МОДЕЛИ В ФИЗИКЕ И ТЕХНОЛОГИИ

УДК: 004.85,004.93,539.1.05

# Нейросетевая реконструкция треков частиц для внутреннего CGEM-детектора эксперимента BESIII

## Г. А. Ососков<sup>1</sup>, О. В. Бакина<sup>1</sup>, Д. А. Баранов<sup>1</sup>, П. В. Гончаров<sup>1,2</sup>, И. И. Денисенко<sup>1</sup>, А. С. Жемчугов<sup>1</sup>, Ю. А. Нефедов<sup>1,а</sup>, А. В. Нечаевский<sup>1</sup>, А. Н. Никольская<sup>3</sup>, Е. М. Щавелев<sup>3</sup>, Л. Ван<sup>4</sup>, Ш. Сунь<sup>4,5</sup>, Я. Чжан<sup>4</sup>

<sup>1</sup>Объединенный институт ядерных исследований, Россия, 141980, г. Дубна, Московская обл., ул. Жолио-Кюри, д. 6 <sup>2</sup>Государственный университет «Дубна», Россия, 141982, г. Дубна, Московская обл., ул. Университетская, д. 19 <sup>3</sup>Санкт-Петербургский государственный университет, Россия, 199034, г. Санкт-Петербург, Университетская набережная, д. 7-9 <sup>4</sup>Институт физики высоких энергий АН КНР, Китайская Народная Республика, 100049, г. Пекин, ул. Юйцюаньлу, д. 19В <sup>5</sup>Университет Академии наук КНР, Китайская Народная Республика, 100049, г. Пекин, ул. Юйцюаньлу, д. 19А E-mail: <sup>a</sup> Nefedov.Yury@jinr.ru

> Получено 29.07.2020, после доработки — 22.09.2020. Принято к публикации 25.09.2020.

Реконструкция траекторий заряженных частиц в трековых детекторах является ключевой проблемой анализа экспериментальных данных для физики высоких энергий и ядерной физики. Поток данных в современных экспериментах растет день ото дня, и традиционные методы трекинга уже не в состоянии соответствовать этим объемам данных по скорости обработки. Для решения этой проблемы нами были разработаны два нейросетевых алгоритма, использующих методы глубокого обучения, для локальной (каждый трек в отдельности) и глобальной (все треки в событии) реконструкции треков применительно к данным трекового GEM-детектора эксперимента BM@N ОИЯИ. Преимущество глубоких нейронных сетей обусловлено их способностью к обнаружению скрытых нелинейных зависимостей в данных и возможностью параллельного выполнения операций линейной алгебры, лежащих в их основе.

В данной статье приведено описание исследования по обобщению этих алгоритмов и их адаптации к применению для внутреннего поддетектора СGEM (BESIII ИФВЭ, Пекин). Нейросетевая модель RDGraphNet для глобальной реконструкции треков, разработанная на основе реверсного орграфа, успешно адаптирована. После обучения на модельных данных тестирование показало обнадеживающие результаты: для распознавания треков полнота (recall) составила 98 % и точность (precision) — 86 %. Однако адаптация «локальной» нейросетевой модели TrackNETv2 потребовала учета специфики цилиндрического детектора CGEM (BESIII), состоящего всего из трех детектируюцих слоев, и разработки дополнительного нейроклассификатора для отсева ложных треков. Полученная программа TrackNETv2.1 протестирована в отладочном режиме. Значение полноты на первом этапе обработки составило 99 %. После применения классификатора точность составила 77 %, при незначительном снижении показателя полноты до 94 %. Данные результаты предполагают дальнейшее совершенствование модели локального трекинга.

Ключевые слова: реконструкция треков, GEM-детекторы, глубокое обучение, сверточные нейронные сети, графовые нейросети

Исследование выполнено при финансовой поддержке РФФИ и ГФЕН Китая в рамках научного проекта № 19-57-53002. Часть исследования, связанная с адаптацией программы TrackNETv2, выполнена при финансовой поддержке РФФИ в рамках научного проекта № 18-02-40101.

© 2020 Геннадий Алексеевич Ососков, Ольга Валерьевна Бакина, Дмитрий Александрович Баранов, Павел Владимирович Гончаров, Игорь Игоревич Денисенко, Алексей Сергеевич Жемчугов, Юрий Анатольевич Нефедов, Андрей Васильевич Нечаевский, Анастасия Николаевна Никольская, Егор Михайлович Щавелев, Лянлян Ван, Шенсень Сунь, Яо Чжан Статья доступна по лицензии Creative Commons Attribution-NoDerivs 3.0 Unported License.

Чтобы получить текст лицензии, посетите веб-сайт http://creativecommons.org/licenses/by-nd/3.0/ или отправьте письмо в Creative Commons, PO Box 1866, Mountain View, CA 94042, USA. UDC: 004.85,004.93,539.1.05

# Tracking on the BESIII CGEM inner detector using deep learning

G. A. Ososkov<sup>1</sup>, O. V. Bakina<sup>1</sup>, D. A. Baranov<sup>1</sup>, P. V. Goncharov<sup>1,2</sup>, I. I. Denisenko<sup>1</sup>, A. S. Zhemchugov<sup>1</sup>, Yu. A. Nefedov<sup>1,a</sup>, A. V. Nechaevskiy<sup>1</sup>, A. N. Nikolskaya<sup>3</sup>, E. M. Shchavelev<sup>3</sup>, L.-L. Wang<sup>4</sup>, S.-S. Sun<sup>4,5</sup>, Y. Zhang<sup>4</sup>

<sup>1</sup>Joint Institute for Nuclear Research,
6 Joliot-Curie st., Dubna, Moscow Region, 141980, Russia
<sup>2</sup>Dubna State University,
19 Universitetskaya st., Dubna, Moscow Region, 141982, Russia
<sup>3</sup>St. Petersburg State University,
7-9 Universitetskaya Emb., St. Petersburg, 199034, Russia
<sup>4</sup>Institute of High Energy Physics CAS,
19B Yuquan Road, Shijingshan District, Beijing, 100049, China
<sup>5</sup>University of Chinese Academy of Sciences,
19A Yuquan Road, Shijingshan District, Beijing, 100049, China

E-mail: a Nefedov.Yury@jinr.ru

Received 29.07.2020, after completion – 22.09.2020. Accepted for publication 25.09.2020.

