Graph neural network with an attention mechanism for clustering particle tracks by events in the SPD experiment at the NICA accelerator

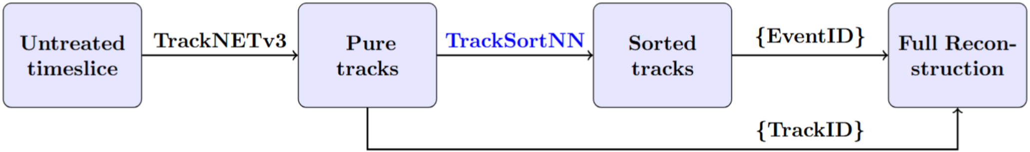
[1,2]Omelianchuk S.S., [2,3]Ososkov G.A., [2]Talochka E.N.

[1]Branch of MSU in Dubna,
[2]JINR LIT
[3]Dubna University

June 03, 2025

Introduction and Motivation

The task is the classification of reconstructed tracks by event
The approach uses deep ML methods for reconstruction and classification.



Results of a **Siamese Neural Network** with an encoder[M.Borisov,P.Goncharov... 2024].

	Fixed	**Unfixed**
Metrics	100 epochs	100 epochs
Precision	0,811	0,574
Recall	0,843	0,651
Accuracy	0,895	0,587

Disadvantages

1. Inability to evaluate with missing hits.
2. Necessity of information about the number of events in the timeslice.

Datasets

General algorithm for timeslice generation:

$$n_{tracks} \sim U[2, n_{tracks}^{max}] \qquad \mathrm{P}(N_{events} = k) = \frac{\overline{N}_{events}^{k}}{k!} e^{-\overline{N}_{events}} \qquad n_{fakes} = 0$$

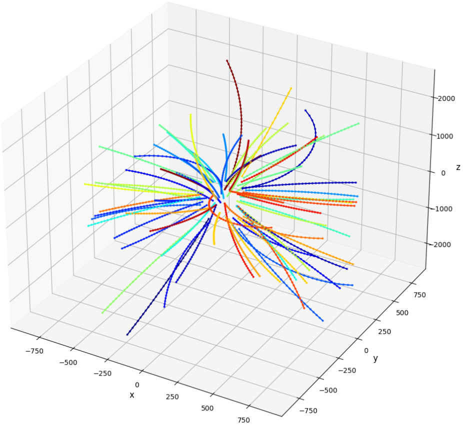
Tracks from SPD simulation:

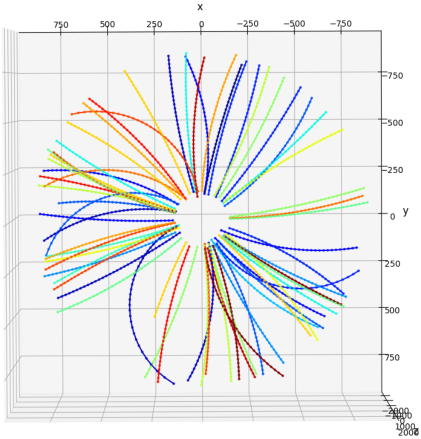
- Key characteristics:
  1. All hits in a track are limited to $r_\phi^{min} = 150\,mm, \quad r_\phi^{max} = 850\,mm$ and are equidistant in $r_\phi = \sqrt{x^2 + y^2}$.
  2. Each track consists of a large number of hits in the range $(28, 35)$.

Tracks from TrackML:

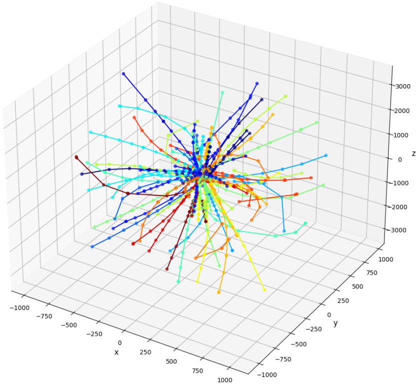
- Key characteristic:
  1. All hits in a track are limited to $r_\phi^{min} = 50\,mm, \quad r_\phi^{max} = 1000\,mm$ and not equidistant in $r_\phi = \sqrt{x^2 + y^2}$.
  2. Each track consists of a small number of hits in the range $(3, 20)$.

