

Разделение траекторий частиц по событиям столкновений пучков накопленных в едином временном срезе в детекторе SPD на ускорителе NICA с помощью графовых нейронных сетей

^{1,2}Омелянчук С.С, ^{2,3}Ососков Г.А, ²Талочка Е.Н

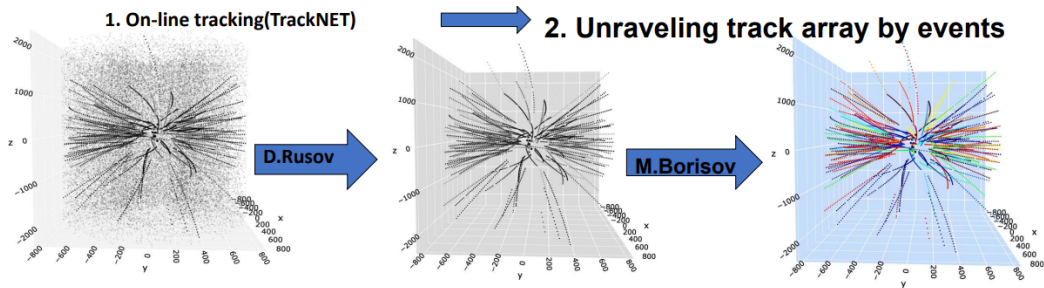
¹Филиал МГУ, г. Дубна,

² ОИЯИ ЛИТ

³ Университет Дубна

Апрель 09, 2025

Задача: классификация трековых событий регистрируемых детектором SPD.
Подход: глубокие нейронные сети для отчистки и классификации.



Процесс обработки экспериментальных данных. Отчистка треков от ложных хитов
TrackNETv3 [O.Bakina,D.Baranov... 2022](#). Классификация отчищенных треков.

Сиамская нейронная сеть, основанная на кодировщике треков и "one-shot" подходе к обучению [M.Borisov,P.Goncharov... 2024](#).

Fixed	
Metrics	100 epochs
Silhouette	0,725
DB	0,655
Precision	0,811
Recall	0,843
F1-score	0,818
Accuracy	0,895

UnFixed	
Metrics	100 epochs
Silhouette	0,342
DB	0,922
Precision	0,574
Recall	0,651
F1-score	0,607
Accuracy	0,587

Недостатки:

- 1 Плохие результаты при работе с нефиксированным числом событий.
- 2 Чувствительность к пропущенным хитам.

Основная идея графовой нейронной сети: работа с данными представленными в виде графа. [LingfeiWu,PengCui... 2022](#)

Преимущества:

- 1 Адаптивность к количеству хитов в треке и событий в таймслейсе.
- 2 Успехи генеративных моделей в биохимии и медицине.
- 3 Успехи графовой модели в реконструкции треков на эксперименте CMS.

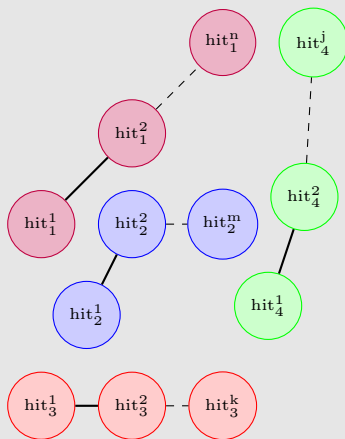
Недостатки:

- 1 Большие графы вычислений.
- 2 Сильная чувствительность к дисбалансу классов.