The reconstruction of charged particle trajectories in tracking detectors is a key problem in the analysis of experimental data for high energy and nuclear physics.

The amount of data in modern experiments is so large that classical tracking methods such as Kalman filter can not process them fast enough. To solve this problem, we have developed two neural network algorithms of track recognition, based on deep learning architectures, for local (track by track) and global (all tracks in an event) tracking in the GEM tracker of the BM@N experiment at JINR (Dubna). The advantage of deep neural networks is the ability to detect hidden nonlinear dependencies in data and the capability of parallel execution of underlying linear algebra operations.

In this work we generalize these algorithms to the cylindrical GEM inner tracker of BESIII experiment. The neural network model RDGraphNet for global track finding, based on the reverse directed graph, has been successfully adapted. After training on Monte Carlo data, testing showed encouraging results: recall of 98% and precision of 86% for track finding.

The local neural network model TrackNETv2 was also adapted to BESIII CGEM successfully. Since the tracker has only three detecting layers, an additional neuro-classifier to filter out false tracks have been introduced. Preliminary tests demonstrated the recall value at the first stage of 99%. After applying the neuro-classifier, the precision was 77% with a slight decrease of the recall to 94%. This result can be improved after the further model optimization.

Keywords: track reconstruction, GEM detectors, deep learning, convolutional neural networks, graph neural networks

Citation: Computer Research and Modeling, 2020, vol. 12, no. 6, pp. 1361–1381 (Russian).

This work was supported in part by Russian Foundation for Basic Research and the State Fund for Scientific Research of China under Contract No. 19-57-53002; The Russian Foundation for Basic Research under Contract No. 18-02-40101.

© 2020 Gennady A. Ososkov, Olga V. Bakina, Dmitry A. Baranov, Pavel V. Goncharov, Igor I. Denisenko, Aleksey S. Zhemchugov, Yury A. Nefedov, Andrey V. Nechaevskiy, Anastasiya N. Nikolskaya, Egor M. Shchavelev, Liang-Liang Wang, Sheng-Sen Sun, Yao Zhang This work is licensed under the Creative Commons Attribution-NoDerivs 3.0 Unported License. To view a new of the license wire the travel (Creative Commons Attribution-NoDerivs 3.0 Unported License.)

To view a copy of this license, visit http://creativecommons.org/licenses/by-nd/3.0/ or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

1363

### Введение

Основной целью экспериментальной физики высоких энергий является изучение структуры атомного ядра и элементарных частиц и закономерностей их взаимных превращений и взаимодействия между собой. Для этого ускоренные ядра атомов или элементарные частицы (протоны, электроны и так далее) сталкиваются либо с атомными ядрами неподвижной мишени, либо со встречным пучком ускоренных ядер или элементарных частиц. В результате столкновения образуются вторичные элементарные частицы, которые регистрируются в детекторах, расположенных рядом с точкой столкновения. В зависимости от конструкции детекторов с их помощью можно измерить траекторию движения и определить энергию, импульс и тип вторичных частиц. Статистическая обработка данных, полученных при большом количестве столкновений (событий), и анализ полученных распределений позволяют выявить физические закономерности взаимодействий и сравнить их с теоретическими предсказаниями.



Рис. 1. Пример события образования и распада чармониевого резонанса в эксперименте BESIII

Одним из важнейших элементов практически любой экспериментальной установки являются трековые детекторы, которые позволяют определить координату прохождения частицы и восстановить трек частицы, то есть траекторию ее движения. Алгоритм восстановления трека часто называют трекингом. В случае если движение происходит в магнитном поле, искривление трека позволяет вычислить импульс частицы. Таким образом, координатное разрешение трековых детекторов оказывается непосредственно связанным с импульсным разрешением установки, что является одной из важнейших характеристик. Другим параметром, влияющим на импульсное разрешение, является количество точек в пространстве, по которым восстанавливается трек. Далее эти точки мы будем называть хитами (от англ. hit — удар, попадание). Количество хитов определяется конструкцией детектора и количеством чувствительных элементов на пути частицы. Для восстановления хита из отклика детектора, как правило, требуется объединение информации из соседних элементов детектора и привлечение геометрической информации об их взаимном расположении.

Для восстановления трека требуется определить, какие сработавшие элементы детектора относятся к одной и той же частице, определить координаты соответствующих точек и затем аппроксимировать их заданной моделью трека. Модель трека в идеальном случае может быть довольно простой. Например, движение заряженной частицы в однородном магнитном поле можно описать обычной спиралью. Однако на практике приходится учитывать эффекты, связанные с неоднородным магнитным полем, рассеянием и ионизационными потерями энергии частиц во время движения и другими явлениями, поэтому модель трека оказывается существенно более сложной.

Тем не менее в случае единственного трека в событии его восстановление обычно не составляет трудностей. Однако, как правило, в событии регистрируется одновременно больше одного трека, и количество треков (множественность) может быть очень большим и достигать нескольких десятков или сотен треков. В этом случае для восстановления треков нужно сначала определить, какие хиты относятся к каждому треку. При этом, в зависимости от конструкции детектора, помимо хитов, обусловленных прохождением искомой частицы, в событии могут присутствовать хиты, вызванные шумом или ложной реконструкцией соответствующих точек. Например, если чувствительный элемент детектора представляет собой проволоку или полоску (например, в случае проволочных газовых детекторов или детекторов GEM) и координата срабатывания определяется по их перекрещиванию, в случае нескольких треков возможные комбинации перекрещиваний чувствительных элементов приведут к образованию ложных хитов, которые мы также будем называть фейками (от англ. fake — ложный). Очевидно, только часть восстановленных хитов будет относиться непосредственно к частице. Таким образом, задача распознавания треков, то есть группирования хитов по принадлежности к определенному треку, становится довольно сложной как с точки зрения применяемых алгоритмов, так и с точки зрения затрат вычислительных ресурсов.

