Neural network based tracking for BM@N and BES-III experiments

Egor Shchavelev[1]

Olga Bakina[2], Dmitriy Baranov[2], Igor Denisenko[2], Pavel Goncharov[2], Yury Nefedov[2], Anastasia Nikolskaya[1], Gennady Ososkov[2], Ekaterina Rezvaya[2], Aleksey Zhemchugov[2]

[1]Saint Petersburg State University
[2]JINR

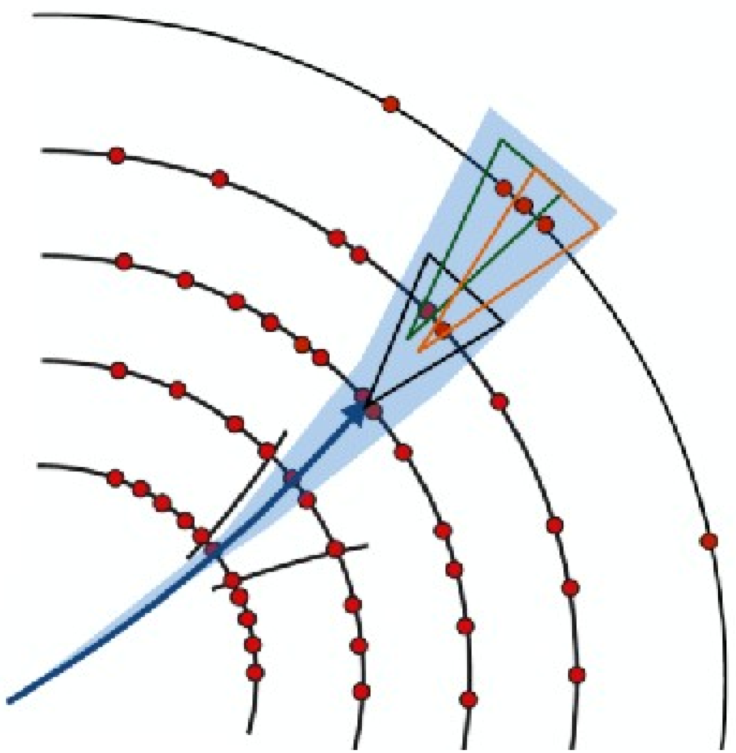
SPD collaboration meeting
June 9, 2021

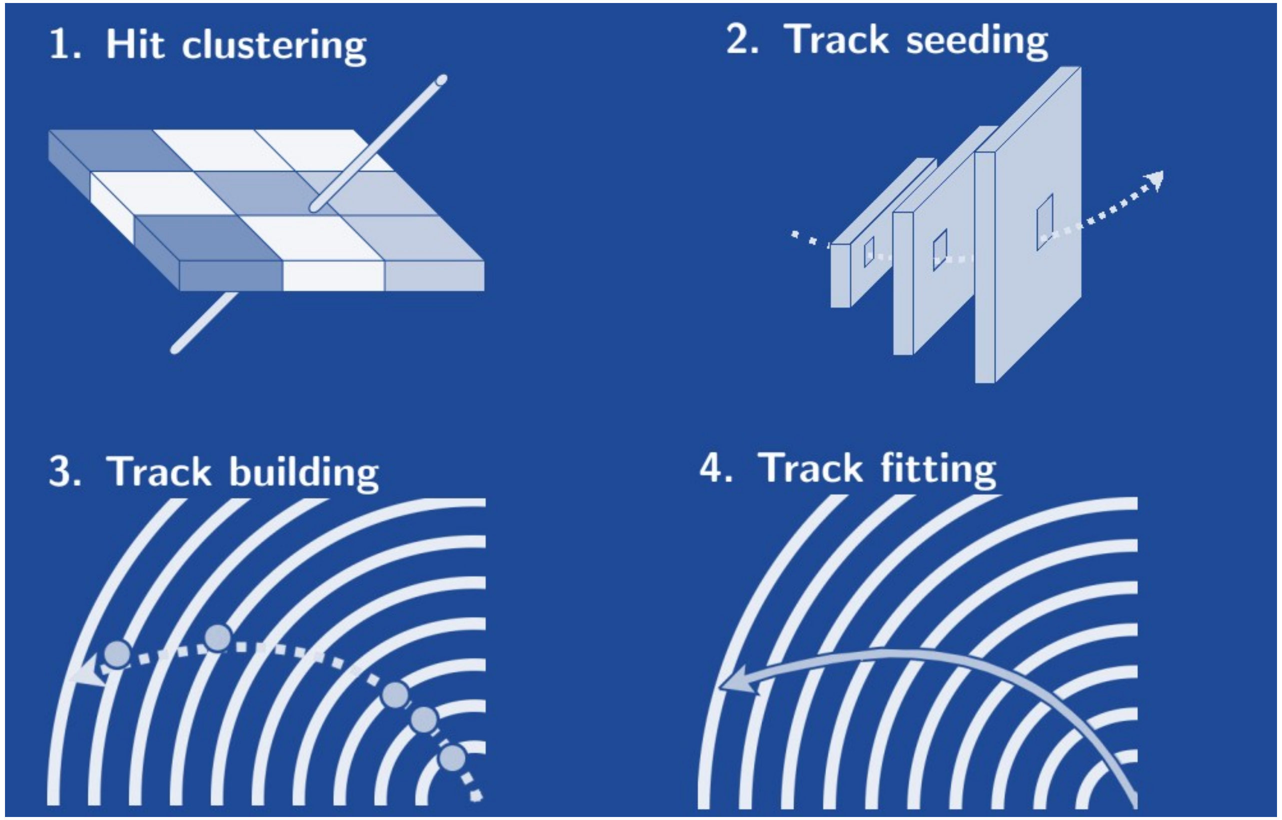
Plan

- What is tracking
- BM@N and BES-III experiments
- Why neural networks
- Local approach
- Global approach
- Results
- Conclusion and outlook
- Possible scheme of neural-based online filter

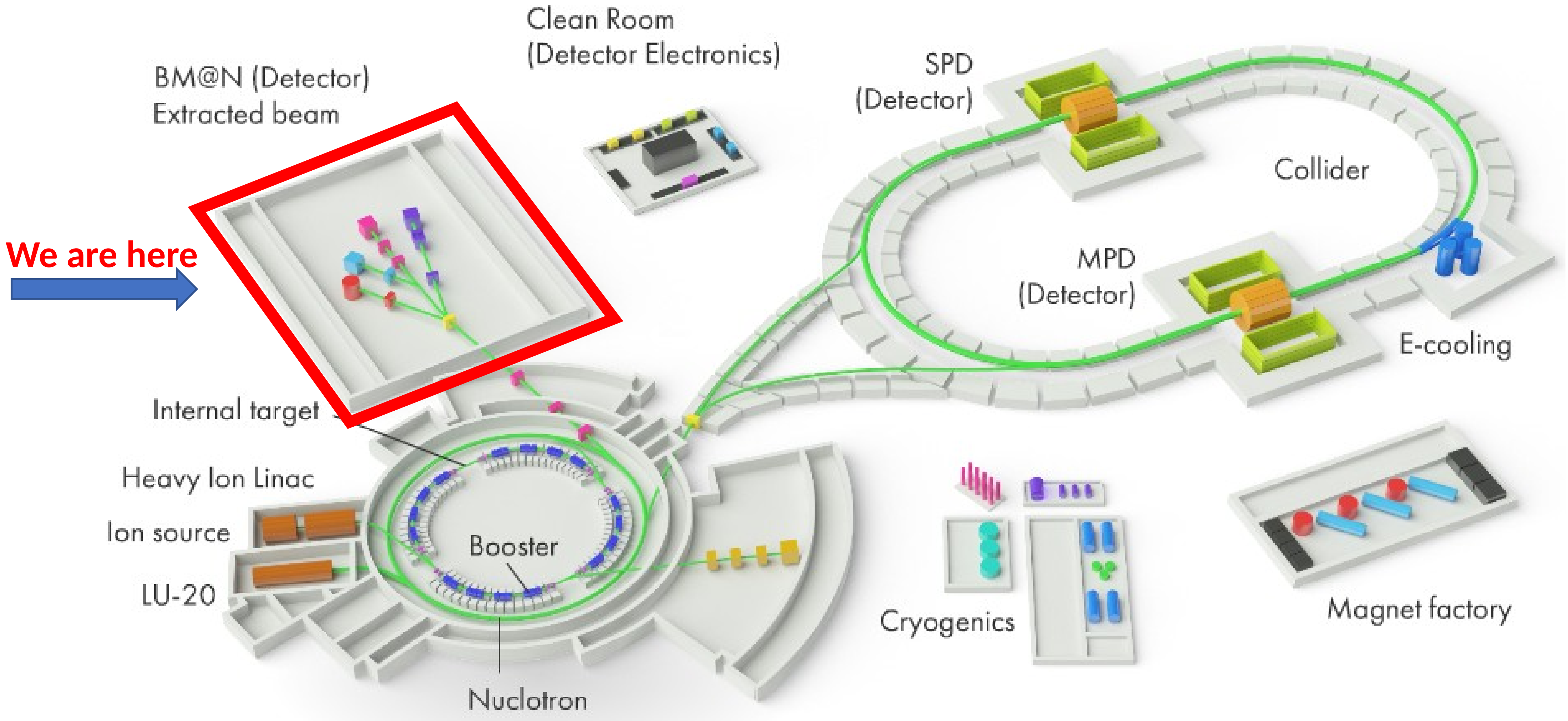
What is tracking?

