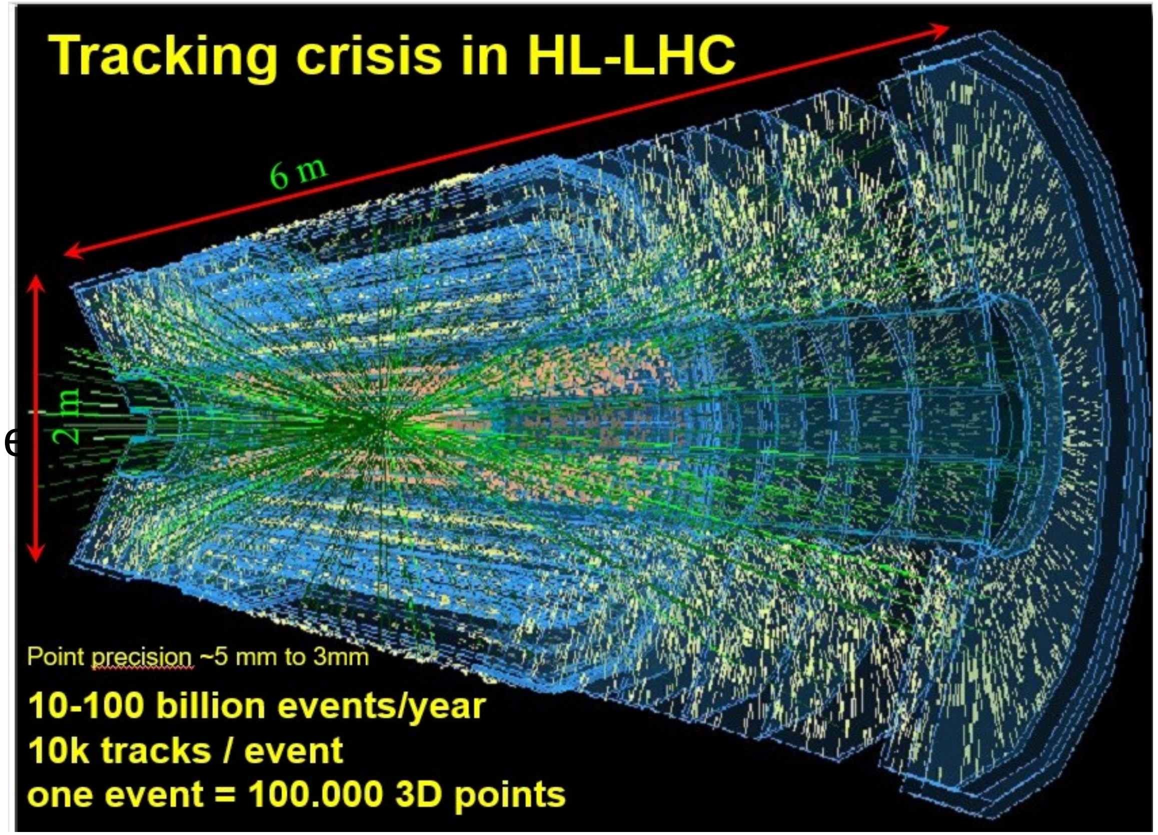
Deep tracking for the SPD experiment

Rusov D., Goncharov P., Ososkov G., Zhemchugov A.

Problem Statement

- **Particle track reconstruction in dense environments such** as the Run-4 detectors of the CERN High Luminosity Large Hadron Collider (HL-LHC) as well as MPD NICA is a **challenging pattern recognition problem**.
- To achieve such high luminosity, the particles are not accelerated individually, but in bunches, so that the moments of collisions occur so close to each other that the event tracks overlap strongly.
- For the SPD experiment, in which events are expected to arrive with a frequency of 3 MHz, the data acquisition is supposed to be performed in time slices, during one time slice up to 40 events with overlapping tracks may appear.
- Of the entire stream of events, only a few percent are of interest to physicists.



