

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
"МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ
М.В.ЛОМОНОСОВА"

ФИЛИАЛ МГУ в г. Дубне

Профиль: Физика элементарных частиц

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

"Разделение треков заряженных частиц по событиям в эксперименте SPD
методами глубокого машинного обучения"

Выполнил студент:

Омелянчук Савелий Сергеевич

(подпись студента)

Научный руководитель:

кандидат физ.-мат. наук

Леонтьев Владимир Викторович

(подпись научного руководителя)

Научный консультант:

кандидат физ.-мат. наук

Войтишин Николай Николаевич

(подпись научного консультанта)

Допущена к защите:

(дата)

Руководитель образовательной
программы:

(подпись руководителя ОП)

г. Дубна
2026 г.

Содержание

	Введение.	3
1	Генерация модельных данных.	5
2	Подход к решению задачи.	9
3	Графовые нейронные сети.	12
	3.1 Графовая сверточная нейронная сеть (GCN).	13
	3.2 Графовая нейронная сеть с механизмом внимания (GAT).	14
	3.3 Графовая изоморфная нейронная сеть (GIN).	15
4	Модель Track-GANN.	17
	4.1 Graph Encoder.	17
	4.2 Edges Classifier.	18
5	Иерархическое обучение	21
6	Лувенский метод	23
7	Результаты	25
	7.1 Метрики оценки качества	25
	7.2 Результаты SPD-GANN5	27
	7.3 Результаты Track-GANN5	32
	Выводы.	37
	Заключение.	38
	Список использованных источников.	41

Введение

SPD (*Spin Physics Detector*) — это планируемый эксперимент на коллайдере NICA (*Nuclotron-based Ion Collider fAcility*), предназначенный для изучения спиновой структуры адронов и различных непертурбативных явлений в КХД (квантовой хромодинамике). В качестве сталкивающихся частиц в эксперименте будут использоваться поляризованные адроны. Проектная светимость на коллайдере NICA ожидается на уровне значений порядка $10^{32} \text{ см}^{-2}\text{с}^{-1}$ в протон-протонных столкновениях а энергия в системе центра масс $\sqrt{s} = 10\text{--}27 \text{ ГэВ}$ [1].

Поляризованные пучки лёгких адронов являются ключевой особенностью установки SPD. Данная характеристика в совокупности с высокой светимостью обуславливает ряд специфических задач относящихся к сбору и обработке данных. Ожидаемая частота событий в протон-протонных столкновениях составляет $\sim 4 \text{ МГц}$, а скорость потока «сырых» данных — около 20 ГБ/с [1]. Электроника детектора непрерывно оцифровывает сигналы и передаёт данные на вычислительные узлы с соответствующими временными метками. В таком режиме работы детектор ориентирован на обработку временных срезов (интервалов интегрирования) вместо традиционного считывания «событие за событием». Каждый временной срез представляет собой совокупность нескольких физических событий.

Следовательно, основными инструментами для обработки столь значительных объёмов данных являются алгоритмы, адаптированные для работы с событиями, сильно перекрывающимися в пространстве и времени. В совокупности с высокой скоростью потока данных это приводит к необходимости использования методов, совмещающих в себе как высокую точность и скорость, так и возможность параллельных вычислений. В частности, эффективным решением выступают глубокие нейронные сети, выполняющие

вычисления на графических процессорах (GPU).

Процесс полного восстановления событий включает несколько этапов. На первом этапе осуществляется фильтрация ложных срабатываний (хитов) и реконструкция истинных траекторий в рамках одного временного среза. Вторым этапом является классификация треков по принадлежности к конкретным событиям. Настоящее исследование сфокусировано на втором этапе. Ранее для этой задачи была разработана сиамская нейронная сеть [2], однако она демонстрировала высокую эффективность исключительно при фиксированном числе событий во временном срезе, что не соответствует реальным условиям эксперимента SPD. В данной работе архитектура GNN применяется к задаче классификации треков по событиям. Графовые нейронные сети (GNN, *Graph Neural Network*) уже применялись для реконструкции траекторий заряженных частиц в физике высоких энергий. Настоящая работа посвящена разработке и тестированию модели GNN для восстановления меток событий для каждого трека в пределах отдельных временных срезов в эксперименте SPD.

1. Генерация модельных данных.

Поскольку экспериментальные данные на текущий момент недоступны в связи с тем, что установка ещё не введена в эксплуатацию, в настоящей работе используется модельная выборка. Генератор данных разработан на основе проектной конфигурации и характеристик эксперимента SPD [1] и подробно описан в работе [3]. Алгоритм реализован в виде *Python*-скрипта, аппроксимирующего спиральные траектории движения частиц.

Программный комплекс генерирует временные срезы, состоящие из набора событий, количество которых распределено по закону Пуассона. Каждое событие включает в себя совокупность траекторий заряженных частиц, число которых распределено равномерно в заданном диапазоне. В свою очередь, каждый трек представляет собой последовательность хитов (пространственных точек срабатывания детектора). Каждому хиту соответствует определённое время регистрации в слоях детектора, определяемое как время дрейфа. Предполагается, что в эксперименте SPD измерение этого времени будет осуществляться для каждой частицы с определенной точностью.

В алгоритме генерации модельных данных временные интервалы внутри среза вычисляются в соответствии со следующими выражениями:

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

где $i = 0, \dots, N_{\text{ev}} - 1$ — индекс события; N_{ev} — общее количество событий во временном срезе; t_d^i — центр временного интервала дрейфа для i -го события; Δt_d — полная ошибка детектора (принимается одинаковой для всех событий); t_L^i и t_R^i — левая и правая границы временного интервала дрейфа для i -го события соответственно; r — параметр перекрытия.

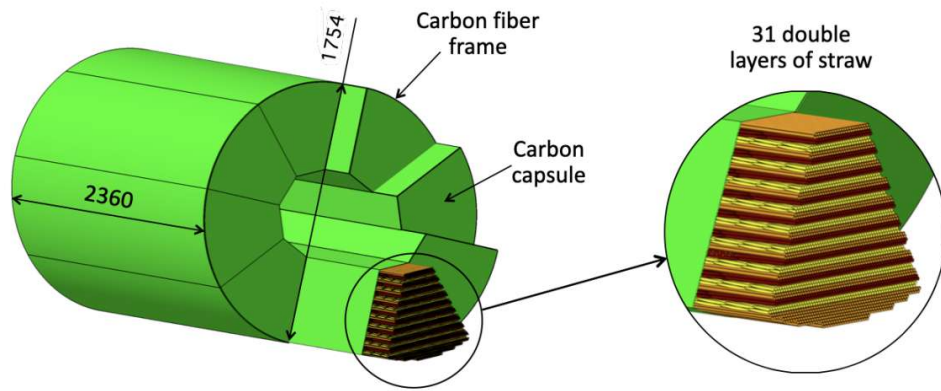


Рисунок 1. Геометрия SPD Straw Tracker [1].

Параметр перекрытия является регулируемым и физически связан с разрешающей способностью детектора: увеличение r эквивалентно снижению временного разрешения. С точки зрения структуры данных данный параметр определяет, с каким количеством соседних событий перекрывается по времени дрейфа событие i . Таким образом, треки из события i могут иметь временное наложение только на треки из $2r$ окружающих событий. Это упрощает задачу классификации, так как трек из события с индексом i заведомо не может принадлежать одному событию с треками из событий с индексами $i \pm (r + 1), \dots, N$ в силу их значительного временного разнесения.

В работе используются два типа временных срезов, различающихся структурой треков. Первый тип — так называемые временные срезы SPD (рис. [2]), в которых траектории частиц моделируются с помощью аппроксимации спирали с учётом среднеквадратичного отклонения, соответствующего проектным характеристикам трекингowej системы состоящей из дрейфовых трубок (Straw Tracker) [1] (рис. [1]). В данном исследовании использовалась конфигурация детектора, включающая 35 станций с эффективностью регистрации 95%.

Второй тип — временные срезы TrackML (рис. [3]), сформированные из треков публичного набора данных TrackML [4]. Данные такого типа представляют большую сложность для алгоритмов реконструкции, так как траектории в среднем обладают большей кривизной и содержат меньшее количество

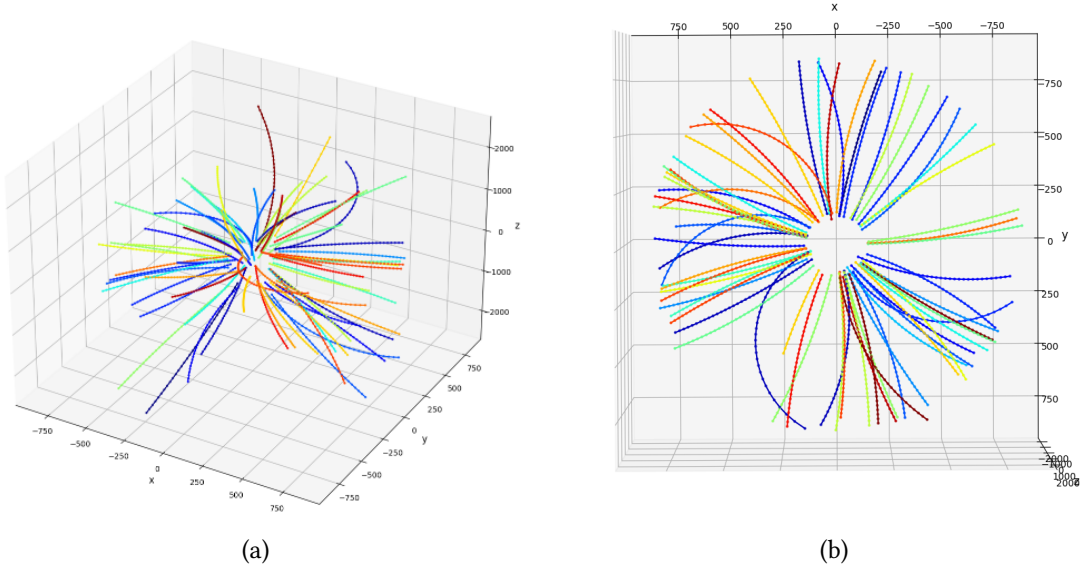


Рисунок 2. Пример SPD временного среза со средним числом событий $\bar{N}_{ev} = 15$ и максимальным числом треков в событии $n_{tr}^{max} = 6$.

хитов. Кроме того, в отличие от срезов SPD, хиты в наборе TrackML не являются эквидистантными по радиальной координате $r_\phi = \sqrt{x^2 + y^2}$. Основные различия между двумя наборами данных систематизированы в таблице [1](#).

	Хитов в треке	Смещение вершин (мм)	r_ϕ (мм)	Закрученность
SPD треки	(28, 35)	(−300, 300)	(75, 850)	низкая
TrackML треки	(3, 20)	(−350, 350)	(50, 1000)	высокая

Таблица 1. Сравнение треков из базы данных TrackML и из SPD симуляции.

В данной работе рассматриваются временные срезы, не содержащие ложных хитов (шумов). Данное допущение обосновано высокой эффективностью рекуррентных нейронных сетей при очистке данных от фоновых срабатываний, продемонстрированной в работах [\[5, 6\]](#). Временные срезы включают в среднем 15 событий; при этом количество треков в одном событии варьируется от 2 до 6. Параметр перекрытия r принимает целочисленные значения в диапазоне от 2 до 5 включительно.