Tracking or track finding is a process of **reconstruction the particle's trajectories** in high-energy physics detector by connecting the points – hits – that each particle leaves passing through detector's planes. Tracking includes track seeding and track building phases.



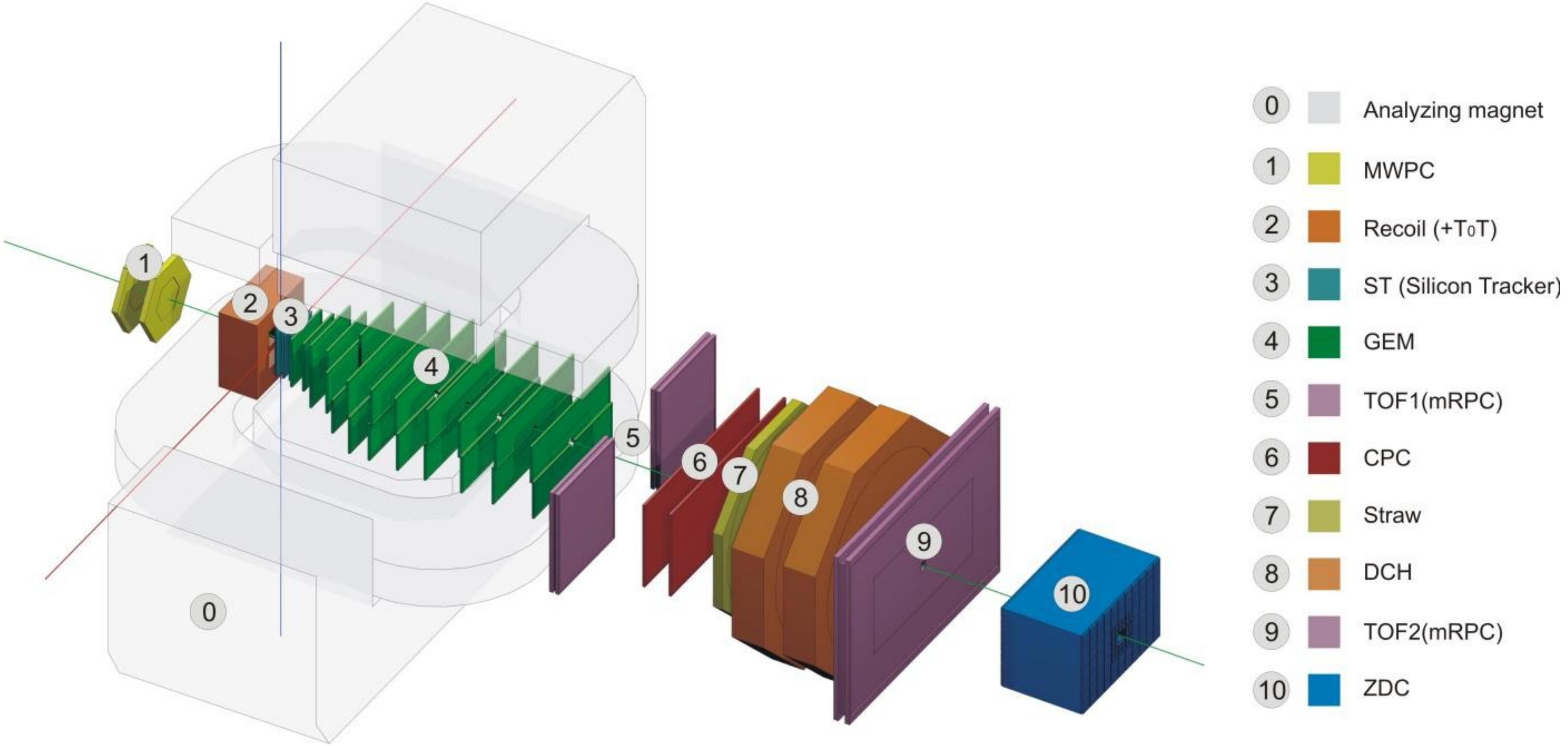


Tracking for BM@N



General view of the NICA complex with the experiments MPD, SPD, BM@N

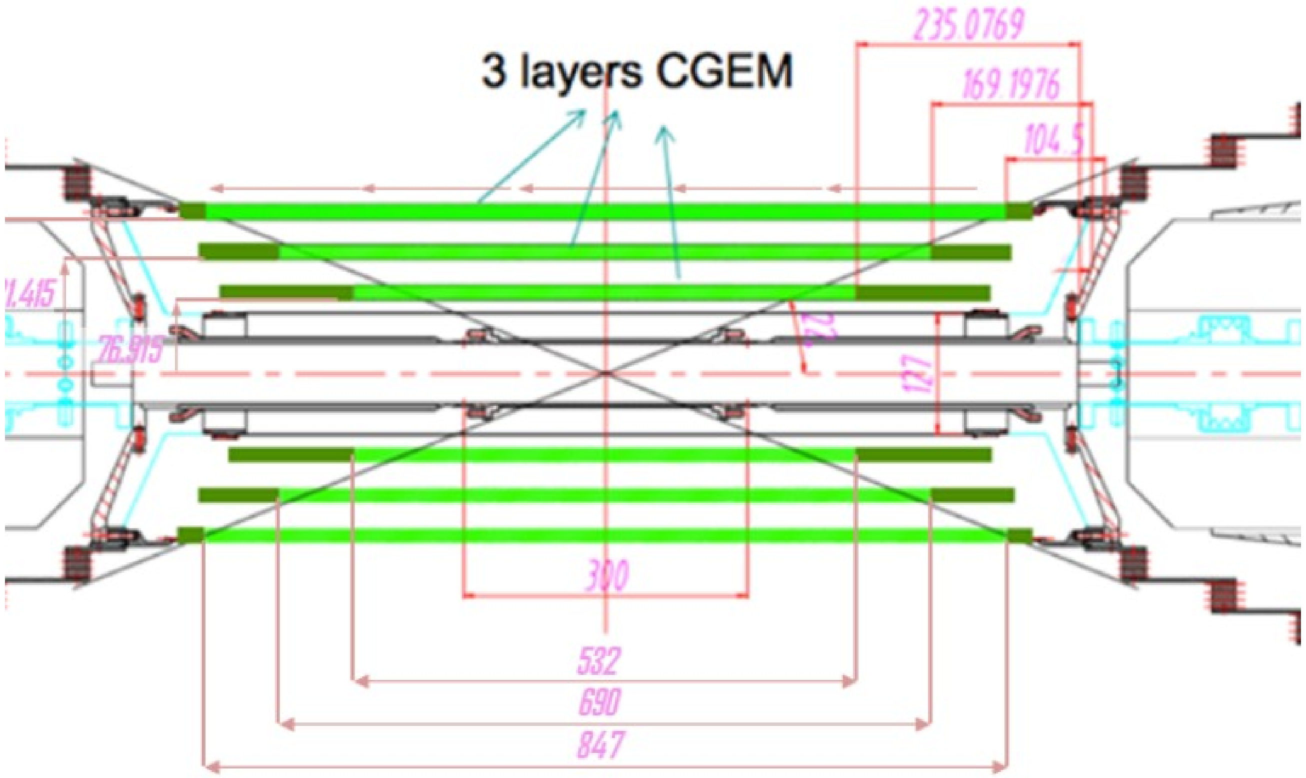
Baryonic Matter at Nuclotron (BM@N)



