Mathematical Modeling and Computational Physics, 2024
20–25 Oct 2024, Yerevan, Armenia

On Overcoming the Crisis of Particle Tracking in High-Luminosity Experiments using Deep Learning Methods

Gennady Ososkov

Meshcheryakov Laboratory of Information Technologies, Joint Institute for Nuclear Research

Professor at the Dubna University
141980 Dubna, Russia

email: gososkov@gmail.com
http://gososkov.ru

Data and Technology Before Electronic Data acquisition. Outdated methods and problems still relevant today

- Bubble chambers with photo and manual measurements
- Measurin gsemi-automatic machines and viewing tables
  Scanning machines: HPD, Spiral Reader, Sweepnik, AELT-2
- In CERN IBM-360/44 then CDC-6600, programming in FORTRAN language
- In JINR lamp KIEV, M-20, Ural-2, semiconductor Minsk-2, programs in machine code

Methods of mathematical statistics and calculations

- Hypothesis and Parameter EstimationMethods
- Least Squares Methods
- Monte Carlo method
- Robust regression
- Radon-Hough transform
- Fourier Analysis and Wavelet Analysis
- Pattern Recognition
- Perseptrons and Hopfield Neural Networks

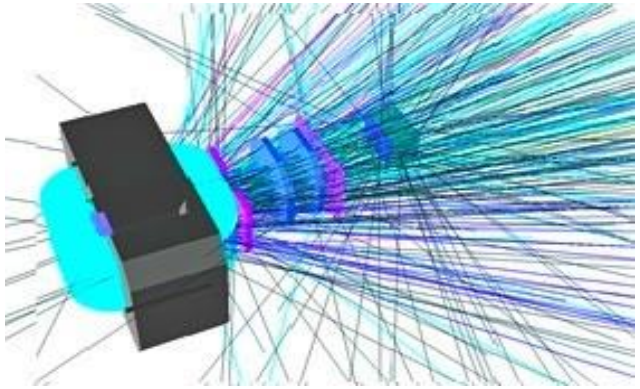
Math problems, that are still relevant today

- Estimation of buffer memory capacity under pulse loading
- Modeling of physical processes by generators of random numbers with specified distributions
- Recognition of smooth lines in a noise background
- Rejection of outliers in curve fitting
- Mathematical apparatus of calibration transformations
- Separation of close overlapping signals
- Finding small resonant peaks on a large noise substrate

Data, technologies and challenges in experimental HEP today

- LHC and NICA colliders
- Electronic data acquisition
- Pixel and strip track detectors
- World Wide Web - Internet
- Computer farms and supercomputers
- Distributed Computing, GRID, WLCG
- Machine and Deep Learning

Modern experiments with electronic data acquisition



BM@N Experiment. Strip GEM detector inside a magnet

TPC track detector inside an MPD magnet.
A simulated event from the interaction of gold ions generating thousands of tracks is shown

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

Tasks: reconstruction of events from measurement data in track and other detectors

Data measured in experiments and problem statements

Condensed
Barion
Matter

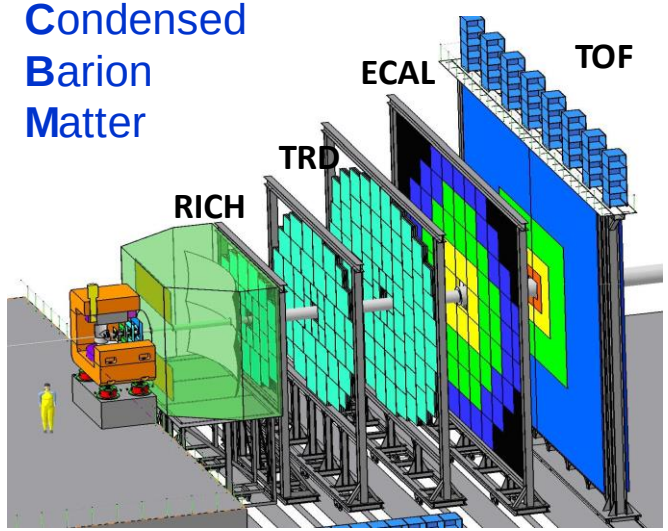
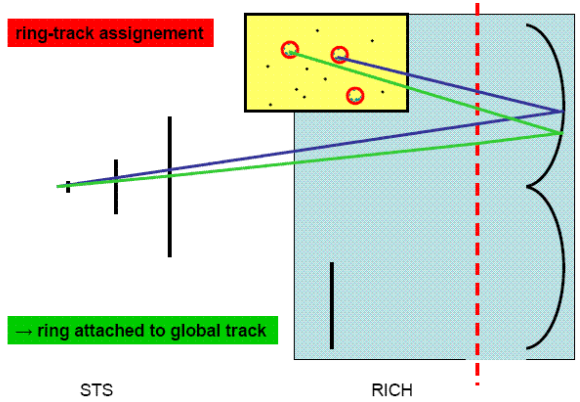


Схема установки CBM



Schematic diagram of the RICH detector of Cherenkov radiation

CBM experiment (Germany, GSI, to be launched in 2025)

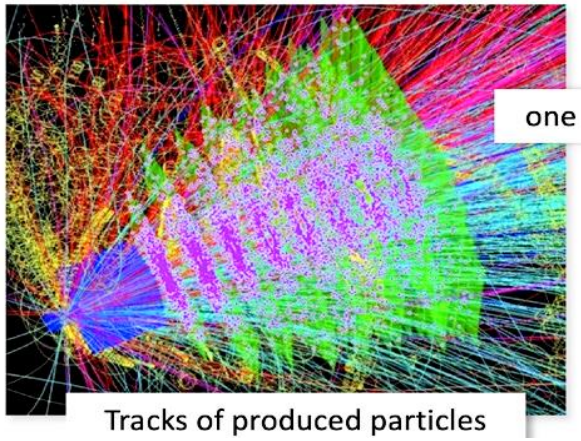
Data rate:

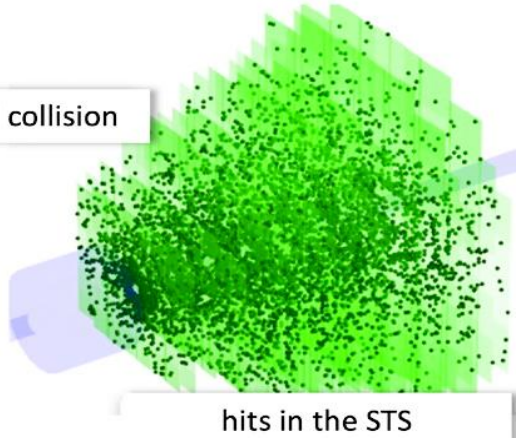
10^7event/sec,

~1000 tracks/event

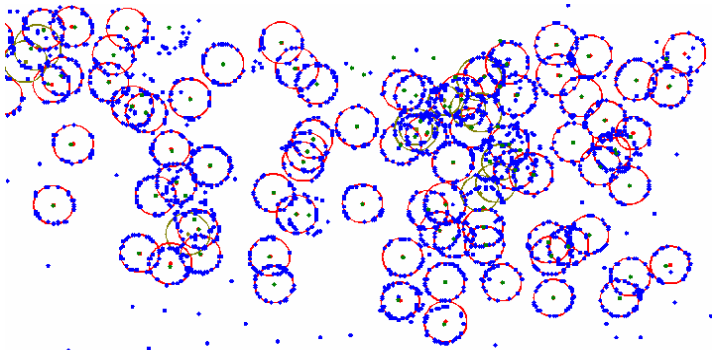
~100 numbers per track

Total: 1 terabyte/sec!