Структурно каждый временной срез представляется в виде таблицы [2](#). Она содержит координаты хитов (x, y, z) , идентификатор $track_id$, определя-

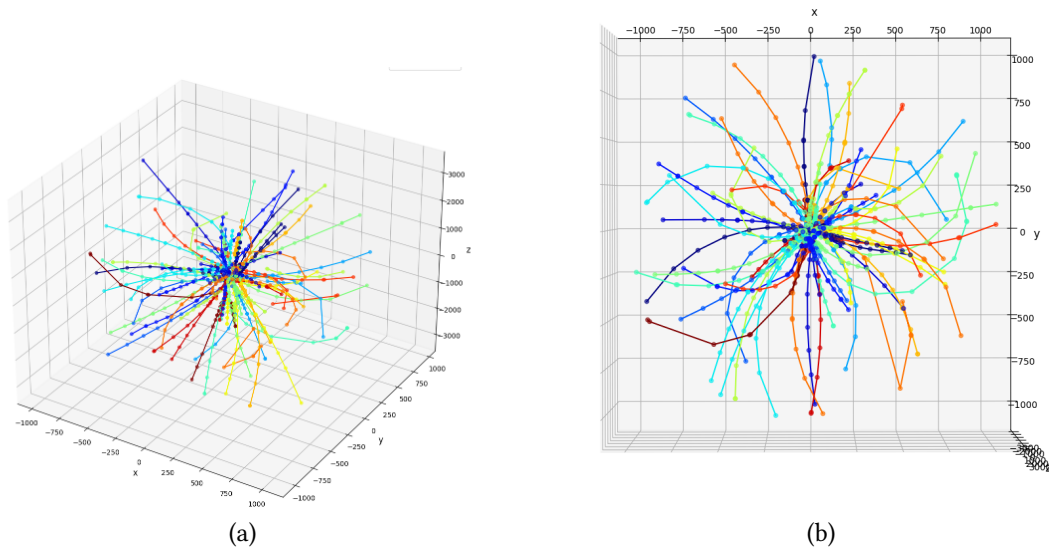


Рисунок 3. Пример TrackML временного среза со средним числом событий $\bar{N}_{ev} = 15$ и максимальным числом треков в событии $n_{tr}^{max} = 6$.

ющий принадлежность хита к конкретной траектории, а также переменные $time_left$ и $time_right$, задающие левую и правую границы интервала времени дрейфа для каждого трека. Целевой переменной является идентификатор $event_id$, указывающий на принадлежность трека к определённому событию.

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

Таблица 2. Схематичное представление одного временного среза

2. Подход к решению задачи.

Представление набора данных трекинговых систем в виде графовых структур является распространённым подходом [7, 8]. Данная методика эффективна при решении задач реконструкции траекторий и фильтрации фоновых срабатываний (шума детектора). В подходах, описанных в работах [9, 10, 11], хиты связываются между собой на основе определённых эвристических правил, формируя граф. Впоследствии нейронная сеть классифицирует рёбра, исключая те, что связаны с ложными хитами, после чего изолированные

узлы удаляются. Таким образом, оставшиеся связные последовательности хитов интерпретируются как истинные треки. Качественная реализация данной концепции обеспечивает высокую точность реконструкции даже в условиях большой множественности событий.

Тем не менее, этот подход обладает рядом недостатков. Алгоритм демонстрирует высокие показатели, если хиты истинного трека расположены достаточно близко друг к другу. Однако в случаях, когда значительная часть сегментов траектории не была зарегистрирована детектором и трек оказывается фрагментированным, эффективность реконструкции существенно снижается. В работе [12] предложен метод корректной обработки таких случаев: идея заключается в кодировании отдельных сегментов трека в узлы графа высшего порядка с последующей классификацией рёбер уже в этой структуре. Данная модификация позволила значимо повысить эффективность восстановления траекторий.

Применительно к задаче кластеризации треков по событиям, методика, предложенная в работе [2], основывается на получении векторных представлений (эмбеддингов, *embeddings*) отдельных треков в n -мерном пространстве признаков с помощью сиамской нейронной сети. Однако данный подход оказался малоэффективным в условиях произвольного числа событий во временном срезе. Ограничение обусловлено необходимостью применения классических алгоритмов кластеризации на финальном этапе: итоговый результат критически зависел от выбора конкретного алгоритма и его гиперпараметров. Кроме того, архитектура указанной сети не позволяла обрабатывать треки с произвольным числом хитов.

В предлагаемом подходе объединены идеи формирования векторных представлений (эмбеддингов) для каждого трека и представления этих треков в качестве узлов графа более высокого уровня. Будем называть такую структуру *суперграфом*. Основная концепция заключается в предваритель-

ном кодировании признаков каждого трека с помощью GNN и последующем построении супер-графа, где узлами являются закодированные треки, а рёбра соединяют пары треков, чьи интервалы времени дрейфа перекрываются. Формально условие существования ребра e_{ij} между i -м и j -м треками можно записать следующим образом:

$$\exists e_{ij} \iff (T^i \cap T^j \neq \emptyset), \quad (2)$$

где $T^k = [t_L^k, t_R^k]$ — временной интервал дрейфа k -го трека. Задача модели состоит в классификации рёбер супер-графа и исключении «ложных» связей. Ложными считаются рёбра, соединяющие треки, принадлежащие разным физическим событиям; остальные рёбра классифицируются как истинные. Таким образом, исходная задача многоклассовой кластеризации редуцируется к задаче бинарной классификации рёбер графа.

Алгоритм реализации подхода:

1. Построение первичных графов: для каждого трека формируется граф, где узлами являются хиты, а рёбра соединяют их последовательно от центра детектора к периферии.
2. Кодирование и генерация супер-графа: треки преобразуются в векторные представления (узлы супер-графа), связи между которыми устанавливаются согласно правилу (2).
3. Классификация рёбер: фильтрация ложных связей в супер-графе с помощью обученной модели графовой нейронной сети.
4. Финальная сегментация: удаление оставшихся аномальных рёбер и выделение кластеров (событий) с помощью алгоритмов поиска сообществ на полученном разреженном графе.

3. Графовые нейронные сети.

Графовые нейронные сети (GNN, *Graph Neural Networks*) предназначены для обработки данных, представленных в виде графовых структур. Большинство современных архитектур GNN основано на парадигме передачи сообщений (MP, *Message Passing*) и относится к классу сетей MPNN (*Message Passing Neural Networks*). Слои такой нейронной сети обеспечивают итеративный обмен информацией между узлами, соединёнными рёбрами, что позволяет на каждом шаге обновлять векторные представления (эмбединги) узлов и рёбер с учётом локальной топологии графа. В общем виде процесс обновления состояния узла описывается следующим выражением:

$$h_v^{(l+1)} = f_\phi^{(l)} \left(h_v^{(l)}, \bigoplus_{u \in \mathcal{N}(v)} g_\theta^{(l)}(h_u^{(l)}) \right), \quad (3)$$

где l — номер слоя; $h_v^{(l)}$ — вектор признаков узла v на слое l ; $\mathcal{N}(v)$ — множество соседних узлов для v ; $g_\theta^{(l)}$ — функция формирования сообщения (обычно линейное преобразование с параметрами θ); $\bigoplus$ — оператор агрегации, инвариантный к перестановкам (например, сумма, среднее или максимум); $f_\phi^{(l)}$ — функция обновления состояния узла с параметрами ϕ .

Выражение (3) лежит в основе большинства современных архитектур GNN. Различия между конкретными реализациями заключаются преимущественно в выборе видов функций f_ϕ , g_θ и способе агрегации $\bigoplus$. Обновление представления узлов может производиться итеративно внутри каждого слоя l по правилу (3) с числом итерации k_l .

3.1. Графовая сверточная нейронная сеть (GCN).

Свёрточная графовая нейронная сеть (GCN, *Graph Convolutional Network*) [13] выполняет обновление представлений узлов согласно следующему правилу:

$$h_v^{(l+1)} = \sigma \left(W^{(l)} \sum_{u \in \mathcal{N}(v) \cup \{v\}} \frac{1}{\sqrt{\deg(v) \deg(u)}} h_u^{(l)} \right), \quad (4)$$

где $h_v^{(l)}$ — вектор признаков узла v на слое l ; σ — нелинейная функция активации; $W^{(l)}$ — обучаемая матрица весов слоя l ; $\deg(v)$ — степень узла v ; $\mathcal{N}(v)$ — множество непосредственных соседей узла v .

GCN использует детерминированную функцию агрегации с симметричной нормировкой, что существенно снижает вычислительные затраты. Для повышения выразительности модели вместо увеличения числа итераций внутри одного слоя предпочтительным является наращивание количества последовательных слоёв. Это обеспечивает оптимальный баланс между вычислительной сложностью и способностью модели извлекать высокоуровневые признаки.

К ключевым особенностям архитектуры GCN относятся:

- Симметричная нормировка: использование степеней узлов позволяет сбалансировать вклад соседей с разной связностью.
- Структурная зависимость: веса рёбер определяются исключительно топологией графа, а не значениями признаков узлов.
- Изотропность: все соседние узлы учитываются с одинаковой относительной важностью (в рамках нормировки), что может ограничивать эффективность модели на графах со сложными гетерогенными связями.

3.2. Графовая нейронная сеть с механизмом внимания (GAT).

Графовая нейронная сеть с механизмом внимания (GAT, *Graph Attention Network*) [14] выполняет обновление представлений узлов согласно следующему правилу:

$$h_v^{(l+1)} = \sigma \left(\sum_{u \in \mathcal{N}(v) \cup \{v\}} \alpha_{vu}^{(l)} W^{(l)} h_u^{(l)} \right), \quad (5)$$

где $h_u^{(l)}$ — вектор признаков узла u на слое l ; σ — нелинейная функция активации; $W^{(l)}$ — обучаемая матрица весов; $\mathcal{N}(v)$ — множество соседей узла v ; $\alpha_{vu}^{(l)}$ — коэффициент внимания, определяющий важность соседа u для узла v на слое l . Коэффициенты внимания вычисляются с использованием функции *softmax*:

$$\alpha_{vu}^{(l)} = \frac{\exp(e_{vu}^{(l)})}{\sum_{k \in \mathcal{N}(v) \cup \{v\}} \exp(e_{vk}^{(l)})}, \quad (6)$$

где веса связей $e_{vu}^{(l)}$ (энергия внимания) определяются как:

$$e_{vu}^{(l)} = \text{LeakyReLU} \left(\mathbf{a}^{(l)T} \left[W^{(l)} h_v^{(l)} \parallel W^{(l)} h_u^{(l)} \right] \right). \quad (7)$$

Здесь $\parallel$ обозначает операцию конкатенации векторов, а $\mathbf{a}^{(l)}$ — обучаемый вектор параметров механизма внимания. Данная архитектура обладает преимуществом перед GCN благодаря адаптивному агрегированию: вклад каждого соседа взвешивается динамически в зависимости от его признаков, что существенно повышает гибкость модели.

К ключевым особенностям GAT относятся:

- Адаптивное взвешивание: обучаемые коэффициенты внимания позво-

ляют модели выделять наиболее релевантные связи в графе.