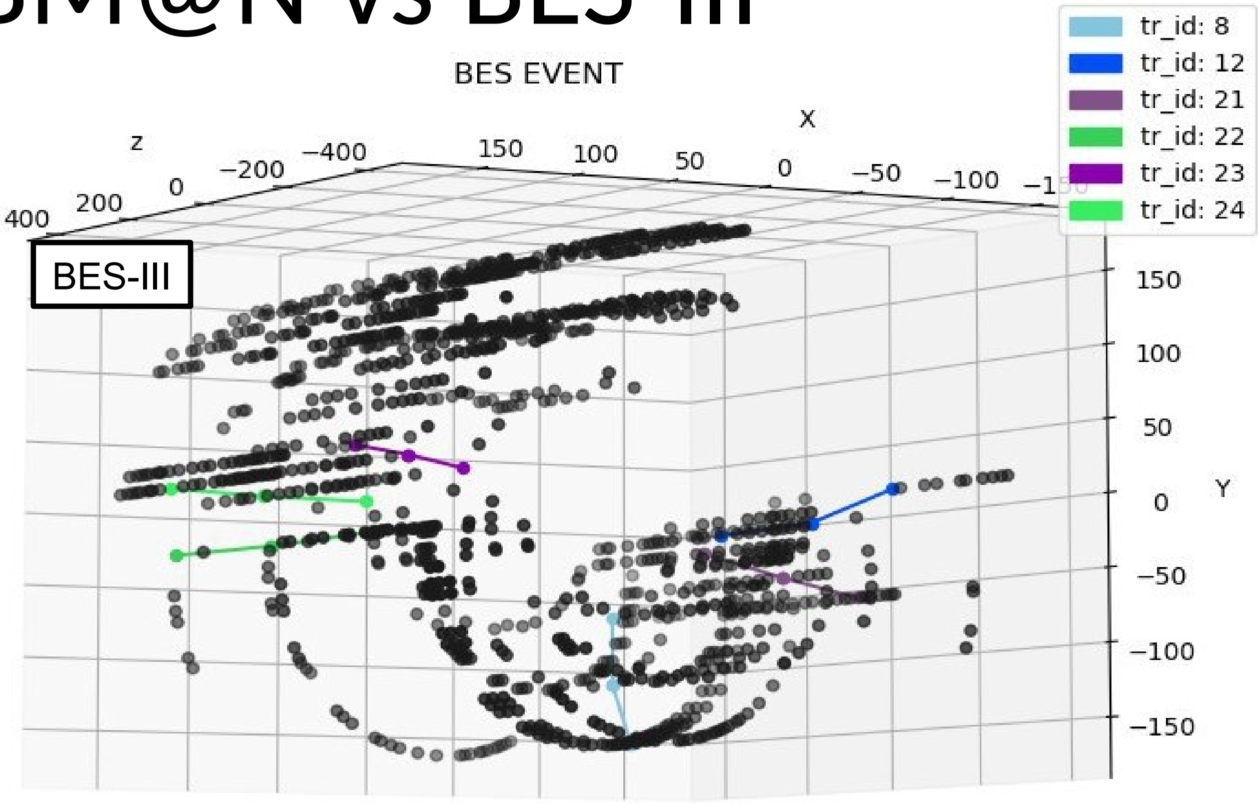
- Our problem is to reconstruct tracks registered by the GEM vertex detector with 6 GEM-stations (RUN 6, spring 2017) inside the magnet.
- All data was simulated in the BmnRoot framework with LAQGSM generator.

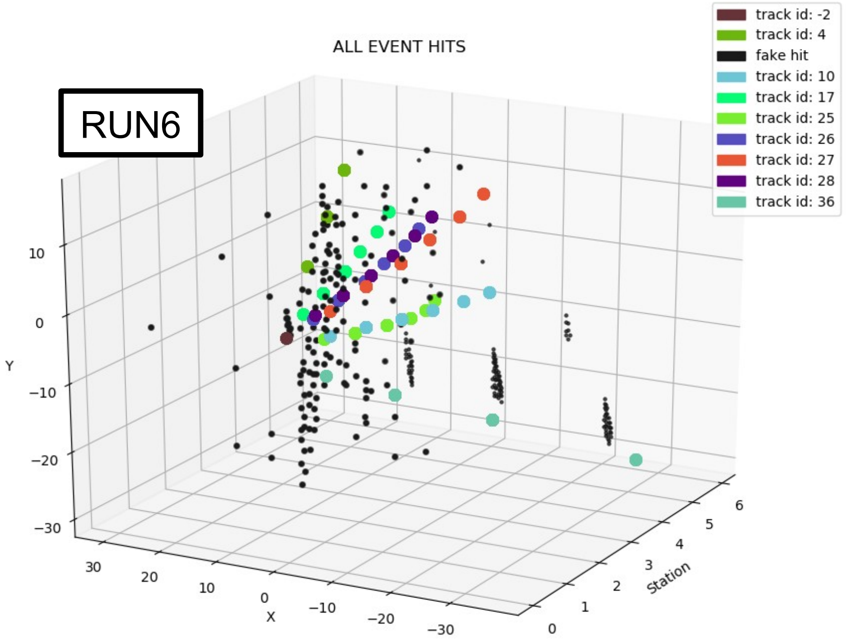
Tracking for BES-III

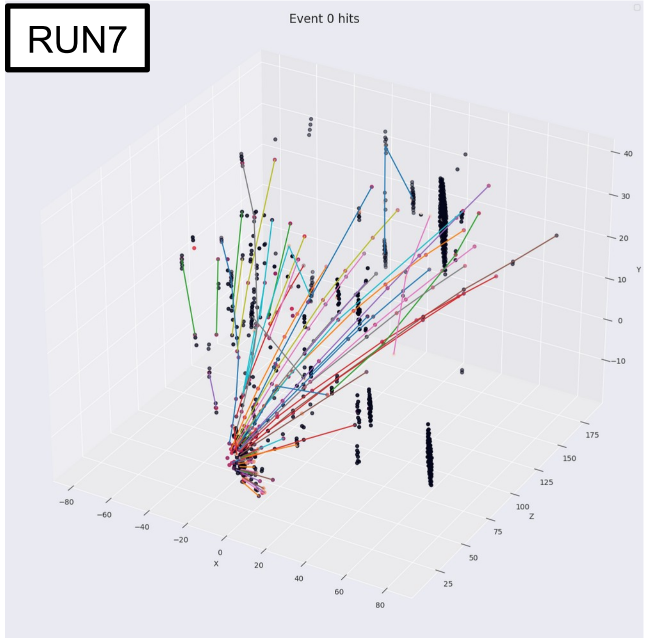
- Inner tracking detector of the BES-III collider experiment in Beijing IHEP.
- Consists of 3 cylindrical layers
- BES-III CGEM Monte-Carlo simulation was used followed by a basic clustering algorithm to reconstruct CGEM hits



BM@N vs BES-III







- Similarities:
  - Both detectors are strip-based GEM detectors
  - A lot of “fake” hits (about O(N^2) of real hits)
  - Comparable events multiplicities (avg. <15 tracks per event)
- Differences:
  - Number of stations: 3 for BES-III and 6 for BM@N (run6 configuration)
  - Environment: collider for BES-III and fixed-target for BM@N

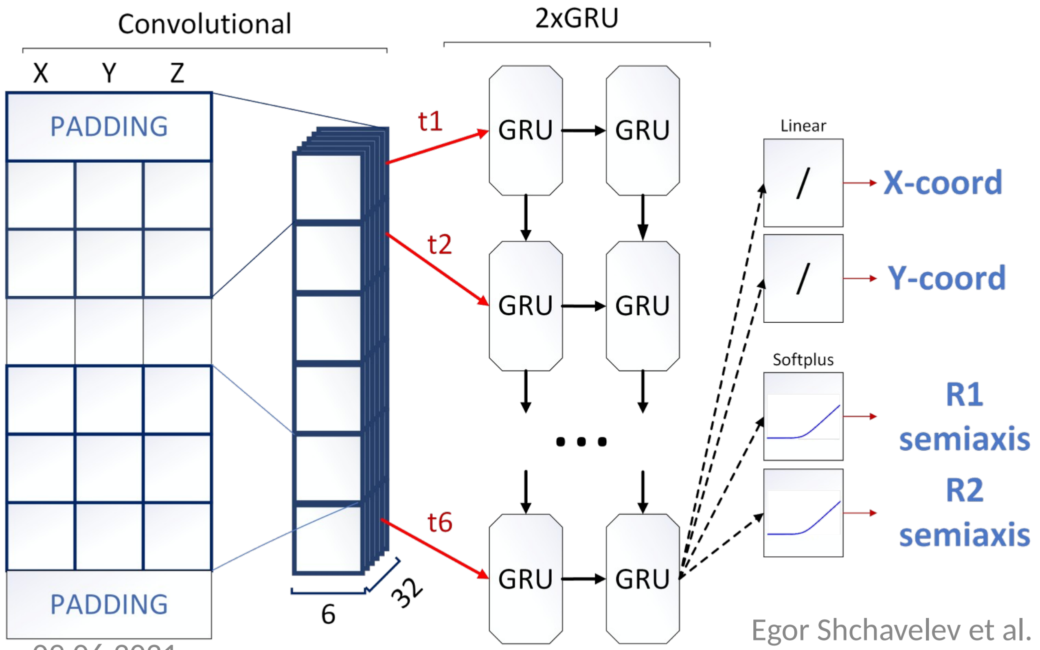
Why neural networks?

