Graph neural network application to the particle track reconstruction for data from the GEM detector

Shchavelev Egor

Pavel Goncharov, Gennady Ososkov, Dmitriy Baranov

Saint Petersburg State University

egor.schavelev@gmail.com

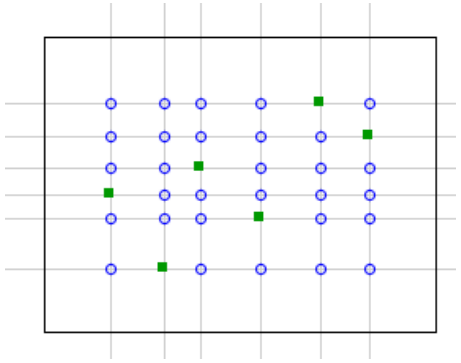
Plan

- BM@N vs LHC tracking problem;
- Event as a graph conception;
- Graph neural network approach (HEPTrkX) and its application for GEM detectors;
- Minimum spanning tree approach;
- Preliminary results for the GNN for the GEM-detector.

Problems of microstrip gaseous chambers

The main shortcoming is the appearance of fake hits caused by extra spurious strip crossings

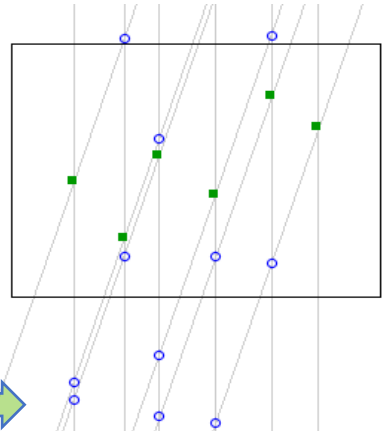
For n real hits one gains n^2 - n fakes

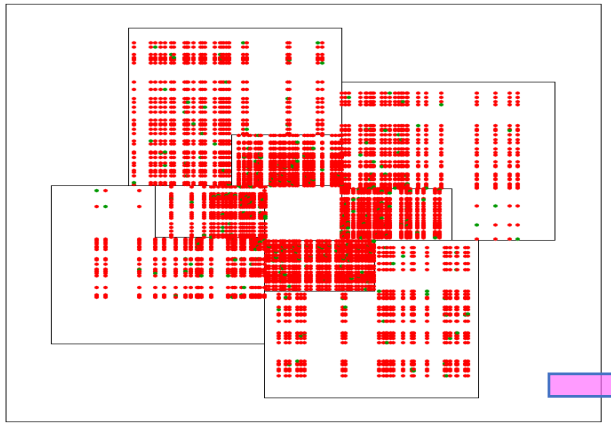


- **Real hit (electron avalanche center)**

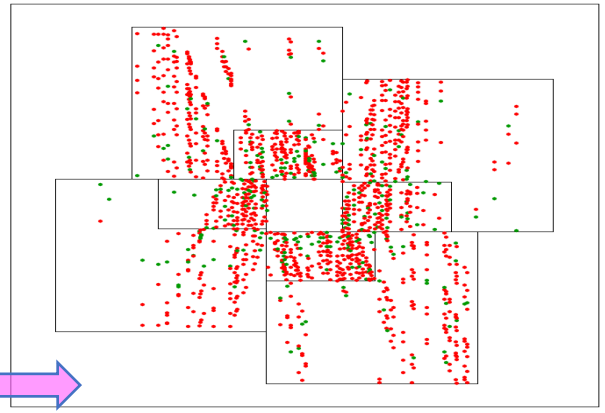
○ **- Spurious crossing**

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



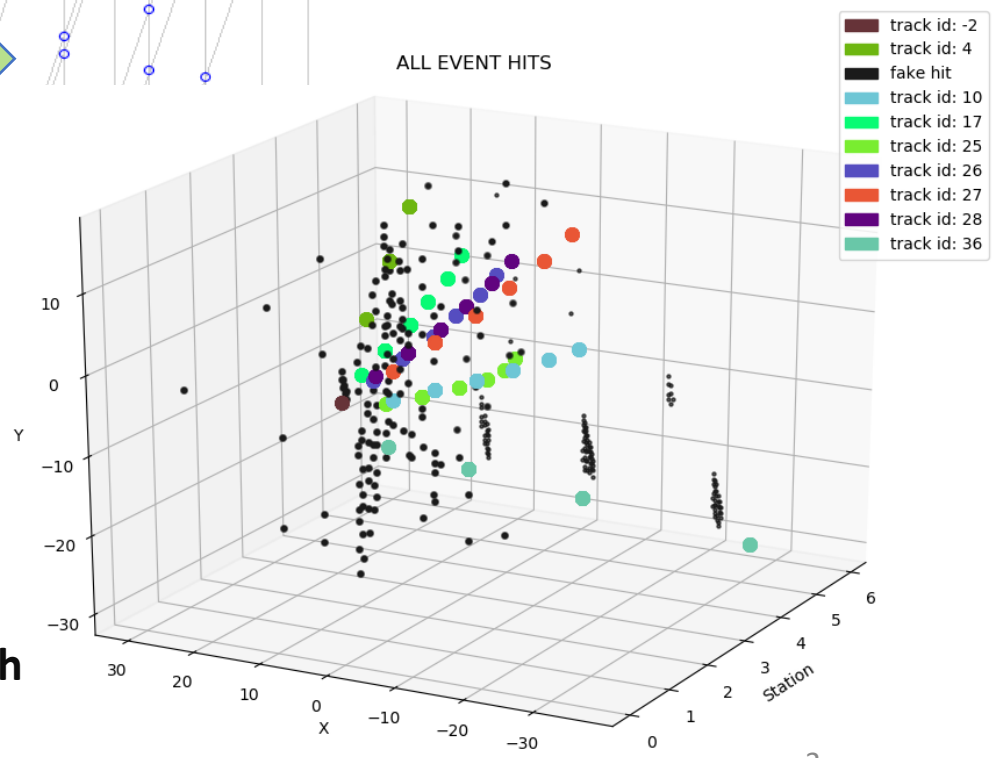


*Angle between strips **90 degrees**, UrQMD event Au-Au, 4 A·GeV*



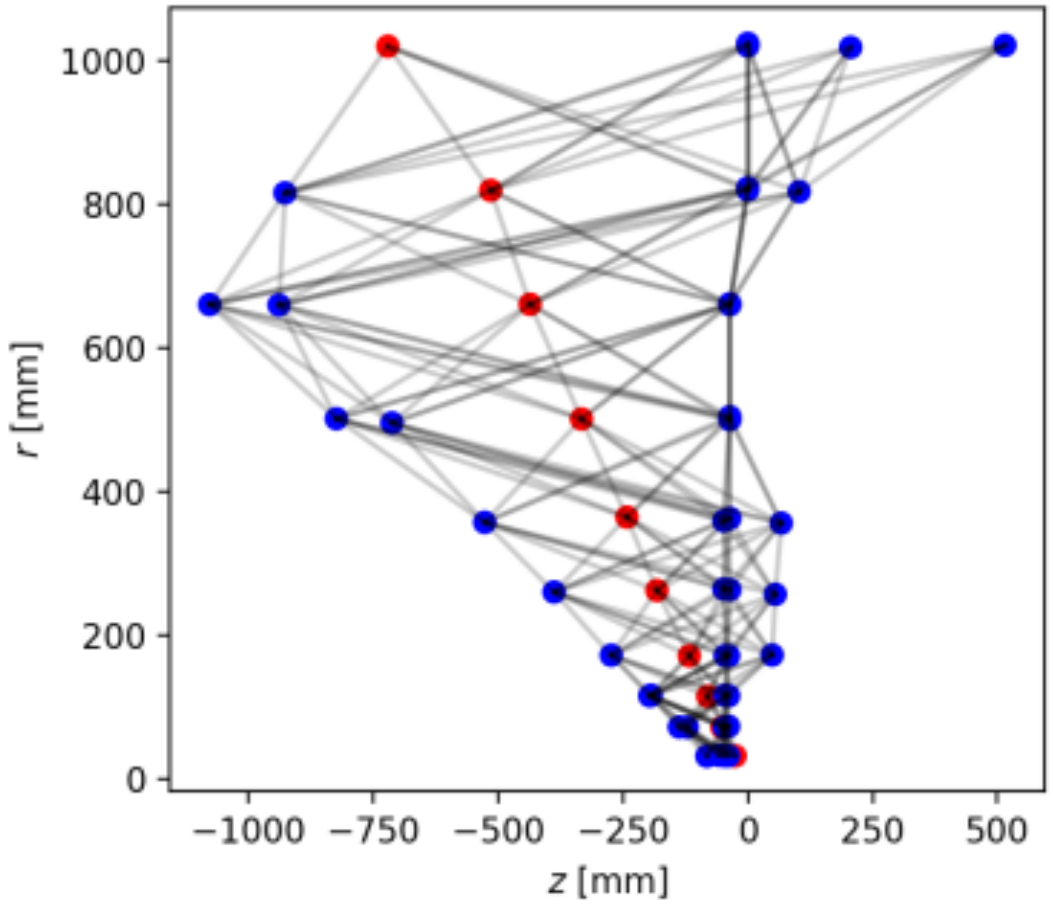
- hit - fake

*Angle between strips **15 degrees**, UrQMD event Au-Au, 4 A·GeV*



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

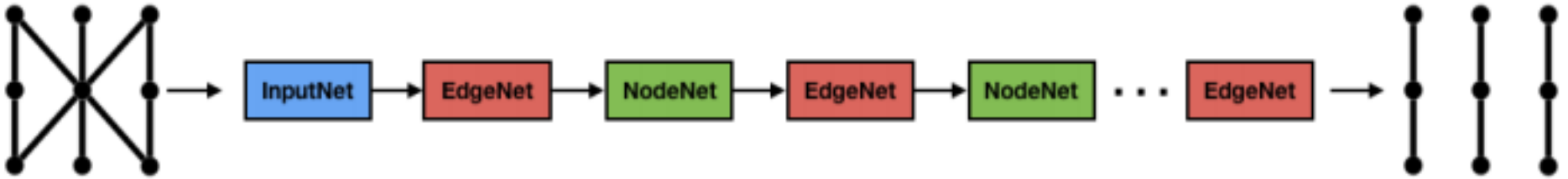
Graph neural network approach at LHC



- Introduced by HEPTrkX project at LHC;
- Graph neural network based on Interaction Networks approach adapted for particle tracking;