Алгоритмы трекинга эволюционировали вместе с развитием экспериментальных установок и технологий регистрации частиц в трековых детекторах, переходя на новый уровень с появлением новых методов ускорения частии и более сложных типов детекторов. В экспериментах 60-х годов с пузырьковыми камерами события в них фиксировались на стереофотографиях и точки на треке вводились в компьютер оператором, ведущим курсор вручную вдоль траектории движения частицы, наблюдаемой под микроскопом. С ростом потока данных появились сканирующие устройства типа «Спиральный измеритель» [Gouache, 1972], в котором оператор ставил точку в вершину события, откуда шло сканирование снимка по спирали радиальной щелью с передачей в компьютер координат всех зарегистрированных точек. Трекинг на основе этой информации потребовал применения преобразований координат для «спрямления» треков [Hansroul et al., 1988] и использования модификации преобразования Хафа для их обнаружения [Никитин, Ососков, 1986]. Несмотря на преимущества, связанные с простотой реализации алгоритмов, основанных на координатных преобразованиях, ни они, ни преобразование Хафа не позволяют эффективно работать в трех измерениях, к тому же используемое в этих методах допущение об однородности магнитного поля выполняется далеко не всегда. С появлением электронных экспериментов данные измерений стали оцифровываться и сразу поступать прямо в компьютер. После многоэтапной фильтрации и процедур юстировки (алайнмента) наступало время трекинга, среди многих методов которого стоит отметить первые применения аппарата искусственных нейронных сетей, как многослойных персептронов, так и полносвязных нейросетей Хопфилда, предложенных Б. Денби и К. Петерсоном [Denby, 1987]. Несовершенство этих нейросетевых алгоритмов, резкое падение их эффективности с ростом зашумления трековых данных и множественность событий в коллайдерных экспериментах привели к появлению модификации нейро-

сетевых алгоритмов Хопфилда, названных эластичными нейросетями [Gyulassy, Harlander, 1991]. К сожалению, эффективное применение эластичных методов трекинга сильно затруднено необходимостью выбора начального приближения.

Однако с дальнейшим развитием коллайдеров и компьютерных технологий наиболее эффективным оказался ставший классическим метод трекинга, использующий фильтр Калмана [Frühwirth, 1987], поскольку он позволяет учитывать неоднородность магнитного поля, многократное рассеяние и потери энергии при прохождении частицы через среду детектора. Фильтр Калмана (Kalman Filter — KF) выполняет экстраполяцию начального состояния трека (как правило, первые три точки) во все последующие чувствительные элементы детектора. Для проверки гипотезы о том, что очередной хит вблизи экстраполированного трека принадлежит этому треку, используется критерий  $\chi^2$ . Далее процедура повторяется с учетом нового хита.

В случае если в событии много треков, лежащих в пространстве близко друг к другу, используется комбинаторная версия фильтра Калмана (Combinatorial Kalman Filter – CKF) [Mankel, 1997]. Для всех треков, для которых существует несколько возможных продолжений, CKF формирует ветвления, пытаясь экстраполировать исходные треки в нескольких направлениях. В конце процедуры ложные треки бракуются по критерию  $\chi^2$ .

Несмотря на успех КF и множество оптимизаций, позволяющих существенно сократить время на поиск начального состояния [Abt et al., 2002; Baranov et al., 2016], данный метод обладает целым рядом недостатков, связанных с его вычислительной сложностью и существенными трудностями при реализации параллельных вычислительных схем формирования треков. Особенно сильно недостатки KF проявляются в экспериментах с огромной множественностью треков. Это, например, проект NICA [Trubnikov et al., 2008] с тяжелыми ионами (см. рис. 2) или эксперименты с большой светимостью, как, например, на Большом адронном коллайдере (Large Hadron Collider — LHC) [Aad et al., 2008]. К 2024 году на LHC планируется проведение четвертого запуска, физики ожидают около 10 тысяч треков в событии, что непременно означает кризис классического трекинга.

В попытке снизить вычислительную сложность КF был предложен метод с применением клеточного автомата для задания начального состояния фильтрации Калмана [Mankel, 1997]. Однако программа, реализующая применение клеточного автомата, не документирована, что весьма затрудняет ее использование.



Рис. 2. Моделирование события в эксперименте BM@N

В противовес классическим алгоритмам трекинга существует семейство адаптивных методов реконструкции треков, включающее искусственные нейронные сети. Выше были отмечены применения нейросетей Хопфилда для трекинга, среди которых можно указать на успешное их использование для распознавания треков в эксперименте EXCHARM [Ососков и др., 1999], а также их эластичных модификаций, страдающих затруднениями с выбором начального приближения. Однако в настоящее время с интенсивным развитием параллельных вычислительных инфраструктур подлинную популярность приобретают методы глубокого обучения нейросетей ввиду своей способности к обнаружению скрытых нелинейных зависимостей в данных и возможности распараллеливания операций линейной алгебры, лежащих в основе этих методов.

Перспектива применения методов глубокого обучения нейросетей в задаче реконструкции треков порождает проблему создания эффективных алгоритмов «глубокого трекинга», превосходящих по скорости классические алгоритмы вроде KF.

Одной из попыток найти эффективный с точки зрения распознавания и быстрый в плане обработки метод реконструкции треков, использующий алгоритмы машинного обучения, стало соревнование TrackML [Amrouche et al., 2019]. Перед участниками соревнования стояла задача решить проблему трекинга для модельных данных, приближенных к реальным экспериментальным данным с LHC. Каждое событие содержало порядка  $10^5$  хитов для  $10^4$  частиц; хиты представлены трехмерной координатой в декартовой системе координат; для моделирования событий использовалось  $t\bar{t}$ -взаимодействие. Необходимо было реконструировать как минимум 90% истинных треков.