- Therefore, it is **necessary to develop an intelligent online filter to sift out uninteresting events.**

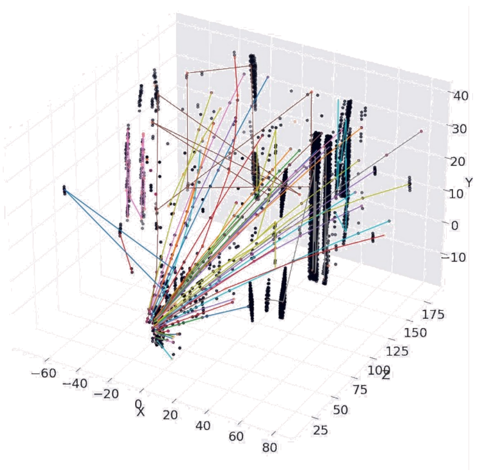
Deep Learning Comes to the Rescue

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?

Reconstructed event

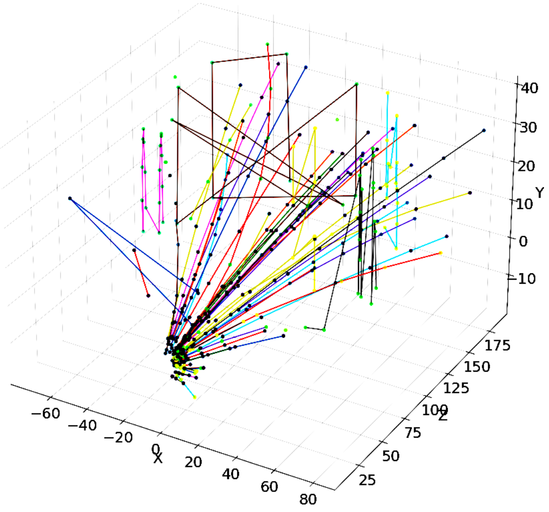


Neural network (online)

High recall, low precision
Very fast

Kalman filter (offline)

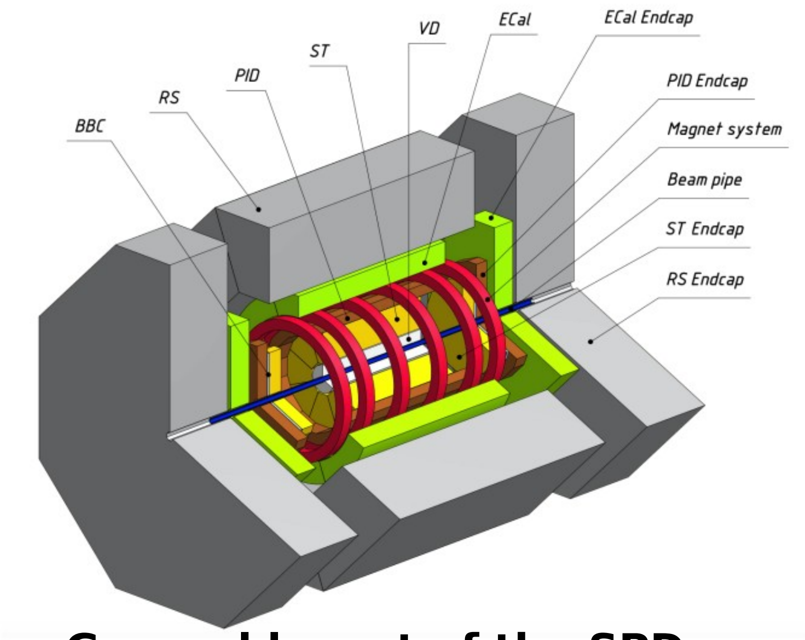
High recall, high precision
Too slow



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).
Event data from SPD comes in the form of time slices with a length of 10 ms and about 40 events in time slice will be produced.
On average there are **200 tracks per time slice and 1100 hits (real and fake) per station**.