- Hits are represented as nodes of the graph;
- Nodes are fully connected between adjacent layers;
- Model is evaluated by iterating over ‘Edge’ and ‘Node’ networks.

GNN inside. Input network.

- The GNN consists of three main parts – Input Network, Node Network, Edge Network.
- Event-graph is being interpreted as 4 matrices:
  - **X** – matrix of the node features ($N \times M$) where N is the count of nodes and M is the count of feature nodes. In our situation, we use the **hit coordinates as features**, so $\boldsymbol{M = 3}$;
  - **Ri** – matrix of the edges which are ending in the corresponging nodes with the size $N \times E$ (E is the count of edges). In this matrix $\mathrm{Ri}[i, j] = \mathbf{1}$ **if the edge** with the index $\boldsymbol{j}$ **ends on the node** with the index $\boldsymbol{i}$ and 0 otherwise;
  - **Ro** – matrix of the edges which are starting in the concrete nodes. Has the same properties as Ri but it is for output edges;
  - **Y** – neural network labels with the size of ($1 \times E$). $\mathrm{Y}[j] = \mathbf{1}$ if the edge with the index j belongs to the **real track** and **0** otherwise.
- Then, we have the 'Input network'. It is an MLP with 1 layer and Tanh activation function. The X matrix is applying to the Input network. The output of the Input network is moved to the 'Edge-Node' Network iterations.