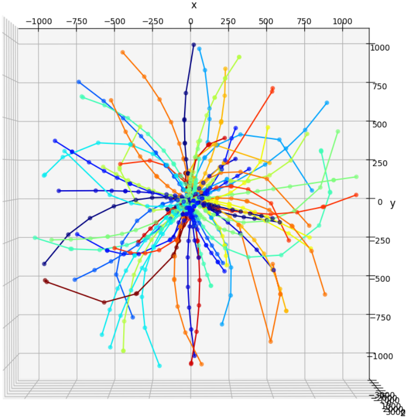
SPD Timeslices





TrackML Timeslices





Graph Neural Networks

The main idea is use GANN for permutation-invariant computations.

Advantages:

1. Independence from the number of hits in a track and the number of events in the timeslice.
2. Success of the graph model in track reconstruction at LHC[arXiv:2007.13681v2].

Disadvantages:

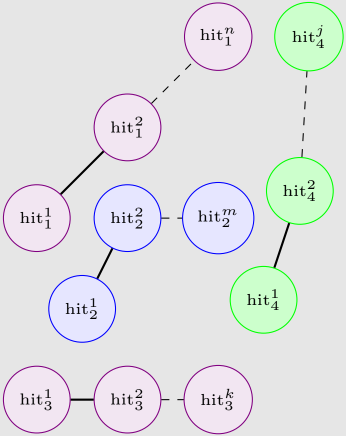
1. Large computational graphs.
2. Sensitivity to class imbalance.

Graph representation

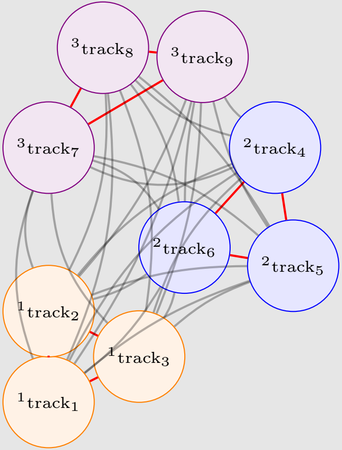
Timeslice

Timeslice				
x	y	z	tr_{id}	ev_{id}
-118.7	26.4	278.6	0	0
-138.8	31.3	301.7	0	0
-158.9	36.3	325.0	0	0
-21.6	128.4	26.4	1	0
-25.6	148.7	11.5	1	0
-29.3	169.1	-3.4	1	0
110.3	101.0	63.5	2	1
125.7	114.7	42.3	2	1
141.2	128.2	20.8	2	1
156.9	141.6	-0.5	2	1