Победителем в соревновании TrackML стала команда «Top Quarks» [Amrouche et al., 2019], предложив решение, основанное на подборе хороших дублетов и триплетов хитов при помощи логистической регрессии с последующей экстраполяцией спиралью. Это решение, как и другие алгоритмы из верхушки турнирной таблицы, не используют полноценно методы глубокого обучения, а лишь являются модификациями уже существующих классических методов трекинга.

Цель данного исследования состояла в обобщении алгоритмов и программ реконструкции треков, разработанных на основе методов глубокого обучения в эксперименте ВМ@N ОИЯИ с фиксированной мишенью, для внутреннего поддетектора BESIII СGEM коллайдерного эксперимента BESIII ИФВЭ Пекин.

#### Предыдущие исследования

Одной из первых попыток авторов данной статьи применить глубокие нейронные сети к задаче реконструкции треков стала работа [Baranov et al., 2017]. Был предложен двухэтапный подход к реконструкции треков частиц.

На первом этапе использовался метод комбинаторного пространственного поиска трековкандидатов параллельно по двум координатным проекциям. На рис. 3 представлена схема процедуры формирования треков-кандидатов на первом этапе двухэтапного трекинга. В проекции на XZ-плоскость треки представляют собой почти прямые линии; соответственно, для поиска траекторий можно сформировать коридор, в пределах которого запустить процедуру бинарного поиска для экстраполяции трека. В проекции на плоскость YZ треки имеют форму полуокружности. Чтобы выбрать корректные треки в этой проекции, необходимо наложить ограничение в виде критерия гладкости кривой (критерий синусов) — все это выражается в функции допустимости [Baranov et al., 2017].

В результате выполнения первого этапа формируется список из треков-кандидатов, которые на втором этапе разделяются на две группы: истинные и ложные треки. Для разделения треков используется глубокая рекуррентная нейронная сеть-классификатор. Глубокие нейросети, называемые рекуррентными (Recurrent Neural Networks – RNN), содержат в себе обратные



Рис. 3. Процедура пространственного поиска треков в алгоритме двухэтапного трекинга

связи, позволяющие сохранять информацию. Вместо одного слоя RNN-сети содержат несколько взаимодействующих слоев, которые меняют информацию в ячейке с помощью фильтров или вентилей (gates), позволяющих пропускать, задерживать или стирать информацию на основании задаваемых условий. Такая нейронная сеть фактически обладает некоей памятью, чтобы, исходя не только из настоящего состояния объекта, но и из его прошлого, моделировать его последующее состояние. С помощью несколько упрощенного механизма вентилей для RNN созданы управляемые рекуррентные блоки (англ. Gated Recurrent Units, GRU). Подробно об архитектуре нейросети, реализующей двухэтапный трекинг, можно прочитать в [Baranov et al., 2017].

Несмотря на то что двухэтапный подход позволил достичь полноты в 97.5 % на тестовой выборке, в которой оба класса треков-кандидатов распределены равномерно, двухэтапный подход не отличается высокой скоростью работы. Узким местом в алгоритме является первый этап построения списка всех возможных треков-кандидатов.

Чтобы ускорить процедуру поиска треков-кандидатов, оба этапа были объединены в единую рекуррентную нейросеть, TrackNET [Baranov et al., 2019]. Модель TrackNET представляет собой модификацию сети-классификатора из метода двухэтапного трекинга. Основа этой модификации состоит в добавлении четырех специальных нейронов, выполняющих предсказание эллиптической области, в пределах которой необходимо искать продолжение трека-кандидата на следующей чувствительной плоскости (станции) детектора. Два нейрона предсказывают центр эллиптической области, а два других — размеры полуосей. Еще один, пятый выходной нейрон отвечает за классификацию треков-кандидатов на истинные и ложные. Таким образом, при объединении механизма экстраполяции и проверки гипотезы о том, что набор хитов на входе сети принадлежит истинному треку и соответствует критерию гладкости кривой, фактически воссоздается поведение алгоритма КF. В отличие от КF модель TrackNET можно отнести к классу адаптивных, поскольку физические параметры, описывающие трек, нейросеть аппроксимирует при помощи синаптических весов, которые «выучивает» в процессе тренировки. Кроме того, в процессе модификации двунаправленные слои GRU [Cho et al., 2014] заменены на однонаправленные и отключен механизм dropout. (Dropout [Srivastava et al., 2014] - алгоритм, используемый при обучении нейросети, который выполняет случайное прореживание нейронов. Каждый вес либо обнуляется с вероятностью p, либо умножается на 1/p с вероятностью 1 - p.)

Несмотря на многообещающие результаты работы модели, TrackNET обладает целым рядом недостатков. Функция ошибки нейросети (loss function), которая минимизируется при обучении, имеет три параметра, весьма критичных к неточностям их настройки. Кроме того, нейросеть тяжело обучается из-за сильной несбалансированности в данных — трудно соблюсти баланс между хорошей экстраполяцией и классификацией в условиях, когда число ложных треков на порядок больше, чем число истинных треков. Чтобы получить тренировочную выборку для модели TrackNET, необходимо воспользоваться алгоритмом поиска треков-кандидатов из первой стадии двухэтапного метода трекинга, что требует больших вычислительных затрат. Модель на самом деле не является сквозной единой моделью, а представляет собой три модели для случаев, когда на входе два хита и невозможно предсказать принадлежность истинному треку (классификационная часть не работает), когда число хитов на входе равно числу станций и не нужно проводить экстраполяцию (регрессионная часть не работает) и промежуточный случай, когда регрессионная часть работает вкупе с классификационной.