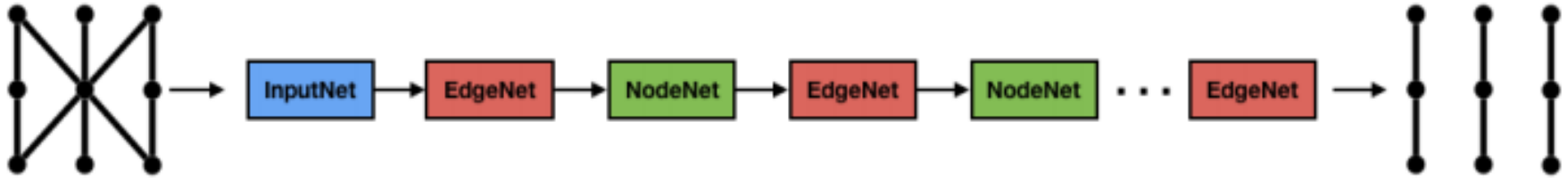


GNN Inside. Edge network. Node network.

- **Edge network** is the MLP with 2 layers. The activation function between layers is the same Tanh, but the activation of the output layer is the **Sigmoid function** which predicts the probability that concrete **edge is the true edge**.
- Edge network is a network which computes weights for edges of the graph.
  For each edge, **it selects** the associated nodes' **features** (from multiplying input data by Ri and Ro matrices) and then **applies network layers** with sigmoid activation.

- **Node network** is the MLP with 2 layers and Tanh activations. It **computes new** node **features** on the graph.
- For each node, it **aggregates** the **neighbor** node **features** (separately on the input and output side), **and combines** them **with** the node's **previous features** in a fully-connected network **to compute** the **new features**.

- This networks can be applied one after another as the iteration cycle. **Count of the iterations** is one of the **hyperparameters** of the full neural network.

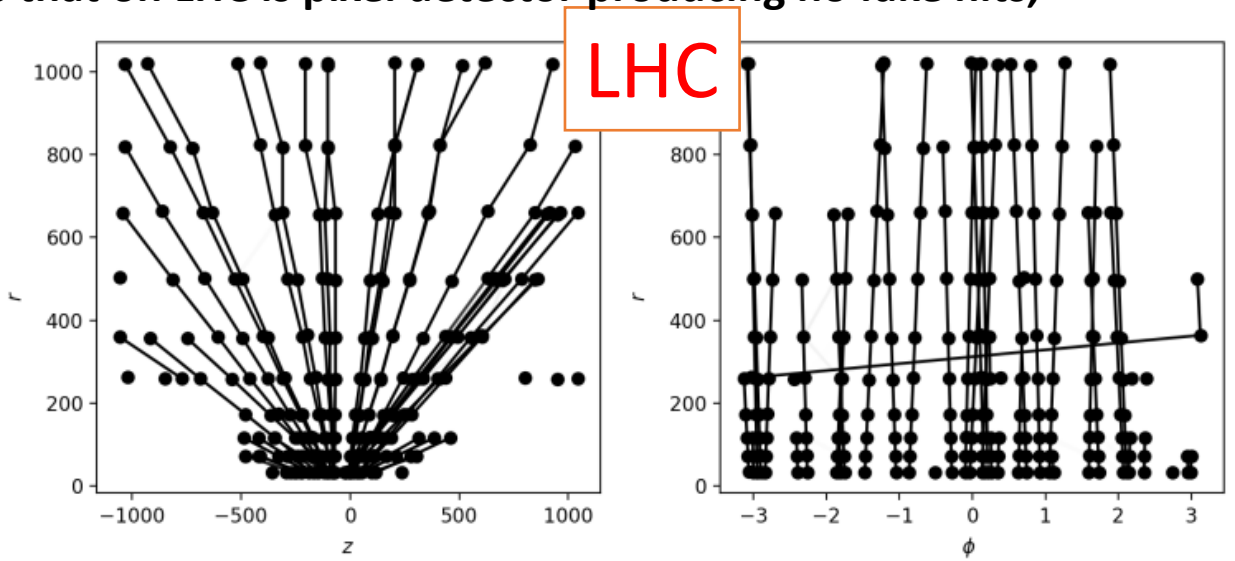


Graph neural network approach at LHC vs GEM

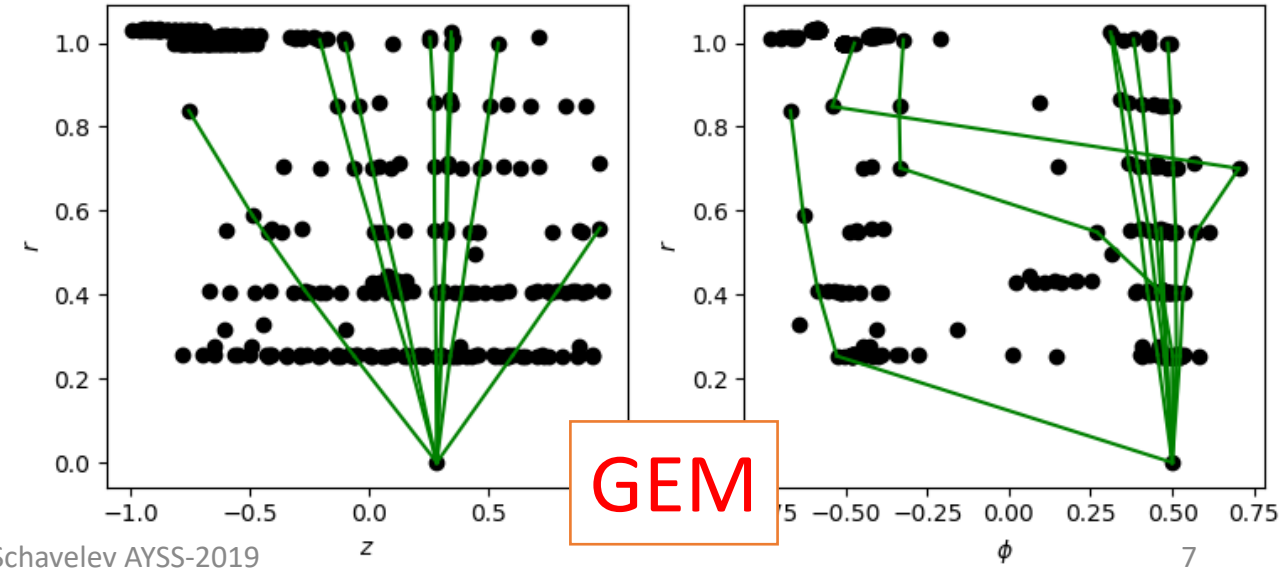
The main difference between the LHC and GEM data is that on LHC is pixel detector producing no fake hits, but in GEM - the majority of hits are fakes