HitGraph

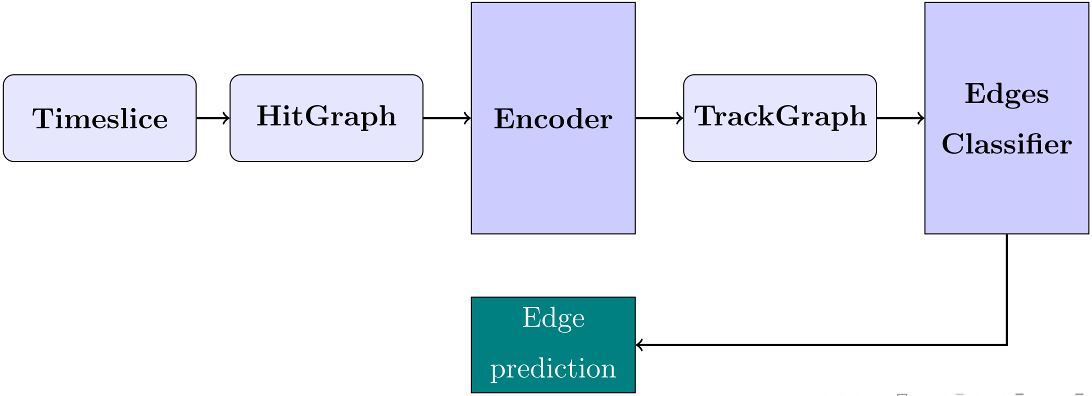


TrackGraph



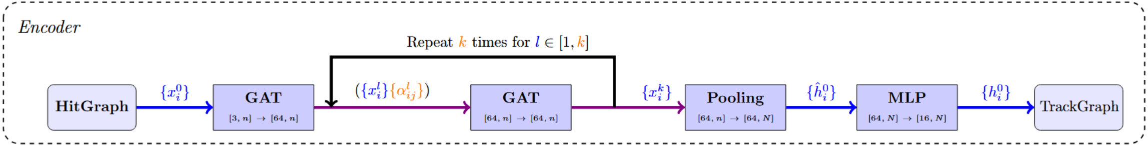
Model GANN

The **general architecture** of the model contains two main blocks: **Encoder** and **Classifier**.



Encoder

Encoder architecture:

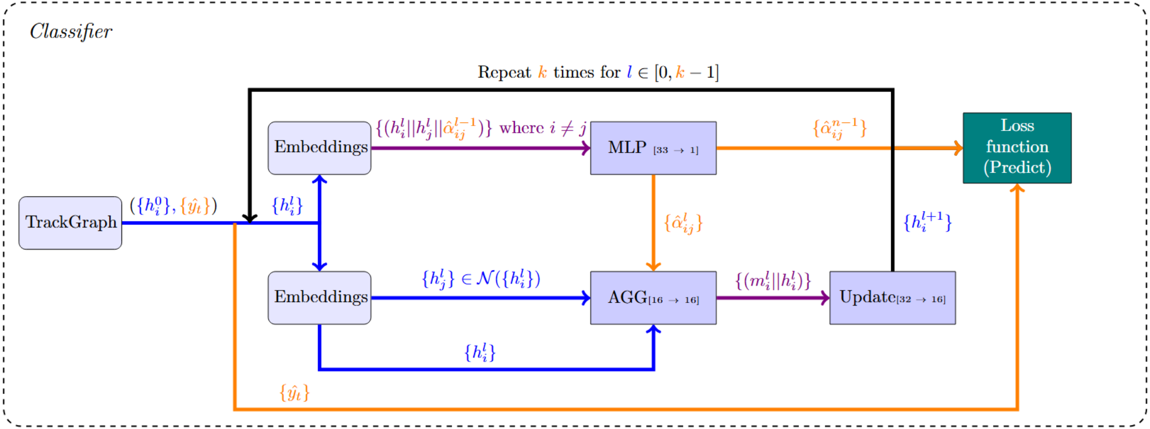


The encoder works as follows:

$$x_i^l = \sum_{j \in \mathcal{N}(i)} \alpha_{ij}^l \mathbf{W}^l \cdot \mathrm{Tanh}\Big(\sum_{k \in \mathcal{N}(j)} \alpha_{jk}^{l-1} \mathbf{W}^{l-1} x_k^{l-1}\Big)\Big) \qquad \hat{h}_i^0 = \frac{1}{N_i} \sum_{j=1}^{N_i} x_j^k$$

$$\alpha_{ij}^l = \frac{\mathrm{Exp}[\mathrm{LeakyReLU}(\mathbf{a}^T[\mathbf{W}^l x_i^l + \mathbf{W}^l x_j^l])]}{\sum_{k \in \mathcal{N}(i)} \mathrm{Exp}[\mathrm{LeakyReLU}(\mathbf{a}^T[\mathbf{W}^l x_i^l + \mathbf{W}^l x_k^l])]}$$

Edges Classifier



Loss function and Metrics

- Balanced Focal Loss:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\omega_i(1-p_t)^{\gamma}\ln(p_t) \qquad \omega_i = y_i\omega_p + (1-y_i)\omega_n$$

where y_i is the truth label of edge, $p_t = p_i$ if the edge is false, $p_t = (1-p_i)$ if the edge is true, p_i is the model prediction, N is the number of edges, $\omega_p, \omega_n, \gamma$ are adjustable parameters.