В процессе совершенствования модели TrackNET авторы данной статьи пришли к мысли о том, что процедура предсказания эллиптической области для поиска следующего хита трека уже содержит необходимую информацию о гладкости кривой. Эти выводы привели к созданию второй версии модели TrackNETv2 [Goncharov et al., 2019], описание которой приведено ниже в данной статье. TrackNETv2 является наиболее перспективным алгоритмом локального адаптивного трекинга, так как по сути своего функционирования эта модель выполняет роль обучаемого нейросетевого фильтра Калмана. В то же время модель не требует затрат на сложный предварительный этап поиска треков-кандидатов, необходимых для инициации вектора параметров фильтра Калмана. Еще одним преимуществом TrackNETv2 является пригодность к параллельным вычислениям, присущая нейросетевым алгоритмам, что крайне важно в условиях массовой обработки.

Модель TrackNETv2 относится к локальным методам трекинга, так как не использует данных о всей картине события и работает с каждым треком отдельно. У локальных подходов к трекингу есть очевидный недостаток: локальные методы не позволяют оценить глобальную картину события, увидеть зависимость между отдельными треками или группами треков. Также нет и прямой возможности отследить такие явления, как вторичные вершины, как это делается в глобальных методах трекинга, выполняющих распознавание треков среди истинных и ложных хитов сразу по всей картине события.

В работе [Farrell et al., 2018] исследователями из ЦЕРН в качестве глобального алгоритма трекинга был предложен метод графовых нейронных сетей (Graph Neural Networks — GNN) для реконструкции треков на LHC. Графовый подход предлагает рассматривать событие целиком в виде направленного графа (рис. 4), где хиты являются его узлами, а ребра связывают хиты между соседними детектирующими элементами, если они относятся к одному треку-кандидату. Задача нейросети заключается в прореживании этого графа путем классификации отдельных ребер. После завершения классификации остается только сгруппировать хиты по трекам, к которым они принадлежат. В работе [Farrell et al., 2018] получен практически безупречный результат распознавания в 99.4 % полноты по классификации ребер. Однако данный результат был получен только для треков с ограничением по углу разлета и, кроме того, без указания точности (recoil),



Рис. 4. Иллюстрация графового (GNN) представления. Узлами являются хиты (цветные кружки), а ребра, их соединяющие, образуют треки-кандидаты. Хиты соединяются, если они совместимы по некоторым критериям [Farrell et al., 2018]

то есть процента ложных треков, «просочившихся» в итоговый результат работы нейросети. В более поздней работе [Ju et al., 2020] те же авторы смогли снять большую часть ограничений и в результате оценили полноту трекинга в 95% при точности в 96% для диапазона энергий регистрируемых частиц от 100 MeV до 5 GeV.

Авторами данной статьи применение GNN было рассмотрено в рамках решения задачи трекинга на эксперименте BM@N [Shchavelev et al., 2019]. Основной проблемой при адаптации уже существующего для LHC подхода на случай с трековым детектором типа GEM (от англ. Gas Electron Multiplier) стало наличие большого количества так называемых фейков (ложных хитов), которые рождаются при срабатывании сразу нескольких стрипов чувствительной зоны детектора. При этом количество фейков ~  $O(n^2)$ , где n — количество сработавших стрипов. Более подробно об особенностях GEM-детекторов речь пойдет в следующей секции данной статьи на примере нового внутреннего трекера эксперимента BESIII [BESIII, 2014]. Непосредственно сама модель GNN для трекинга в условиях использования GEM-детектора описана далее в этой статье.

### Внутренний трекер эксперимента BESIII

Эксперимент BESIII [Ablikim et al., 2010] проводится в ИФВЭ АН КНР на коллайдере ВЕРСІІ. Задачей эксперимента является изучение свойств тау-лептонов, очарованных частиц и состояний чармония, которые образуются в электрон-позитронных столкновениях. В результате взаимодействия в каждом событии, как правило, образуется до 20 заряженных треков (в среднем от 2 до 8-10, в зависимости от типа события). Восстановление треков заряженных частиц важно для всех элементов физической программы эксперимента [Asner et al., 2010]. В одноименном детекторе BESIII они восстанавливаются в дрейфовой камере, которая состоит из двух частей: внешней и сменной внутренней. Внутренняя часть находится ближе к пучку и сильнее подвержена старению вследствие радиации, что приводит к постепенному ухудшению эффективности детектирования заряженных частиц. Проблема старения стала актуальной в последние годы работы эксперимента [Lavezzi et al., 2017]. Для замены внутренней части дрейфовой камеры был предложен трекер на основе трех цилиндрический GEM-детекторов (CGEM-IT) [BESIII, 2014]. Принципы работы GEM-детекторов подробно описаны в работе [Sauli, 2016]. При проходе заряженной частицы через рабочий объем детектора газовая смесь, которой он заполнен, ионизируется. Электроны под действием внешнего электрического поля дрейфуют к считывающей плоскости и проходят через три каскада газового усиления в GEM-пленках. В результате на считывающих электродах наводится электрический импульс, который усиливается и регистрируется. Считывание осуществляется двумя слоями микрополосок (стрипов), расположенных с шагом 650 микрон под углами в 30 и 45 градусов (в дальнейшем X- и V-слои, стрипы в X-слое параллельны оси пучков сталкивающихся частиц). Одна частица может формировать сигнал одновременно на нескольких расположенных рядом стрипах.

Для восстановления хитов на первом этапе трекинга восстанавливаются кластеры сработавших стрипов в каждом слое. Координата кластера может восстанавливаться из сработавших стрипов как средняя координата (бинарный режим), средневзвешенная с зарядом, а также может определяться по времени их срабатывания. На втором этапе хиты восстанавливаются комбинаторно по кластерам в X- и V-слоях. В данной работе используется кластеризация в бинарном режиме, в деталях описанная в работе [Denisenko, Ososkov, 2019].

