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

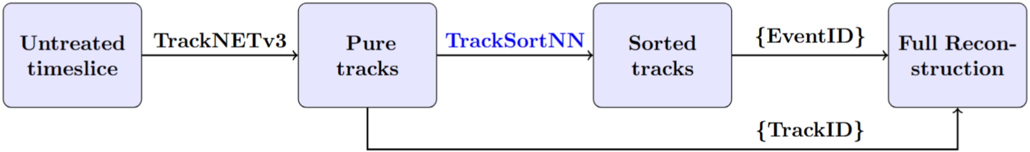
[1,2]Omelianchuk S.S., [2,3]Ososkov G.A., [2]Talochka E.N.

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

September 10, 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.

Graph Neural Networks

The basic idea is to transform the tracks into a supergraph nodes where event classification will be performed.

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.

General algorithm for timeslice generation:

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

Tracks from SPD simulation:

- Key characteristics:
  1. All hits in a track are limited to $r_{\phi}^{min} = 150\,mm$, $r_{\phi}^{max} = 850\,mm$ 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:

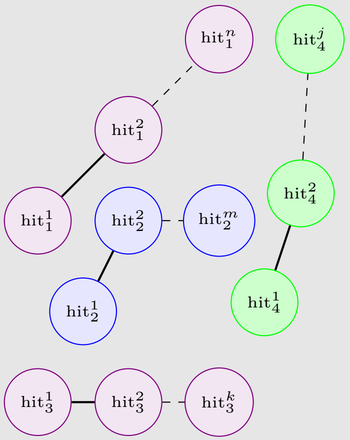
- Key characteristic:
  1. All hits in a track are limited to $r_{\phi}^{min} = 50\,mm$, $r_{\phi}^{max} = 1000\,mm$ and not equidistant in $r_{\phi} = \sqrt{x^2 + y^2}$.
  2. Each track consists of a small number of hits in the range $(3, 20)$.

Data

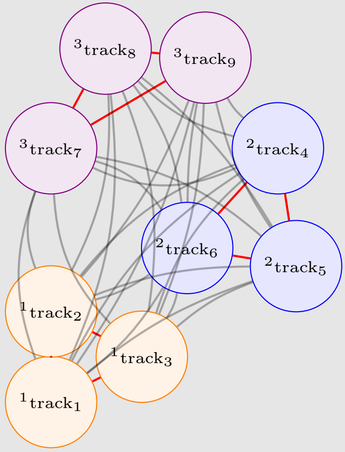
x	y	z	track_id	event_id	time_left	time_right
-35.64	-132.12	-115.31	0	9	0.384	0.999
-41.67	-151.88	-113.44	0	9	0.384	0.999
-47.79	-171.59	-111.46	0	9	0.384	0.999
-54.03	-191.24	-109.38	0	9	0.384	0.999
-32.40	-552.65	247.86	1	0	0.000	0.307
-36.00	-573.12	254.47	1	0	0.000	0.307
-39.79	-593.44	261.17	1	0	0.000	0.307
-43.74	-613.80	268.04	1	0	0.000	0.307
-47.87	-634.24	274.66	1	0	0.000	0.307
-52.16	-654.65	281.30	1	0	0.000	0.307
210.12	710.05	-1167.82	2	13	0.692	1.00
210.25	731.33	-1202.57	2	13	0.692	1.00
209.99	752.79	-1237.73	2	13	0.692	1.00
209.35	774.46	-1272.85	2	13	0.692	1.00
208.41	795.95	-1307.93	2	13	0.692	1.00