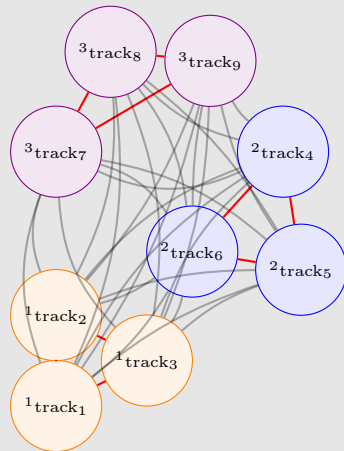
Timeslice

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

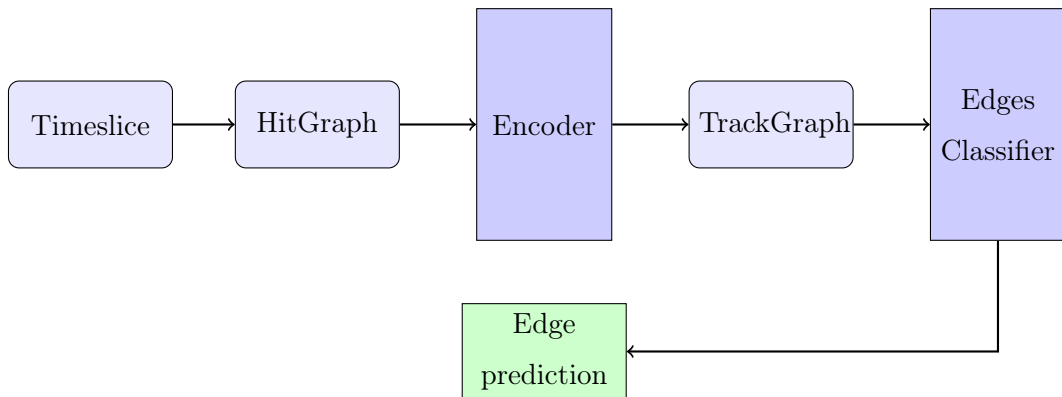
HitGraph



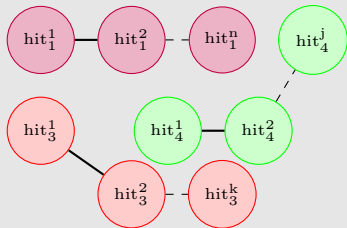
TrackGraph



Общая архитектура модели включает в себя 2 основных блока: Encoder и Classifier.



HitGraph*



$$\mathbf{H}^* = [\text{NHits}, \text{NFeatures}^*]$$

$$\mathcal{E}^* = [2, \text{NEdges}^*]$$

$$\mathbf{T}^* = [\text{NHits}]$$

Pooling+MLP

- 1 Преобразование формы графа:

$$\text{globalMaxPool}(\mathbf{H}^*, \mathcal{E}^*, \mathbf{T}^*) \rightarrow (\mathcal{H}^*, \mathcal{E}^*, \mathbf{T}^*)$$

$$\text{где: } \mathcal{H}^* = [\text{NTracks}, \text{NFeatures}^*]$$

- 2 Окончательное кодирование треков:

$$\text{MLP}(\mathcal{H}^*) \rightarrow \mathcal{H}$$

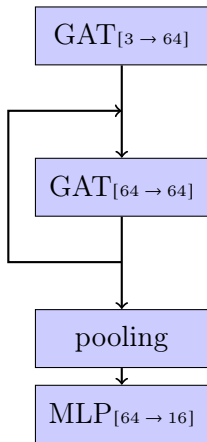
$$\text{где } \mathcal{H} = [\text{NTracks}, \text{Embeddings}]$$

- 3 Построение нового Суперграфа:

$$\mathcal{G} = (\mathcal{H}, \mathcal{E}, \mathbf{T}) \quad \mathbf{T} = [\text{NEdges}]$$

после полного кодирования имеем TrackGraph

Encoder

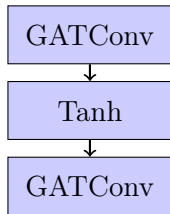


Графовый сверточный слой с механизмом внимания
GATConv:

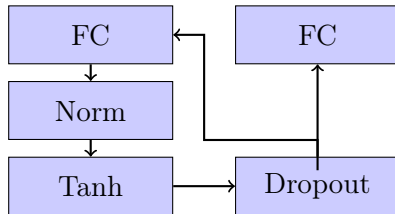
$$h_i^l = \sum_{j \in \mathcal{N}(i)} \alpha_{ij} W h_j^{l-1}$$

$$\alpha_{ij} = \frac{\text{Exp}[\text{LeakyReLU}(a^T [W h_i + W h_j])]}{\sum_{k \in \mathcal{N}(i)} \text{Exp}[\text{LeakyReLU}(a^T [W h_i + W h_k])]}$$