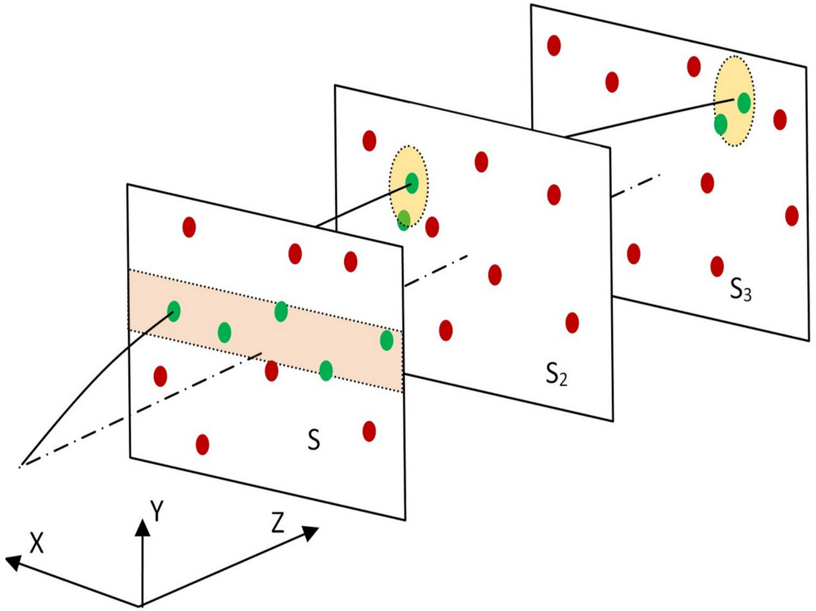
- Modern experiments produce vast amount of data
- Kalman filter (KF) based tracking approaches can not handle such giant data volumes in a reasonable time
- KF algorithms are local, sequentially following tracks station-by-station, therefore they cannot assess the global picture of an event, to see the dependence between individual tracks or groups of tracks and have exponentially growing complexity with increasing the event multiplicity

- Modern Neural Network (NN) based approaches are coming to rescue - they can be easily run in parallel on GPU achieving comparable results in terms of purity and efficiency

Local approach: TrackNETv2

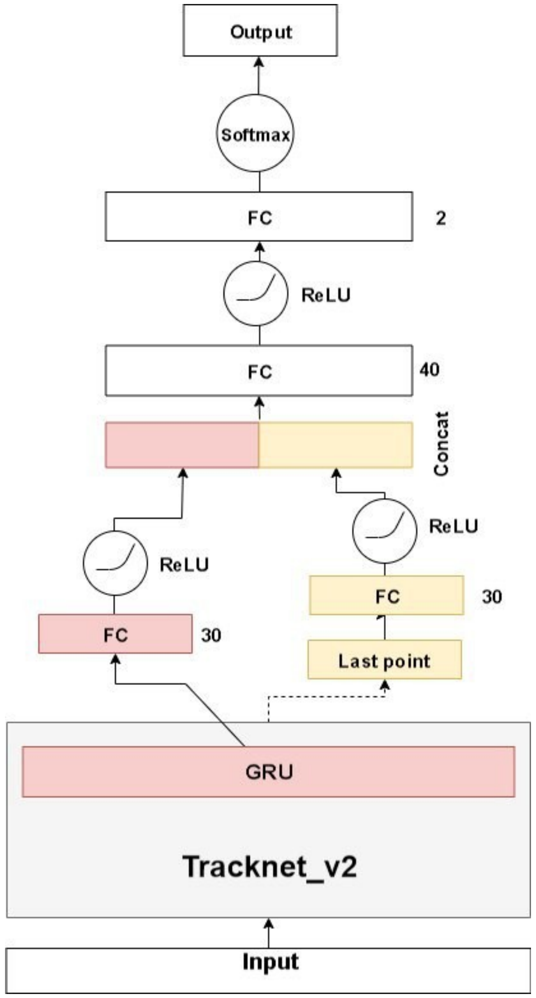
- Input of the network is coordinates of track-candidate points
- The goal of TrackNETv2 is to **predict the center of ellipse on the next coordinate plane**, where to search for track-candidate continuation and **predict the semiaxis of that ellipse.**
- If there is a hit inside such ellipse, we can run the algorithm again to predict an ellipse on the next coordinate plane and so on
- Can be treated as Kalman Filter "analogue" powered by neural networks