Главным недостатком считывающей плоскости, спроектированной на основе стрипов, является неизбежное появление фиктивных пересечений (фейков), в случае если число треков больше одного. На рис. 6 представлен пример образования фейковых хитов в случае двух треков. В общем случае число ложных пересечений пропорционально квадрату числа треков.



Рис. 5. Цилиндрический трековый детектор CGEM-IT эксперимента BESIII, состоящий из трех детектирующих цилиндров (layers)



Рис. 6. Процесс формирования ложных хитов. Узкий стереоугол между стриповыми слоями позволяет убрать некоторые фейки за границы детектора

Другой отличительной особенностью эксперимента BESIII от эксперимента BM@N с фиксированной мишенью является то, что регистрация частиц производится в  $4\pi$ -геометрии. Встречные пучки сталкиваются между собой внутри цилиндрического детектора CGEM-IT, и регистрируются практически все заряженные частицы, родившиеся во взаимодействии. В эксперименте BM@N пучок сталкивается с веществом неподвижной мишени, расположенной перед детектором, и треки регистрируются только в узком конусе по направлению исходного движения пучка.

Пример Монте-Карло-события, смоделированного в детекторе CGEM-IT, показан на рис. 7. Несмотря на сравнительно малое число треков в событиях, в данных присутствует очень много шумов и фейков, что отчетливо видно на рис. 7. Наличие всего трех станций детектора усугубляет задачу отсеивания этих шумов, поскольку по трем точкам в пространстве нельзя с высокой точностью восстановить кривые треков, закручивающиеся по винтовой линии в магнитном поле детектора.

Методы TrackNETv2 и GNN имеют большой потенциал для реконструкции треков в эксперименте BESIII в условиях столь сильного зашумления, а также показывают перспективные результаты при тестировании на данных Монте-Карло-моделирования для эксперимента BM@N с аналогичным типом детектора [Goncharov et al., 2019; Shchavelev et al., 2019; Goncharov et al., 2020].



Рис. 7. Событие, полученное Монте-Карло-моделированием, в детекторе CGEM-IT эксперимента BESIII

## Особенности TrackNETv2 для цилиндрических трековых детекторов

Нейросетевая модель для трекинга, TrackNETv2, изначально была предложена для реконструкции треков в эксперименте BM@N с фиксированной мишенью [Goncharov et al., 2019].

Особенность работы TrackNETv2 заключается в том, что модель экстраполирует начальное состояние трека на последующие станции путем последовательного предсказания эллиптической области для поиска следующего хита трека. На рис. 8 представлена схема архитектуры модели TrackNETv2. На вход сети подается матрица, каждая строка которой представляет собой набор параметров хита трека-кандидата. Первая строка матрицы соответствует параметрам хита с первой станции, вторая строка — со второй станции и так далее в порядке следования станций. Матрица имеет фиксированный размер, число строк на единицу меньше числа станций. В случае когда на вход подается трек, число хитов которого меньше, чем фиксированный размер матрицы, оставшиеся входные значения заполняются нулями (на рисунке обозначено как padding).

На выходе сеть возвращает координаты центра и размеры полуосей эллипса, в котором на следующей станции детектора необходимо будет искать продолжение трека. Для предсказания координат центра используется линейная функция активации. Для предсказания полуосей



Рис. 8. Схема архитектуры модели TrackNETv2

используется функция активации softplus [Dugas et al., 2020], что обеспечивает строго положительный размер полуосей эллипса.

Функция ошибки нейросети TrackNETv2 описывается уравнением

$$J = \lambda_1 \cdot \sqrt{\frac{x - x'}{R_1} + \frac{y - y'}{R_2}} + \lambda_2 \cdot R_1 R_2,$$
(1)

где  $\lambda_1$ ,  $\lambda_2$  — веса для каждой части уравнения; x', y' — координаты центра эллипса, предсказанного сетью; x, y — координаты следующей точки реального сегмента трека;  $R_1$ ,  $R_2$  — полуоси предсказанного эллипса.

В процессе обучения TrackNETv2 пытается сместить центр предсказанного эллипса как можно ближе к координате следующего хита в треке, при этом сокращая размер полуосей как можно больше, но так, чтобы истинный хит трека обязательно лежал в пределах эллипса — за это отвечает первая часть функции.

В эксперименте BM@N все станции детектора идут последовательно, одна за другой, и для предсказания эллипса на каждой следующей станции в качестве входных параметров для нейросети используются не только исходные координаты хитов трек-кандидата, но и *z*-координата следующей станции относительно текущего хита.

Для того чтобы удобным образом представить данные CGEM-IT-детектора для TrackNETv2, избежав смены архитектуры нейросети и функции ошибки, целесообразно перейти в цилиндрические координаты с последующей нормализацией значений. Переход к цилиндрическим координатам происходит по следующей формуле:

$$Z = \sqrt{x^2 + y^2}, \quad X = \operatorname{atctg}\left(\frac{y}{x}\right), \quad Y = z, \tag{2}$$

где арктангенс вычисляется в диапазоне  $[-\pi, \pi]$  с учетом знаков *у* и *х*.

Новые координаты (X, Y, Z) используются как входные параметры нейросети TrackNETv2. Заметим, что выбор радиальной компоненты в качестве Z-координаты обусловлен тем, что детектирующие цилиндры CGEM-IT располагаются соосно, и, соответственно, пересечение трека с ними приводит к последовательному изменению координаты Z.

На рис. 9 изображен результат перехода от декартовых координат в цилиндрические.





Важно отметить, что координата X периодическая и измеряется от  $-\pi$  до  $\pi$ , что означает, что при достижении  $|X| \simeq \pi$  эта величина меняет знак, и трек как бы появится на следующей станции уже с другой стороны. Всего таких треков ~ 1.5% от общего числа смоделированных треков в событии. В процессе проведения тренировки сети такие треки не убирались из выборки. Задача сети — понять, что X-координата периодическая, и попытаться верно предсказать эллипс на следующей станции детектора.