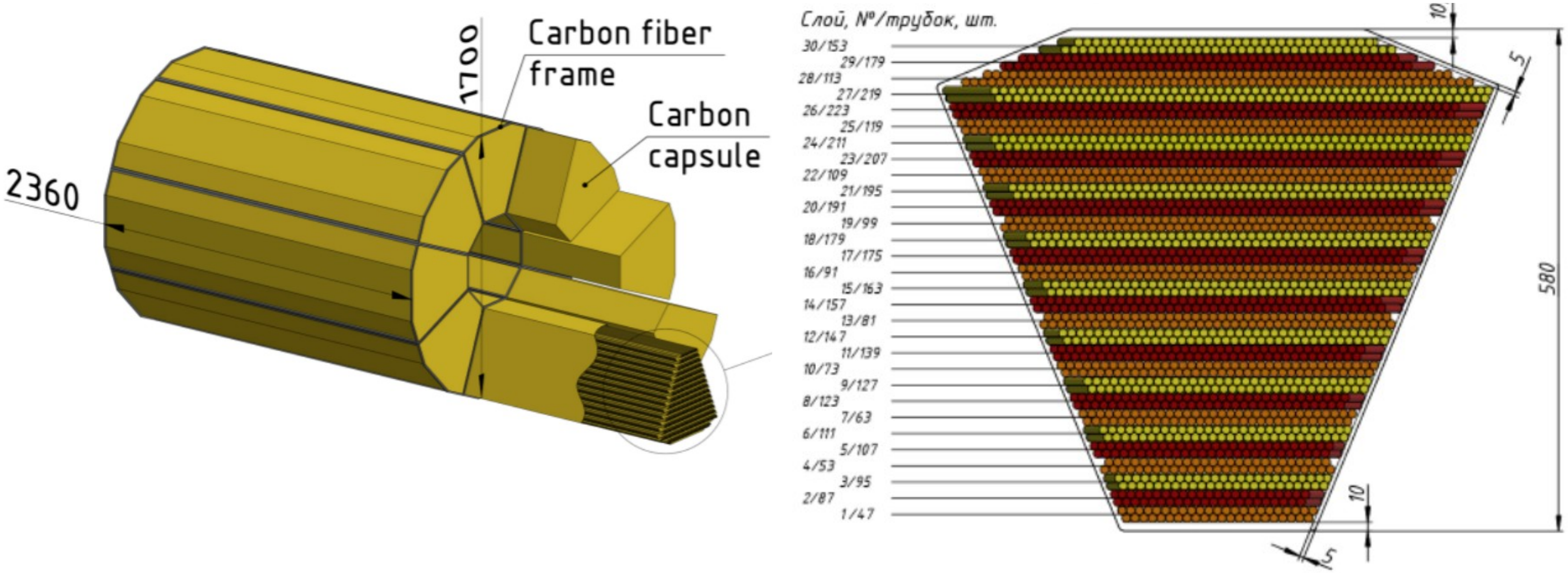
As an online filter, the algorithm must allow processing over 100 time slices per second.