Tracks of produced particles

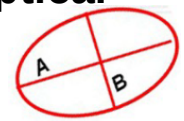


View of simulated Au+Au events in the vertex detector



A fragment of the photodetector data.
Average 1200 points forming 75 rings

CBM problems solved by machine learning methods: recognition of all these RICH tracks and rings and estimation of their parameters, taking into account their overlaps, noise and optical distortions leading to elliptical ring shapes (ellipse fitting), particle identification, analysis of the invariant mass spectra of short-lived particles, search for resonances



Until 2015, all these problems were solved using single hidden layer perceptrons, Hopfield neural networks, Kalman filter, robust methods and application of wavelet analysis.
Deep learning awaited new computer technologies

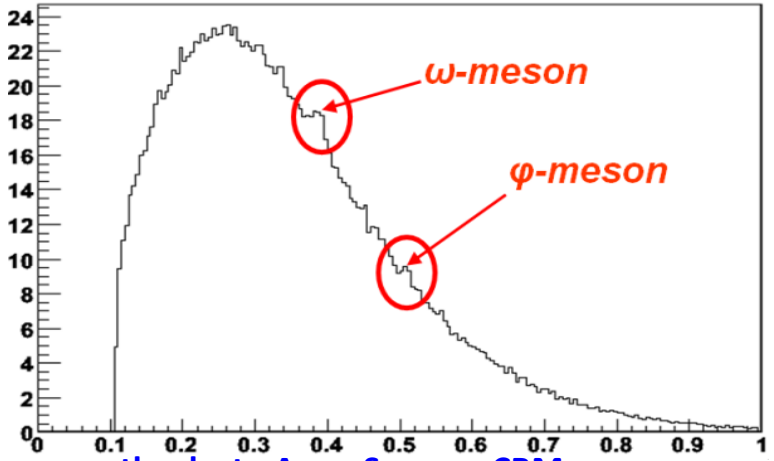
Main stages of data analysis in the current HEP experiments

- Collect data from many channels on many sub-detectors (mln/sec)
- Decide whether to read or discard the event (different levels of triggers)
- Reconstruct event (collect all information)
- Send data to storage
- Analyze them
  - distortion correction: calibration, alignment
  - hit finding, tracking, vertex search,
  - recognizing Cherenkov rings,
  - removal of false objects (fakes)
  - analysis algorithms from physicists-users
  - data reduction

Machine learning methods used

- Hough transformations,
- cellular automata,
- Kalman filter,
- artificial neural networks,
- robust estimation,
- wavelet analysis, etc.

- Detailed modeling of all experiment processes
  - Interaction of the beam with the target or colliding particle
  - scattering when particles pass through detectors
  - distortions during digitization, etc.
- Comparison of theory and physical parameters obtained from experimental results
  - Analysis of invariant mass spectra of short-lived particle, resonances



thanks to Anna Senger, CBM

- Utilize modern computing tools to achieve the highest speed and scalability of processing

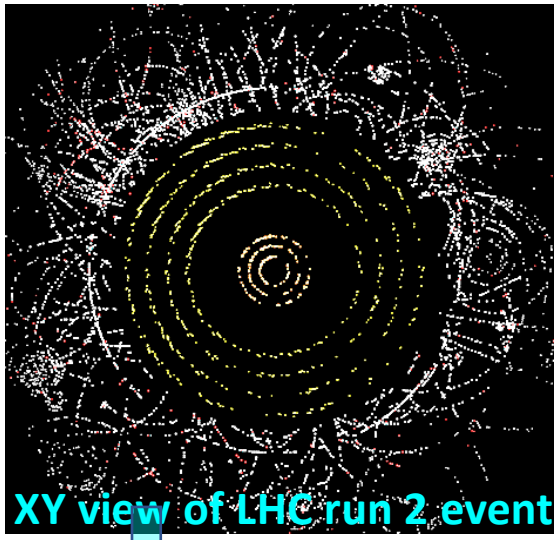
The inevitability of a worldwide Internet-based distributed computing network (Worldwide LHC Computing Grid -WLCG)
Parallel programming of optimized algorithms Grid-cloud technologies which changed considerably HEP data processing concept
See *Scientific data management in the coming decade* https://dl.acm.org/doi/10.1145/1107499.1107503

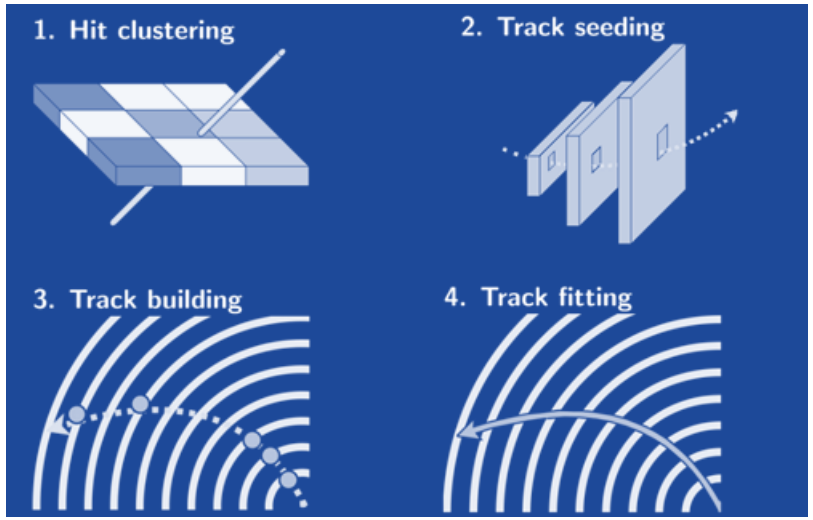
On machine learning methods on the example of tracking task as a key problem of event reconstruction in HEP

The reconstruction should determine the vertex coordinates and particle trajectories (tracks) for each event.

What is tracking?

Tracking or track recognition is the process of reconstructing particle trajectories in a HEP detector by following and connecting the hit points (a hit is the reconstructed detector response) that each particle leaves behind as it passes through the detector planes.





The main problem of modern tracking is the high luminosity of collider beams, i.e., the megahertz data arrival rate and the beam structure with bunches.

The tracking procedure includes phases: (1) obtaining hits (hit clustering), (2) constructing candidate tracks - sets of hits with calculated parameters (so called *seeds*), (3) track-following and (4) their fitting by the equation of the particle motion in the magnetic field.

Evolution of tracking methods

It all started back in the era of bubble chambers, when events were recorded on stereo photographs and entered into a computer manually. Then came devices for automatically scanning photographs and transferring the coordinates to a computer. One of such scanning automatons was the "Spiral Reader", in which the operator put a point at the event vertex, from where the image was scanned spirally by a narrow light slit, perceiving signals only from the event tracks

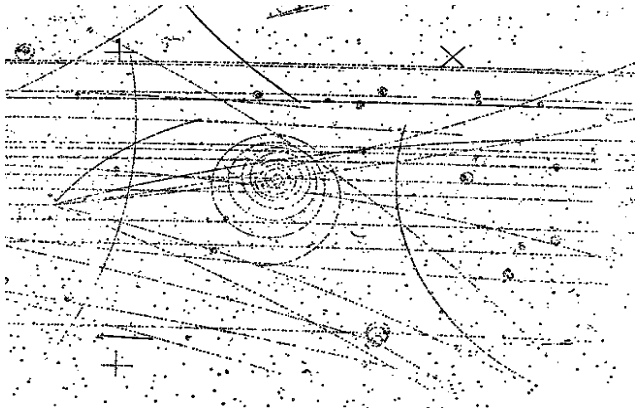
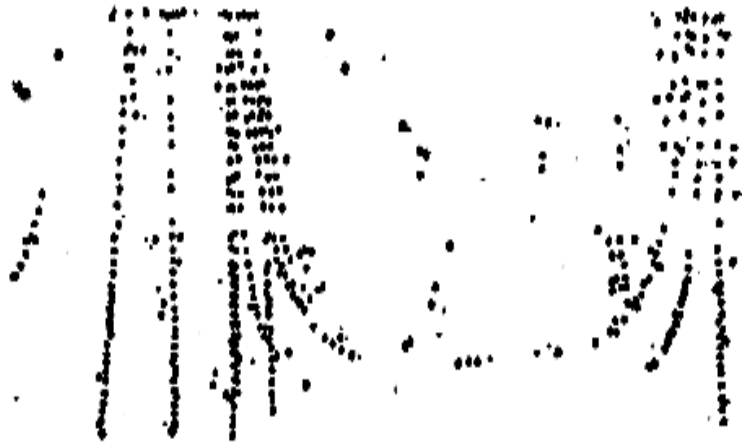
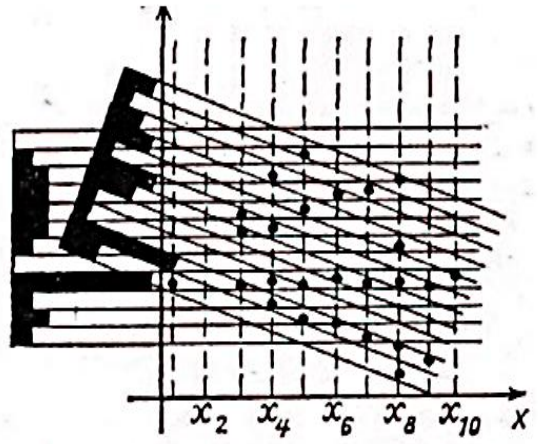


Photo of one projection in 2-meter hydrogen bubble chamber.



Its digitization in polar coordinates.