Тестирование модели TrackNETv2, обученной на данных моделирования для BESIII, выявило существенный недостаток: нейросеть с высокой полнотой ~ 99 % находит истинные треки, но при этом захватывает множество ложных треков. На каждый истинный трек в событии нейросеть находит 10 фейковых треков. Визуальное сравнение реконструированных треков показало, что результаты работы модели логичны, но количество станций слишком мало, чтобы отсеять ложные треки прослеживанием.

Для решения вышеописанной проблемы TrackNETv2 явно не хватает информации о «вершине» события (о пространственном положении точки изначального столкновения частиц в коллайдере). Если модель обучить определять треки-кандидаты, отвечающие условию гладкости и берущие свое начало из первичной вершины, это улучшит классификацию на ложные и истинные треки. Такое обучение возможно, поскольку данные сами по себе в скрытой форме содержат информацию о вершине события, так как большинство треков, которые мы хотим восстановить, летят именно из нее.

Для этого нами разработана сеть-классификатор с архитектурой, графическое представление которой приведено на рис. 10. Используется два входных сигнала: с выхода второго GRUслоя TrackNETv2 (GRU) и набор координат хита, определенного как завершение трека (Last point). Каждый сигнал подается на полносвязный слой (FC — Fully Connected). Затем полученные векторы конкатенируются (Concat), и результирующий сигнал идет на вход следующего полносвязного слоя (FC). Во всех вышеописанных слоях используется функция активации



Рис. 10. Схема классификатора для TrackNETv2

ReLU [Xu et al., 2015]. Последний FC-слой нейросети, отвечающий за классификацию, имеет функцию активации Softmax [Goodfellow et al., 2016], и результат работы классификатора (Output) можно трактовать как вероятность, что трек-кандидат является истинным треком.

Модель-классификатор обучается на данных, полученных в результате применения уже натренированной модели TrackNETv2. Все треки-кандидаты, которые TrackNETv2 реконструировала как истинные треки, помечаются метками: 0 — для ложных треков, 1 — для истинных. Данные для обучения сохраняются в формате триплетов, где первые два элемента — это эмбеддинги TrackNETv2 и координаты последнего хита каждого трека-кандидата, а последний элемент триплета — метка принадлежности к классу истинных или ложных треков.

При таком подходе к формированию тренировочного набора данных для классификации истинных треков очень мало, и необходимо проводить расширение набора с помощью дублирования объектов минорного класса, иначе модель не сможет выявить отличительные признаки истинных треков. В нашем случае для каждого истинного трека было создано пятнадцать дубликатов и они помещены в тренировочную выборку. Параметры расширенного обучающего набора приведены в таблице 1.

Начальное число хитов	1 385 991
Число хитов после дублирования	1 979 988
Начальная доля истинных треков	1.6 %
Доля истинных треков после дублирования	20%

Таблица 1. Характеристики обучающего набора для классификатора

Модель TrackNETv2 с последующим применением классификатора получила название TrackNETv2.1. Результаты ее тестирования приведены ниже в данной статье.

## **GNN с реверсивным ориентированным графом**

Использование GNN для трекинга данных с GEM-детектора в эксперименте BM@N показало перспективные результаты [Shchavelev et al., 2019]. В GNN все хиты события, находящиеся на смежных станциях, соединяются между собой ребрами, образуя направленный граф, где хиты являются узлами этого графа. Событие эксперимента BM@N, представленное в виде такого направленного графа для входного блока GNN, показано на рис. 11.

#### Схема работы Graph Neural Network

Представим событие как граф, в котором узлы — это хиты. Ребрами соединим все узлы между детектирующими слоями. Внутри одного слоя узлы не соединяются. Граф выражается в виде четырех матриц.

- *X* матрица параметров узлов графа размером *N*×*M*, где *N* количество узлов, а *M* количество параметров. Координаты хита используются как параметры узла, поэтому *M* = 3.
- *Ri* матрица ребер, которые заканчиваются в соответствующих узлах графа, размером  $N \times E$  (*E* количество ребер). В этой матрице *Ri*[*i*, *j*] = 1, если ребро с индексом *j* входит в узел с индексом *i*, и ноль в противном случае.
- *Ro* матрица ребер, которые исходят из узлов. Обладает теми же свойствами, что и *Ri*, но описывает исходящие ребра.
- Y вектор размером E. Y[j] = 1, если ребро с индексом j принадлежит реальному треку, и 0 в противном случае.



Рис. 11. Событие эксперимента BM@N, представленное в виде графа. Зеленые узлы соответствуют истинным хитам, черные узлы — фейковым. Оранжевые ребра выделяют сегменты реальных треков, все остальные ребра черные

GNN состоит из трех основных частей: сетей Input, Node и Edge. Сеть Input — многослойный персептрон (MLP) с активацией гиперболическим тангенсом (*th*). На вход сети Input подается матрица X. Задача сети состоит в том, чтобы, опираясь на пространственную информацию хитов, выделить среди нее признаки, которыми в дальнейшем будут оперировать сети Edge и Node. Выход сети Input подается на итерации «Edge–Node»-сетей (рис. 12).



Рис. 12. Схема работы GNN (рисунок взят из статьи [Farrell et al., 2018]). Граф события поступает на вход Input Network и проходит цепочки Edge–Node-итераций

Сеть Edge представляет собой двухслойную полносвязную нейросеть, задача которой состоит в вычислении весов ребер графа на основании признаков, ассоциированных с ребром узлов. Между слоями сети функция активации — гиперболический тангенс. На выходном слое используется сигмоида, которая определяет, является ли ребро графа сегментом истинного трека.

Сеть Node устроена аналогичным образом, но имеет другую задачу. Эта сеть пересчитывает признаки узлов графа, привлекая признаки соседних узлов и используя веса всех входящих в узел и выходящих из него ребер, полученные предыдущей сетью Edge. Эти сети могут быть расположены друг за другом в виде итераций. Таким образом, изначальная информация в каждом узле графа с каждой итерацией распространяется по ребрам графа, комбинируясь с информацией других узлов. Количество таких итераций является одним из глобальных параметров (гиперпараметров) данной модели. Большое число итераций увеличивает количество алгебраических вычислений, что влияет на производительность сети. Это необходимо учитывать при выборе количества итераций.