Graph representation

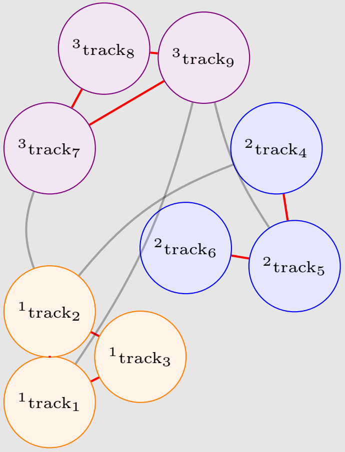
HitGraph



TrackGraph

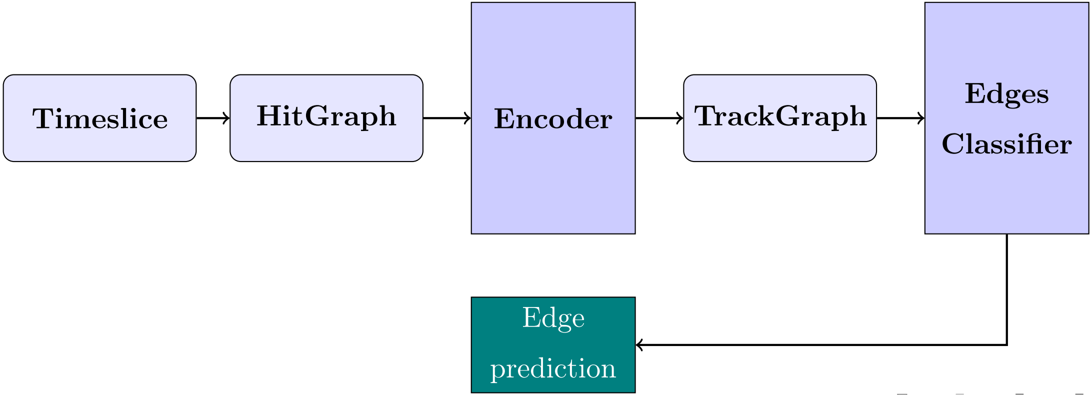


FinalGraph



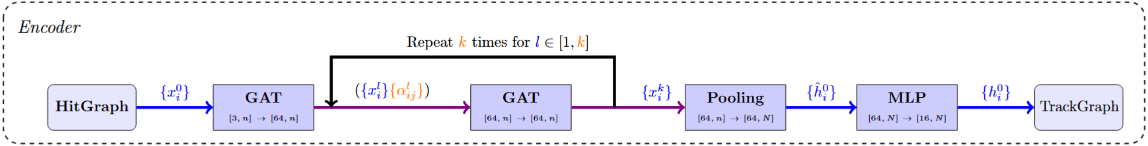
Model GANN

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



Encoder

Encoder architecture:



The encoder works as follow$^{arxiv.org/abs/1710.10903}$:

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

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

Graph construct

For each track in timeslice are a **drift time** intervals generation according to the formulas:

$$t_d^i = \frac{i}{N_{ev} - 1}, \qquad \Delta t_d = \frac{t_d^{r+1} - t_d^0}{2} - \varepsilon, \qquad t_{L/R}^i = t_d^i \mp \Delta t_d$$

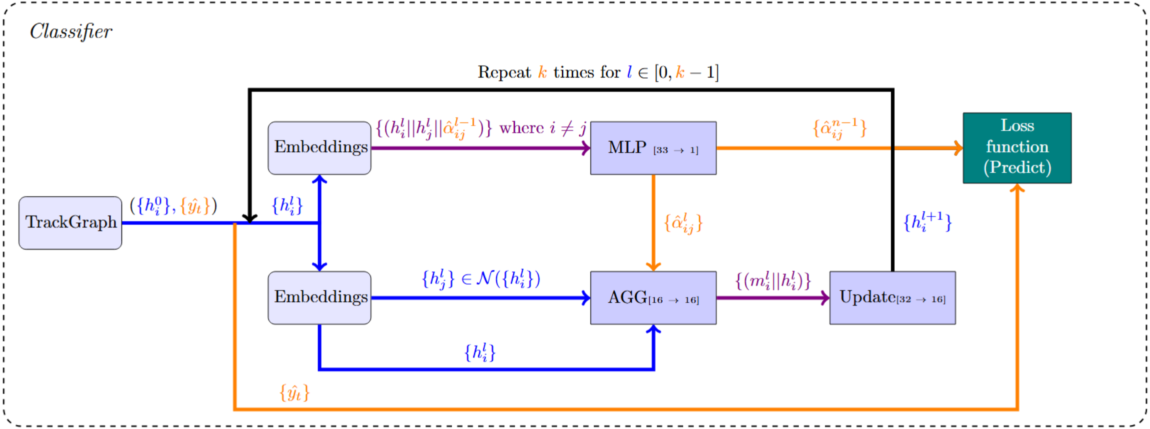
where r - the parameter that limits the number of candidates in one event for each track. Its physical meaning is the time resolution of the detector. This parameter is set manually during the data generation stage and determines the initial number of false edges in the SuperGraph.