- Индуктивность: возможность эффективной работы с графами, структура которых не была полностью известна на этапе обучения.
- Универсальность: применимость к сложным топологиям с разнородными типами связей за счёт неявного определения весов рёбер.

3.3. Графовая изоморфная нейронная сеть (GIN).

Графовая изоморфная сеть (GIN, *Graph Isomorphism Network*) [15] выполняет обновление представлений узлов согласно следующему правилу:

$$h_v^{(l+1)} = \text{MLP}^{(l)} \left((1 + \epsilon^{(l)}) \cdot h_v^{(l)} + \sum_{u \in \mathcal{N}(v)} h_u^{(l)} \right), \quad (8)$$

где $h_v^{(l)}$ — вектор признаков узла v на слое l ; $\epsilon^{(l)}$ — обучаемый (или фиксированный) параметр, регулирующий вклад собственного состояния узла; $\mathcal{N}(v)$ — множество непосредственных соседей узла v ; $\text{MLP}^{(l)}$ — многослойный перцептрон с параметрами ϕ_l [16], выступающий в роли функции обновления.

Данная архитектура обладает максимальной выразительной способностью среди всех MPNN и теоретически эквивалентна по мощности одномерному тесту изоморфизма графов Вейсфейлера–Лемана (1-WL тест). В прикладных задачах GIN может уступать GAT по эффективности из-за отсутствия механизма адаптивного (анизотропного) взвешивания соседей, однако превосходит её в способности различать структурные топологии графа. Использование многослойного перцептрона вместо простой линейной трансформации существенно расширяет пространство гипотез модели и позволяет улавливать нелинейные зависимости в локальной структуре графа.

К ключевым особенностям GIN относятся:

- Инъективная агрегация: использование суммирования (*sum*) вместо усреднения или максимизации, что позволяет различать мультимножества соседних узлов и обеспечивает максимальную теоретическую выразительность.
- Регулируемый вклад центрального узла: наличие параметра $\epsilon^{(l)}$ позволяет гибко настраивать баланс между собственными признаками узла и информацией от его окружения.
- Глубокое обновление: применение MLP в качестве функции преобразования признаков значительно повышает репрезентативность модели по сравнению с однослойными свертками.
- Высокая сложность: значительная выразительная способность может приводить к переобучению на графах с простой структурой или малым объёмом данных, что обуславливает необходимость тщательной регуляризации.

4. Модель Track-GANN.

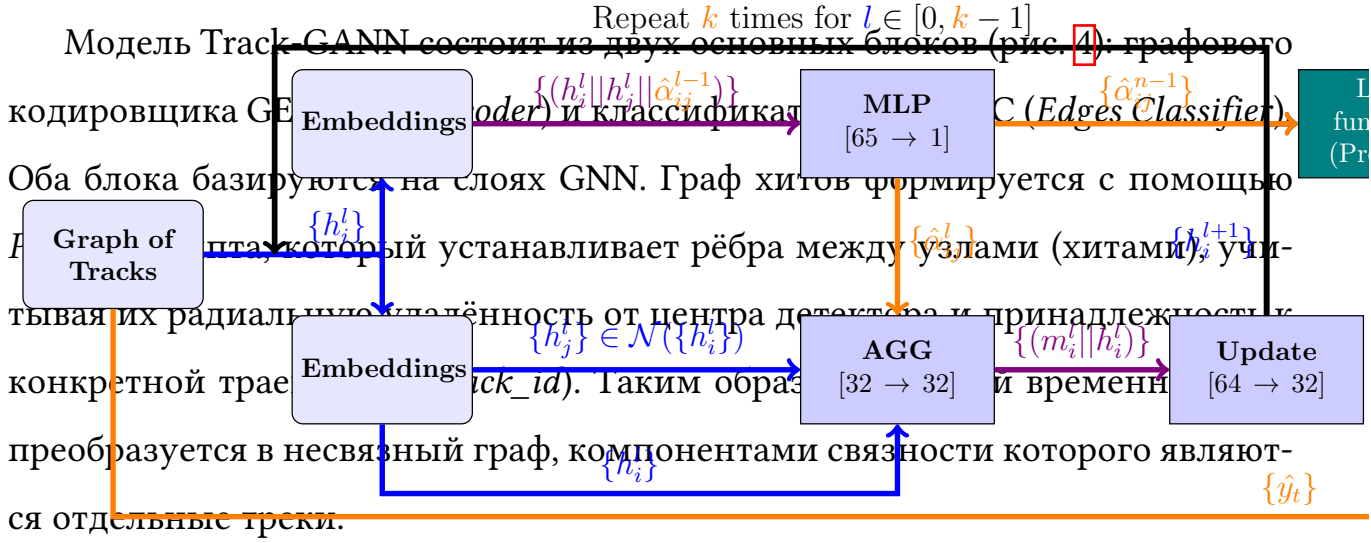
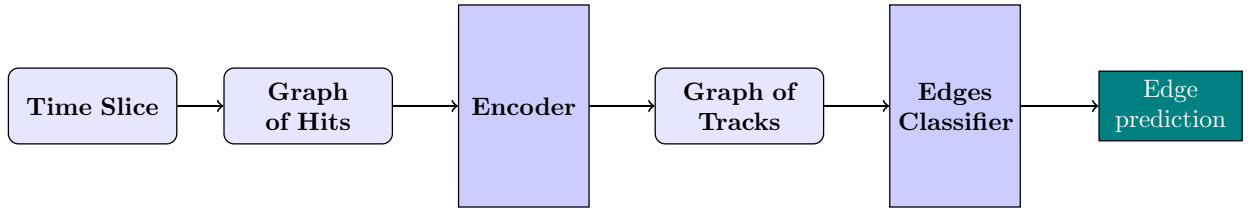


Рисунок 4. Архитектура модели Track-GANN.

4.1. Graph Encoder.

Блок Graph Encoder принимает на вход матрицу признаков, где каждая строка соответствует хиту и содержит его декартовы координаты (x, y, z) , а также матрицу смежности графа. Архитектура GE (рис. 5) включает четыре основных компонента:

- Входной графовый слой: реализован на базе архитектуры GAT с размерностью $[3, 64]$. Слой проецирует исходные признаки хитов в скрытое пространство и вычисляет начальные коэффициенты внимания.
- Скрытый графовый слой: слой GAT с размерностью $[64, 64]$. Данный блок выполняет итеративную агрегацию признаков, обновляя пред-

ставления хитов с учётом их окружения. В текущей конфигурации глубина агрегации равна 6, однако данный параметр является настраиваемым гиперпараметром модели.

- Слой пулинга: выполняет усреднение векторов признаков всех хитов, принадлежащих одному треку. Данная операция позволяет осуществить переход от локальных признаков отдельных точек детектирования к интегральным векторным представлениям (эмбедингам) целых треков.
- Полносвязный блок: представляет собой многослойный перцептрон (MLP) [16] с архитектурой [64, 128, 128, 64, 32]. Для предотвращения переобучения и стабилизации градиента к скрытым слоям применяются метод исключения узлов (*dropout*) с вероятностью 0.1, послойная нормализация (*layer normalization*) и функция активации $\tanh$. Данный блок формирует итоговое компактное представление векторов признаков треков.

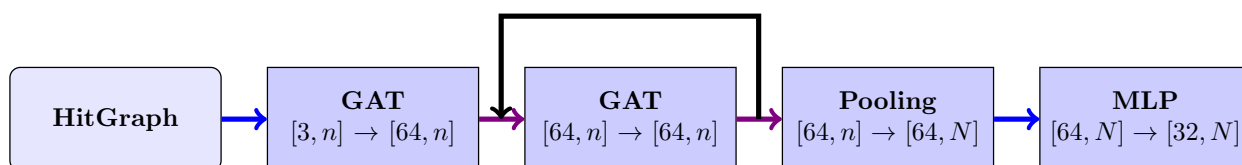


Рисунок 5. Архитектура блока Graph Encoder (GE).

4.2. Edges Classifier.

Классификатор рёбер EC (*Edges Classifier*) принимает на вход суперграф, узлами которого являются векторные представления треков, полученные в блоке GE, а рёбра сформированы согласно правилу (2). Модуль EC базируется на гибридной архитектуре, сочетающей механизмы внимания GAT и

структурную мощность GIN. Основная задача ЕС заключается в предсказании вероятности истинности каждого ребра в диапазоне $[0, 1]$, где значение 1 соответствует истинному ребру (треки принадлежат одному физическому событию), а 0 — ложному. Таким образом, блок ЕС реализует бинарную классификацию рёбер суперграфа.

Архитектура ЕС (рис. 6) включает четыре ключевых компонента:

- Первичный классификатор рёбер: полносвязная сеть (MLP) с конфигурацией слоёв $[64, 128, 128, 1]$. К скрытым слоям применяются *dropout* (0.1), послойная нормализация и активация $\tanh$. Блок вычисляет начальные веса рёбер (коэффициенты внимания):

$$\alpha_{ij}^{(0)} = \sigma \left(\text{MLP} \left(h_i^{(0)} \parallel h_j^{(0)} \right) \right), \quad (9)$$

где $h_i^{(0)}$ — вектор признаков i -го узла на входной итерации; σ — сигмоидная функция активации.

- Механизм агрегации: реализует парадигму *Message Passing* с использованием взвешенного суммирования. Сообщение для узла i формируется как:

$$m_i^{(l)} = \sum_{k \in \mathcal{N}(i)} \alpha_{ik}^{(l)} h_k^{(l)}, \quad (10)$$

где $m_i^{(l)}$ — агрегированное сообщение от множества соседей $\mathcal{N}(i)$ на итерации l .

- Модуль обновления узлов: MLP с размерностью $[64, 128, 128, 32]$, принимающий на вход конкатенацию текущего вектора признаков узла и полученного сообщения. Обновление состояния выполняется согласно выражению:

$$h_i^{(l+1)} = \text{MLP} \left(m_i^{(l)} \parallel h_i^{(l)} \right). \quad (11)$$

- Классификатор рёбер: MLP с размерностью $[65, 128, 128, 1]$, анализирующий обновлённые представления узлов и значение веса ребра с предыдущей итерации. Уточнённые коэффициенты вычисляются как:

$$\alpha_{ij}^{(l+1)} = \sigma \left(\text{MLP} \left(h_i^{(l+1)} \parallel h_j^{(l+1)} \parallel \alpha_{ij}^{(l)} \right) \right). \quad (12)$$

Итерационный процесс (агрегация, обновление узлов и уточнение рёбер) повторяется 6 раз для достижения стабильных эмбедингов и высокой точности классификации.

