

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

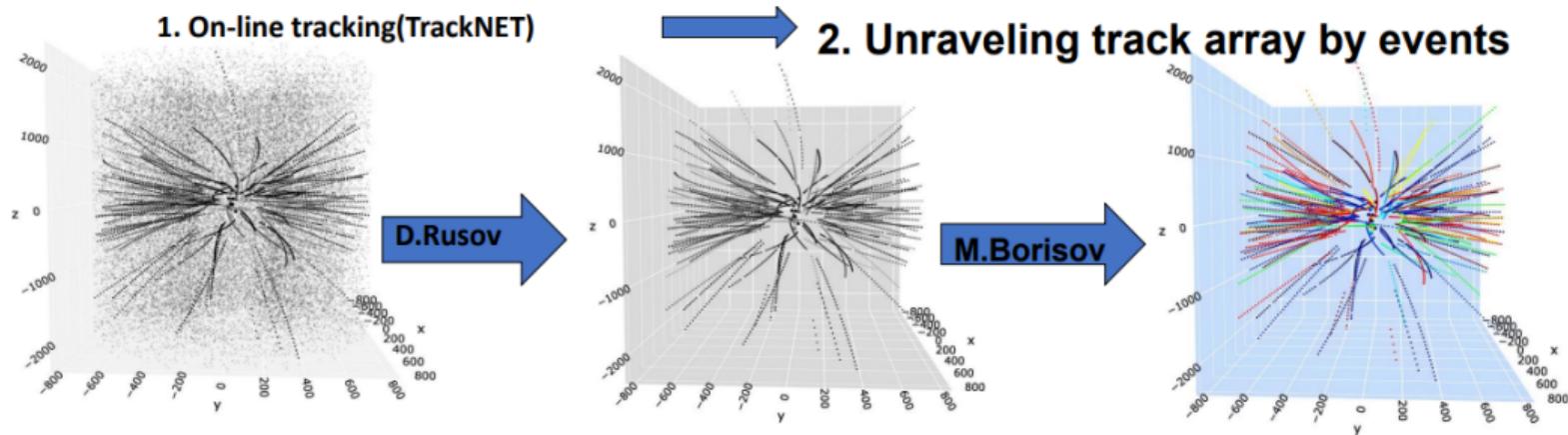
¹Филиал МГУ, г. Дубна,

² ОИЯИ ЛИТ

³ Университет Дубна

Апрель 03, 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](#)

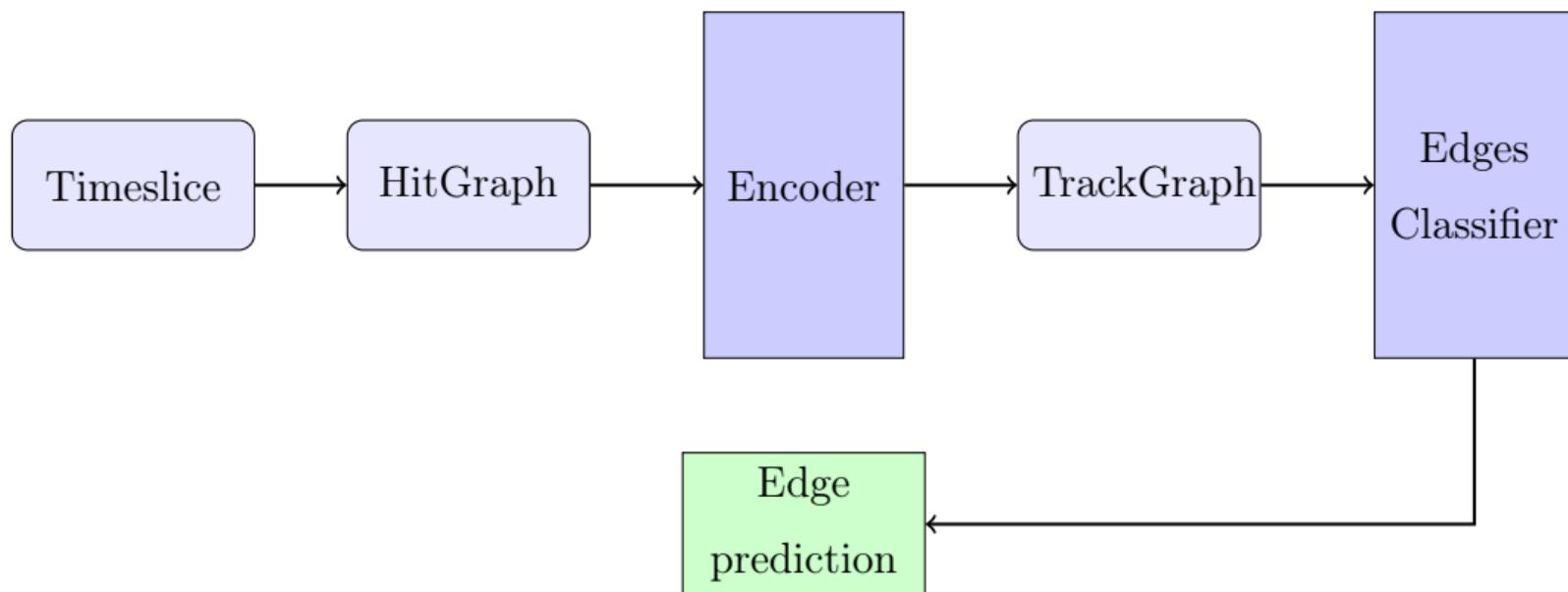
Преимущества:

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

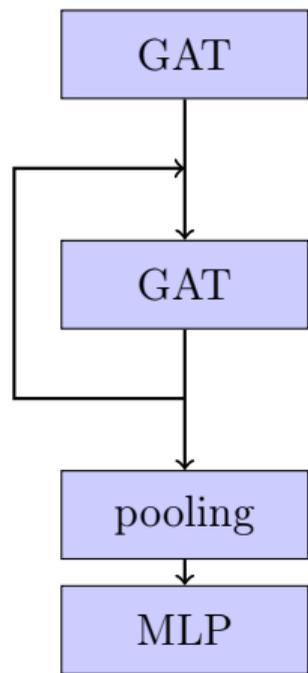
Недостатки:

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

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



Encoder

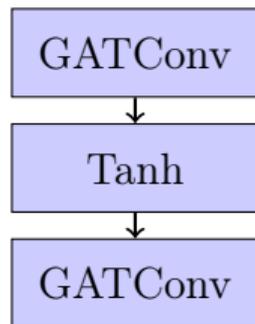


Графовый сверточный слой с механизмом внимания
GATConv:

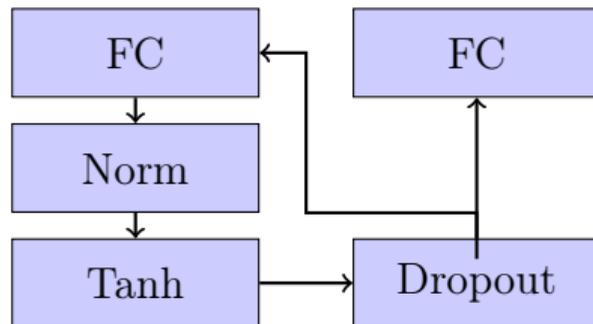
$$h_i^l = \sum_{j \in \mathcal{N}(i)} \alpha_{ij} W h_j^{l-1}$$

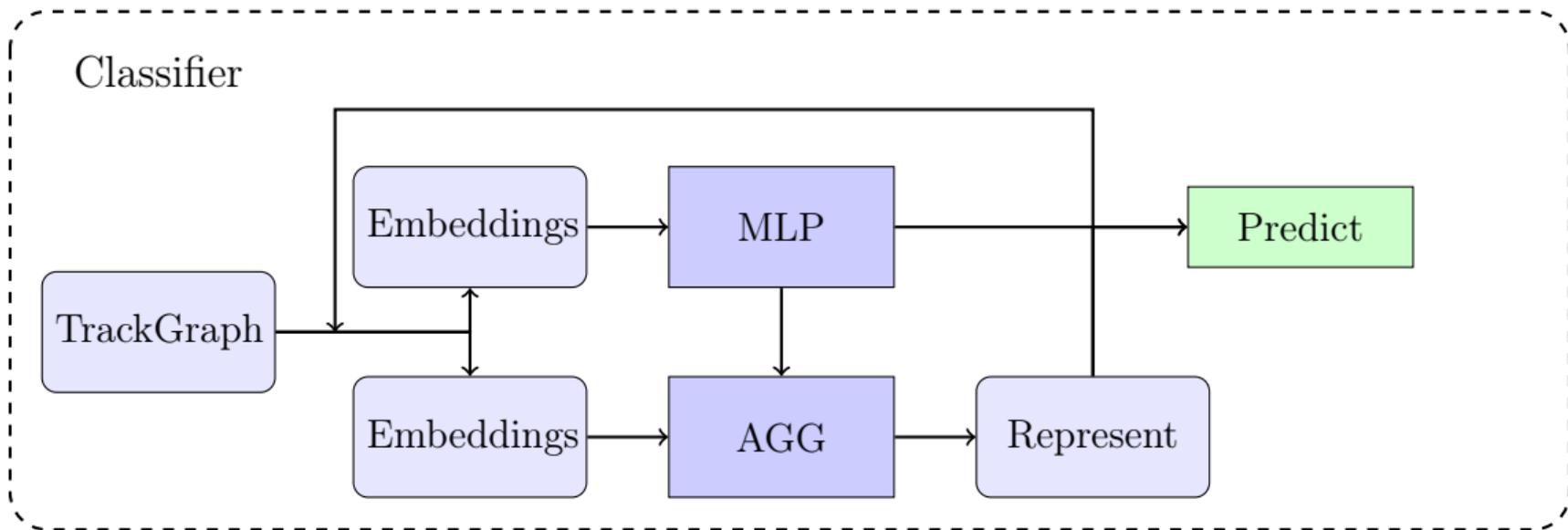
$$\alpha_{ij} = \frac{\text{Exp}[\text{LeakyReLU}(a^T [W h_i + W h_j])]}{\sum_{k \in \mathcal{N}(i)} \text{Exp}[\text{LeakyReLU}(a^T [W h_i + W h_k])]}$$