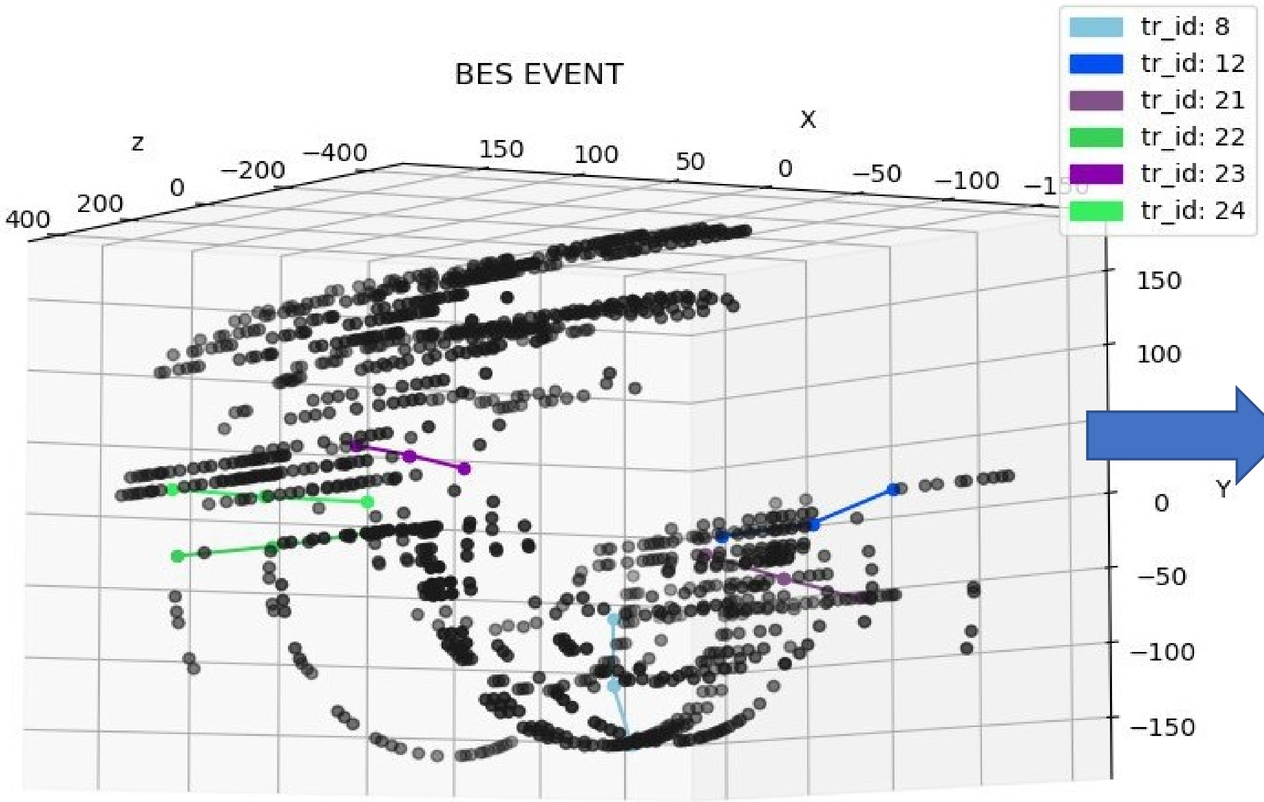
Local approach: TrackNETv3

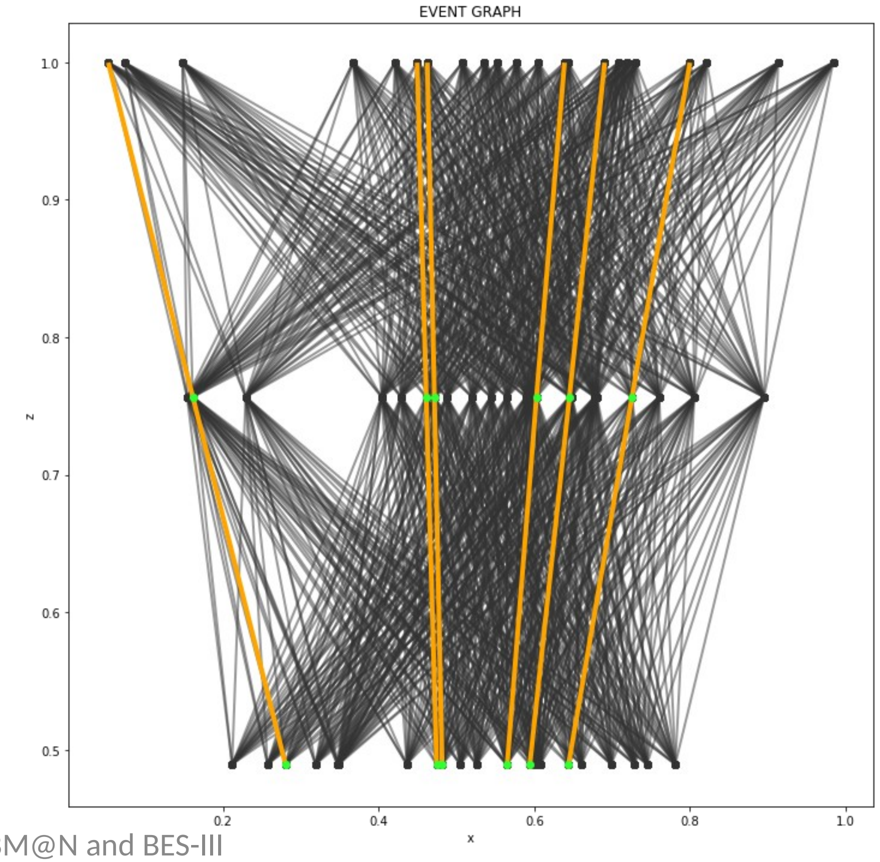
- New block on top of TrackNETv2 to classify track-candidate is real or not.
- To train classifier, following procedure is proposed:
  - For each ellipse, predicted by TrackNETv2 on third and further stations:
    1. If ellipse is empty and doesn't intersect station, track-candidate is added to tracks list
    2. If this candidate is real one, it is labeled as positive, else negative
    3. If real track is not found with this procedure, it is added to positive tracks too



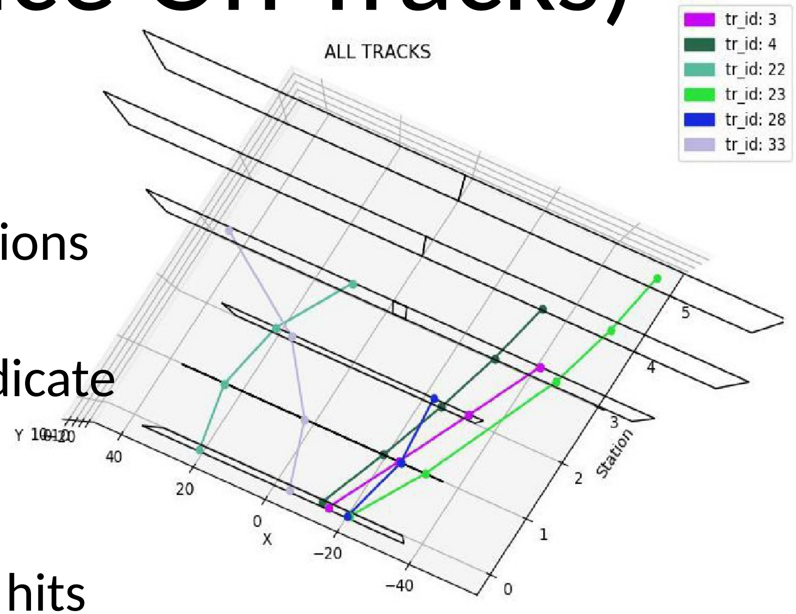
Global approach: GraphNet

- Treats a single event as a graph. Nodes of the events are hits. Hits are fully-connected between adjacent stations.
- Hits features (X, Y, Z) are being propagated inside the Graph Neural Network (GNN) through their connections to the other hits
- Special preprocessing is needed for each event to prepare it for GNN
- Goal of the network is to predict segments which belong to real tracks

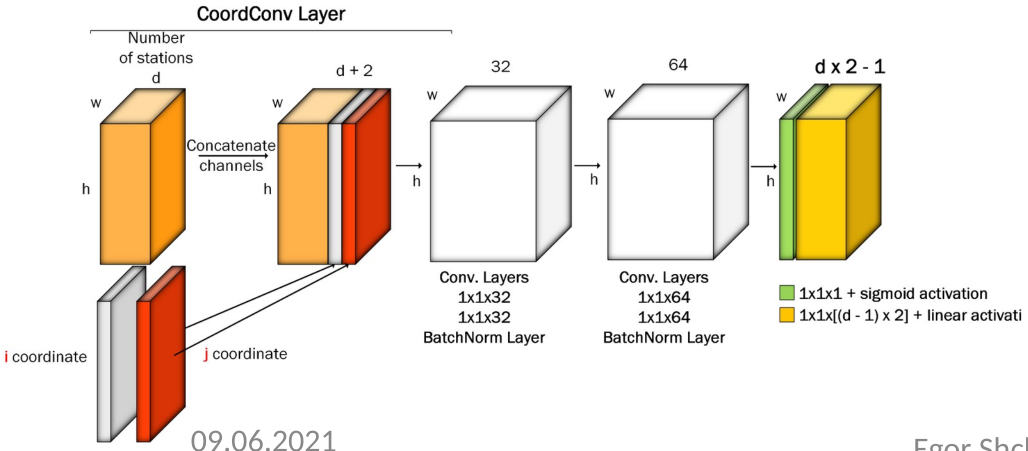


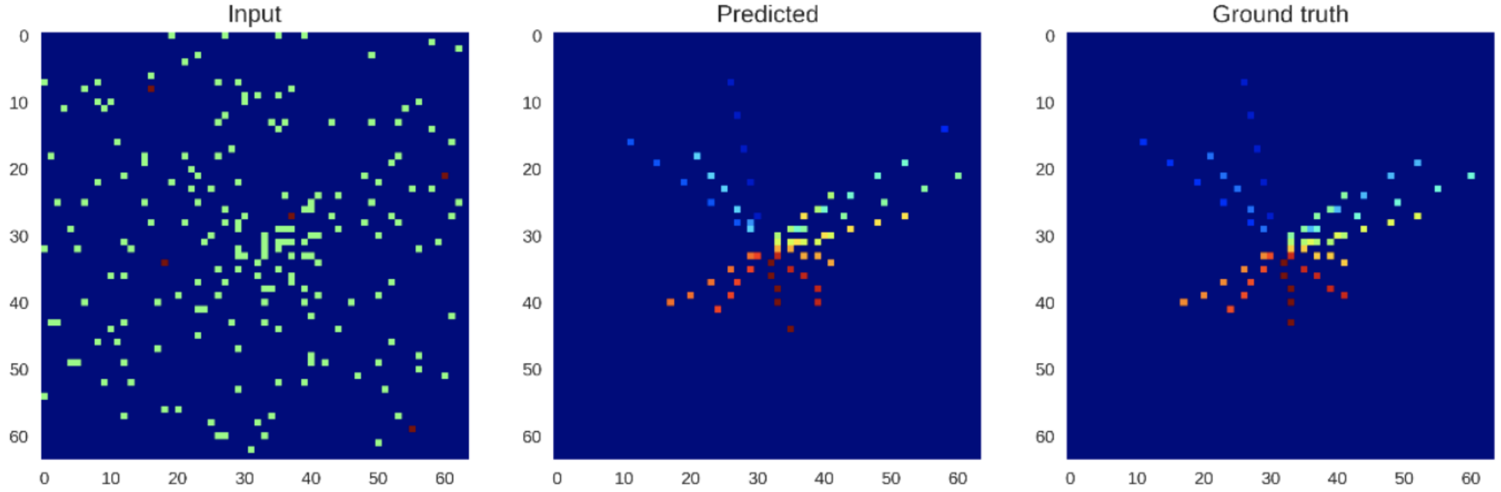


Global approach: U-LOOT (Look Once On Tracks)