- Model performs binary classification of segments between layers
- ‘1’ is for the true segment and ‘0’ is for the fake segment
- Perfect accuracy and results and achieves 97% accuracy, 74% precision, 94% recall (on LHC data)

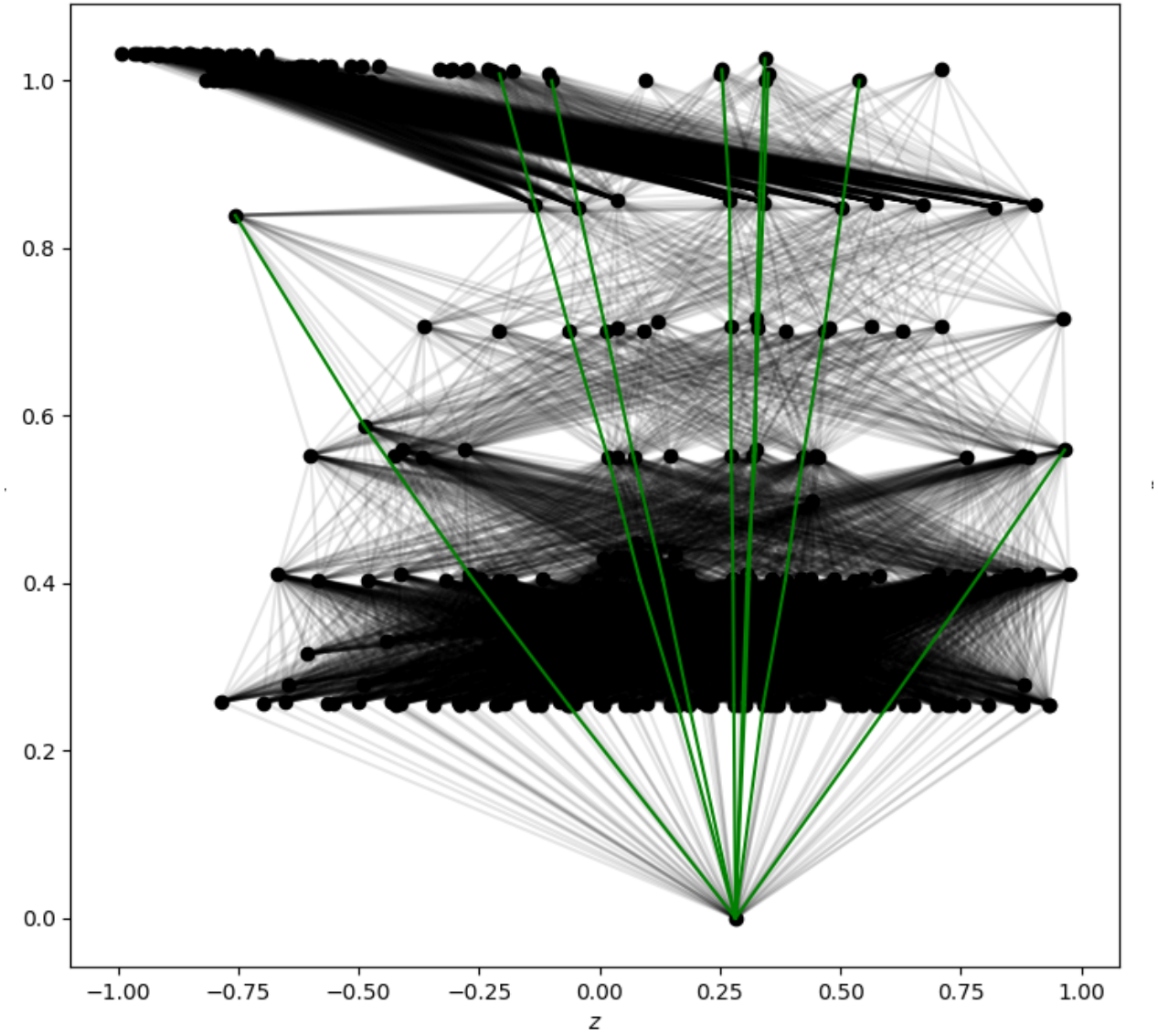
- BUT their **fake-to-real** ratios (about $\mathbf{O}(n)$) for their pixel detector are far from ($\mathbf{O}(n^2)$) for the GEM detectors