General layout of the SPD setup

Layer, № / tubes, pcs

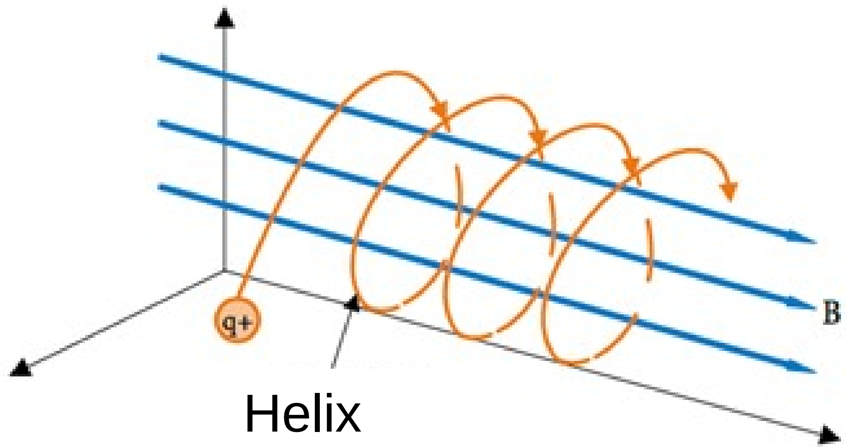
(x2 zoom)
Without cover



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

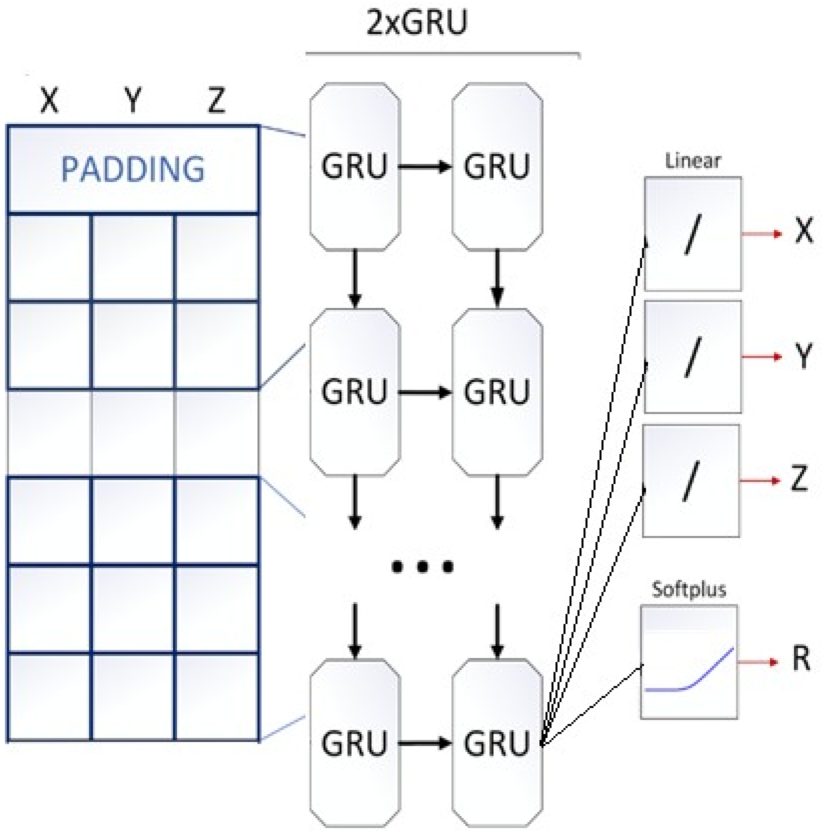
Events Data Generator

- Generator is a simple 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.
- The particle trajectory is represented by a selection of points on a segment of a helix with a helix pitch and radius .
- Detector configuration with 35 stations is considered.



Local Tracking with TrackNETv3. Model overview

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 radius of the sphere where to search for the next hit.
- All hits are placed in the spatial search index (Faiss).
- 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.

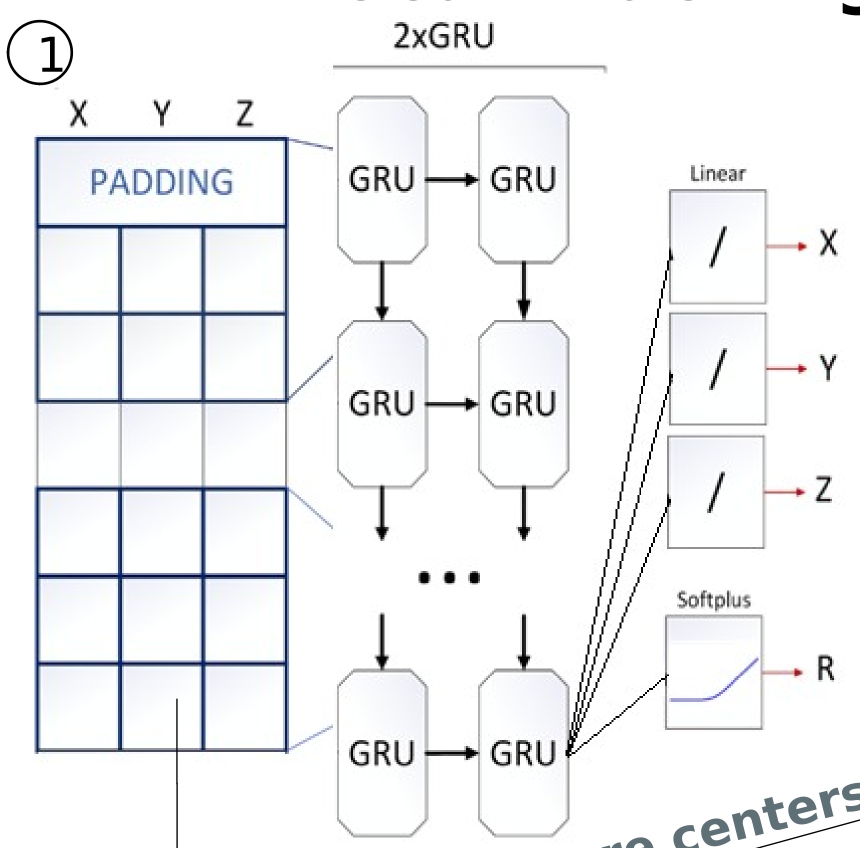
Pros:

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

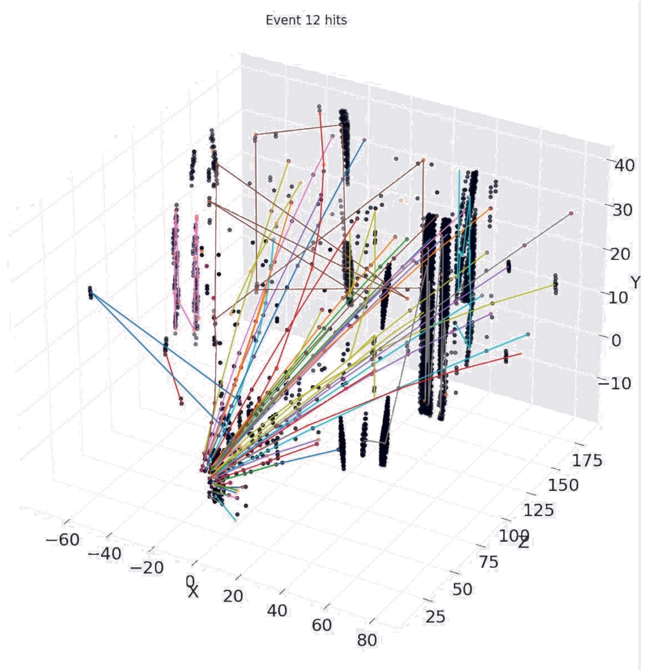
Cons:

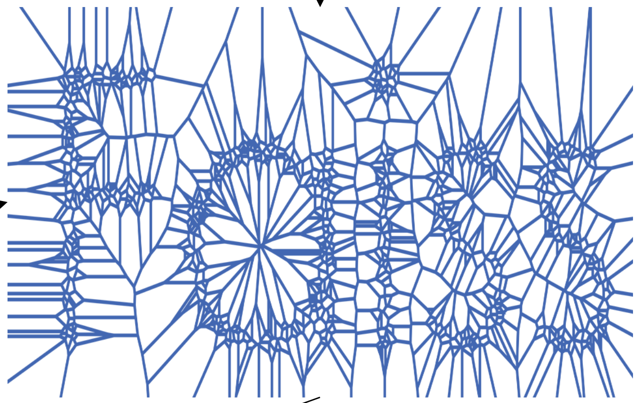
- lot of false positives or so-called ghosts, because of its local nature of prediction

Local Tracking with TrackNETv3. Inference



2 Store all event hits in the index





4 Sphere centers (x, y, z)

3 Sphere predictions (x, y, z, r)

1 nearest hits for each sphere center

5 Check sphere attendance

6 Prolong ca and pass them to the model

As a result

- complexity depends on number of stations
- Start predictions from the 1st hit – much less ghosts

Testing Results. Metrics

Testing setup:

- 25 000 generated events
- Xeon(R) Gold 6148 CPU @ 2.40GHz
- NVIDIA Tesla V100 32GB
- **No tracks with less than 4 hits and tracks with missing hits**