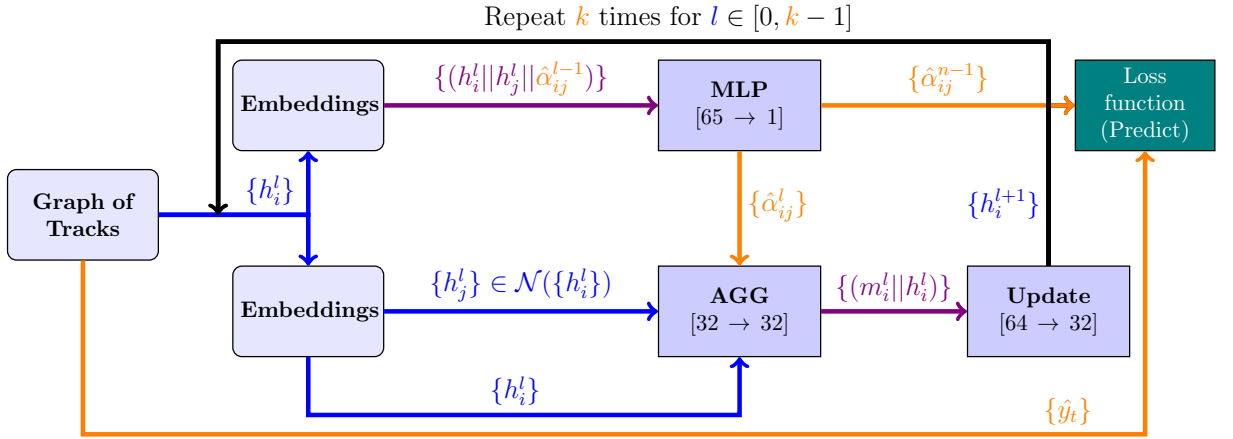
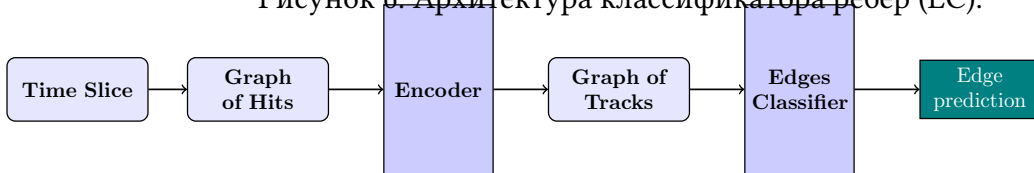


Рисунок 6. Архитектура классификатора рёбер (EC).



5. Иерархическое обучение

В основе стратегии обучения Track-GANN лежит принцип иерархического обучения (*Curriculum Learning*). Данный подход подразумевает последовательное обучение модели на данных возрастающей сложности, что позволяет сформировать устойчивые иерархические представления признаков. В настоящей работе сложность данных регулируется параметром временного перекрытия r (см. раздел [1](#)). Процесс начинается с обучения на временных срезах с минимальным перекрытием ($r = 2$). При достижении плато функции потерь обучение продолжается на выборках с $r = 3$ и так далее, вплоть до целевого значения $r = 5$, которое соответствует ожидаемому временному разрешению детектора SPD.

На каждом этапе среднее число событий во временном срезе фиксировалось на уровне $\bar{N}_{\text{ev}} = 15$. Стоит отметить, что иерархия сложности может быть дополнительно расширена за счёт увеличения $\bar{N}_{\text{ev}}$. Обучение каждой итерации проводилось в течение 100 эпох с размером батча, равным одному временному срезу.

Для минимизации ошибки использовался оптимизатор AdamW [\[17\]](#), а в качестве целевой функции — сбалансированная фокальная функция потерь (BFL, *Balanced Focal Loss*) [\[18\]](#), определяемая выражением:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \alpha_i (1 - p_{t,i})^\gamma \ln(p_{t,i}), \quad (13)$$

где N — общее число рёбер; γ — параметр фокусировки; $p_{t,i}$ принимает значение p_i для истинных рёбер ($y_i = 1$) и $1 - p_i$ для ложных ($y_i = 0$); p_i — предсказание модели. Весовой коэффициент α_i определяется как ω_p для положительного класса и ω_n для отрицательного.

Выбор данной функции обусловлен её адаптивностью к изменению ба-

ланса классов на разных этапах обучения. Поскольку увеличение параметра r приводит к росту доли ложных рёбер в суперграфе, корректировка весов ω_p и ω_n позволяет поддерживать баланс между ложноположительными (FP) и ложноотрицательными (FN) срабатываниями. Параметр γ регулирует вклад «простых» примеров (в которых модель уверена) в общий градиент: при увеличении γ веса хорошо классифицированных объектов снижаются, заставляя сеть фокусироваться на пограничных и сложных случаях.

Необходимость иерархического подхода продиктована экстремальной несбалансированностью исходной выборки. При попытке обучения «с нуля» на данных с максимальным перекрытием количество ложных рёбер в суперграфе на несколько порядков превышает число истинных. В таких условиях из-за механизмов плотной агрегации информации в GNN процесс оптимизации быстро сходится к тривиальному локальному минимуму, при котором модель классифицирует абсолютно все рёбра как ложные.

6. Лувенский метод

Модель Track-GANN не всегда способна полностью исключить все ложные рёбра в суперграфе, что будет продемонстрировано в разделе с результатами. В связи с этим применение эвристических методов на заключительном этапе обработки суперграфа является необходимым этапом постобработки. Существует множество алгоритмов поиска сообществ на графах, таких как CNMA (*Clauset-Newman-Moore Algorithm*) [19], LPA (*Label Propagation Algorithm*) [20], GNA (*Girvan-Newman Algorithm*) [21] и другие. Выбор конкретной эвристики является свободным параметром предложенной методологии.

В настоящей работе был выбран Лувенский метод (Louvain method) [22]. Данный алгоритм является итеративной модификацией CNMA и базируется на максимизации модулярности графа Q , определяемой следующим образом:

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j), \quad (14)$$

где A_{ij} — вес ребра между узлами i и j ; $k_i = \sum_j A_{ij}$ — взвешенная степень узла i ; $m = \frac{1}{2} \sum_{i,j} A_{ij}$ — суммарный вес всех рёбер в графе; $\delta(c_i, c_j)$ — символ Кронекера, принимающий значение 1, если узлы i и j принадлежат одному сообществу ($c_i = c_j$), и 0 в противном случае.

Модулярность графа можно представить как сумму вкладов отдельных сообществ $Q = \sum_c Q_c$. Выражение для модулярности конкретного сообщества c выводится из формулы (14):

$$Q_c = \frac{\Sigma_{in}}{2m} - \left(\frac{\Sigma_{tot}}{2m} \right)^2, \quad (15)$$

где Σ_{in} — сумма весов рёбер внутри сообщества c ; Σ_{tot} — сумма весов рёбер, инцидентных узлам сообщества c .

Алгоритм выполняется итеративно и состоит из двух этапов:

1. Максимизация модулярности: изначально каждый узел рассматривается как отдельное сообщество. Для каждого узла i вычисляется изменение модулярности ΔQ при его перемещении в сообщество соседа j . Узел перемещается в то сообщество, которое обеспечивает максимальный положительный прирост ΔQ . Процесс повторяется последовательно для всех узлов до тех пор, пока локальный максимум модулярности не будет достигнут.
2. Агрегация графа: сообщества, сформированные на первом этапе, стягиваются в супер-узлы нового графа. Веса рёбер между новыми узлами определяются как сумма весов рёбер между соответствующими сообществами, а веса петель (self-loops) супер-узлов — как сумма весов внутренних рёбер исходных сообществ.

Данные этапы повторяются до стабилизации значения Q . В рамках данной работы была реализована версия Лувенского метода, адаптированная для работы с тензорными представлениями данных в среде *PyTorch* [23]. Это позволило интегрировать алгоритм непосредственно в вычислительный граф модели, обеспечив высокую производительность без необходимости конвертации данных в сторонние форматы.

7. Результаты

Модель, обученную на временных срезах с определенным параметром перекрытия r , в дальнейшем будем обозначать как GANN r . Процесс обучения начинается с набора данных при $r = 2$; модель GANN2, обученная на этой выборке, служит стартовой точкой иерархического обучения (см. раздел 5). Далее GANN2 дообучается на временных срезах с $r = 3$, формируя новую модель GANN3. Данный цикл последовательного дообучения продолжается до получения GANN5 как для SPD, так и для TrackML временных срезов.

Для идентификации моделей используются следующие префиксы: «SPD-» для конфигураций, обученных на временных срезах SPD, и «TrackML-» для моделей, обученных на данных TrackML. Одна эпоха обучения модели GANN r , включающая 800 временных срезов, занимала в среднем 3,5, 3,8, 4,0 и 5,2 минуты для параметров перекрытия $r = 2, 3, 4, 5$ соответственно на графическом процессоре NVIDIA V100 GPU.

Окончательным результатом данной работы являются модели, обученные на SPD и TrackML временных срезах при параметрах $\bar{N}_{ev} = 15$ и $r = 5$, обозначенные как SPD-GANN5 и TrackML-GANN5. Тестирование и валидация моделей также производятся на соответствующих независимых наборах данных.

7.1. Метрики оценки качества

Для оценки качества классификации рёбер в суперграфе используются стандартные метрики бинарной классификации — точность (*Precision*) и полнота (*Recall*):

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}, \quad (16)$$

где TP (*True Positives*) — количество правильно классифицированных истинных рёбер; FP (*False Positives*) — количество ложных рёбер, ошибочно классифицированных как истинные; FN (*False Negatives*) — количество истинных рёбер, ошибочно классифицированных как ложные.

Для оценки качества итоговой кластеризации после применения эвристических методов поиска сообществ используются те же метрики, дополненные показателем общей точности (*Accuracy*):

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \quad (17)$$

В контексте сравнения предсказанного разбиения A с эталонным (ground truth) разбиением B , величины TP, TN, FP, FN рассчитываются на основе попарного сопоставления объектов (треков):

- $TP = \sum_{i=1}^m \sum_{j=1}^n \frac{n_{ij}(n_{ij}-1)}{2}$ — число пар объектов, отнесённых к одному кластеру как в разбиении A , так и в B .
- $FN = \sum_{j=1}^n \frac{b_j(b_j-1)}{2} - TP$ — число пар объектов, принадлежащих одному кластеру в эталоне B , но разнесённых по разным кластерам в предсказании A .
- $FP = \sum_{i=1}^m \frac{a_i(a_i-1)}{2} - TP$ — число пар объектов, отнесённых к одному кластеру в предсказании A , но принадлежащих разным кластерам в эталоне B .
- $TN = \frac{N(N-1)}{2} - (TP + FP + FN)$ — число пар объектов, находящихся в разных кластерах в обоих разбиениях.

Здесь N — общее число треков во временном срезе; a_i — размер i -го кластера в разбиении A ; b_j — размер j -го кластера в разбиении B ; n_{ij} —

количество общих объектов для кластеров A_i и B_j ; m и n — общее количество сформированных кластеров в разбиениях A и B соответственно.

7.2. Результаты SPD-GANN5

Динамика изменения функции потерь модели SPD-GANN5 в течение заключительных 100 эпох обучения на тренировочном и тестовом наборах данных представлена на рис. 7. Значения функции потерь нормированы к диапазону $[0, 1]$, где единице соответствует значение лосс-функции, зафиксированное на момент завершения предыдущей стадии иерархического обучения (см. раздел 5).

Как следует из графиков на рис. 7, ошибка на тренировочном множестве снизилась более чем в два раза, а на тестовом — приблизительно в два раза. Наблюдается интенсивное уменьшение ошибки на начальном этапе, после чего кривые выходят на плато. Данный характер сходимости свидетельствует о том, что выбранный лимит в 100 эпох для данного этапа обучения является консервативным: существенное снижение ошибки на временных срезах с $r = 5$ достигается уже к 10-й эпохе. Тем не менее, фиксированная длительность в 100 эпох была принята в качестве универсального параметра для обеспечения гарантированной минимизации функции потерь на каждом из этапов иерархического обучения.