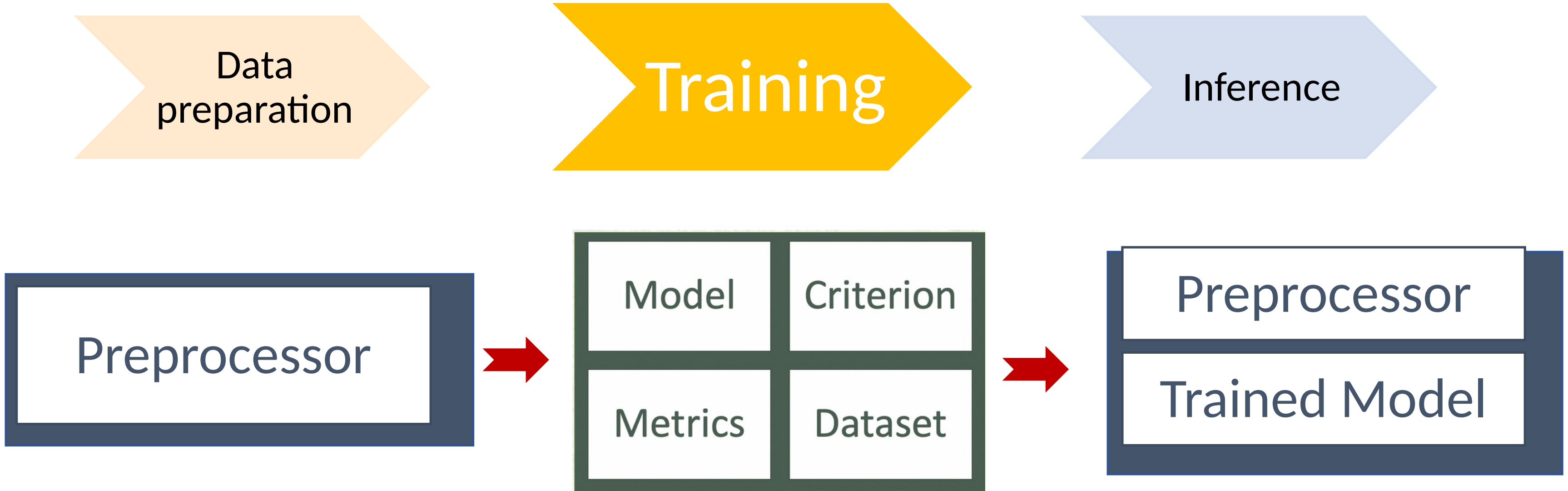
- Interpret an event as an image
- Images have 3D format: Height+Width+RGB
- The main idea is to use OZ dimension instead of RGB channels, and dimensions correspond to the size of the largest station
- Data from each station is a sparse matrix of zeros and ones, where ones indicate hits appearance (fakes too)
- **Events have 3D format too: Height+Width+Stations**
- Goal of the network is to predict the "image mask" (which leaves only true hits on the first station) and coordinates offsets for each station for such hits
- It is already developed as U-LOOT extension for initial event vertex reconstruction task(!)





Data and training (thanks Ariadne!)

- Networks were trained mainly on 3 datasets:
  - MC data from Run6/7 BM@N
  - MC data from BES-III
- All networks were trained on > 100k events
- The first verification result are quite preliminary but promising
- All work is done inside the **Ariadne** framework (developed by our team)
- Ariadne is the **first open-source library for particle tracking** based on deep learning methods.
- Ariadne development is currently in progress and it is not feature-complete yet



Test results

TrackNet	recall	precision	hit efficiency	track efficiency
BES-III	**97.74**	67.78		
BM@N			**99.67**	**98.70**

TrackNETv2	Track efficiency
BM@N RUN7	90.1

GraphNet	recall	precision	hit efficiency	track efficiency	Processing speed
BES-III	96.23	90.64	92.61	89.17	120
BM@N	96.	85.	94.	87.	30

U-LOOT	recall	precision	Processing speed
BM@N	96.90	97.19	150

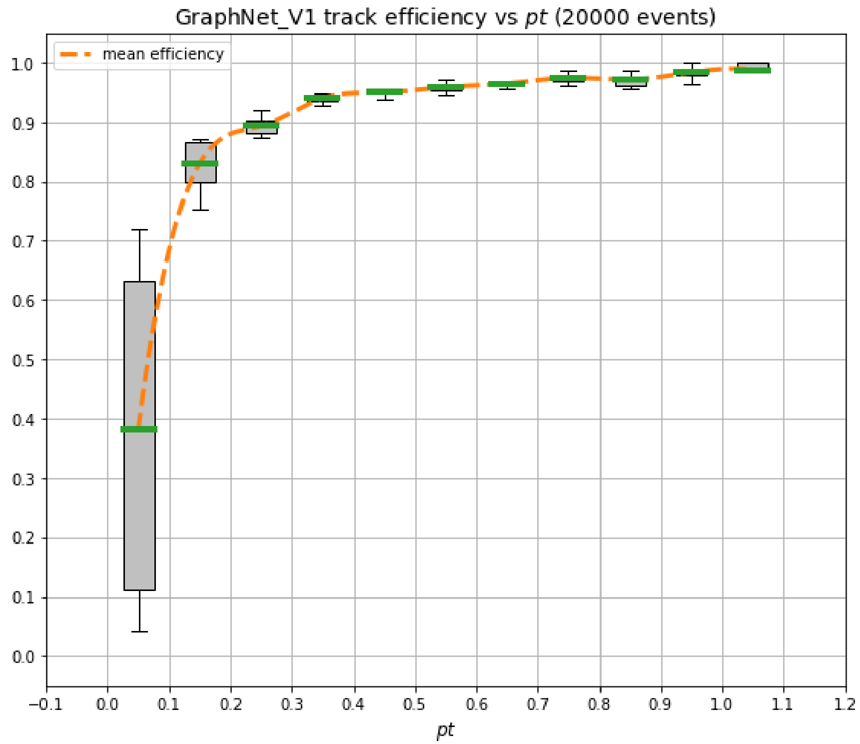
LOOT (vertex find)	MAE
BES-III	1.15 cm

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|x_i - y_i|^2$$

Metrics definition:

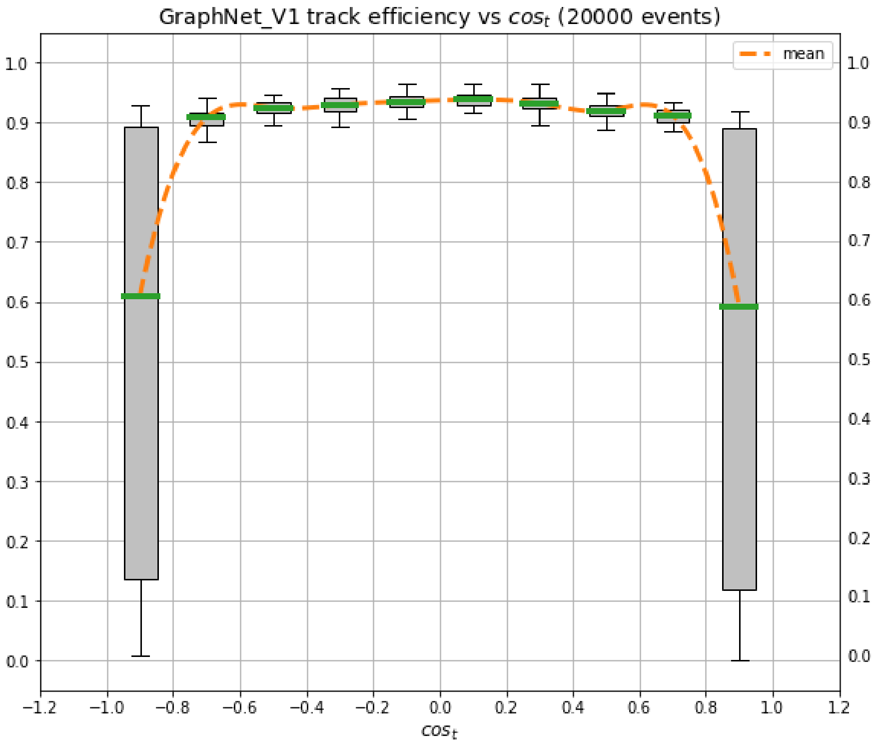
- **Recall** (percent of the found real segments)
- **Precision** (percent of the found true segments interpreted as true segments)
- **Hit efficiency** (percent of true hits found by network out of all true hits in a single event)
- **Track efficiency** (percent of full tracks without gaps found by network out of all tracks in a single event)
- **Processing speed** (events per second)