Rotated histogram method

When the era of electronic experiments arrived, measurement data were digitized and fed directly into a computer. After multi-stage filtering and alignement procedures, it was time for tracking.
One of the first tracking methods back in 1988 was the Hopfield neural network method

Classification of Track Reconstruction Methods

Local Tracking

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

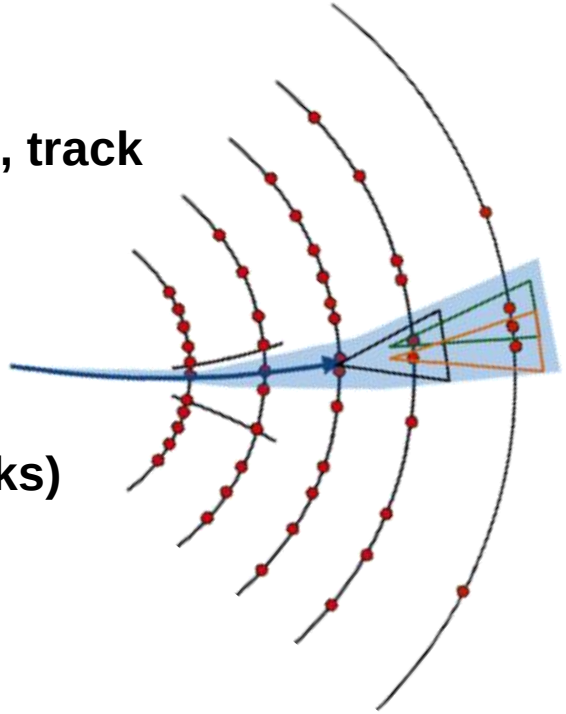
Examples: Road following, Cellular Automaton, Recurrent NN

Pros:

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

Cons:

- **Requires post-processing for ft 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.

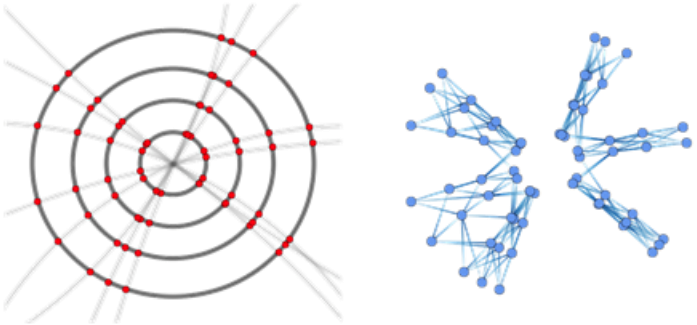
Examples: Graph Neural Networks, Hopfield network, Point Cloud Processing.

Pros:

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

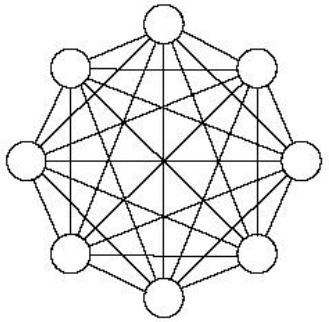
Cons:

- **High memory requirements (entire event as input)**



Graph Neural Networks in Particle Physics

Hybrid Tracking Methods are often useful, taking advantage of both approaches by first applying local tracking and then applying global tracking to the combined output of all recognized event tracks



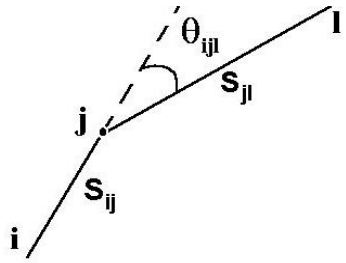
One of the first applications of neural nets in HEP (back in 1988) was event reconstruction using Hopfield neural networks

CHALLENGE. There is a set of N experimental points in the plane. It is required to select (recognize) among them those that belong to continuous smooth curves (tracks).

Hopfield neural network (HNN) is a fully connected network of binary neurons s_i **with symmetric weight matrix** $w_{ij} = w_{ji}$, $w_{ii} = 0$**. The evolution of the CNN leads it to some state of stable equilibrium. The energy functional of the network is the bilinear Lyapunov function** $E(s) = -\frac{1}{2} \sum_{ij} s_i w_{ij} s_j$.

Hopfield's theorem: as a result of evolution, E(s) decreases to local minima corresponding to the stability points of the network.

Mean-field theory, network thermalization and the mechanism of "simulated annealing" are used to find the global minimum E.



Method of segments

The energy functional (Denby and Peterson, 1988) consists of two parts: $E = E_{cost} + E_{constraint}$, **where**

$$E_{cost} = -\frac{1}{2} \sum_{ijkl} \delta_{jk} \frac{\cos^m \theta_{ijl}}{r_{ij} r_{jl}} v_{ij} v_{kl},$$

w_{ijkl}

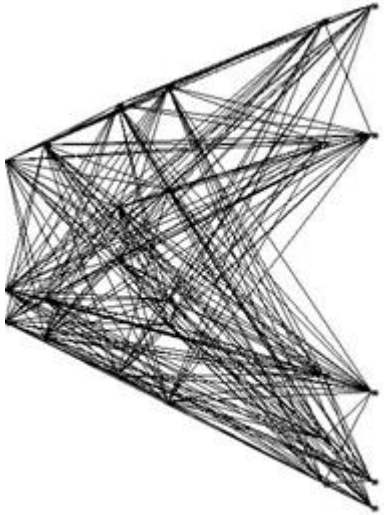
The neuron s_{ij} **is introduced as a directed segment connecting points i, j …**

encourages connections of neurons belonging to the same track, i.e. short adjacent segments with a small angle between them.

$E_{constraint}$ **prohibits both inter-track connections (bifurcations) and excessive growth in the number of tracks themselves.**

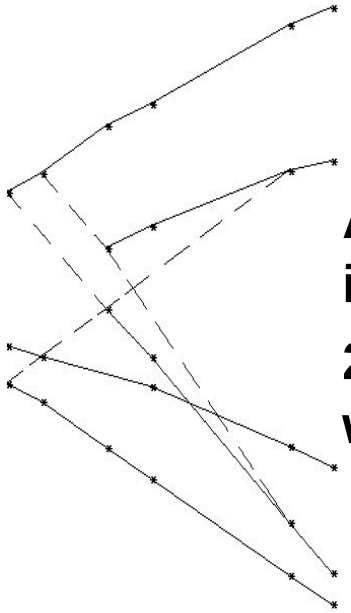
Example of HNN application for recognizing short-lived particle events

EXCHARM experiment (Protvino, 90s) - problem: unlike Denby-Petersen, allow bifurcations but do not allow mass branching of tracks



At iteration zero. a total of 244 *neurons*

Note: the appearance of even a single noise point would result in ~80 additional interfering neurons



After 30 iteration:

26 neurons with v_{ij} >0.5

However, it was just the practical application of HNN for tracking that showed such unacceptable disadvantages of this approach as slowness of the network evolution process, high probability of the network energy function trapping a local minimum, and excessive sensitivity of HNN to noise. In addition, the known equation of particle motion in a magnetic field was not taken into account.

Elastic tracking. Kalman filter advantages and disadvantages

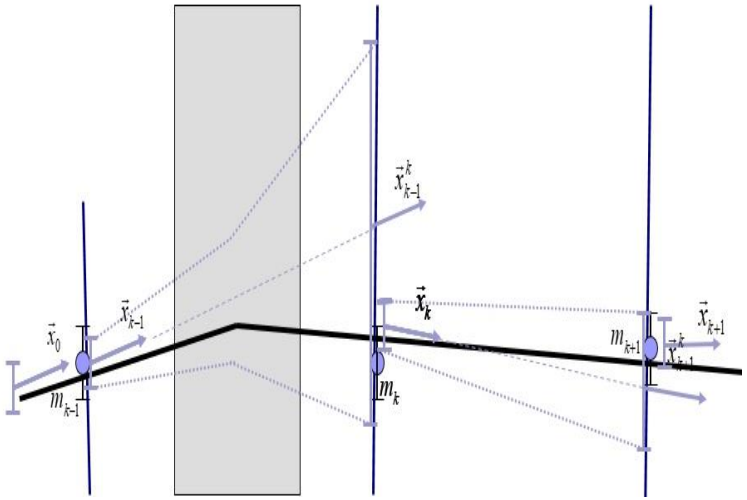
More successful were the attempts to overcome these difficulties with the help of "elastic tracking" methods using HNN, which combined the stages of recognition and fitting of the desired tracks.

However these methods required a laborious combinatorial search procedure and ceased to be effective with the increasing complexity of experiments and the multiplicity of events in them.