Для бинарной классификации рёбер необходимо определить значение порога принятия решения, позволяющее бинаризовать предсказания модели, назовем этот порог *порогом бинаризации*. С этой целью строится график зависимости метрик Precision и Recall от величины порога отсечения. Оптимальным в данной работе считается значение порога, при котором кривые Precision и Recall пересекаются. Такой подход позволяет минимизировать разрыв между полнотой и точностью, обеспечивая наиболее сбалансирован-

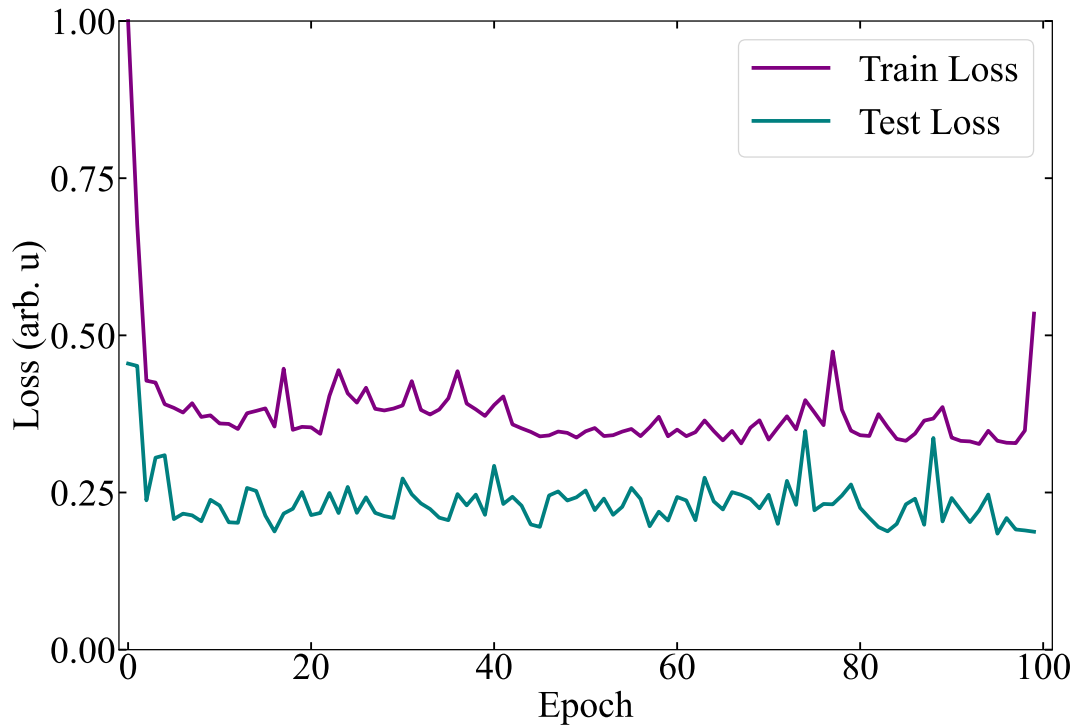


Рисунок 7. Динамика функции потерь модели SPD-GANN5 на тренировочном и тестовом наборах данных.

ные предсказания.

Из рис. 8 следует, что оптимальное значение порога находится вблизи 0,5. При этом в точке пересечения кривых показатели Precision и Recall достигают 97%. Столь высокие значения свидетельствуют о том, что модель эффективно классифицирует как истинные, так и ложные рёбра, сохраняя устойчивость к ошибкам обоих типов.

В качестве комплексной оценки качества бинарной классификации модели SPD-GANN5 в данной работе используется ROC-кривая (*Receiver Operating Characteristic curve*) [24], представленная на рис. 10. Кривая строится в осях FPR (*False Positive Rate*) и TPR (*True Positive Rate*), которые определяются следующим образом:

$$TPR = \frac{TP}{TP + FN}, \quad FPR = \frac{FP}{FP + TN}, \quad (18)$$

где параметры TP, TN, FP, FN соответствуют определениям из форму-

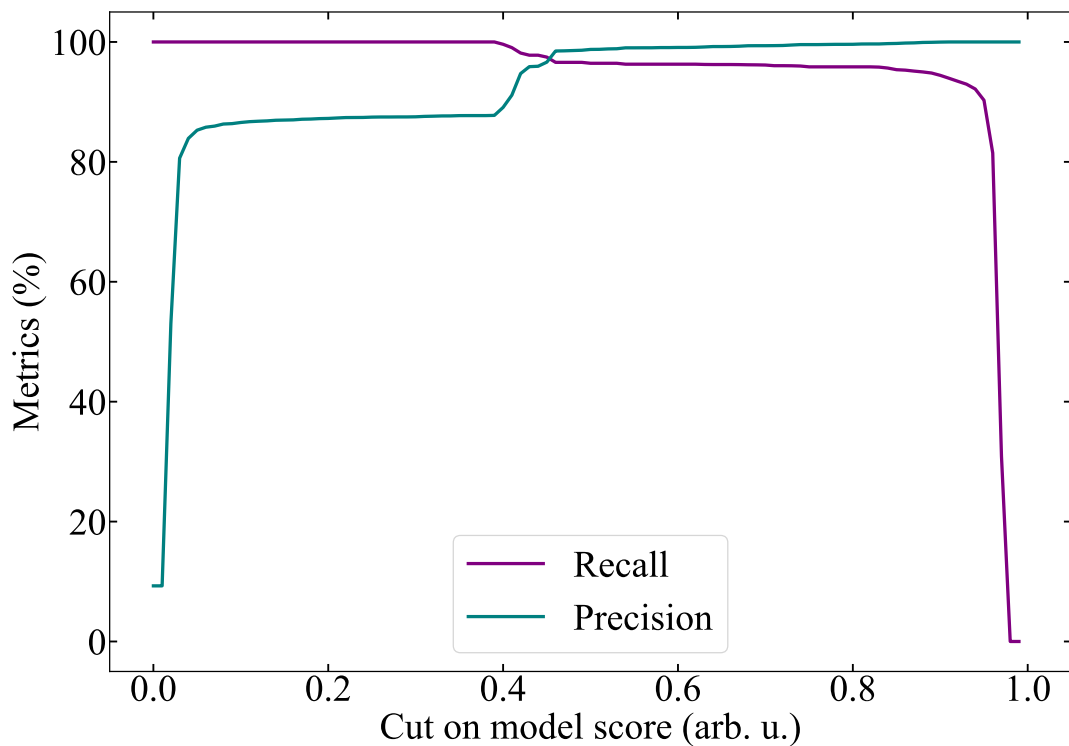


Рисунок 8. Зависимость Recall и Precision модели SPD-GANN5 от значения порога разделения вероятностей рёбер.

лы (16). В отличие от скалярных метрик, ROC-кривая наглядно демонстрирует компромисс между долей правильно классифицированных истинных рёбер и долей ошибочно классифицированных ложных рёбер. Классификация считается идеальной, если кривая проходит через точку $(0, 1)$, а площадь под ней (AUC, *Area Under the Curve*) равна 1.

Как следует из рис. 10, полученная кривая практически достигает точки $(0, 1)$, а значение AUC отличается от единицы лишь на величину порядка 10^{-4} , что подтверждает высокую дискриминационную способность модели при разделении истинных и ложных рёбер.

Преимущества стратегии иерархического обучения моделей SPD-GANN r наглядно иллюстрирует рис. 9. На графиках видно, что показатели качества начинают снижаться при достижении критического значения параметра перекрытия r . Последовательное дообучение моделей позволяет не только сдвинуть этот порог в сторону больших значений r , но и повысить точность классификации на данных с низким уровнем временного перекрытия.

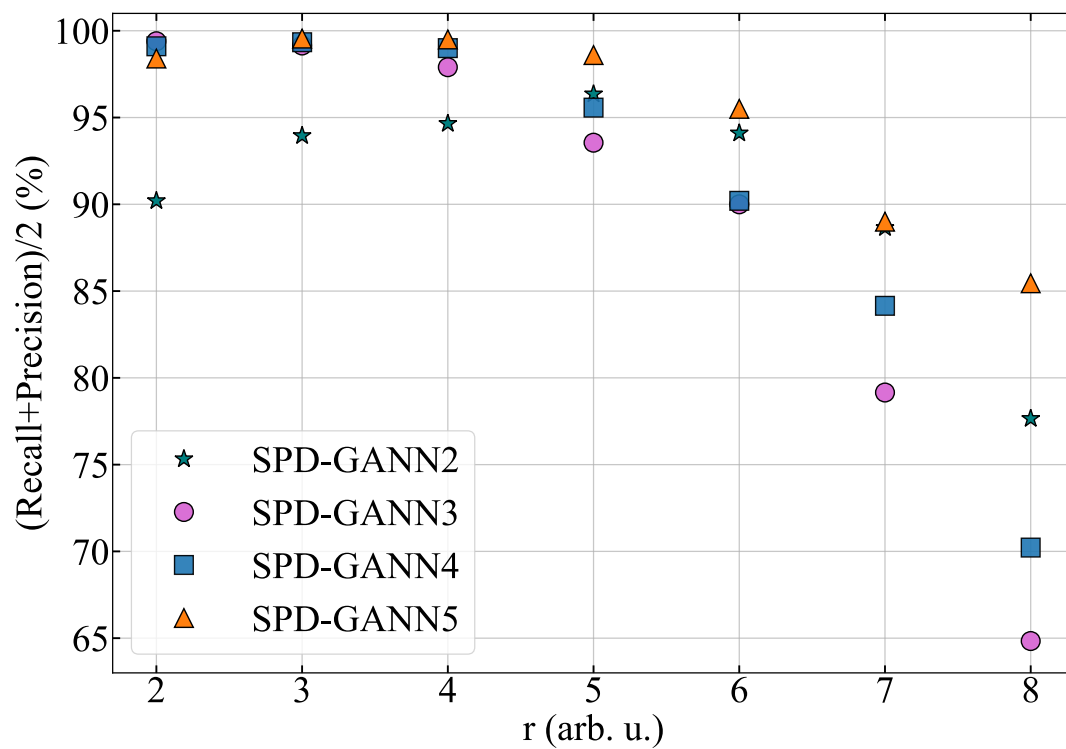


Рисунок 9. Зависимость средних значений метрик классификации моделей SPD-GANN r от параметра перекрытия r .