Graphs with the hits and true track trajectories for both detectors



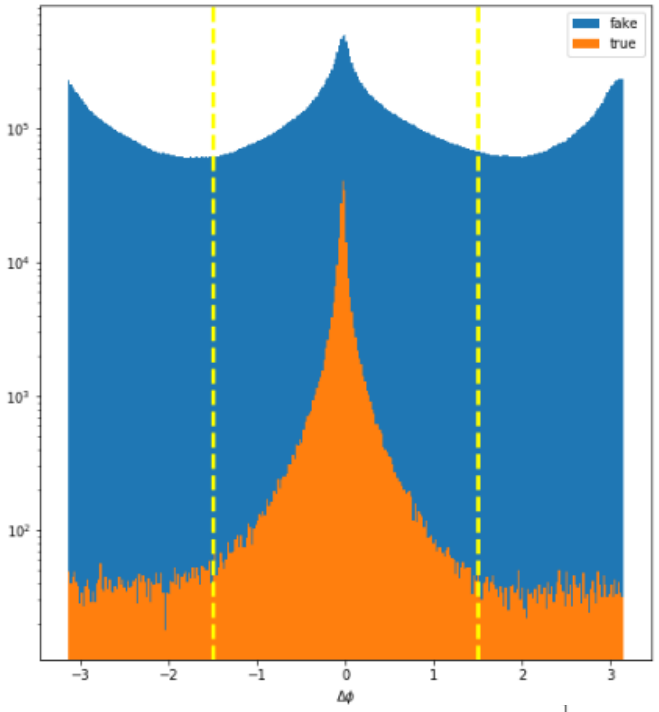
Applying GNN approach for GEM detectors

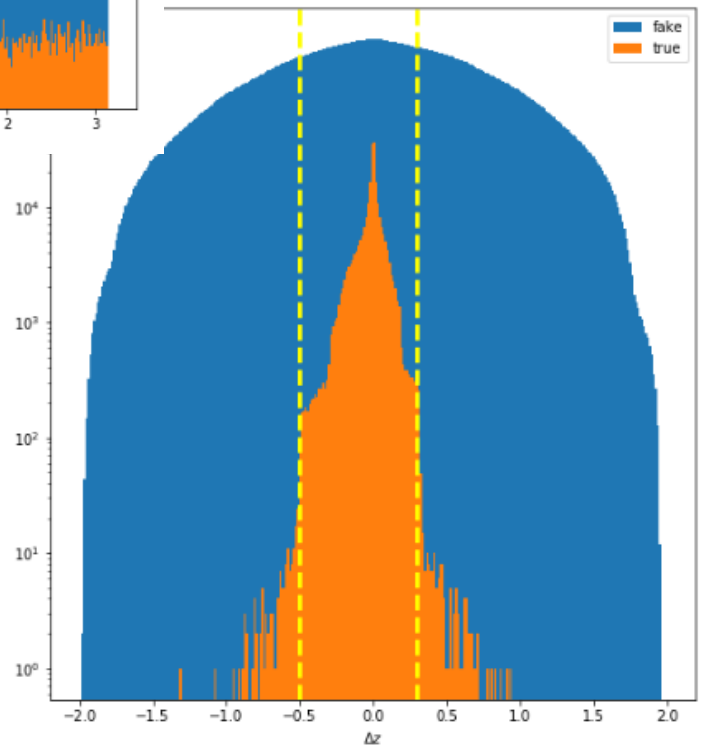


- Straightforward adapting the same method could not achieve even 20% precision with ~60% recall
- A huge amount of fakes is the root cause for network ambiguity
- Quite a lot of the connections can obviously be dropped with some basic criteria
- Also, the event graph can be split by slicing it into the sections

Analyzing the dataset

- We were using the same dataset operated by TrackNet
- For finding the criteria 50k events were preprocessed
- The preprocessing routine is defined by the following steps:
  1. Normalize X, Y, Z coordinates with detectors restrictions
     - X (-83.5, 80.5),
     - Y (-35, 25)
     - Z (32, 194)
  2. Convert X, Y, Z coordinates to the cylinder coordinates r, φ, z
  3. Construct segments pairs and visualize their $\Delta\varphi\Delta z$
  4. Apply found restrictions to the dataset
     - -1.5:1.5 for $\Delta\varphi$
     - -0.5:0.3 for Δz

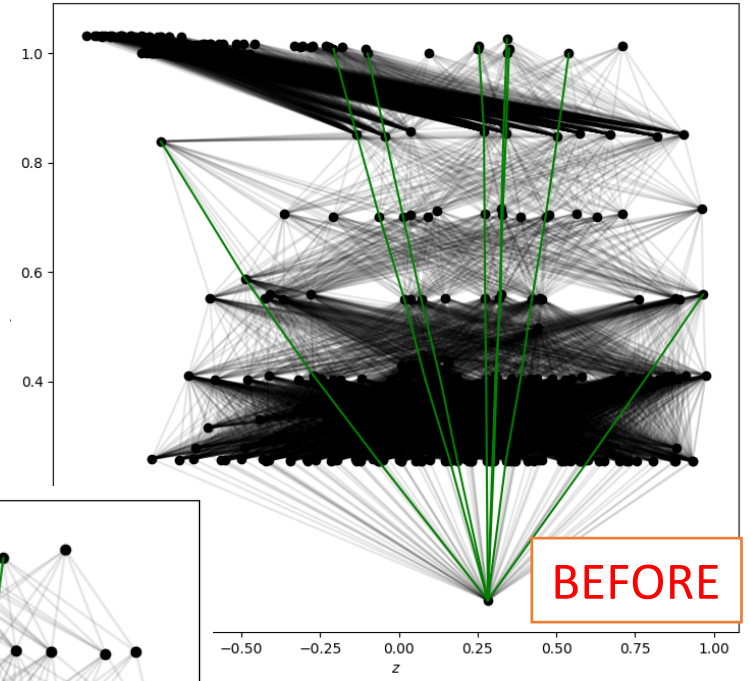


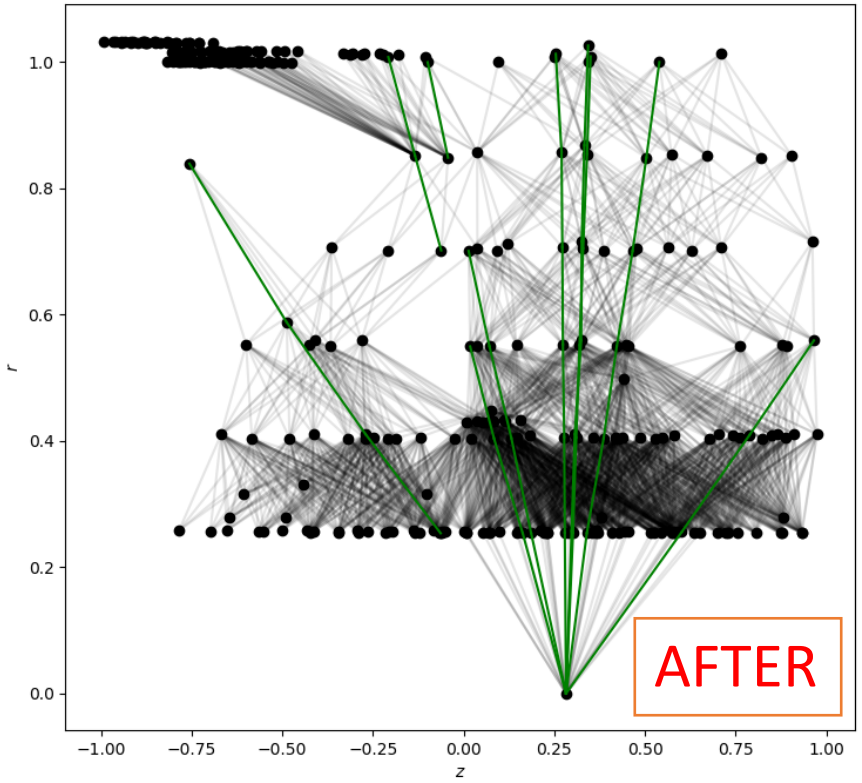


- Dataset fake-to-real segments factor decreased from 1 to 120 to 1 to 60 (**2 times**). After preprocessing we lost only 1.2% of true segments (total **98.8% purity**)