Among many tracking methods, the method using Kalman filter (KF) turned out to be the most effective, because it allows to easily take into account the inhomogeneity of the magnetic field, multiple scattering and energy losses, and the effective procedure of sequential estimation of physical parameters of the studied particle from the data of its measurements in the track detector, allowed to achieve the best results in terms of accuracy compared to other tracking methods

A Kalman filter (KF) is an efficient recursive filter that estimates the state of a linear dynamical system using a series of imprecise measurements

The state vector $\vec{x} = (x, y, t_x, t_y, q/p)^T$ **is iteratively estimated to predict the position of the track on the trace coordinate plane, taking into account changes in the covariance matrix and error corridors.**

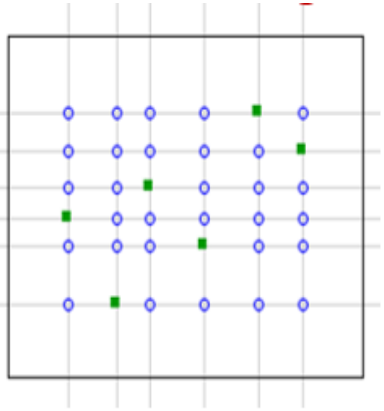


The main disadvantage of KF is the need to know the initial value of the state vector $\vec{X}$, to perform a very slow and cumbersome procedure of so called "seeding"

Furthermore, KF is slow, poorly parallelized and scales poorly!

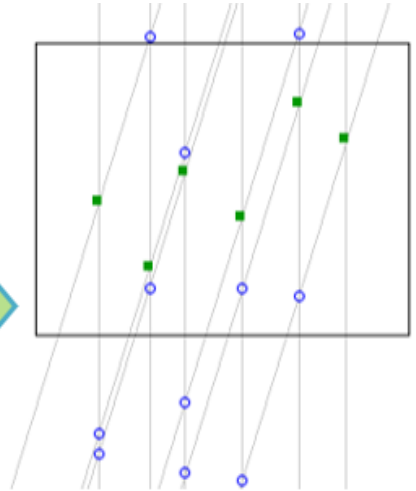
Tracking problems for modern GEM and straw-tube detectors

The main difficulty caused by the specificity of GEM and straw-tube detectors is the appearance of fake counts caused by extra spurious strip crossings **For n real hits one gains n^2-**

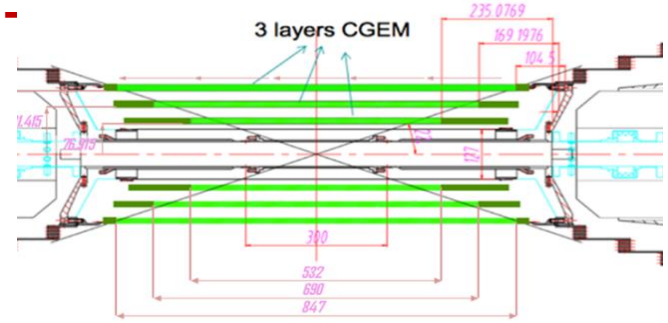


- Real hit (electron avalanche center)
- Spurious crossing

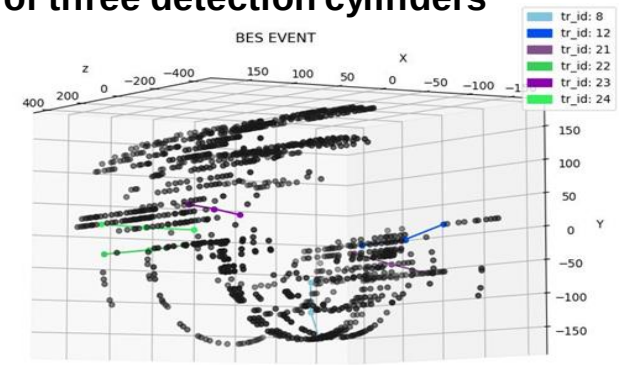
One of ways to decrease the fake number is to rotate strips of one layer on a **small angle** (5-15 degrees) in respect to another layer



Although small angle between layers removes a lot of fakes, pretty much of them are still left



The CGEM-IT internal detector of the collider BESIII experiment, consisting of three detection cylinders



All hits of a simulated event

The second problem is missing counts due to inefficiencies in the detectors. For detectors with a small number of stations, this causes tracking errors leading to false positive tracks (ghosts). In detectors with a small number of stations, skipping one hit out of three does not allow the track to be recovered in the magnetic field.

These problems under ultra-high data arrival rates due to the high luminosity of new experiments inevitably required the development of new tracking methods using deep neural networks

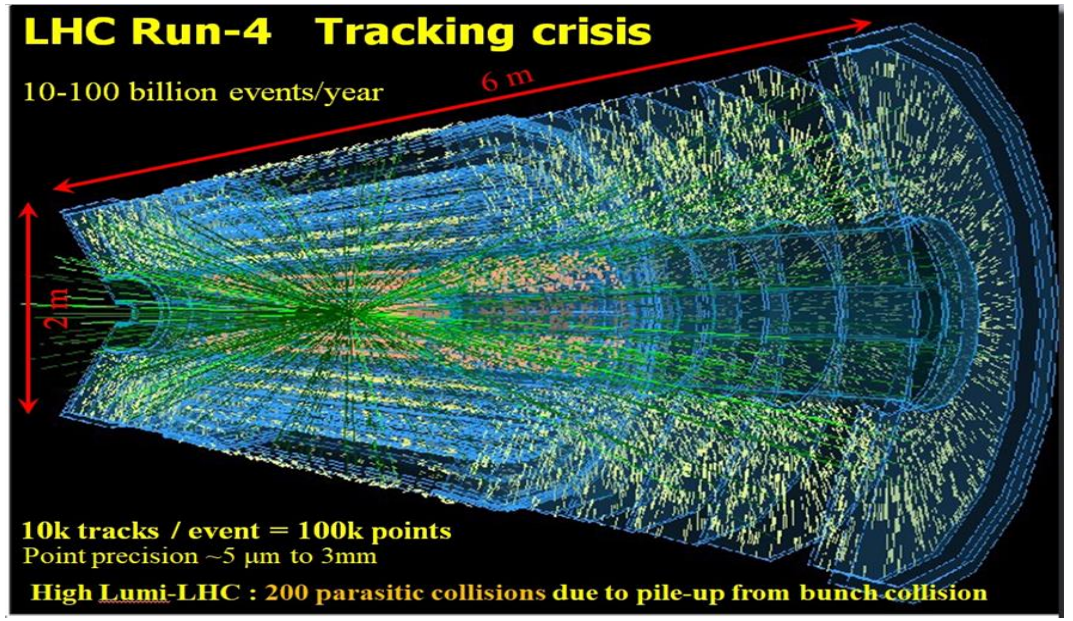
Problem Statement: The Need for Advanced Tracking Methods

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
- Traditional tracking methods cannot handle dense, overlapping particle tracks due to computational complexity and time constraints.

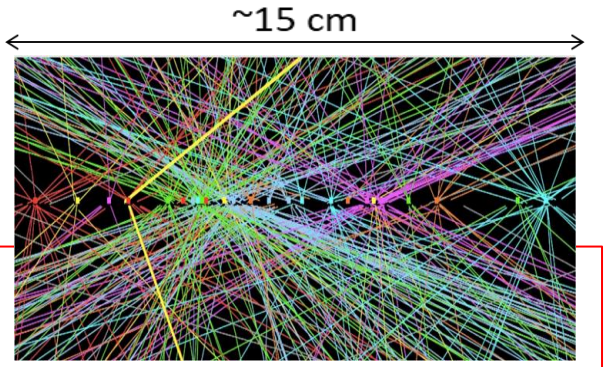
Deep Learning for Efficient Track Reconstruction:

- 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

The same problems are expected also for the NICA megaproject experiments



Under high luminosity conditions, particles are accelerated not individually, but in groups – bunches

To cope with immense data volumes
a new high-throughput deep-learning based approach for tracking is needed

TrackML challenge 2018

In 2018, physicists from CERN and other physics centers around the world, including Russia, staged a competition - the TrackML challenge to solve a machine learning problem for particle tracking in high-energy physics at high luminosity (DOI 10.110 TrackML dataset
9/eScience.2018.00088)

TrackML dataset

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

For this purpose, a source code simulator program is made on the Kaggle platform, where a typical all-pixel LHC tracking detector of 10 layers generates physical events (Pythia ttbar) superimposed on 200 additional collisions. This yields typically 10000 tracks (100000 hits) in each event.

Detailed view of the short strip detector of the TrackML challenge with a simulated event with 200 pile-up interactions

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.