Results GraphNet (track efficiency vs particle momentum), BES-III



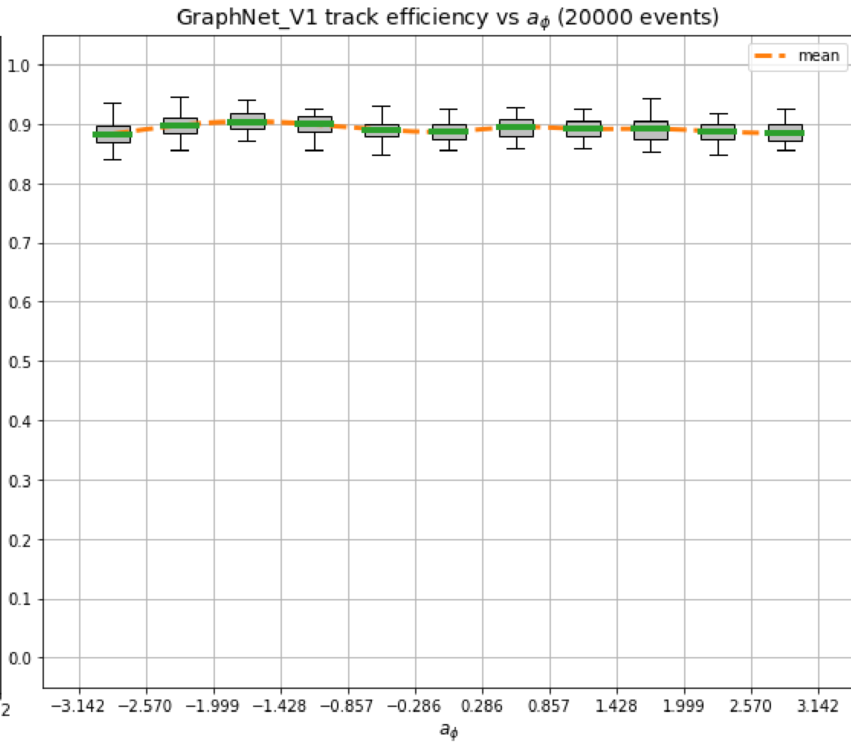
transverse momentum pt

$$pt = \sqrt{px^2 + py^2}$$



track polar angle

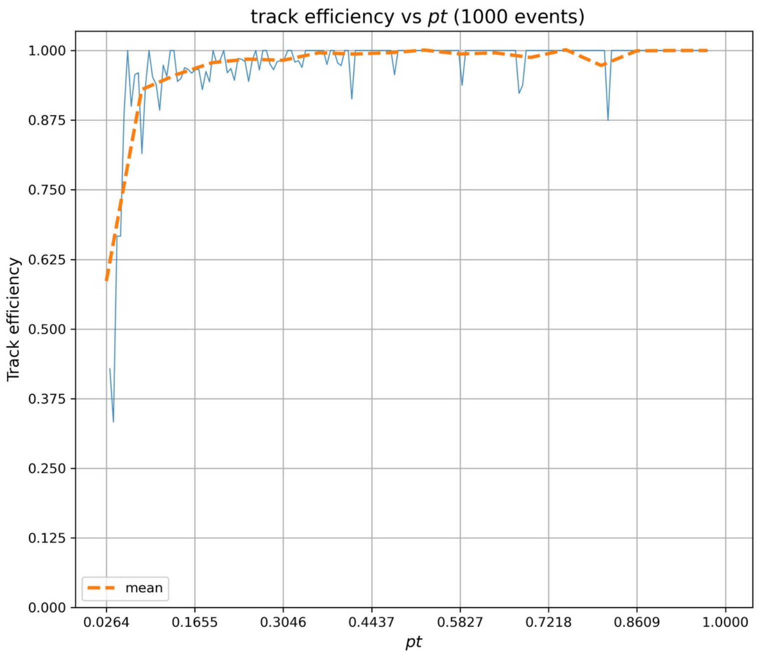
$$cos(\theta) = \frac{pz}{\sqrt{px^2+py^2+pz^2}}$$