Decrease 2 times the ratio of real to fake hits

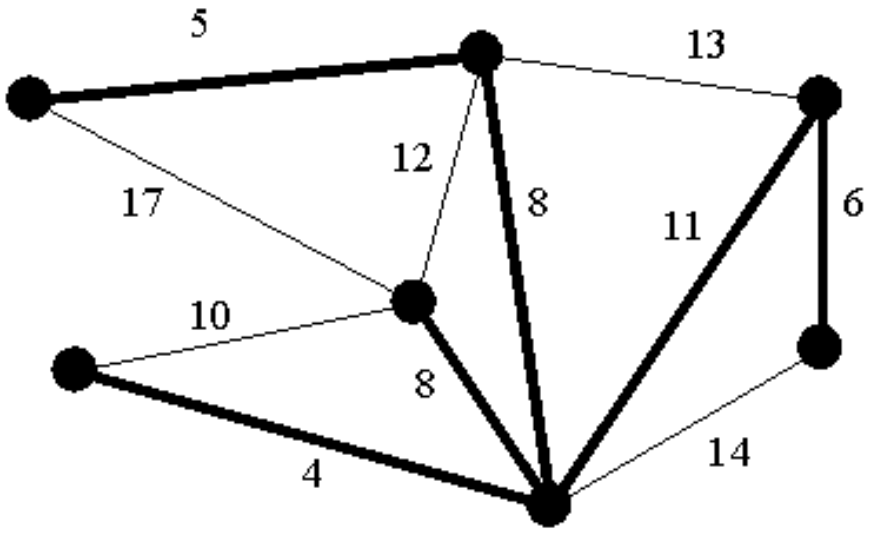
- After applying the criteria cut, the graph is not so overfilled with fakes, but they are still remain
- However, GNN is still struggling to solve this problem
- Now it achieves ~40% precision and 80% recall but it is still not applicable
- Need to radically decrease the fake factor





Minimum spanning tree approach

- **Spanning tree** is a subset of the edges of a connected, undirected graph that connects all the vertices together, without any cycles;
- **Minimum spanning tree** is a spanning tree for an edge-weighted graph with the minimum possible total edge weight;