TrackML challenge results

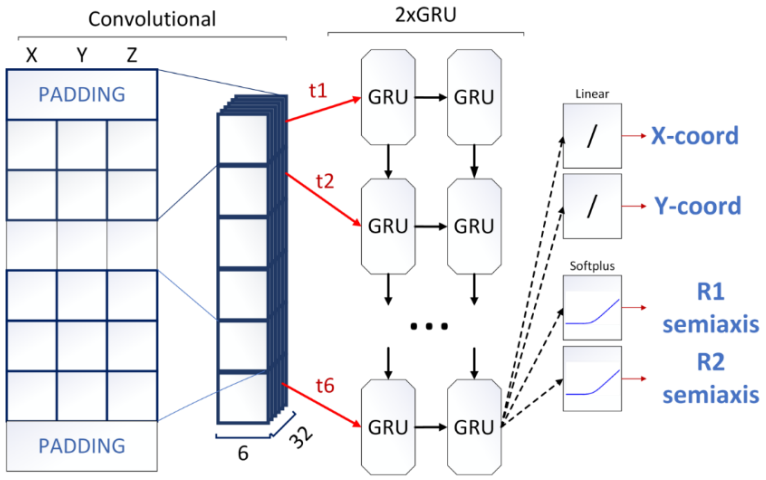
The TrackML competition has stimulated a lot of research where TrackML dataset has been used to train and verify different tracking neuromodels

- **Lots of graph neural network programs, e.g. https://arxiv.org/pdf/2003.11603**
- **There is also some interest in the application of Hopfield neural networks, but in a very different aspect, the slow evolution of the network is proposed to be dramatically accelerated by quantum annealing performed on a quantum D-Wave computer https://doi.org/10.1007/s42484-021-00054-w.**
- **Moreover, it is also proposed to apply quantum annealing to accelerate graph neural networks arXiv:2109.12636v1 [quant-ph] 26 Sep 2021**
- **These works have in many ways stimulated new and quite promising research on deep tracking carried out since 2018 at JINR MLIT for experiments of the NICA and BES-III projects**

Reports on real tracking tasks using LHC RAN 2 and 3 data have already appeared outside the **TrackML challenge** (see, e.g., arXiv:2308.09471v1 [hep-ex] 18 Aug 2023)

Our achievements before the announcement of the TrackML challenge

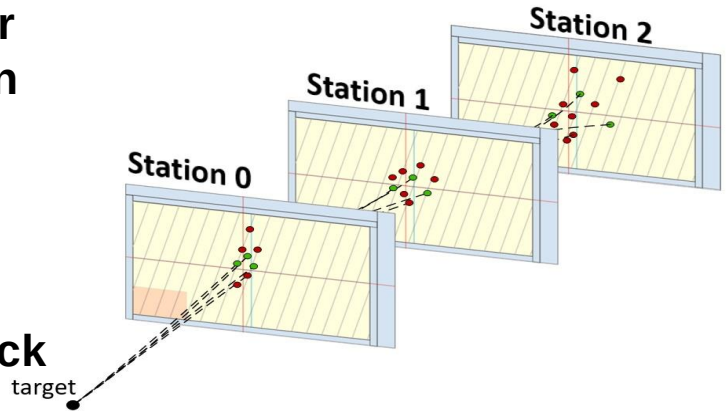
1. Local tracking for the GEM detector of the BM@N experiment is particularly challenging due to the presence of a p number of fake hits, making it extremely difficult to find those hits at subsequent detector stations that are extensions of the processed track.



Scheme of the recurrent TrackNETv2 NN

We used the flexibility of the GRU recurrent neural network architecture, which allowed us to overcome these difficulties and create a new end-to-end TrackNET neural network with a regression part of four neurons. Two neurons predict the ellipse center point on the next coordinate plane where the continuation of the candidate track should be searched for, and two more determine the semi-major axis of this ellipse. (See https://doi.org/10.1063/1.5130102)

This gives us the opportunity to train our model using only the true tracks that can be extracted from the Monte Carlo simulation. Thus, we have obtained a neural network that performs track tfollowing similar to the Kalman filter, although without the final part where track fitting is performed



Used Metrics:

Recall = $\frac{N_{true}^{rec}}{Ntrue}$; Precision = $\frac{N_{true}^{rec}}{Nrec}$

- N_{true}^{rec} - No. of correctly reconstructed true tracks.
- $Ntrue$ – No. of correctly reconstructed true tracks.
- $Nrec$ – Total number of reconstructed tracks.

However, the aforementioned shortcomings of local tracking meant that applying TrackNET to the simulation data of BM@N run 7 gave a good recall of 97% but an precision >50%, which is unacceptable. The situation was later saved by a hybrid tracking approach using a graph neural network in the second run

2. Global tracking. LOOT, BES-III experiment

Goncharov et al http://ceur-ws.org/Vol-2507/130-134-paper-22.pdf

Event as a 3D image in CNN convolutional neural networks.

In CNN, Images are 3d: height + width + RGB;

- **We have data from each station - a sparse matrix of zeros and ones, where the ones indicate the occurrence of hits;**
- **Events are also in 3D format: Height + Width + Stations.**
- **Height and Width are the dimensions of the largest of the stations (usually the last one).**

Our basic idea is to use OZ size instead of RGB channels.

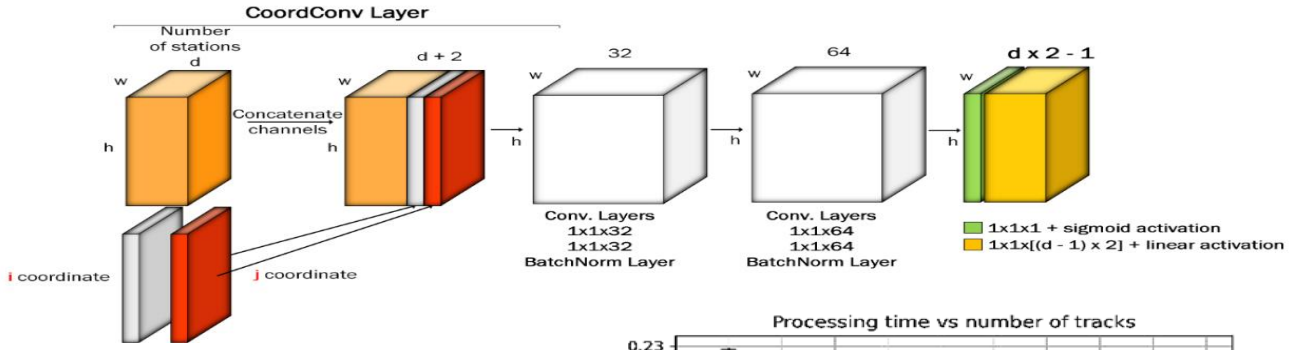
This is a radical new approach to find the coordinates of the event vertex

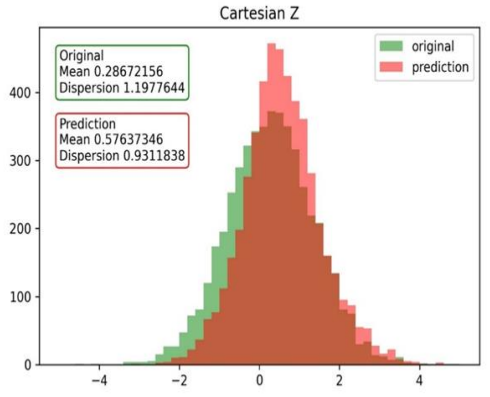
A new neural network model is used

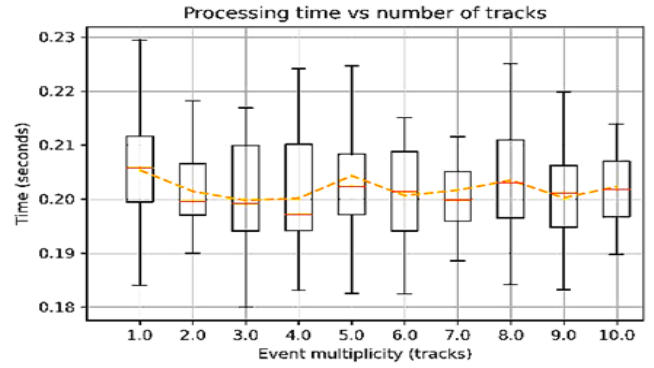
Look Once On Tracks (LOOT)

Since conventional convolutional neural networks cannot learn to find coordinates from input data during training, they are fed as input and then converted into cell indices. The network is trained to predict track continuations to the next layers using a shift procedure

Although the results were good on model data without fakes, adapting to the problems with fakes required the introduction of a new U-Net architecture. As a result of this work, the model after training predicts the Z coordinate of the primary event vertex with an acceptable RMS error of 1 cm







The inference time of the trained model does not depend on the multiplicity of the event

TrackML challenge. Our results

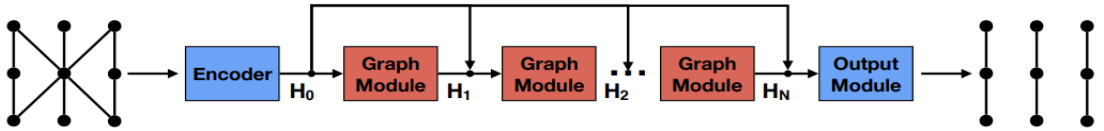
Two more reports will be given today
By Daniil Rusov and Nastya Nikolskaya

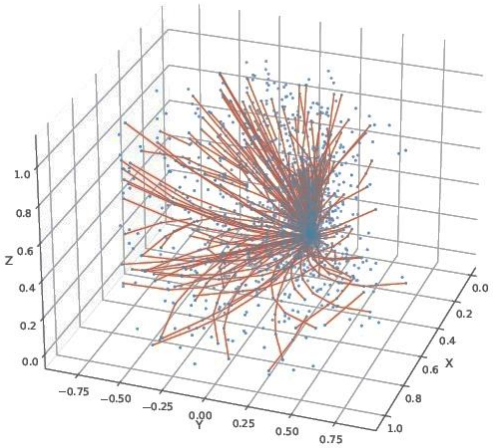
1. Improvement of HEPTrkX GNN model (By Yauheni Talochka)

https://arxiv.org/pdf/2003.11603

The HEPTrkX GNN model has been substantially redesigned.