An edge connects the i-th and j-th tracks only if:

$$[t_L^i, t_R^i] \cap [t_L^j, t_R^j] \neq \emptyset$$

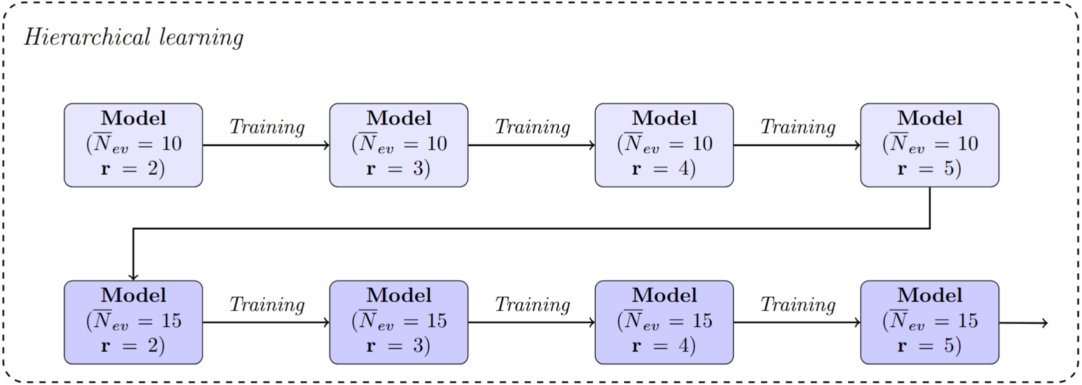
The drift time rate Δt_d is the same for all tracks.

Edges Classifier



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: 800 train, 300 test.



Loss function and Metrics

- Balanced Focal Loss[arxiv.org/abs/1708.02002]:

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

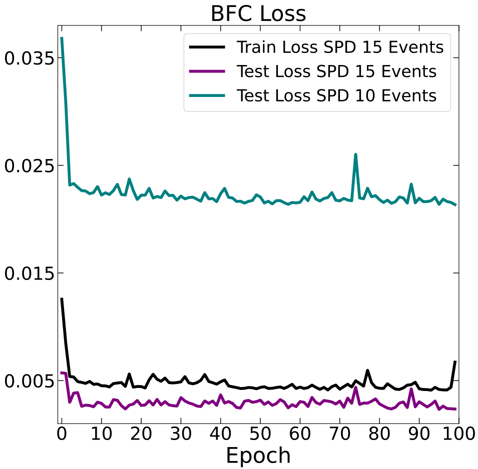
where y_i is the truth label of edge, $p_t = p_i$ if the edge is false, $p_t = (1-p_i)$ if the edge is true, p_i is the model prediction, N is the number of edges, $\omega_p, \omega_n, \gamma$ are adjustable parameters.

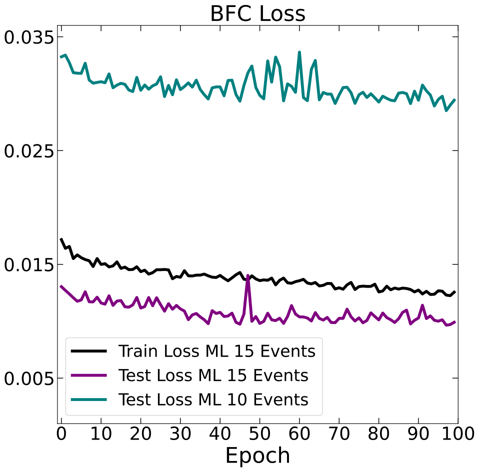
- **Metrics**:

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

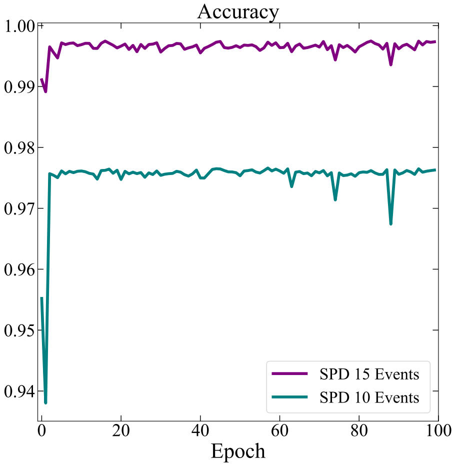
The i-th edge is true if the model's prediction $p_i > 0.5$