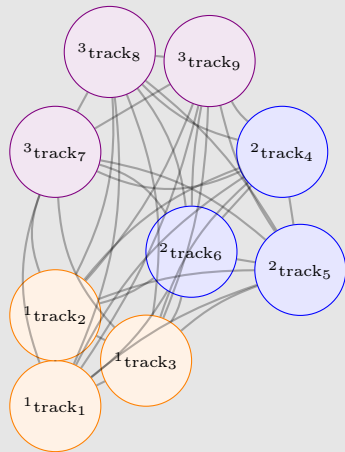
GAT



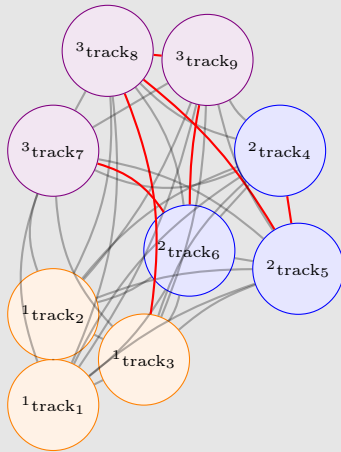
MLP



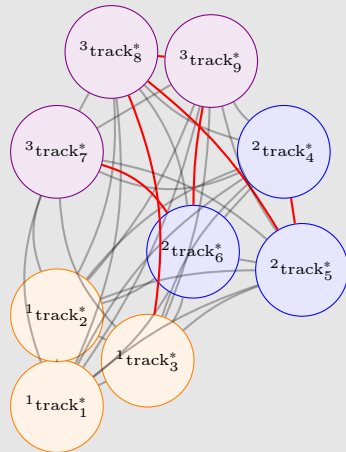
TrackGraph

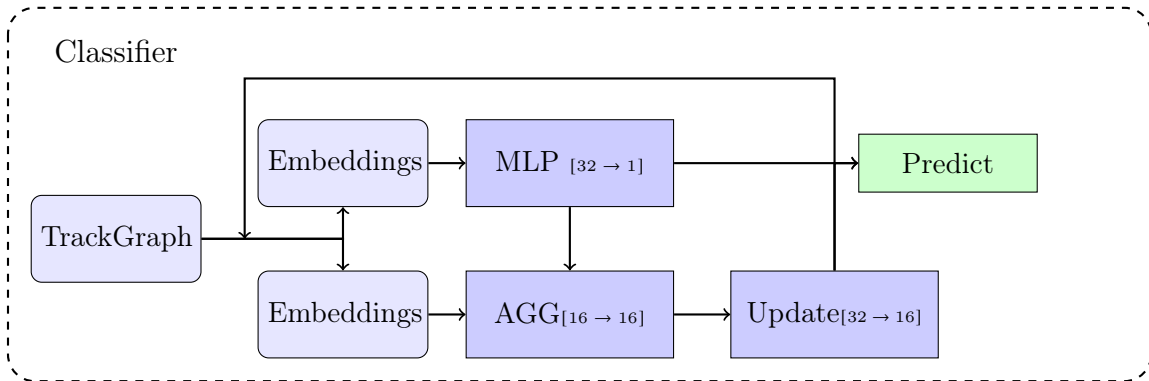


Attention



AGG+Update





$$\alpha_{ij} = \sigma(\text{MLP}(h_i || h_j))$$

$$\text{AGG}(h_i^l) == \sum_{k \in \mathcal{N}(i)} \alpha_{ik} h_k^{(l-1)}$$

$$h_i^l = \text{MLP}(\text{AGG}(h_i^l) || h_i^{(l-1)})$$

■ Balanced Cross Entropy:

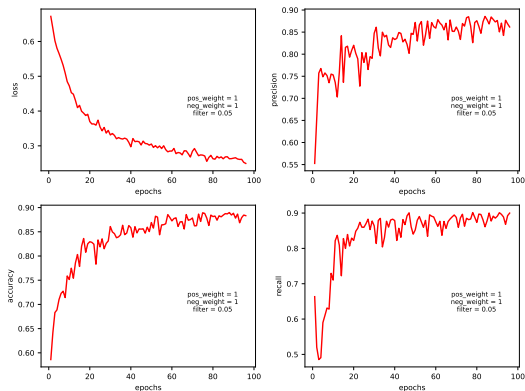
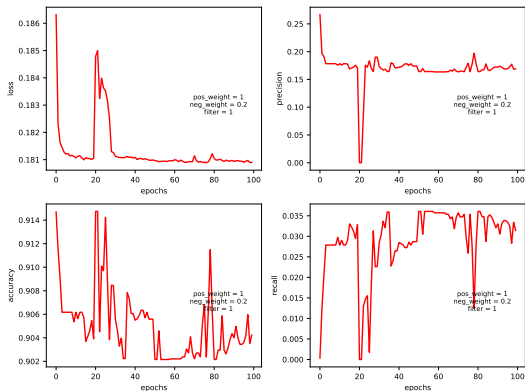
$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \omega_i (y_i \ln(p_i) + (1 - y_i) \ln(1 - p_i)) \quad \omega_i = y_i \omega_p + (1 - y_i) \omega_n$$

где y_i - метка истинности ребра, p_i - предсказание модели, N - число ребер, ω_p и ω_n - веса задаваемые вручную.

■ Метрики качества:

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

i -е ребро считается истинным если предсказание модели $p_i \geq 0.5$



Результат на левом графике получен с учетом: $\omega_p = 1$, $\omega_n = 0.2$.

Результат на правом графике получен с учетом удаления 95% ложных ребер, $\omega_p = \omega_n = 1$.

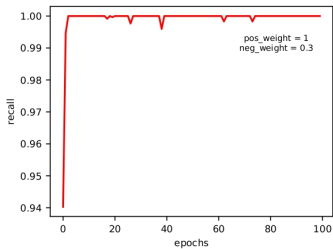
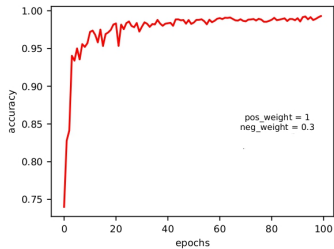
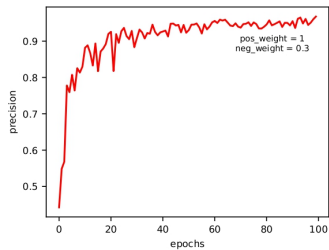
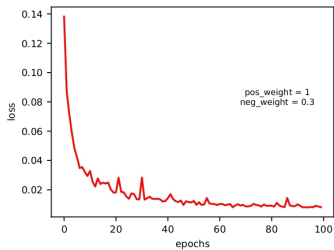
■ Balanced Focal Loss:

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

где y_i - метка истинности ребра, $p_t = p_i$ если ребро ложное, $p_t = (1 - p_i)$ если ребро истинное, p_i - предсказание модели, N - число ребер, $\omega_p, \omega_n, \gamma$ - параметры задаваемые вручную.

■ Время дрейфа:

Соединяем ребрами треки из событий с некоторой "степенью соседства" S_N . На первом тесте: $S_N = 2, \omega_p = 1, \omega_n = 0.3, \gamma = 1$



Успехи:

- 1 Модель одинаково работает с таймслайсами с разным количеством событий.
- 2 Модель показывает хорошие результаты при равенстве классов.

Проблемы:

- 1 Модель плохо работает с несбалансированными классами в наборе данных.

Планы:

- 1 Модификация функции ошибки с учетом дисбаланса между классами.
- 2 Тестировка подходов фильтрации данных на стадии построения Суперграфа

Спасибо за внимание!