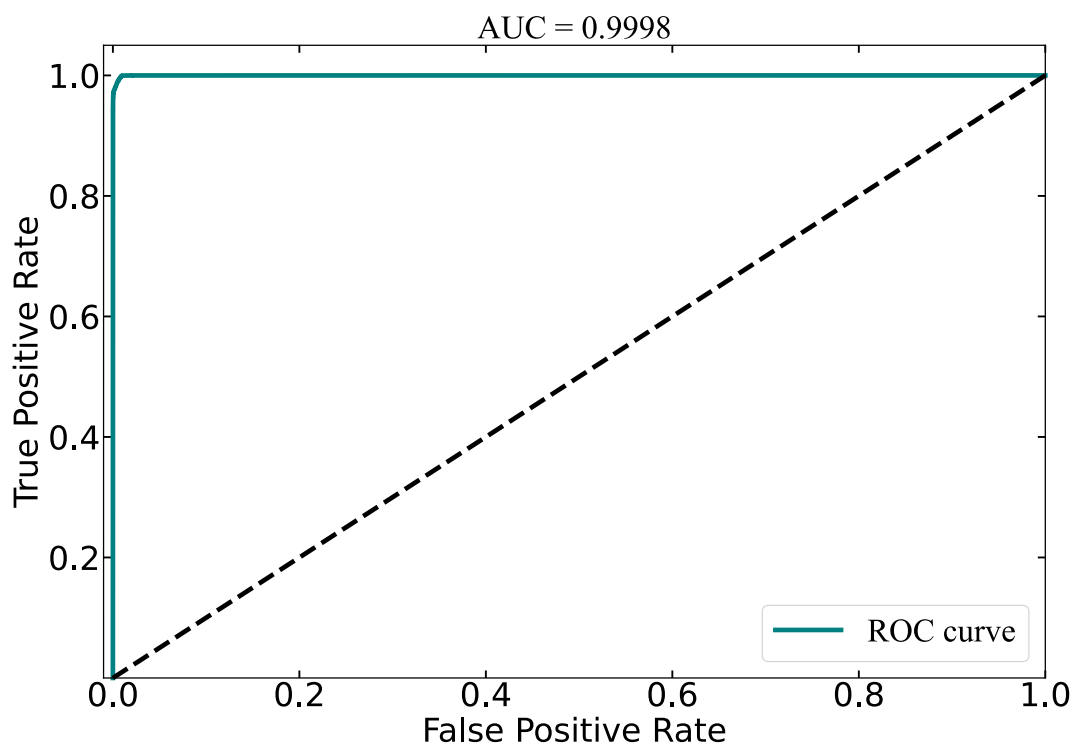


Рисунок 10. ROC-кривая и значение AUC для модели SPD-GANN5.

Распределение предсказаний модели SPD-GANN5 для истинных и ложных рёбер представлено на рис. 11. Значения, предсказанные моделью для истинных рёбер, формируют выраженный пик в области высоких вероятностей (0.9, 1.0), в то время как оценки для ложных рёбер сосредоточены в диапазоне низких значений (0.0, 0.1). Наблюдается незначительное перекрытие распределений, локализованное преимущественно в центральной области.

Характерный наклон (отсечка) распределения оценок истинных рёбер вблизи значения 0,4 обусловлен спецификой иерархического обучения. В процессе последовательного дообучения от стартовой модели SPD-GANN2 до итоговой SPD-GANN5 сформировался выраженный порог, минимизирующий вероятность ложноотрицательных срабатываний. Данная особенность была выработана моделью для минимизации штрафа за пропуск истинных рёбер (ошибки II рода), который был заложен в функции потерь (13) через весовые коэффициенты.

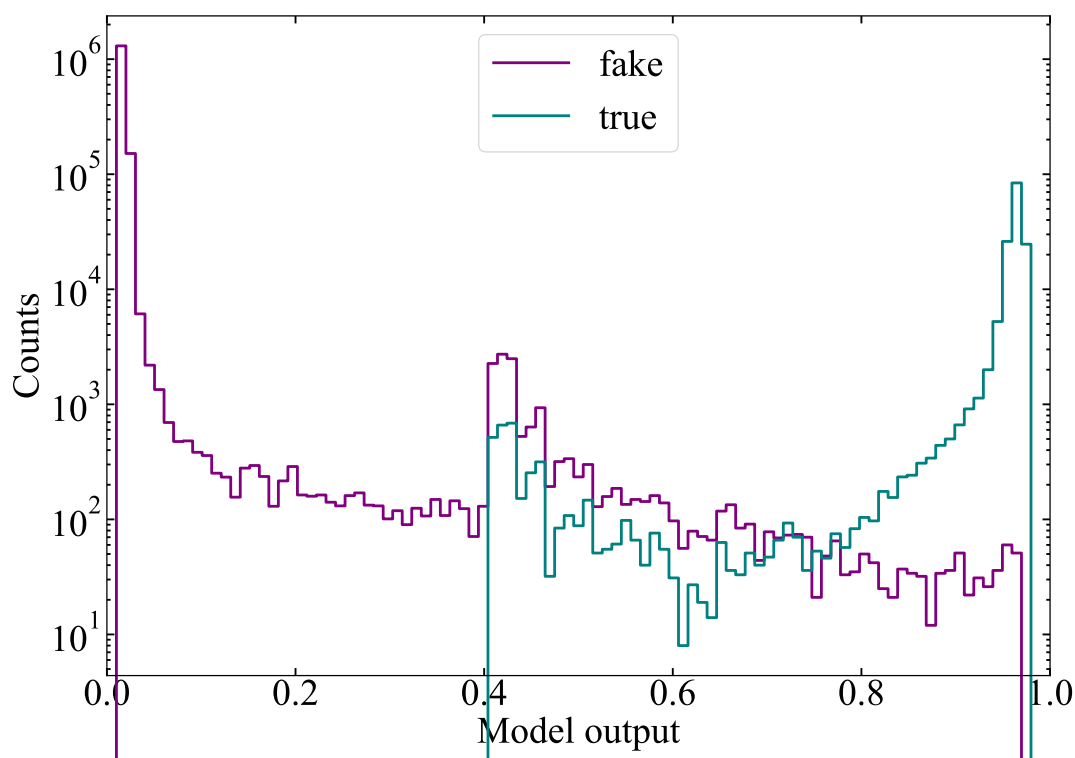


Рисунок 11. Распределения предсказаний модели SPD-GANN5 для истинных и ложных рёбер.

Таким образом, наличие данного порога позволяет сохранить подавляющее большинство истинных рёбер при необходимости, не допуская критического снижения точности. Однако в рамках настоящего исследования изменение порога классификации с 0.5 на 0.4 было признано нецелесообразным, так как точность (*Precision*) при этом снижается с 97.2% до 91.4%, что негативно сказывается на качестве последующей кластеризации. Итоговые метрики качества кластеризации графа после применения Лувенского метода — *Recall*, *Precision* и *Accuracy* — составили 95.6%, 94.7% и 99.3% соответственно.

7.3. Результаты Track-GANN5

Для анализа стабильности предложенного подхода модель GANN r была обучена и протестирована на временных срезах с треками из набора данных TrackML с использованием той же иерархической методики, что и в случае SPD-GANN r . Динамика изменения функции потерь модели TrackML-GANN5 на протяжении заключительных 100 эпох обучения представлена на рис. 12. За указанный период ошибка уменьшилась примерно в 1.3 раза как на тренировочном, так и на тестовом множествах, что указывает на успешную сходимость процесса дообучения.

Зависимость *Recall* и *Precision* от порога разделения для модели TrackML-GANN5 представлена на рис. 13. Оптимальное значение порога бинаризации приблизительно равно 0.5; в точке пересечения кривых обе метрики достигают значения 94%.

Результаты ROC-анализа представлены на рис. 14. Величины *TPR* и *FPR* рассчитывались согласно формулам (18). Кривая практически достигает идеальной точки (0, 1), а значение площади под кривой AUC отличается от единицы на величину порядка 10^{-3} . Данные показатели подтверждают высокую

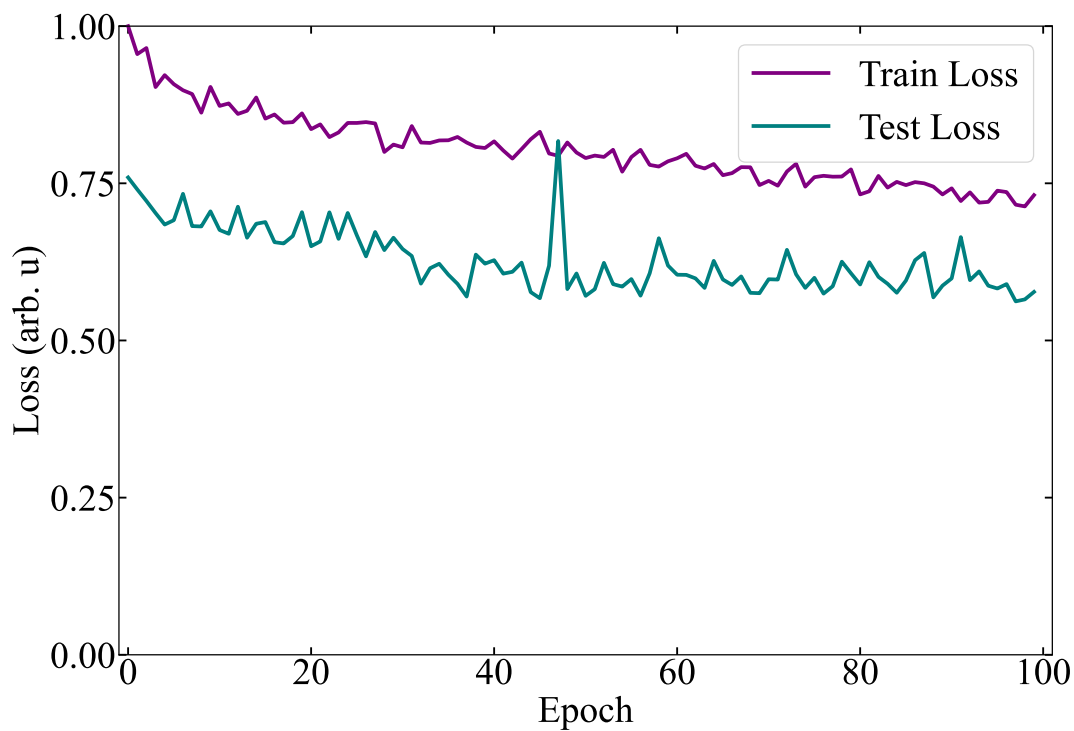


Рисунок 12. Динамика функции потерь TrackML-GANN5 на тренировочном и тестовом наборах данных.

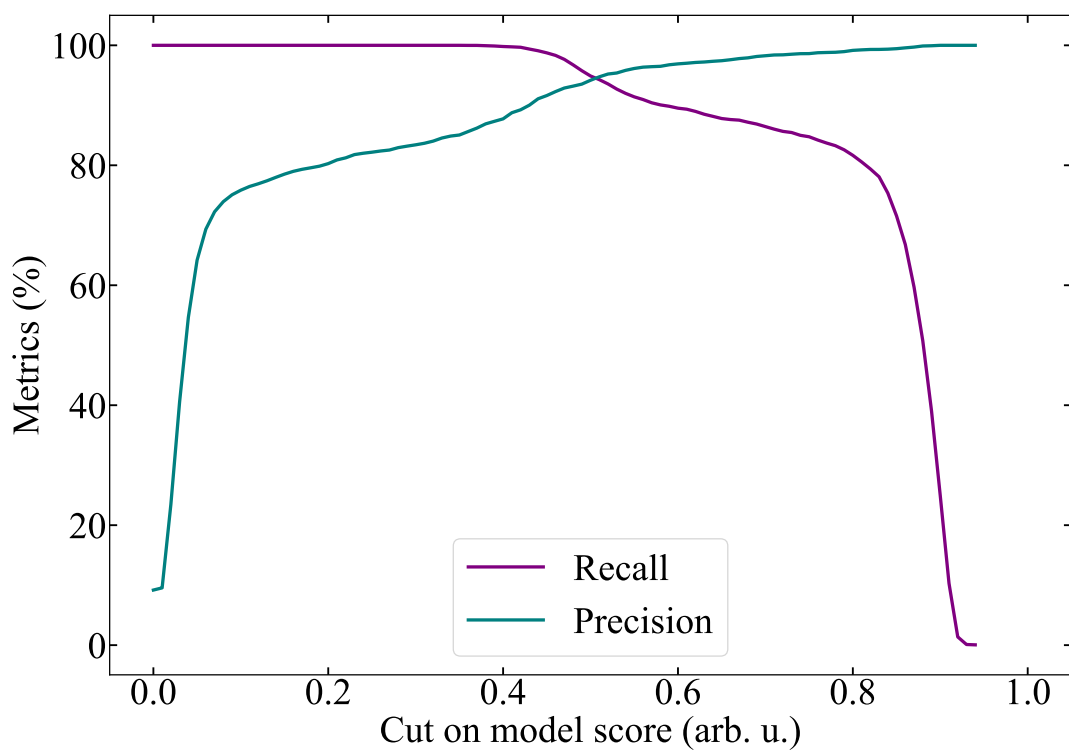


Рисунок 13. Зависимость метрик Recall и Precision модели TrackML-GANN5 от значения порога разделения вероятностей рёбер.