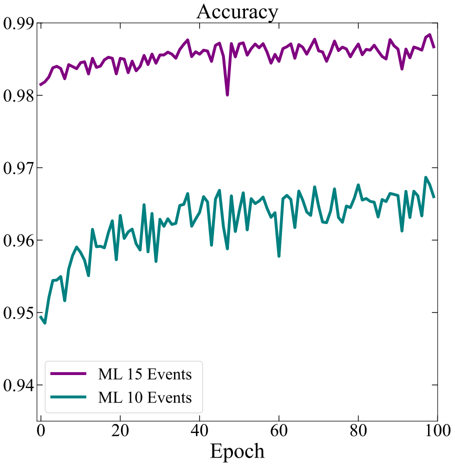
Results for $r = 5$



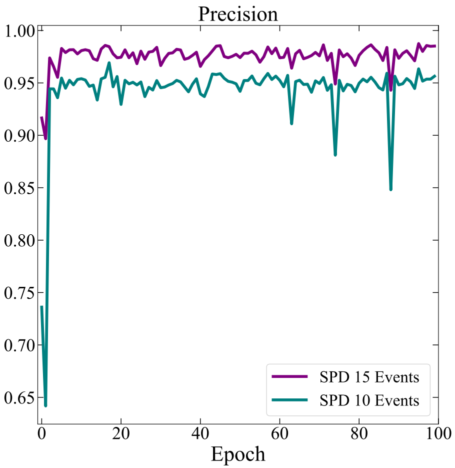


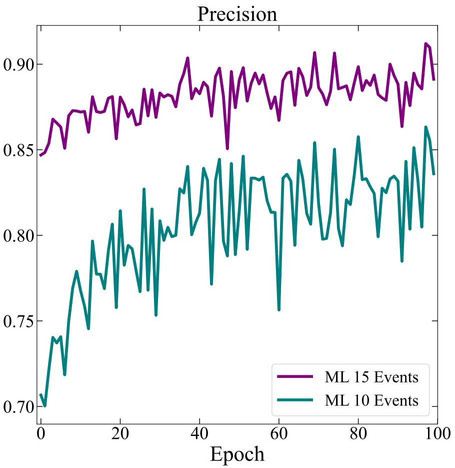
Results for $r = 5$



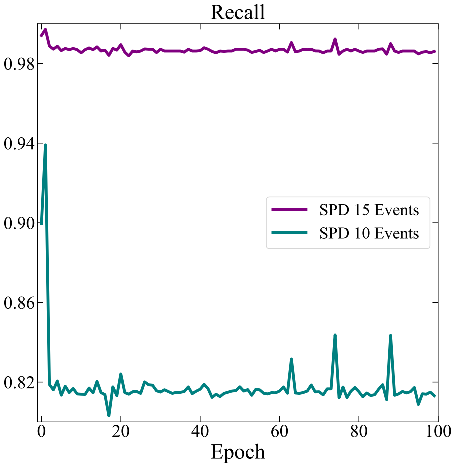


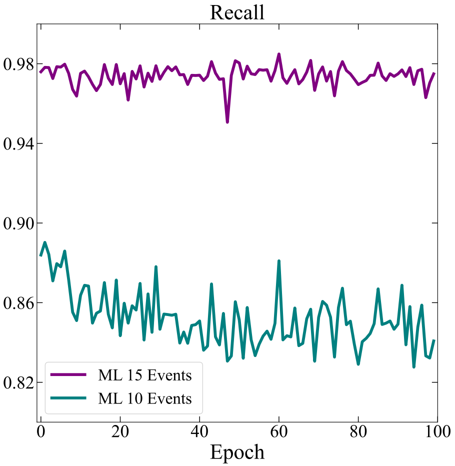
Results for $r = 5$





Results for $r = 5$





Final metrics

The last stage is the final graph clustering. We use the **Louvain method** to remove residual edges[arxiv.org/abs/0803.0476].

ARI			
	$r = 5$	$r = 6$	$r = 7$
$\overline{N}_{ev} = 10$	0.831 ± 0.215	0.675 ± 0.240	0.611 ± 0.225
$\overline{N}_{ev} = 15$	0.950 ± 0.156	0.908 ± 0.157	0.801 ± 0.223

NMI			
	$r = 5$	$r = 6$	$r = 7$
$\overline{N}_{ev} = 10$	0.937 ± 0.075	0.886 ± 0.083	0.874 ± 0.067
$\overline{N}_{ev} = 15$	0.979 ± 0.067	0.971 ± 0.055	0.937 ± 0.076

Table: Quality metrics of the final clustering for different timeslice configurations

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: 4.5 timeslice/sec.
2. The model requires good time resolution in the experiment.