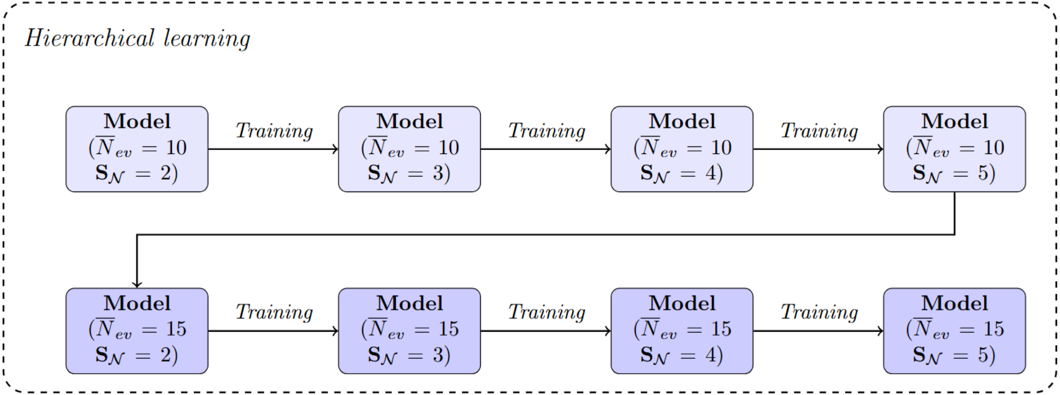
- **Metrics**:

$$\text{Precision} = \frac{TP}{TP+FP} \quad \text{Recall} = \frac{TP}{TP+FN} \quad \text{Accuracy} = \frac{TP+TN}{TP+TP+FP+FN}$$

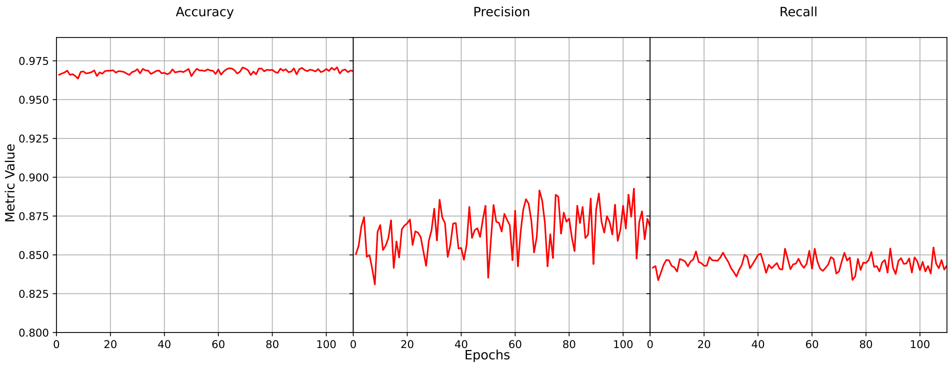
The i-th edge is true if the model's prediction $p_i > 0.5$

Hierarchical learning

The main idea is to train model on simple examples with parameters $S_{\mathcal{N}}, \overline{N}_{ev}$. The model is trained until the accuracy drops below 90%. With an samples: 700 train, 300 test.