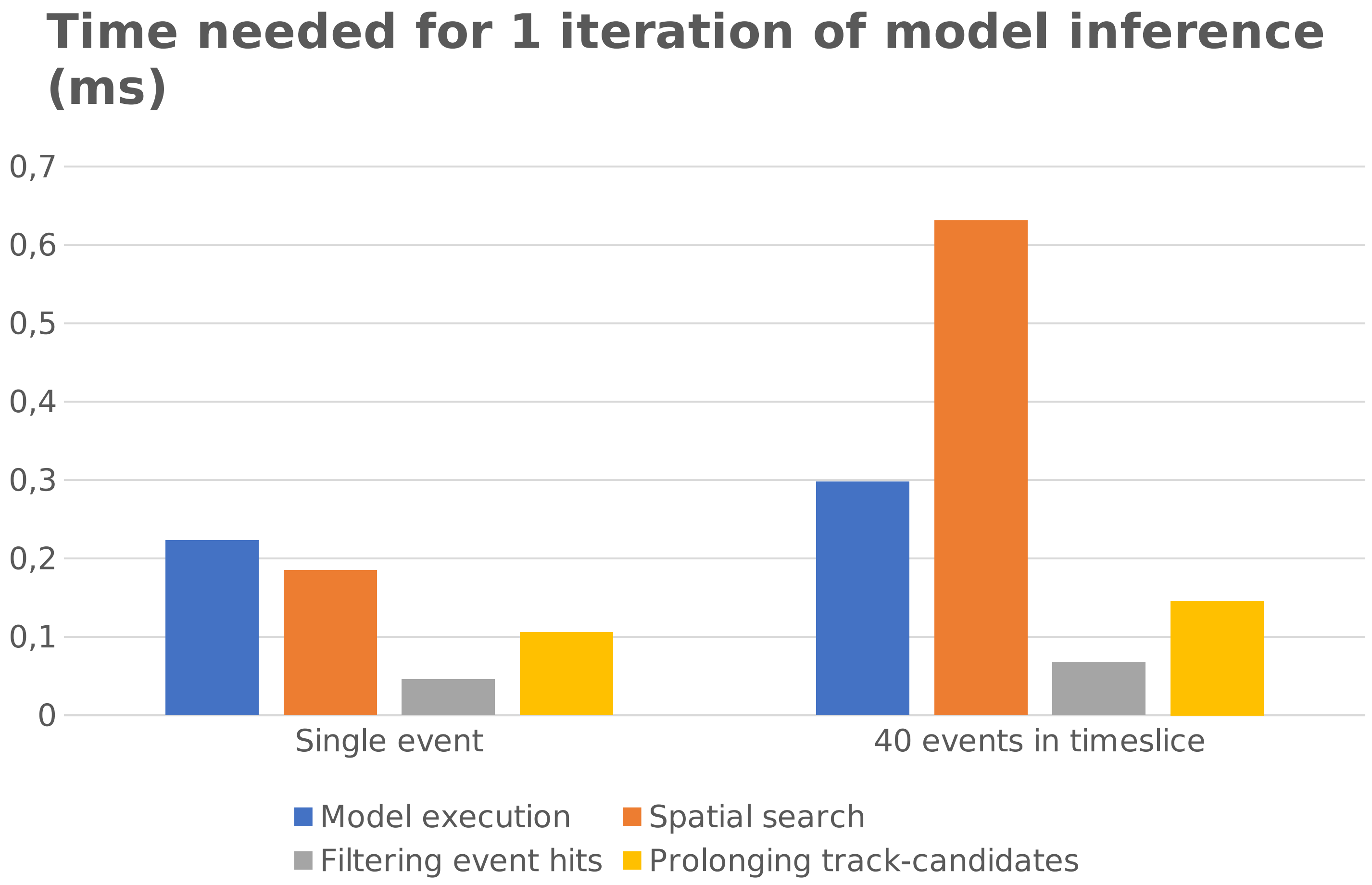
Used metrics:

- \- no. real tracks that the network found
- \- no. all real tracks known from Monte-Carlo
- \- no. all reconstructed tracks

	TrackNETv3 (single event)	TrackNETv3 (40 events in timeslice)
Track efficiency (recall) (%)	99,62	96,78
Track purity (precision) (%)	99,52	88,02
Timeslice/sec		18,46

Testing Results. Time measurement

- **Brute force algorithm for spatial search was used**
- As the number of points in events increases, the time needed to perform a spatial search increases drastically
- Some other algorithms, like Inverted File Index, may improve search speed but require preparation time, so can’t bring any speed-up.



Conclusion and outlook

- Applying the TrackNETv3 model to SPD simulation data showed promising results in terms of track efficiency and purity.

- Research on handling the tracks with missing hits is needed for SPD data (the TrackNETv3 model shown good results for processing such tracks for the BM@N experiment).

- Inference optimization is needed to bring processing speed closer to required level. It can be done by translation the program to C++ or changing the inference algorithm, for example, skipping some stations.

- More complex simulation of SPD events is required.

Deep tracking for the SPD experiment

Rusov D., Goncharov P., Ososkov G., Zhemchugov A.