In particular, the original structure

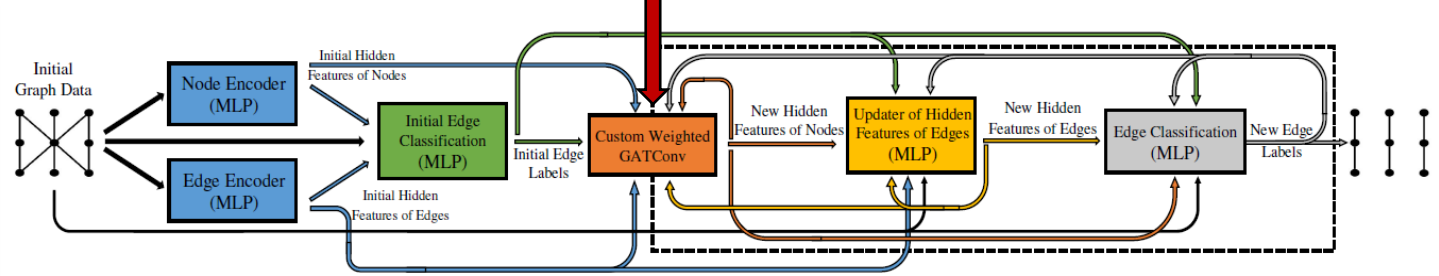




TrackML event graph

has been changed to

PyTorch Geometric



These improvements allowed to reduce the memory to accommodate the event graph CPU-RAM usage from 16 GB to 3GB and keep GPU-RAM usage up to 5GB at the batch size of 8.

Additional studied how much the results deteriorate when uniformly distributed noise of 20% of the data is added.

	w/o noise	20 % of noise
Accuracy	99 %	99 %
Purity & Efficiency	91 %	90 %
AUC	0.996	0.996

Adapting the GNN model to datasets obtained from the MPD experiment of the NICA project, taking into account their specifics, is the next step in the current study.

2. Quantum algorithms for TrackML event reconstruction

By Martin Bures

The approach is similar to the classical Denby-Peterson method, i.e., finding the global minimum for the Hopfield network using the mean-field method, but within the framework of quantum annealing with reformulating the tracking problem as a quadratic unconstrained binary optimization (QUBO) using modern D-Wave type quantum computer in simulation mode on the HybriLIT Cluster

The main unnovation is in using triplets of hits as binary variables instead of just doublets and quadruplets by combining two triplets. It's allowed us to weaken geometric assumptions about vertices, ban zigzagging quadruplets and incorporate in QUBO formulation more physical and geometric aspects.

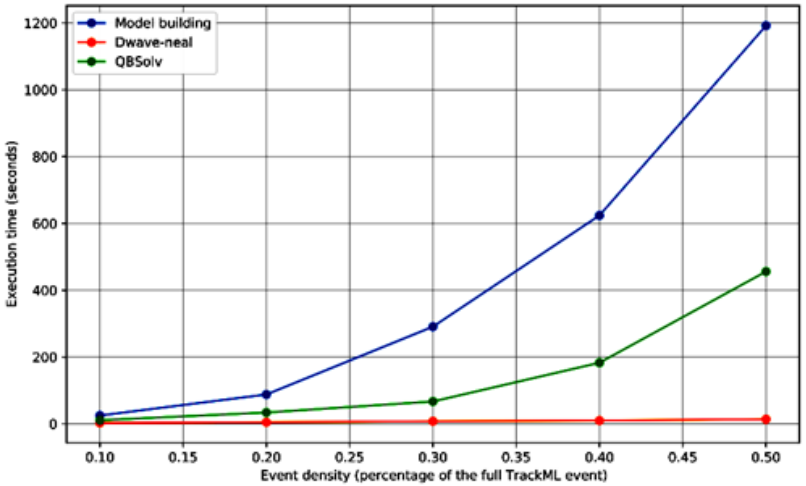
Calculatios use qbsolv (https://github.com/dwavesystems/qbsolv) and the neal library (classical QUBO solver).

Algorithm overview:

1) create potential doublets, filter them, create triplets + quadruplets, build QUBO
2) solve QUBO (qbsolv, neal library)
3) post-processing - determine the kept triplets from the QUBO solution, convert triplets back into doublets, compute metrics.

Event density — what proportion of the full event is used in the tracking algorithm. The TrackML training dataset consists of 100 events, the average number of particles per event is 10 792 ± 1 256. The number of hits per

event density	Num. of tracks	Num. of doublets	Precision (%):	Recall (%):	TrackML score (%)
~10%	818	51482	99.13/99.13	97.86/97.86	98.57/98.57
~20%	1637	188662	98.80/98.57	97.17/97.64	95.94/96.90
~30%	2456	456760	99.12/98.61	97.68/97.26	96.81/96.47
~40%	3274	785624	98.30/96.94	96.75/96.28	96.20/95.80
~50%	4093	1175314	**98.08/95.65**	**96.72/95.37**	95.88/95.01



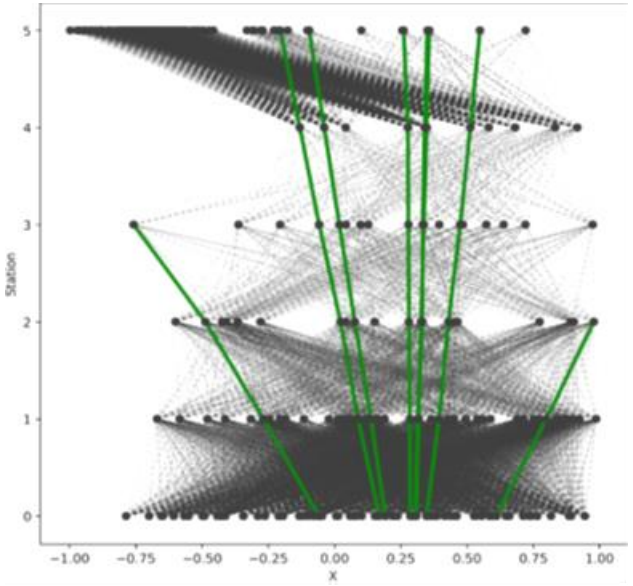
Performance timing as a function of event density

Tracking for data from high luminosity experiments. GNN for BM@N

Application of graph neural networks (GNNs).

By E.Shchavelev

Consider an event as a graph in which the nodes are hits. Nodes between neighboring stations can be connected by edges, which are possible track segments. Nodes are not connected within the same detector layer. The tracking task for graph neural networks (GNNs) can be formulated as a graph edge classification problem - to determine which of the segments belong to real tracks and which ones should be discarded as false.



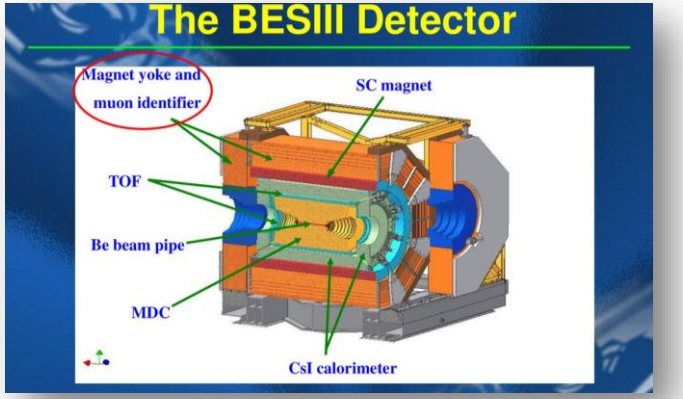
Graphical representation of the C + C, 4 GeV event of the BM@N experiment. Black nodes and edges correspond to the fakes, green nodes and edges to the found tracks

This scheme is similar to the well-known global Denby-Peterson approach with a segmented Hopfield neural network, where the neural network takes a long time to self-train separately for each event. However GNNs, where we need to find edges that are segments of real tracks, can be trained on a sample of event graphs, where these edges are labeled with a binary vector indicating whether a particular edge is true (1) or not (0). This approach has been successfully implemented at CERN for model events from the pixel detector, but our attempts to adapt their GNN for BM@N events with a huge fake background failed due to the resulting memory space issues for loading the graph.