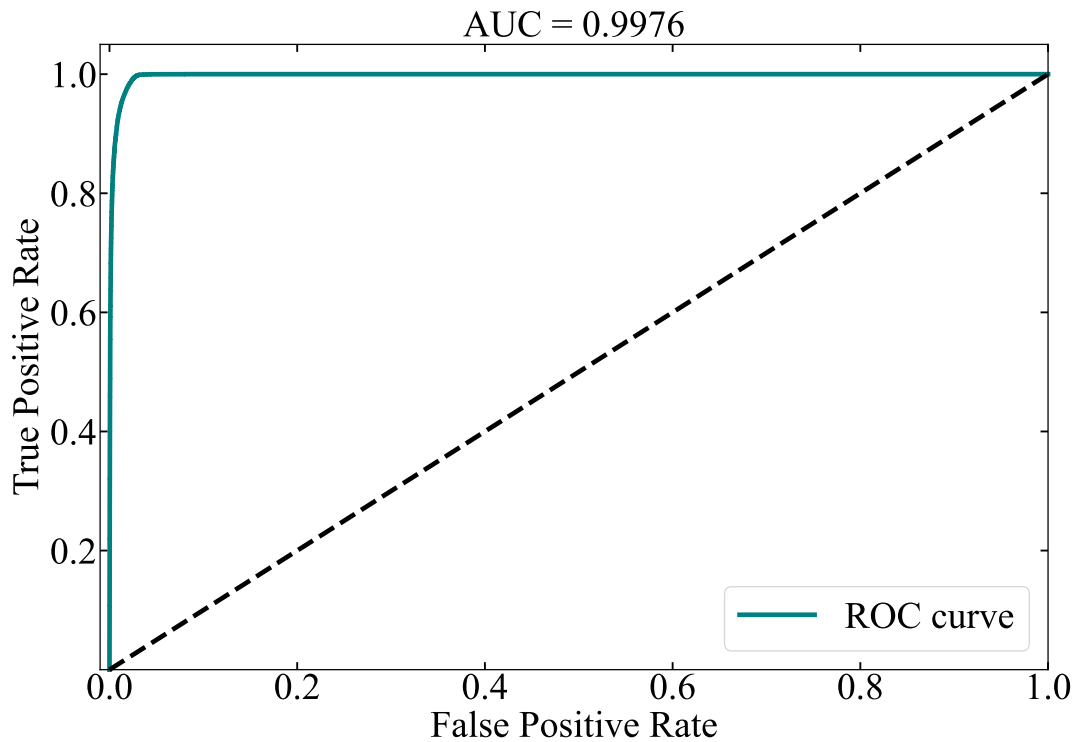


Рисунок 14. ROC-кривая и значение AUC для модели TrackML-GANN5.

эффективность модели TrackML-GANN5 в задаче бинарной классификации рёбер суперграфа.

Преимущества иерархического подхода для данных TrackML наглядно иллюстрирует рис. 15, где показана устойчивость метрик при варьировании параметра перекрытия r .

Распределение предсказаний модели TrackML-GANN5 для истинных и ложных рёбер представлено на рис. 16. Как и в случае с SPD-GANN5 (см. раздел 7.1), значения для истинных рёбер формируют выраженный пик в области $(0.9, 1.0)$, в то время как оценки для ложных рёбер сосредоточены в диапазоне $(0.0, 0.1)$.

Наблюдается характерный порог отсечения (наклон) распределения истинных рёбер вблизи значения 0.35. Его природа аналогична модели SPD-GANN5 и обусловлена адаптацией сети к минимизации ложноотрицательных срабатываний в процессе иерархического обучения. Снижение порога разделения с 0.5 до 0.35 приводит к падению Precision с 94.7% до 85.1%, что

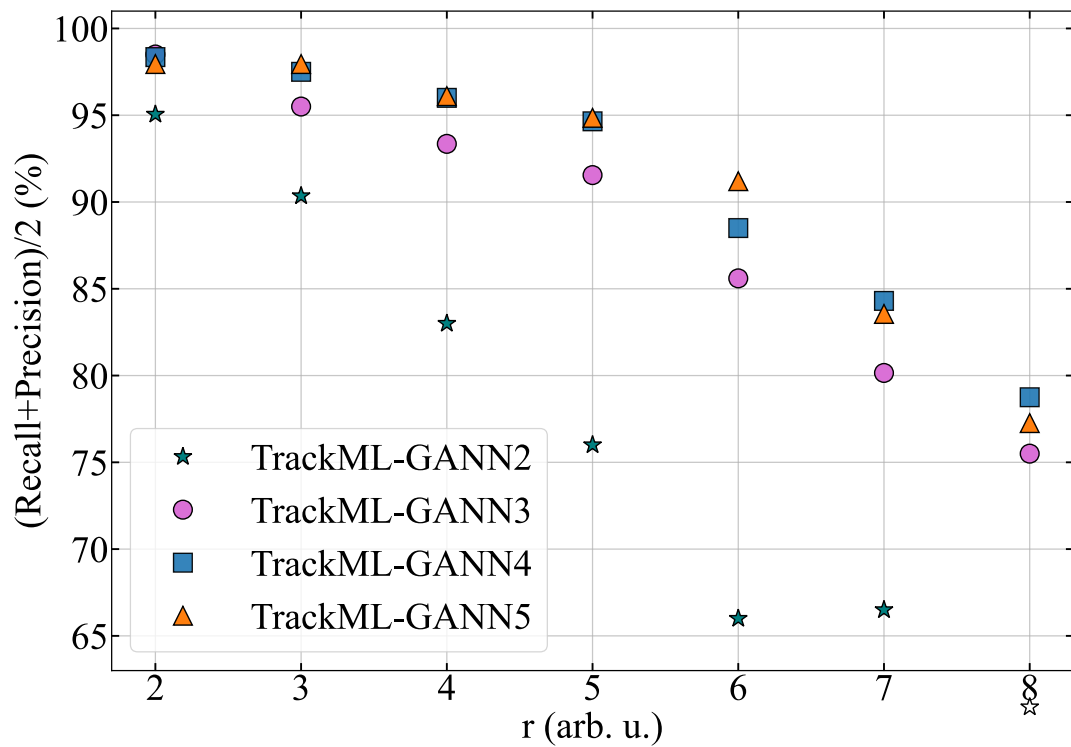


Рисунок 15. Зависимость средних значений метрик классификации моделей TrackML-GANN r от параметра перекрытия r .

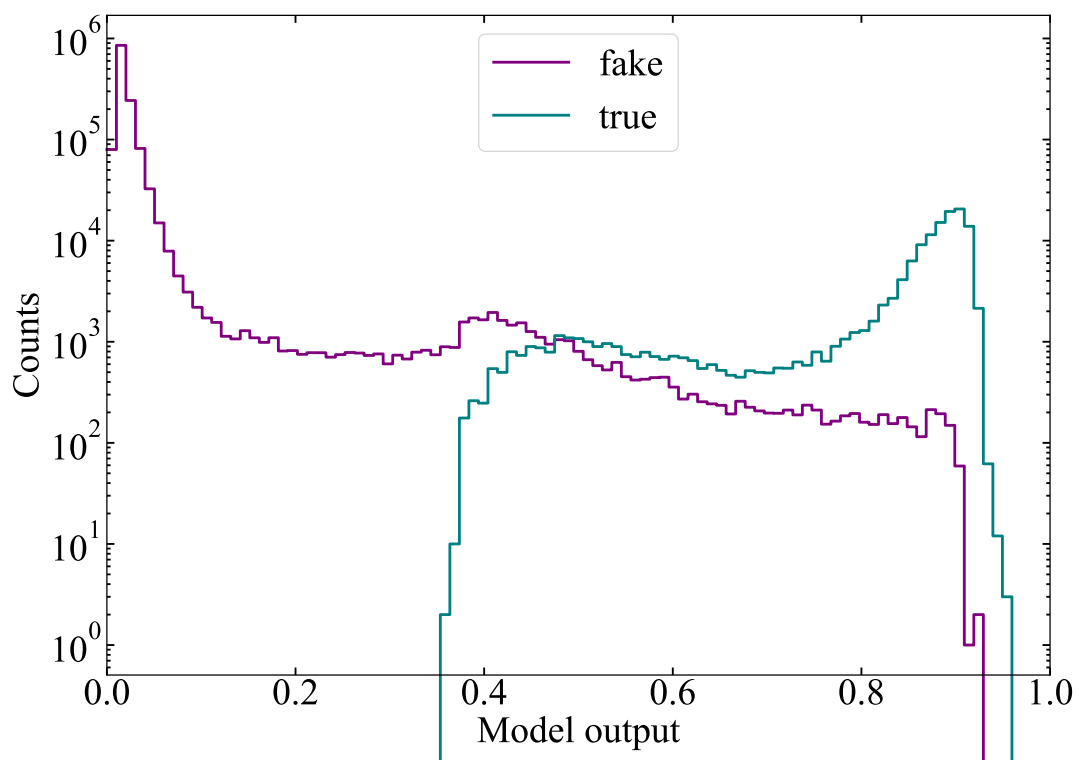


Рисунок 16. Распределения предсказаний модели TrackML-GANN5 для истинных и ложных рёбер.

существенно ухудшает чистоту последующей кластеризации треков.

Выводы

Результаты работы позволяют сделать вывод о значительном преимуществе подхода, основанного на архитектуре GANN, над подходом на базе сиамской нейронной сети, предложенным в работе [2]. Общая точность кластеризации SPD-GANN5 превышает точность сиамской сети на десятки процентов в сценариях с нефиксированным числом событий в временном срезе.

Представленный в работе подход обладает рядом ключевых преимуществ как перед классическими алгоритмами, так и перед методами машинного обучения, ранее применявшимися к задаче классификации треков по событиям:

- Точность: графовая нейронная сеть Track-GANN достигает 95% точности кластеризации.
- Инвариантность: благодаря графовой структуре модель способна эффективно обрабатывать произвольное число событий, треков и хитов.
- Масштабируемость: иерархический подход к обучению позволяет адаптировать модель под различные конфигурации эксперимента без существенной потери эффективности.
- Скорость: перенос всех вычислений на GPU обеспечивает высокую скорость обработки — порядка десятков временных срезов в секунду.