GAT



MLP





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

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

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

■ Balanced Cross Entropy:

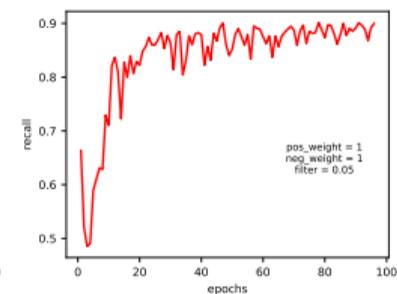
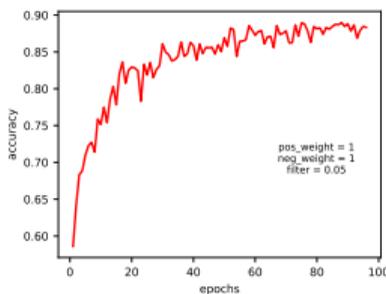
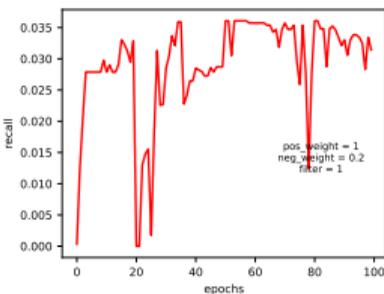
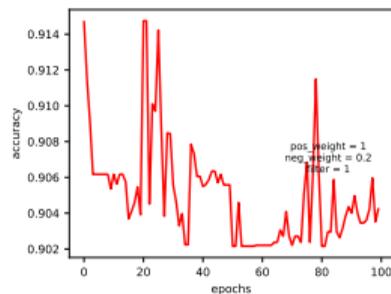
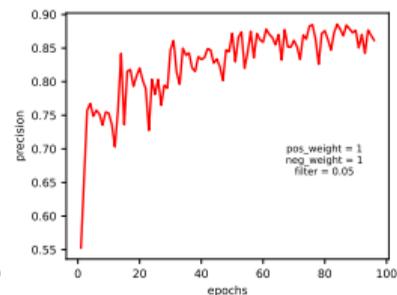
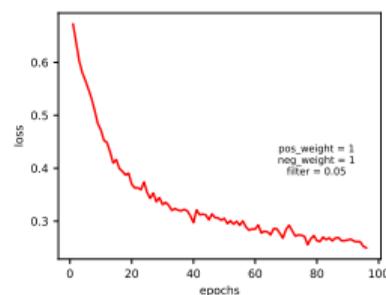
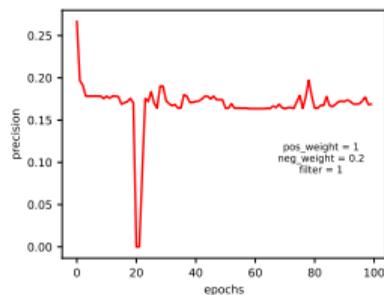
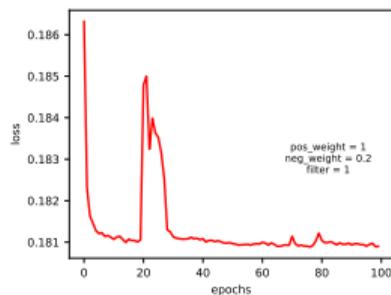
$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \omega_i (y_i \ln(p_i) + (1 - y_i) \ln(1 - p_i)) \quad \omega_i = y_i \omega_p + (1 - y_i) \omega_n$$

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, $\omega_p = \omega_n = 1$.

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 Добавление алгоритма динамической кластеризации/фильтрации для естественного удаления большинства ложных ребер на стадии обучения.

Спасибо за внимание!