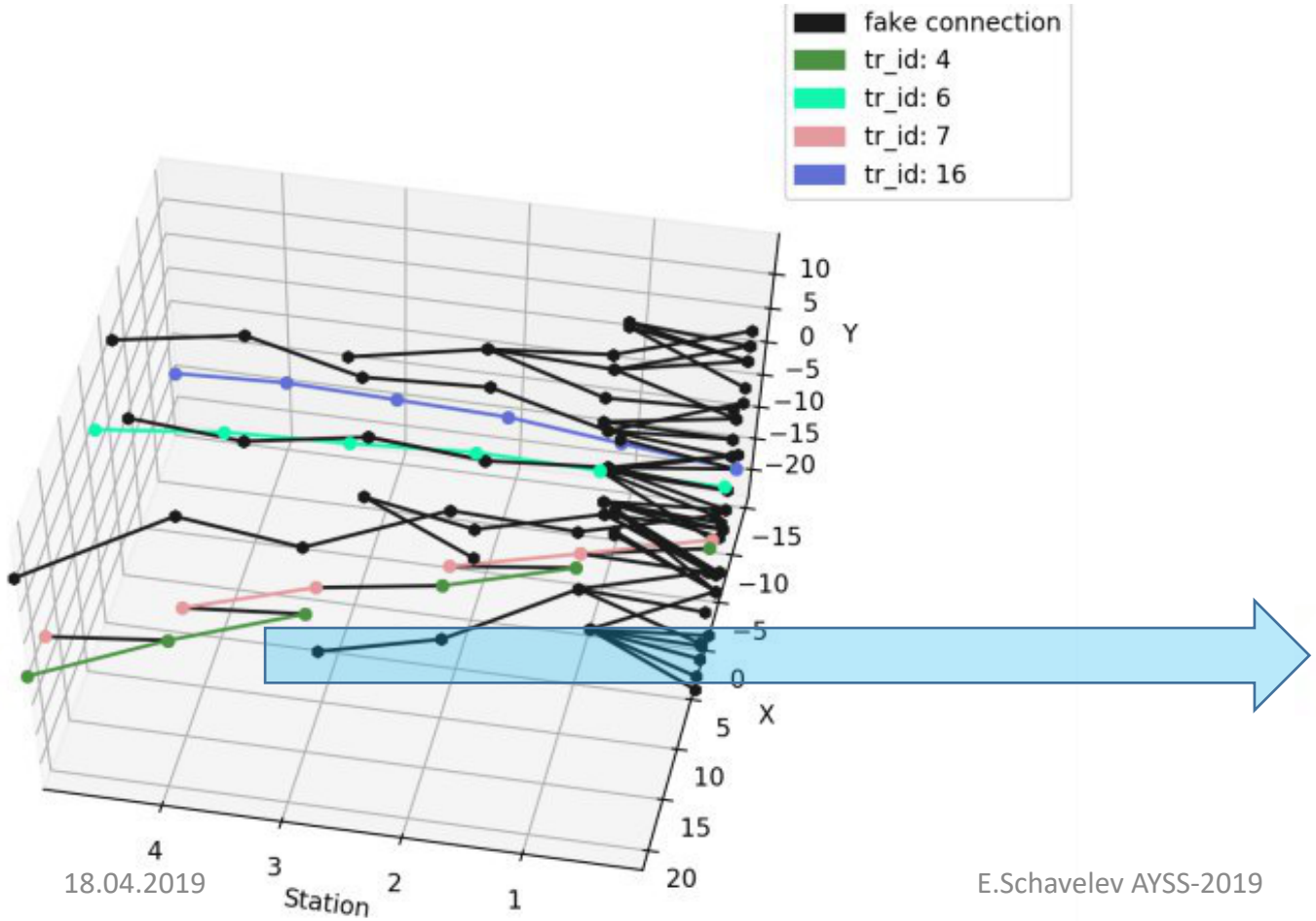


Minimum spanning tree in a arbitrary graph

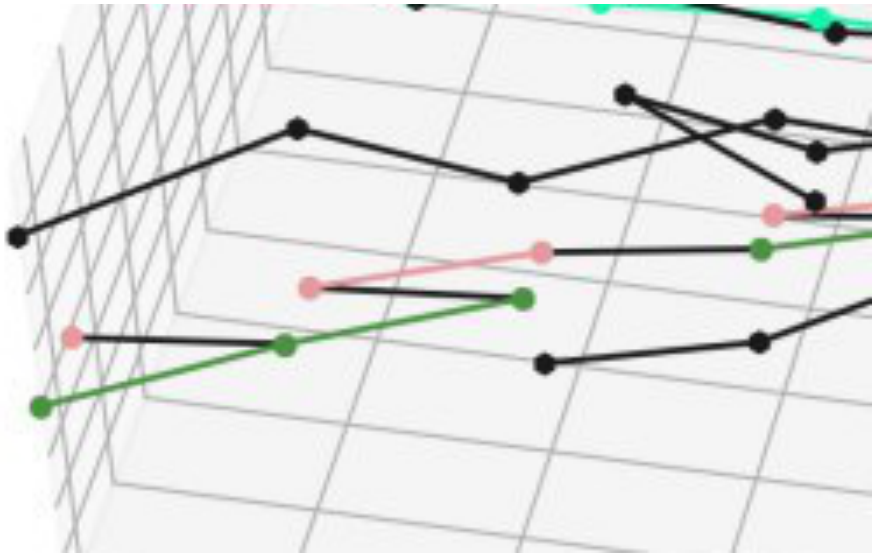
- We are taking current event-as-a-graph representation and introducing graph **edge weight** as a $\frac{cos^m(\varphi_i-\varphi_j)}{(r_i-r_j)^n}$ where:
  - i, j are indexes of adjacent layers hits;
  - φ, r are coordinates of a hit in a cylinder coordinate system;
  - m, n are arbitrary odd integer exponents.
- Also, we observed that **Euclidian distance** in cylinder coordinate space between two hits can be used as graph edge weight and it provides almost the same results.

Applying minimum spanning tree (MST)

- First results of applying MST to the XYZ event graph representation were already promising:



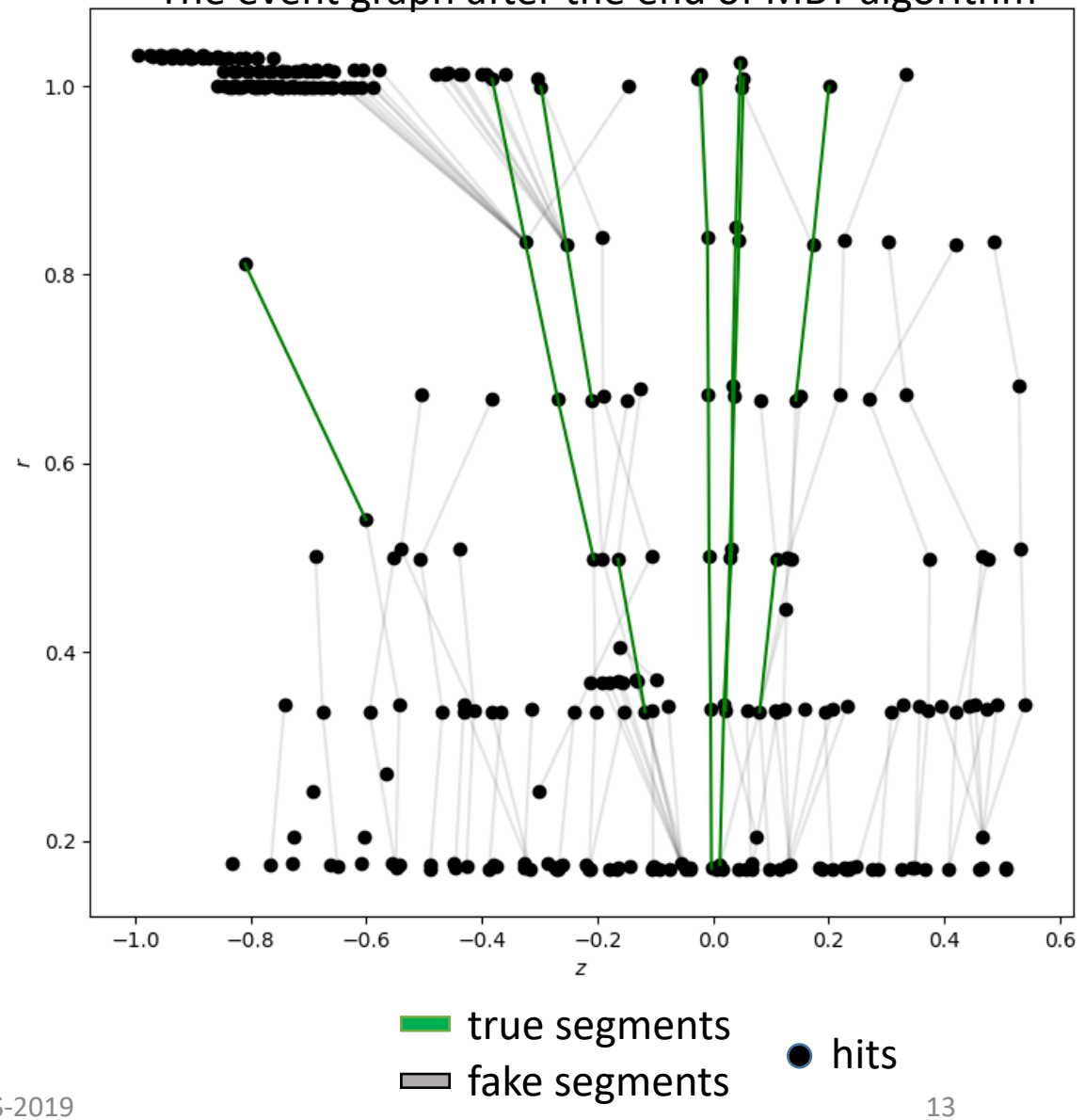
- But the graph is undirected, so the tree is considering “backward” connecting $(n+1)$-th layers to the n-th ones. So we need the directed version of the same algorithm.