Заключение

В рамках данной работы была предложена, разработана и протестирована архитектура на основе графовых нейронных сетей (GNN), предназначенная для кластеризации траекторий заряженных частиц по событиям во временных срезах эксперимента SPD. Предложенный иерархический подход позволил существенно повысить эффективность разделения треков по первичным событиям в условиях высокой плотности данных.

Разработанная модель GANN r , несмотря на более высокую вычислительную сложность по сравнению с решениями, представленными в [2], обладает инвариантностью к числу событий во временном срезе. Благодаря специфике вычислительного графа, архитектура GANN r демонстрирует хорошую масштабируемость при минимальных потерях эффективности на больших графах.

На симулированных данных SPD при параметрах $\overline{N}_{ev} = 15$ и $r = 5$ была обучена модель SPD-GANN5, метрики полноты (*Recall*) и точности (*Precision*) которой достигают 97% при оптимальном пороге разделения 0.5. Для финальной кластеризации суперграфа был адаптирован эвристический Лувенский метод поиска сообществ, полностью реализованный для работы с тензорными данными в среде *PyTorch*. Итоговые показатели качества кластеризации (после применения Лувенского метода) для модели SPD-GANN5 составили: *Precision* — 95.6%, *Recall* — 94.7% и *Accuracy* — 99.3%. Производительность модели на этапе инференса составляет порядка десятков временных срезов в минуту на графическом процессоре NVIDIA V100 GPU.

Для верификации устойчивости предложенного метода было проведено кросс-тестирование на данных международного соревнования TrackML. Модель TrackML-GANN5 продемонстрировала значения *Recall* и *Precision* на уровне 94%. Полученные результаты подтверждают высокую обобщающую

способность и надежность разработанной архитектуры. Также на обоих наборах данных была экспериментально доказана эффективность стратегии иерархического обучения. Предлагаемый алгоритм ориентирован на офлайн-обработку данных. Все исходные коды, конфигурационные файлы и веса предобученных моделей размещены в открытом репозитории [25].

Результаты работы были представлены и апробированы на следующих конференциях, посвященных физике высоких энергий и методам машинного обучения:

- Весенняя школа по информационным технологиям, 9-10 апреля, 2025 год, ЛИТ ОИЯИ, устный доклад - Разделение траекторий частиц по событиям столкновений пучков накопленных в едином временном срезе в детекторе SPD на ускорителе NICA с помощью графовых нейронных сетей (награда: "Лучший доклад")
- Deep Learning in Computer Physics (DLCP-2025), 2-4 июля, 2025 год, МГУ, устный доклад - Графовая нейронная сеть с механизмом внимания для кластеризации треков частиц по событиям в эксперименте SPD на ускорителе NICA
- 62st meeting of the PAC for Particle Physics Monday 20 June 2025, постерный доклад - Separation of particle trajectories by events accumulated over a single timeslice in the SPD NICA detector using graph neural networks
- XXV Международная Байкальская летняя школа по физике элементарных частиц и астрофизике, 11 по 18 июля 2025 год, Большие Коты, постерный доклад - Separation of particle trajectories by events accumulated over a single timeslice in the SPD NICA detector using graph neural networks
- Conference of high energy physics (CHEP-2025), 29 сентября -04 октября, 2025 год, Ереван, устный доклад - Graph Neural Network with an Attention

Mechanism for Clustering Particle Tracks by Events in the SPD Experiment

- Workshop on Quantum Computing and Machine Learning (QCML), 11-14 October, 2025, Cairo, устный доклад - Graph neural network with an attention mechanism for clustering particle tracks by events in the SPD experiment

В дальнейшие планы по развитию предложенного метода входят следующие задачи:

1. Протестировать модель на детальной (*full simulation*) симуляции, максимально приближенной к реальным условиям и геометрии детектора эксперимента SPD.
2. Адаптировать блок GE (*Graph Encoder*) для задачи прямой реконструкции физических параметров отдельных частиц (импульс, тип частицы) непосредственно во временном срезе.
3. Подготовить и опубликовать цикл научных работ, посвященных применению глубоких графовых нейронных сетей для офлайн-обработки данных полученных на экспериментах физики высоких энергий.

Студент _____ Омелянчук С. С.

Научный руководитель _____ Леонтьев В. В.

Руководитель образовательной программы _____ Матвеев В. А.

«_____» _____ 2026 г.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Abazov V. et al. Technical Design Report of the Spin Physics Detector at NICA // Natural Sci. Rev. — 2024. — Vol. 1. — P. 1. — 2404.08317.
- [2] Unraveling Time-Slices of Events in the SPD Experiment / Borisov M., Goncharov P., Ososkov G., and Rusov D. // [Physics of Particles and Nuclei](#). — 2024. — 06. — Vol. 55. — P. 453–455.
- [3] [The Particle Track Reconstruction based on deep Neural networks](#) / Baranov D., Mitsyn S., Goncharov P., and Ososkov G. // European Physical Journal Web of Conferences. — EDP. — 2019. — July. — Vol. 214 of European Physical Journal Web of Conferences. — P. 06018. — 1812.03859.
- [4] [TrackML : a tracking Machine Learning challenge](#) / Golling T., Amrouche S., Kiehn M., Calafiura P., Farrell S., Gray H., Estrade V., Germain C., Gligorov V., Guyon I., Hushchyn M., Innocente V., Salzburger A., Moyse E., Rousseau D., Yilmaz Y., and Vlimant J.-r. // Proceedings of the 39th International Conference on High Energy Physics. — 2019. — 08. — P. 159.
- [5] Deep Learning Methods in High Luminosity Track Reconstruction Scenario: Applying TrackNET to TrackML Challenge / Rusov D., Goncharov P., Zhemchugov A., and Ososkov G. // [Physics of Particles and Nuclei](#). — 2025. — Vol. 56, no. 6. — P. 1599–1603.
- [6] Deep Tracking for the SPD Experiment / Rusov D., Goncharov P., Zhemchugov A., and Ososkov G. // [Phys. Part. Nucl. Lett.](#) — 2023. — Vol. 20, no. 5. — P. 1180–1182.
- [7] Pileup mitigation at the Large Hadron Collider with graph neural networks / Arjona Martínez J., Cerri O., Pierini M., Spiropulu M., and Vlimant J.-R. // [Eur. Phys. J. Plus](#). — 2019. — Vol. 134, no. 7. — P. 333. — 1810.07988.

- [8] Graph Neural Network-based track finding in the LHCb vertex detector / Correia A., Giasemis F. I., Garroum N., Gligorov V. V., and Granado B. // [JINST](#). — 2024. — Vol. 19, no. 12. — P. P12022. — 2407.12119.
- [9] Ju X. et al. Graph Neural Networks for Particle Reconstruction in High Energy Physics detectors // 33rd Annual Conference on Neural Information Processing Systems. — 2020. — 3. — 2003.11603.
- [10] Duarte J., Vlimant J.-R. Graph Neural Networks for Particle Tracking and Reconstruction. — 2020. — 12. — 2012.01249.
- [11] Application of Geometric Deep Learning for Tracking of Hyperons in a Straw Tube Detector / Akram A., Ju X., Papenbrock M., Taylor J., Stockmanns T., and Schönning K. // [Comput. Softw. Big Sci.](#) — 2025. — Vol. 9, no. 1. — P. 17. — 2503.14305.
- [12] Hierarchical Graph Neural Networks for Particle Track Reconstruction / Liu R., Calafiura P., Farrell S., Ju X., Murnane D. T., and Pham T. M. // 21th International Workshop on Advanced Computing and Analysis Techniques in Physics Research: AI meets Reality. — 2023. — 3. — 2303.01640.
- [13] Kipf T. N., Welling M. Semi-Supervised Classification with Graph Convolutional Networks // CoRR. — 2016. — Vol. abs/1609.02907. — arXiv : [1609.02907](#).
- [14] Graph Attention Networks / Veličković P., Cucurull G., Casanova A., Romero A., Liò P., and Bengio Y. // [arXiv e-prints](#). — 2017. — . — P. arXiv:1710.10903. — 1710.10903.
- [15] How Powerful are Graph Neural Networks? / Xu K., Hu W., Leskovec J., and Jegelka S. // [arXiv e-prints](#). — 2018. — . — P. arXiv:1810.00826. — 1810.00826.
- [16] Rumelhart D. E., Hinton G. E., Williams R. J. Learning representations by back-propagating errors // [Nature](#). — 1986. — Vol. 323, no. 6088. — P. 533–536.

- [17] Loshchilov I., Hutter F. Decoupled Weight Decay Regularization // [arXiv e-prints](#). — 2017. — Nov. — P. arXiv:1711.05101. — 1711.05101.
- [18] Batch-balanced focal loss: a hybrid solution to class imbalance in deep learning / Singh J., Beeche C., Shi Z., Beale O., Rosin B., Leader J., and Pu J. // [Journal of Medical Imaging](#). — 2023. — 06. — Vol. 10.
- [19] Clauset A., Newman M. E. J., Moore C. Finding community structure in very large networks // [Phys. Rev. E](#). — 2004. — Dec. — Vol. 70. — P. 066111. — Access mode: <https://link.aps.org/doi/10.1103/PhysRevE.70.066111>.
- [20] Raghavan U. N., Albert R., Kumara S. Near linear time algorithm to detect community structures in large-scale networks // [Physical Review E](#). — 2007. — Sep. — Vol. 76, no. 3. — Access mode: <http://dx.doi.org/10.1103/PhysRevE.76.036106>.
- [21] Girvan M., Newman M. E. J. Community structure in social and biological networks // [Proceedings of the National Academy of Sciences](#). — 2002. — Vol. 99, no. 12. — P. 7821–7826. — <https://www.pnas.org/doi/pdf/10.1073/pnas.122653799>.
- [22] An Improved Louvain Algorithm for Community Detection / Zhang J., Fei J., Song X., and Feng J. // [Mathematical Problems in Engineering](#). — 2021. — 11. — Vol. 2021. — P. 1–14.
- [23] PyTorch: An Imperative Style, High-Performance Deep Learning Library / Paszke A., Gross S., Massa F., Lerer A., Bradbury J., Chanan G., Killeen T., Lin Z., Gimelshein N., Antiga L., Desmaison A., Köpf A., Yang E., DeVito Z., Raison M., Tejani A., Chilamkurthy S., Steiner B., Fang L., Bai J., and Chintala S. // [arXiv e-prints](#). — 2019. — Dec. — P. arXiv:1912.01703. — 1912.01703.

- [24] Junge M. R. J., Dettori J. R. ROC Solid: Receiver Operator Characteristic (ROC) Curves as a Foundation for Better Diagnostic Tests // [Global Spine Journal](#). — 2018. — Vol. 8, no. 4. — P. 424–429. — PMID: 29977728. <https://doi.org/10.1177/2192568218778294>.
- [25] Omelianchuk S. TrackGANN. — <https://github.com/SavelyOm/TrackGANN>. — 2026. — GitHub repository.