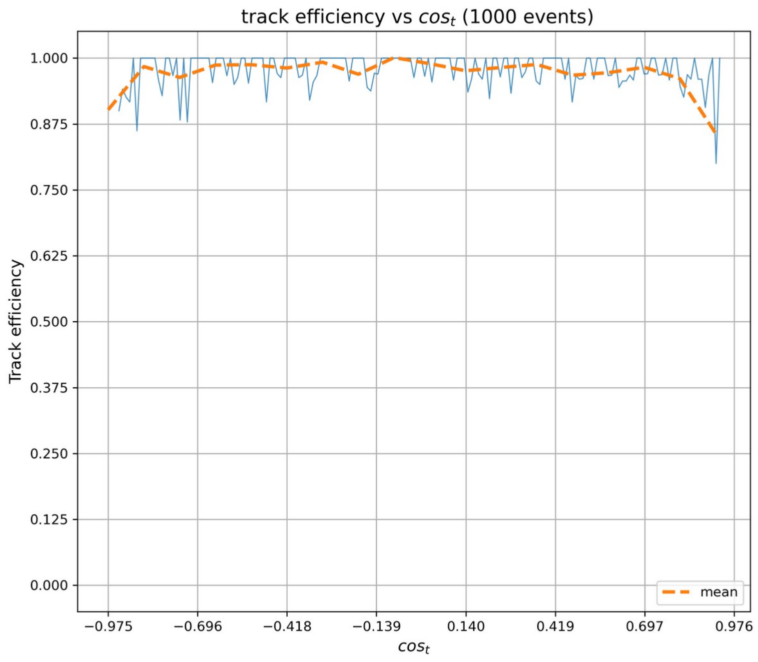


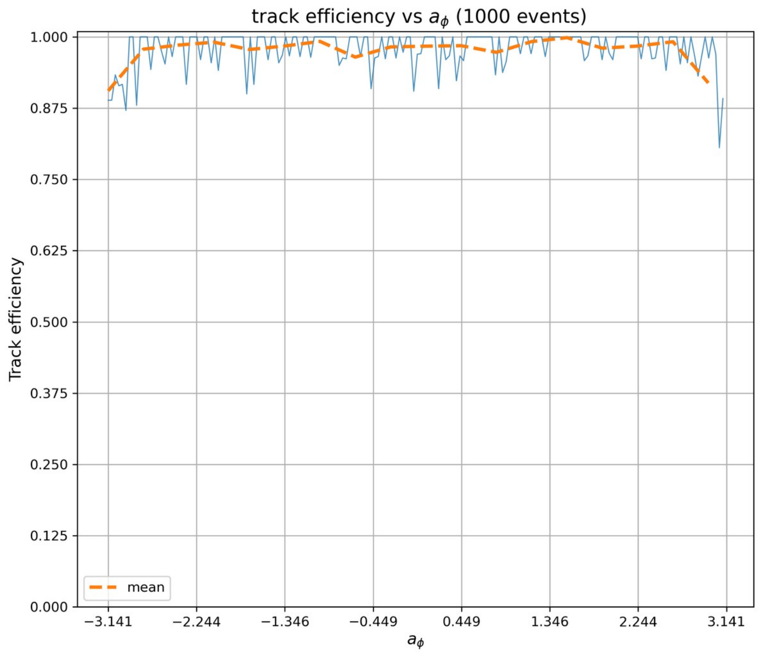
track azimuthal angle φ

$$\varphi = arctan2(px, py)$$

Results TrackNETv2 (track efficiency vs particle momentum), BES-III







transverse momentum pt

$$pt = \sqrt{px^2 + py^2}$$

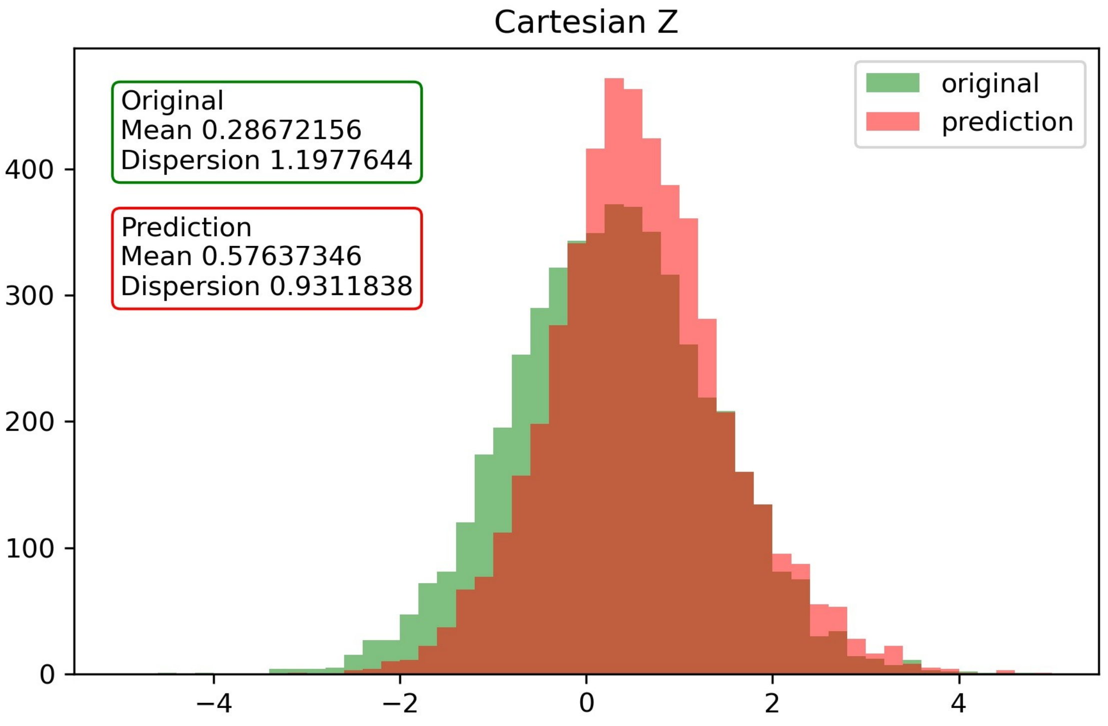
track polar angle

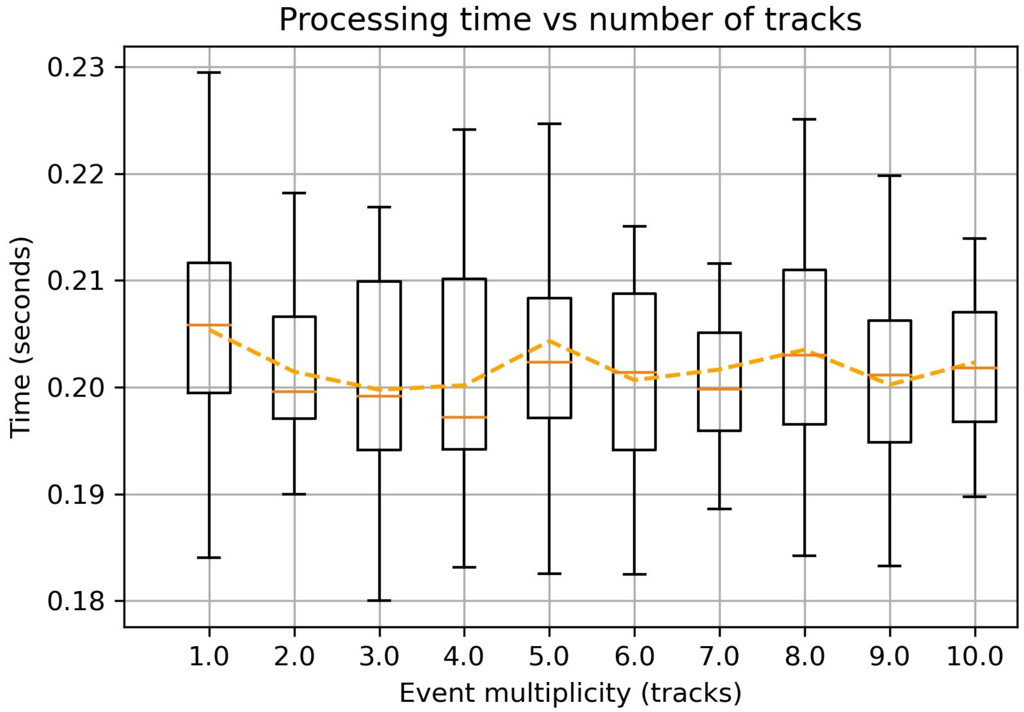
$$cos(\theta) = \frac{pz}{\sqrt{px^2+py^2+pz^2}}$$

track azimuthal angle φ

$$\varphi = arctan2(px, py)$$

Results LOOT (vertex coordinate distribution), BES-III





X and Y coordinates are known for BES III, so we tried to teach the LOOT-model to predict only Z coordinate

Conclusion

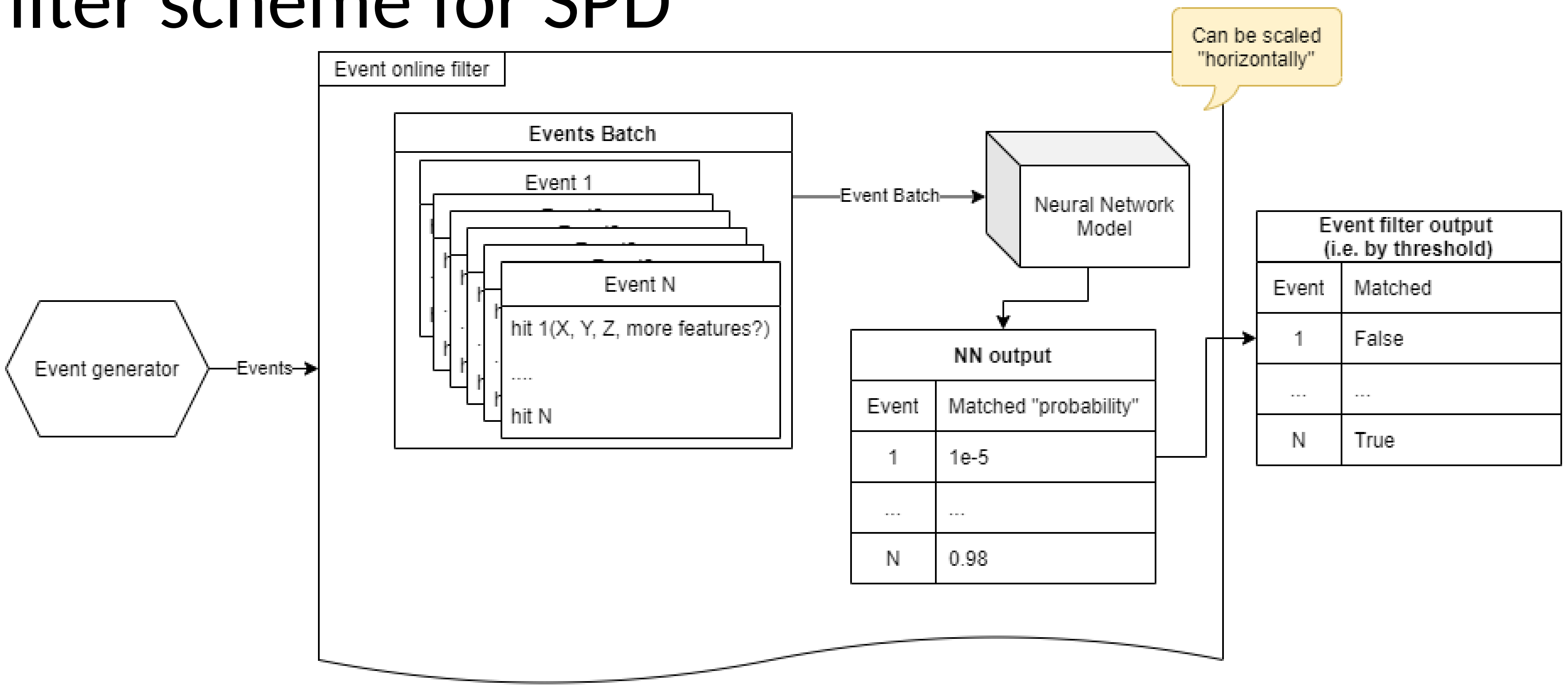
- We successfully implemented and verified our novel approaches on our data;
- Results are already promising, however, based only on MC simulation data;
- We need to try our approaches on the **real** data from **real** experiments;
- Models outperforms KF implementations in terms of speed by factors of magnitude;
- However, global algorithms are quite heavy and could be optimized further in terms of consumed GPU memory.
- Some of our approaches have overall generalization potential: they are producing comparable results for both fixed-target and collider experiments;
- This means that our approaches could be studied to work on more complex detectors (MPD? SPD? and so on..)
- We are almost ready to study the experimental data.

Outlook

- We are currently studying our approaches for the MC data from the BM@N run7
- We have started to study possible approaches for MPD tracking as well (first step is one of the most important steps in machine learning: visualize your data properly!)
- All our study is written with Python3 and does not operate with ROOT at all. Still, no resolution on this problem.
- We have developed some preprocessing algorithms in C++: got speed-up by 100x!

MPD event visualization with Ariadne (30 fps, python3, no ROOT)

Outlook. Proposed ML-powered event online filter scheme for SPD



- The goal of the Neural Network is to predict whether concrete event is “interesting” or not.
- No track reconstruction at all.
- NN model must be slim and robust enough to satisfy a huge event throughput rate (millions per second).

Neural network based tracking for BM@N and BES-III experiments

Egor Shchavelev[1]

Olga Bakina[2], Dmitriy Baranov[2], Igor Denisenko[2], Pavel Goncharov[2], Yury Nefedov[2], Anastasia Nikolskaya[1], Gennady Ososkov[2], Ekaterina Rezvaya[2], Aleksey Zhemchugov[2]

[1]Saint Petersburg State University
[2]JINR

SPD collaboration meeting
June 9, 2021