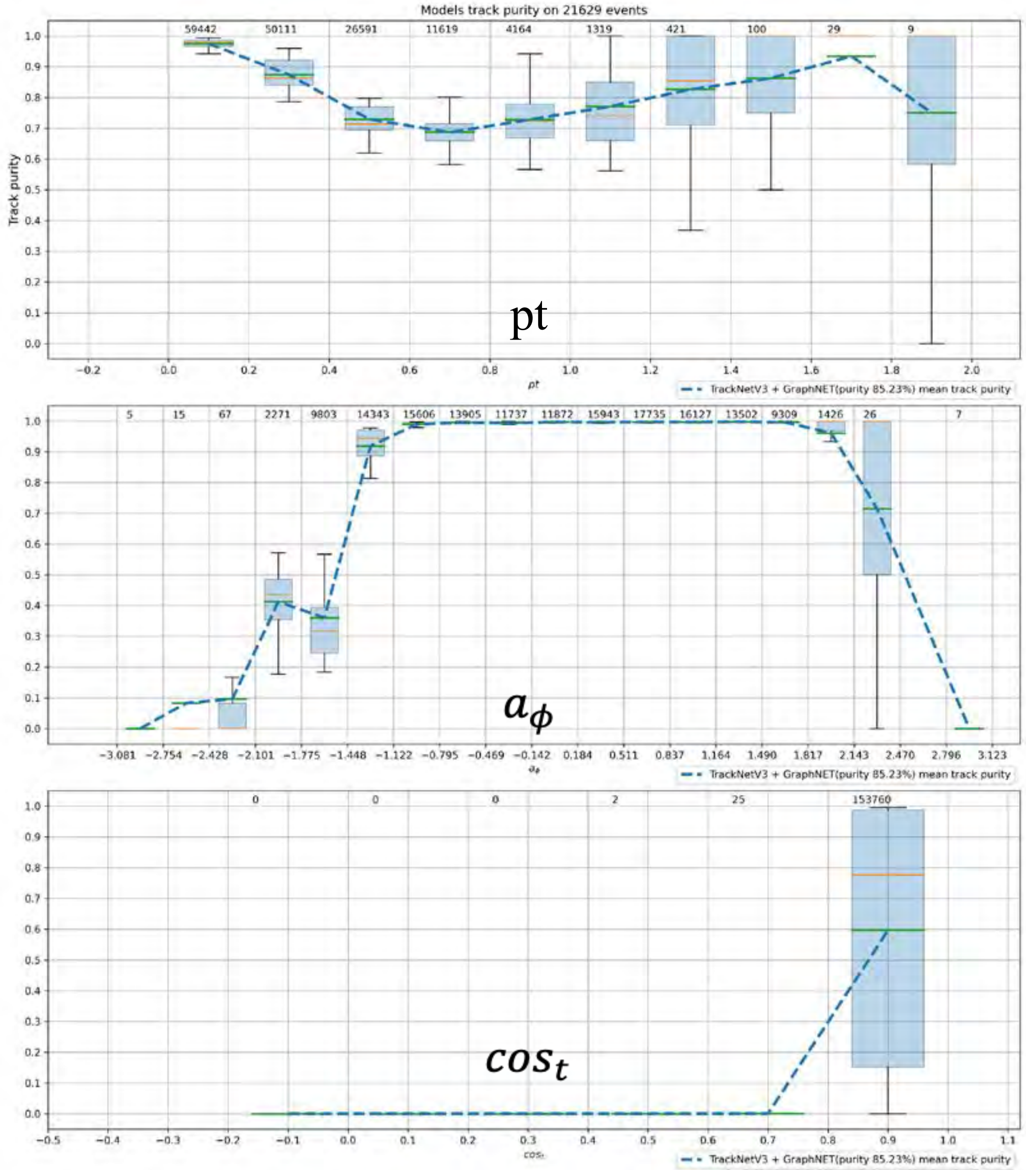
	TrackNETv3	TrackNETv3 + GraphNet (pos. weight = 24)	TrackNETv3 + GraphNet (pos. weight = 10)
Track efficiency (recall) (%)	97,67	94,89	93,34
Track purity (precision) (%)	2,35	74,98	85,23
Event/sec	12,78	7,56	7,56

Results analysis

Track Efficiency (recall)



Track Purity (precision)



Conclusion and Outlook

Discussion

- The current GNN model achieves great results on generated data for SPD, but still more sophisticated simulation is required.
- To increase precision of the models for BM@N we tried to fit each track-candidate by helix to filter out tracks with hi-square greater then a threshold. Attempts to use the helix fitting did not show any improvement.
- The results presented for BM@N were achieved for the event data with short tracks, tracks with holes and spinning tracks being excluded.
- Handling tracks with holes is a specific process which requires extra tricks. For example, during the TrackNETv3 inference we can decide where to place a virtual point instead of prolonging the candidate with the hits from the predicted ellipse (in progress).
- Tracks fitting with parameters calculation is required for better analysis of the results.

Acknowledgment

The calculations were carried out on the basis of the HybriLIT heterogeneous computing platform (LIT, JINR).
[G. Adam et al., CEUR Workshop Proc., Vol. 2267, 638-644 (2018).]

Deep neural network applications for particle tracking at the BM@N and SPD experiments

Goncharov Pavel, Rusov Daniil, Lubchenkov Leonid, Shchavelev Egor, Ososkov Gennady