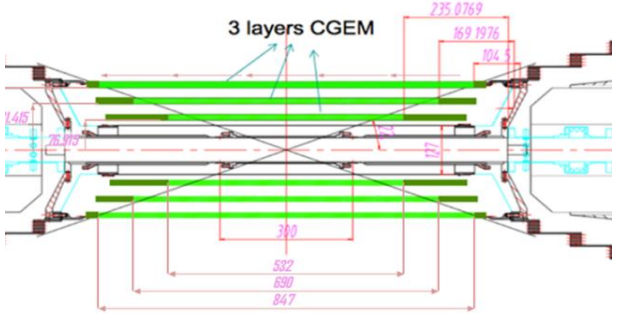
These problems disappeared when GNN was applied to the TrackNET output data in the second stage of tracking. By receiving as input an event represented as a graph of all candidate tracks generated in the first stage, GNN produced an acceptable tracking performance as a result

2 Tracking for data from high luminosity experiments. GNN for BES-III

By E.Shchavelev

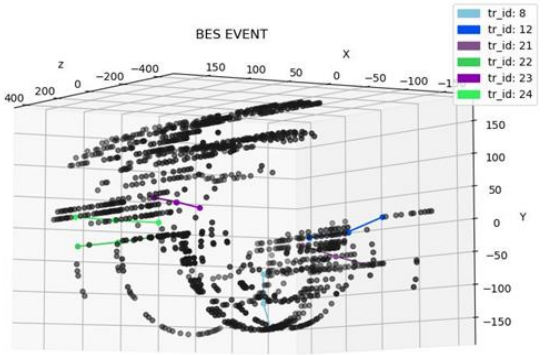


collider BESIII experiment



The CGEM-IT internal detector of the collider BESIII experiment, consisting of three detection cylinders

The presence of fakes and missed hits necessitated the use of a different type of GNN



All hits of a simulated event

The event graph is inverted into a linear digraph when edges are represented by nodes and nodes of the original graph are represented by edges. In this case, information about the curvature of track segments is embedded in the edges of the graph, making it easier to recognize tracks in a sea of fakes and noise. **In the process of training, the network receives as input an inverse digraph with labels of true edges - real track segments. The already trained GraphNet neural network as a result associates each edge with the value *x ϵ [0,1]* in the output. True path edges are those edges for which *x* is greater than some given threshold *(> 0.5)*. (http://ceur-ws.org/Vol-2507/280-284-paper-50.pdf)**

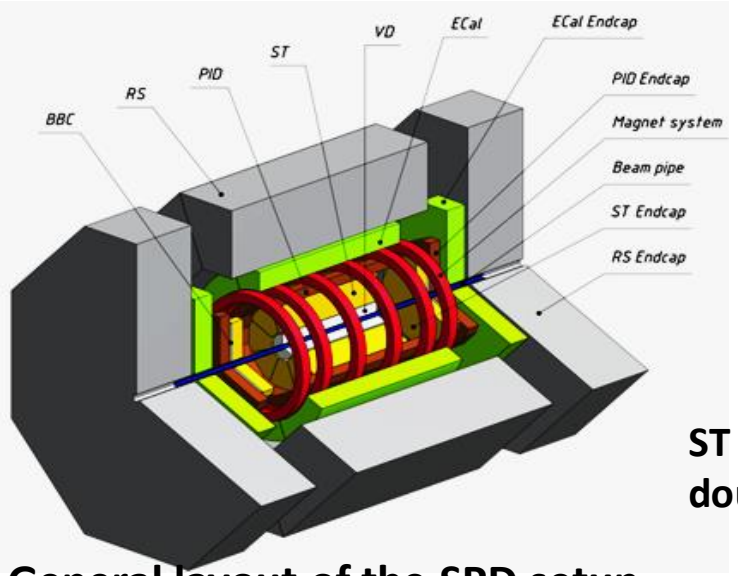
Tracking efficiency estimates. Evaluation of **accuracy** as a share of found tracks to the total number of candidate tracks is useless and even dangerous, because our sample is very unbalanced. It is accepted to use two metrics - **recall and precision. Recall** is the fraction of true tracks that the model was able to correctly reconstruct by finding all its hits. **Precision** is the fraction of true tracks among those that the model reconstructed

GraphNet	recall	precision
BES-III	96.23	90.64

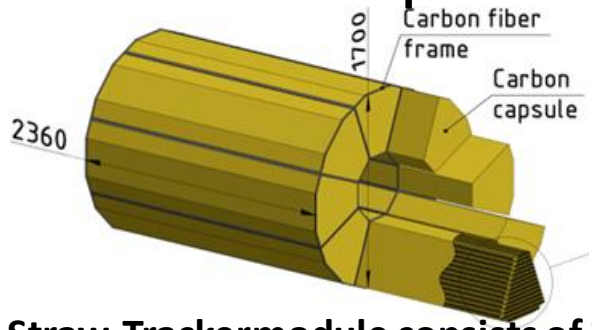
Tracking for data from high luminosity experiments. SPD NICA

SPD (Spin Physics Detector) is being developed to study the spin structure of proton, deuteron and other spin-related phenomena using polarized beams of protons and deuterons at collision energies up to 27 GeV and luminosities up to 10^{32} cm $^{-2}$ s $^{-1}$.

Event data from the SPD will be received at 3 MHz as 10 ms time-slice data, with up to 40 events in each time-slice, i.e., one time-slice will contain up to 200 tracks and ~ 1100 hits per station

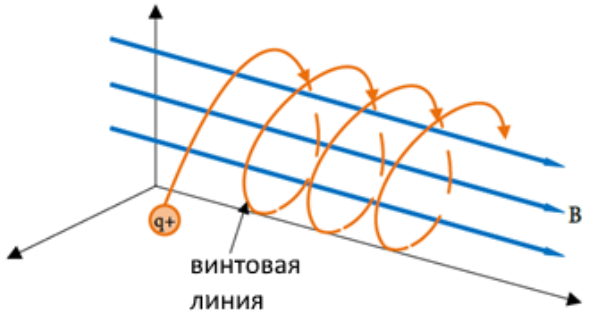


General layout of the SPD setup



ST - Straw-Trackermodule consists of 35 double layers of straw-tubes

Reconstruction of events from a dataset of time frames is required. For this purpose it is planned to develop an <u>algorithm for online filter</u> to process at least 100 time-slice per second



The calculations were performed according to <u>a simplified simulation scheme</u>:

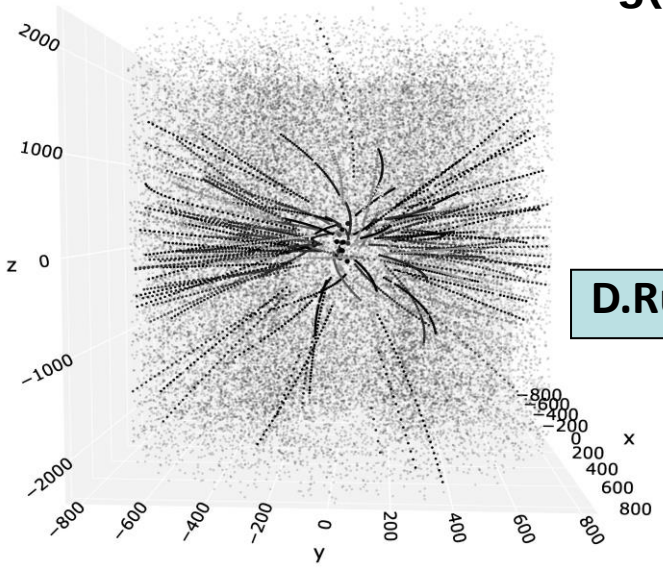
- Python script generates events with 1-10 random tracks.
- Transverse momentum: 100-1000 MeV/c (uniform).
- Random vertex coordinates within the collision area.
- Trajectories follow a helical path, defined by the pitch and radius equations.
- Simulated detector with 35 stations.
- Fake and noise hits simulated using randomly sampled points in detector space.