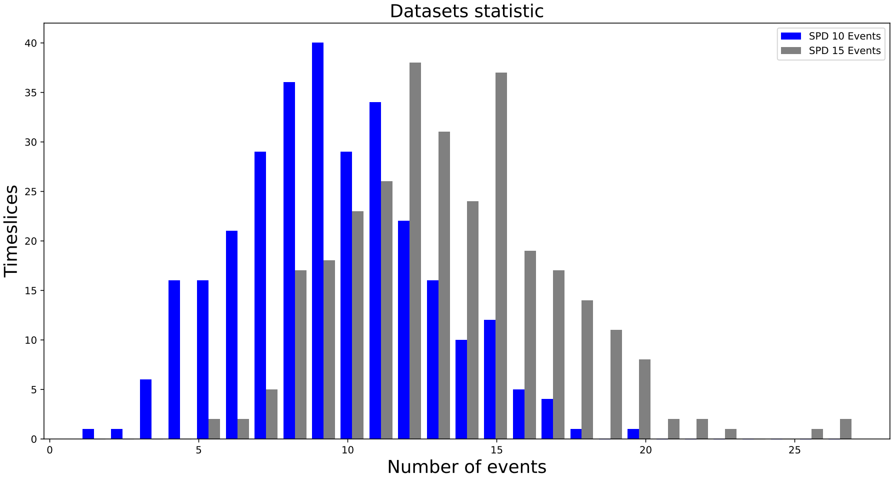
Prospects:

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

Thank you for your attention!

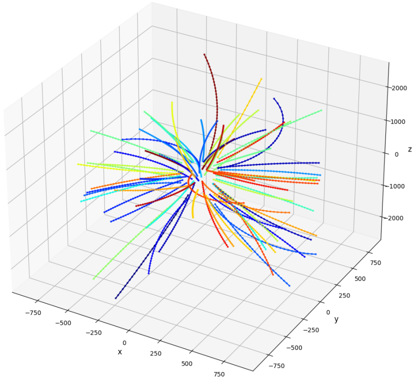
TrackGANN: https://github.com/SavelyOm/TrackGANN/

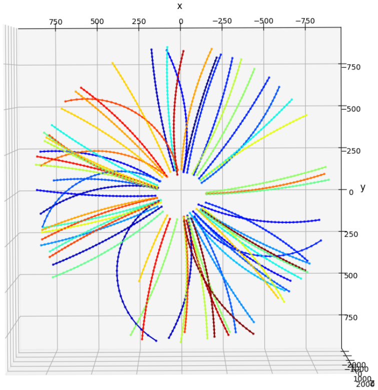
Backup



Datasets statistic
SPD 10 Events
SPD 15 Events
Timeslices
Number of events
0
5
10
15
20
25
30
35
40
0
5
10
15
20
25

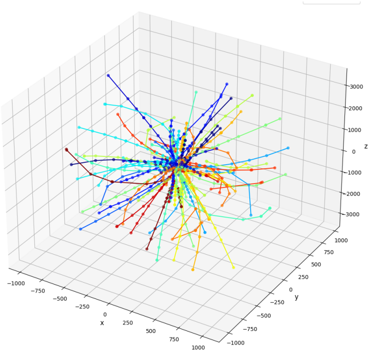
Backup

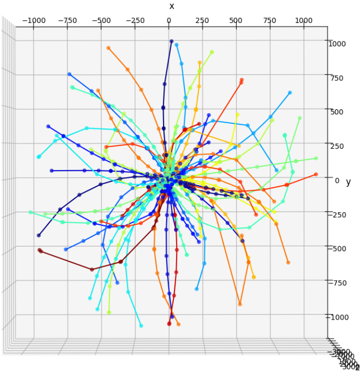




SPD timeslices with the parameters $\overline{N}_{ev} = 15$ and $n_{tr}^{max} = 6$

Backup





TrackML custom timeslices with the parameters $\overline{N}_{ev} = 15$ and $n_{tr}^{max} = 6$