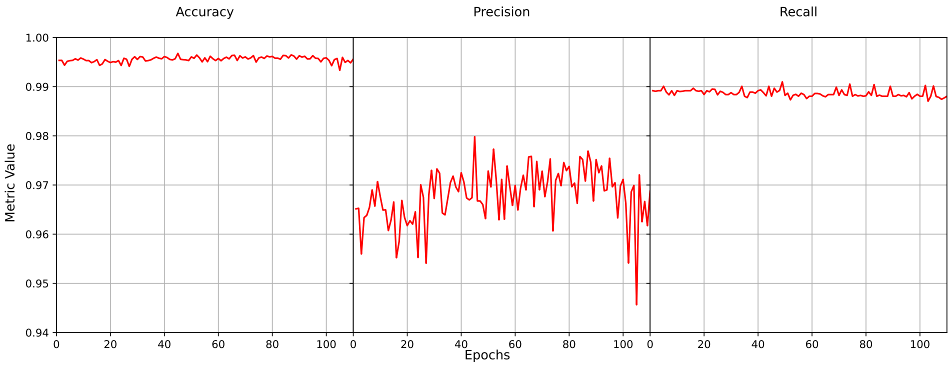


Results



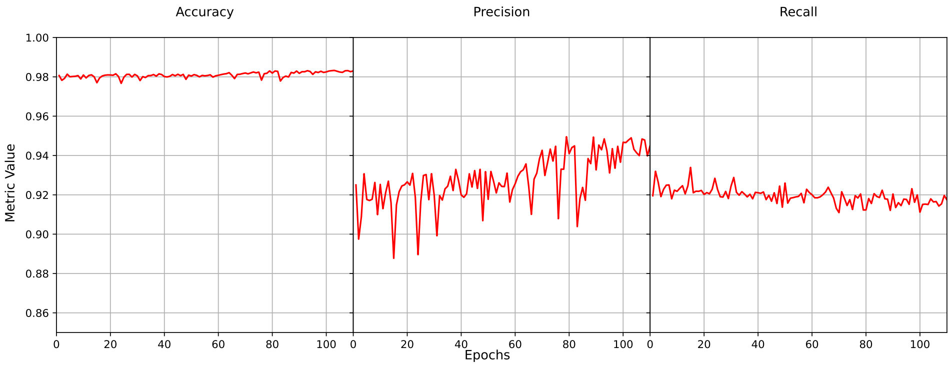
Metrics of the trained model with SPD simulation ($\overline{N}_{ev} = 10, S_{\mathcal{N}} = 5$), parameters ($\omega_p = 1, \omega_n = 0.4, \gamma = 1.5$), on SPD timeslices with ($\overline{N}_{ev} = 10, S_{\mathcal{N}} = 5$)

Results



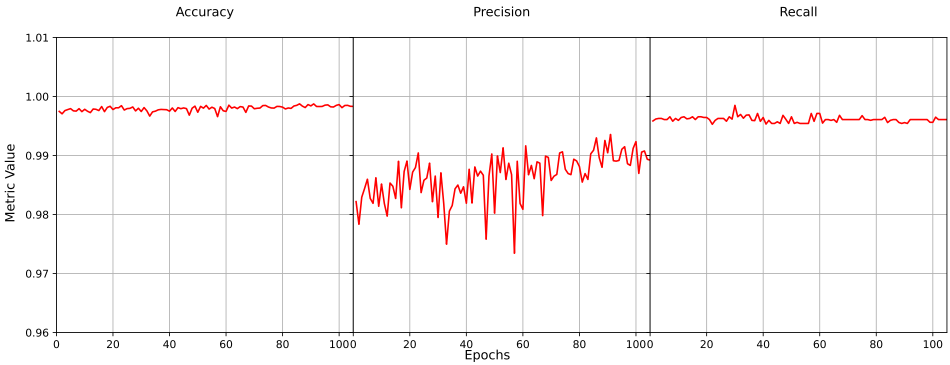
Metrics of the trained model with SPD simulation ($\overline{N}_{ev} = 15, S_{\mathcal{N}} = 5$), parameters ($\omega_p = 1, \omega_n = 0.6, \gamma = 1.5$), on SPD timeslices with ($\overline{N}_{ev} = 15, S_{\mathcal{N}} = 5$)

Results



Metrics of the trained model with TrackML timeslices $(\overline{N}_{ev} = 10, S_{\mathcal{N}} = 4)$, parameters $(\omega_p = 1, \omega_n = 0.5, \gamma = 1.5)$, on TrackML timeslices with $(\overline{N}_{ev} = 10, S_{\mathcal{N}} = 4)$

Results



Metrics of the trained model with TrackML timeslices ($\overline{N}_{ev} = 15, S_{\mathcal{N}} = 4$), parameters ($\omega_p = 1, \omega_n = 0.6, \gamma = 1.5$), on TrackML timeslices with ($\overline{N}_{ev} = 15, S_{\mathcal{N}} = 4$)

Conclusions

Successes:

1. The model demonstrates excellent metrics in the initial simulation.
2. The model does not require information about the number of events for correct evaluations.
3. Hierarchical learning improves results on both complex and simple examples.

Limitations:

1. Computing speed on an Nvidia V100 Tesla GPU: 4.0 timeslice/sec.
2. The model requires good time resolution in the experiment.

Prospects:

1. Testing the model on a complete and rigorous simulation of the SPD experiment.

Thank you for your attention!