Deep tracking for SPD NICA timeslice data

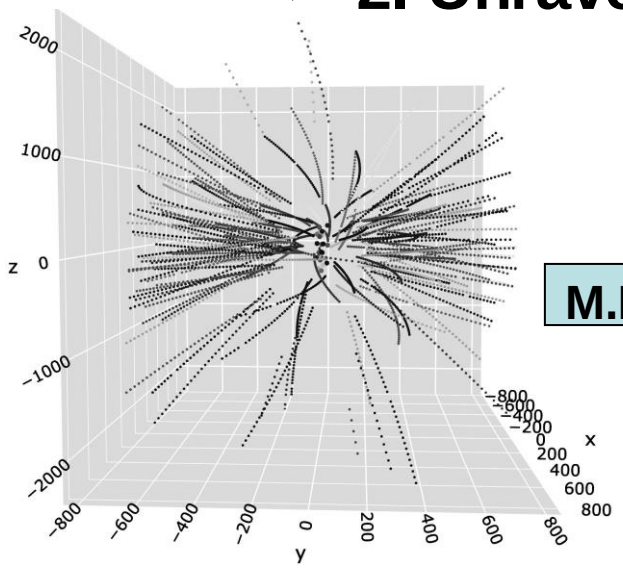
The main problems in SPD tracking are a huge number of fake signals, missed counts due to inefficiency of detectors and "left-right" ambiguity of straw-tubes Introducing appropriate complications in the TrackNET program inevitably slows down its work and reduces its efficiency.

Reconstruction of events from the time-slice dataset was performed in two stages

1. On-line tracking(TrackNET)

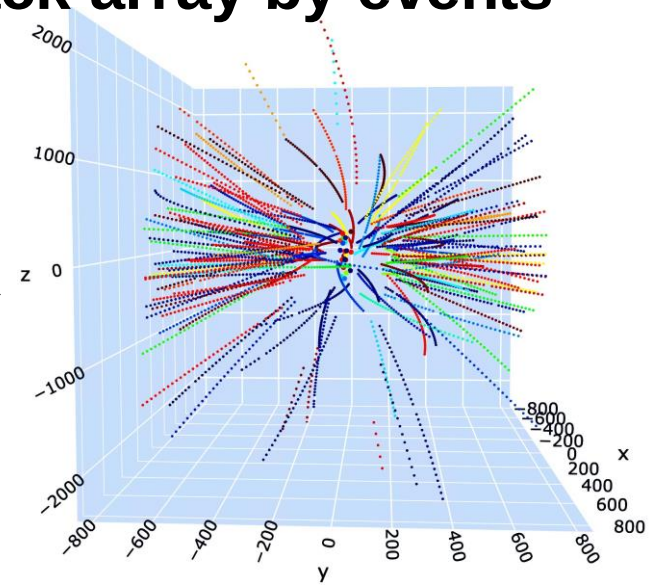


D.Rusov



M.Borisov

2. Unraveling track array by events



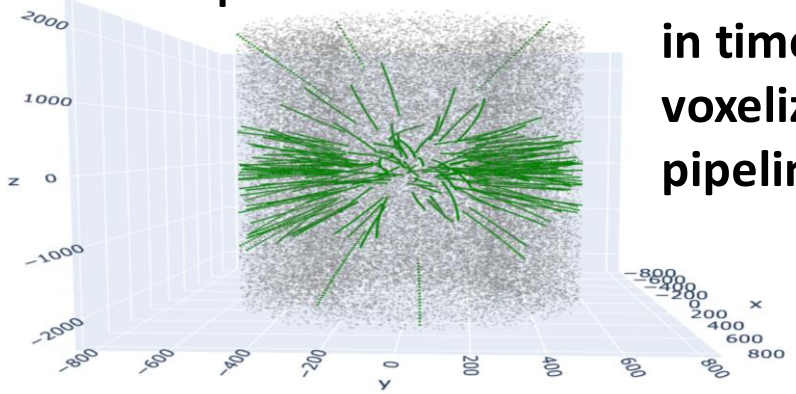
By fine-tuning TrackNET on the GOVORUN supercomputer, a processing speed of ~2000 model events per second with acceptable tracking efficiency was achieved

The event unraveling algorithm is based on clustering of feature vectors, obtained using Siamese neural network. The result is quite promising, but requires improvement due to insufficiently low efficiency.

New approaches using the Point cloud transformer (PCT) neural net

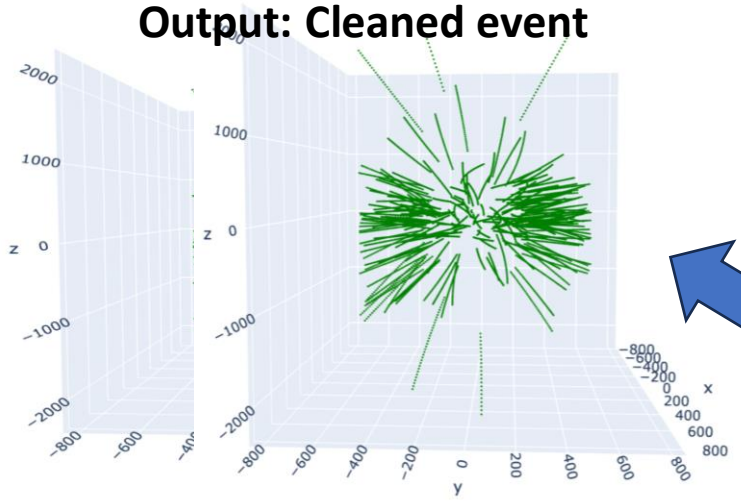
1. PCT to determine the number of tracks in an event (By Anastasia Nikolskaya)

Input: Raw event

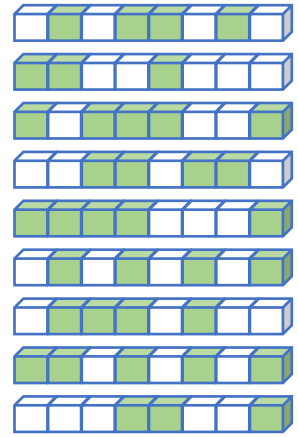


2.PCT for removing fakes in time-slice. SPD event voxelization and software pipeline

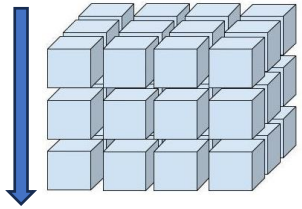
Output: Cleaned event



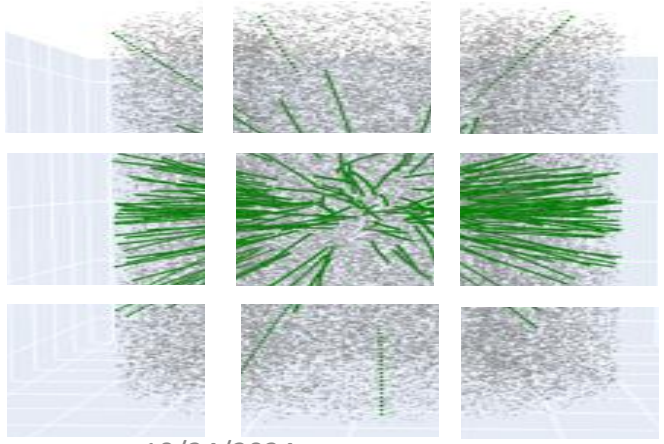
Combine everything into one event

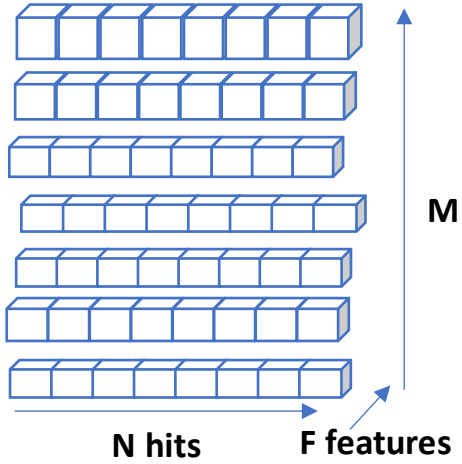


Divide the detector space into M voxels, i.e. into smaller subspaces



Take the hits from each voxel and form a batch of M x N x F subsamples, considering that each subsample is a mini-event





PCT= Point cloud transformer

PCT model

Classifying hits into true and fake ones

NUMERICAL RESULTS

Events number	Not effective	Voxel number	Precision	Recall
40	2%	512	0.98	0.98

Conclusion and outlook

- The radical projects of recent years for high luminosity experiments (HL-LHC) and NICA, pose the difficult problem of particle track reconstruction in dense media, which requires the development of new deep tracking algorithms and their parallelization on supercomputers.
- The TrackML challenge in many ways stimulated further development of deep tracking methods already tested in LIT (TrackNet, Loot) and induced a number of new approaches. It should be noted that research on the application of neural network models of transformers, which allow, in particular, effectively filter out fake measurements and perform tracking on raw data, bypassing the stage with hits acquisition.
- In the more distant future, attention should also be paid to quantum annealing methods in applications to both global tracking and local tracking methods generalizing Kalman filter algorithms.
- On the wave of success of generative-adversarial neural networks in deepfakes and diploma, publications on their successful application to simulate interactions in FWE experiments should be noted

G.Ososkov

On Overcoming the Crisis of Particle Tracking in High-Luminosity Experiments using Deep Learning Methods

Thank you for your attention

email: gososkov@gmail.com
http://gososkov.ru