Applying minimum branching tree (MBT)

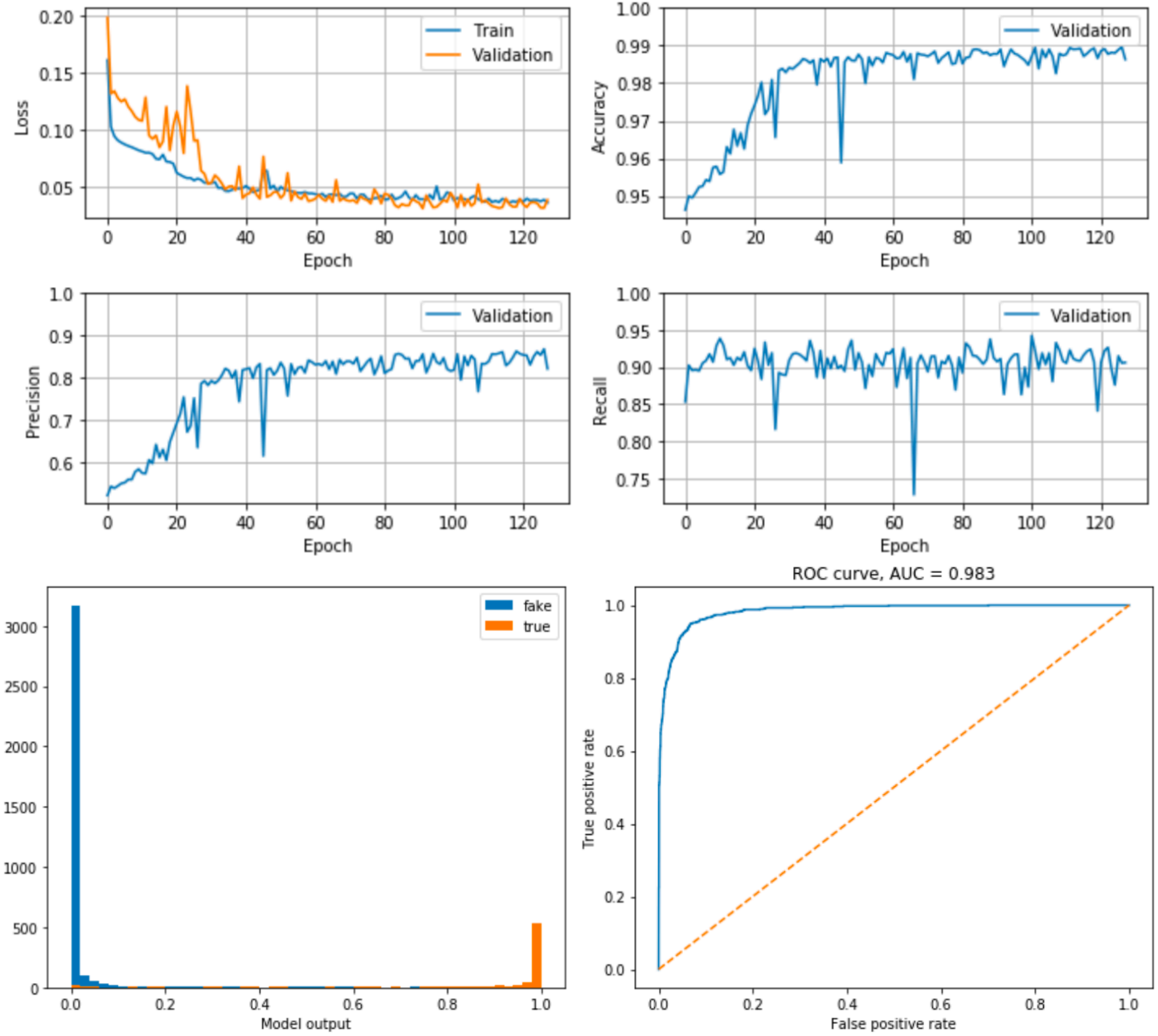
- So now we are dealing with the Minimum branching tree. The graph now is directed and edges are starting on the (N-1)-th station and ending on the N-th.
- Also, we are applying previous methods (reducing edges count via criteria and transforming from $\mathrm{X}, \mathrm{Y}, \mathrm{Z}$ to r, φ, z).
- Now, after preprocessing and MBT fake-to-real edges factor decreased by **12 times**.
- Real edges **“purity”** (amount of true edges left after all steps) is about **~82%**

The event graph after the end of MBT algorithm



Preliminary results of GNN approach for the GEM-based detectors

- After all processing our GNN is achieving on test dataset:
  - **99% accuracy**;
  - **92% recall**;
  - **85% precision**.
- These results were obtained after training on a “pet”-dataset with the size of 16k events;
- Tested on the 1k events;
- Already looks promising.



Conclusion

- Straightforward applying well-proven technologies from the LHC detectors did not provide any outcome;
- While searching solution for adapting GNN for GEM detectors we found interesting MBT approach which works immediately (no training required), provides promising results from start;
- We have significantly reduced the amount of fakes segments in event graphs with combined MBT+GNN approach while preserving 92% true segments left from the spanning tree;
- We are looking forward to improving graph thinning mechanism, because of promising pure-MBT results.

Graph neural network application to the particle track reconstruction for data from the GEM detector

Shchavelev Egor

Pavel Goncharov, Gennady Ososkov, Dmitriy Baranov

Saint Petersburg State University

egor.schavelev@gmail.com