#### Применение Graph Neural Network для трекинга эксперимента BESIII

Авторы работы [Goncharov et al., 2020] показали, что прямое использование модели GNN [Farrell et al., 2018], разработанной для экспериментов на детекторах LHC, не позволяет достичь необходимой точности для GEM-детектора из-за наличия большого количества фейков. Поэтому в работе [Goncharov et al., 2020] был предложен метод препроцессинга графа на основе минимального ветвящегося дерева (Minimal Branching Tree — MBT), позволяющий снизить количество фейковых сегментов в конечном графе. В данной работе MBT заменен на алгоритм создания реверсивного ориентированного графа (реверсного орграфа), ребра которого являются узлами исходного графа, а узлы реверсного графа являются ребрами исходного [Shchavelev et al., 2019]. Такой подход позволил сократить число фейковых ребер в 10 раз, сохранив при этом все истинные ребра.

Перед тем как начать строить граф события, координаты всех хитов переводят в цилиндрические. Первым шагом алгоритма является собственно построение исходного полносвязного направленного графа *G* путем соединения первоначальных хитов на соседних станциях ребрами, как на рис. 11. Узлами такого графа будут хиты события, а ребра есть попарные соединения хитов между двумя последовательными станциями.

После создания графа G выполняется переход к реверсному орграфу  $G_r$ . Узлы графа G являются ребрами  $G_r$ , а ребра G — узлами  $G_r$ . Таким образом, каждое ребро реверсного орграфа  $G_r$  есть «путь», соединяющий 2 ребра графа G.

Ребра  $G_r$  фильтруются на основе статистического анализа исходных данных. Для каждого ребра графа  $G_r$  вычисляется вес по следующей формуле:

$$w_i = \sqrt{(dr_{i+1} - dr_i)^2 + (d\varphi_{i+1} - d\varphi_i)^2 + (dz_{i+1} - dz_i)^2},$$
(3)

где  $dr_j = r_j - r_{j-1}$ ,  $d\varphi_j = \varphi_j - \varphi_{j-1}$ ,  $dz_j = z_j - z_{j-1}$  для i = 1,  $j \in \{1, 2\}$ , и  $r_j$ ,  $\varphi_j$  и  $z_j$  — координаты хитов в цилиндрической системе координат. В обучающий набор данных попадают только те ребра, вес которых w < 0.073. Данное значение было выбрано в ходе статистического анализа исходного набора данных таким образом, чтобы сохранить максимальное число истинных ребер, минимизируя при этом количество фейковых ребер после фильтрации.

Полученные после всех описанных шагов обработки данные используются для обучения и тестирования GNN. При обучении сеть получает на вход реверсный орграф с метками истинных ребер. В результате после обучения нейросеть RDGraphNet (Reversed Directed Graph Neural Network) получает на выходе для каждого ребра значение  $x \in [0, 1]$ . Истинным ребром трека считаются те ребра, для которых x больше некоторого заданного порога. В данной работе порог фиксирован и равен 0.5. Такое значение порога позволяет одновременно добиться высоких значений полноты и точности. Его изменение приведет к улучшению одной метрики за счет ухудшения другой. Таким образом, с помощью порога можно управлять результатом работы алгоритма: можно не потерять истинные треки ценой захвата лишних, ложных треков или же, наоборот, отсеять множество ложно-положительных, но при этом пропустить какое-то количество истинных треков.

Гиперпараметры RDGraphNet для трекинга в детекторе CGEM эксперимента BESIII, выбранные для обучения, приведены в таблице 2.

Стоит отметить, что большая гибкость модели GNN и метода с применением реверсного орграфа позволяет рассчитывать на их успешное применение и в более сложных трековых детекторах, с большим числом детектирующих плоскостей.

Глобальный параметр	Значение
Количество итераций Node-Edge	2
Размерность пространства признаков	5 (координаты $r_j$ , $r_{j-1}$ , $\varphi_j$ , $\varphi_{j-1}$ и порядковый номер станции)
Количество нейронов на скрытом слое MLP	96

Таблица 2. Глобальные параметры (гиперпараметры) RDGraphNet

## Тестирование на Монте-Карло-событиях

Представленные методы были применены для поиска заряженных треков в детекторе CGEM-IT эксперимента BESIII при энергии столкновений в 3.686 ГэВ. В работе использовались результаты Монте-Карло-моделирования событий электрон-позитронной аннигиляции с образованием и распадом резонанса  $\psi(3686)$ . Для каждого события результаты Монте-Карло-моделирования представляют из себя хиты, которые либо ассоциированы с истинными треками заряженных частиц, либо отмечены как фейки. Процедура моделирования хитов подробно описана в работе [Denisenko, Ososkov, 2019]. Нами использовались события, в которых смоделированный трек всегда содержит три хита и между собой треки не пересекаются. Набор данных для обучения и валидации состоял из 200 000 смоделированных событий.

В представленном наборе данных относительное соотношение фейковых и истинных хитов составляет ~ 1:3 (на один реальный хит приходится более 3 фейков), что вносит дополнительную сложность нахождения истинных треков.

В таблице 3 содержатся характеристики набора данных для RDGraphNet и TrackNETv2.1. Заметим, что при обучении моделей нейронных сетей использовалась только часть описанного выше набора данных.

Характеристики	RDGraphNet	TrackNETv2.1		
Обучение				
Всего событий	19 000	134 997		
Всего треков	74 990	686246		
Всего хитов (включая истинные и ложные)	969 929	7 137 867		
Доля фейковых хитов	77%	77 %		
Доля истинных хитов	23 %	23 %		
Валидация				
Всего событий	10 000	57716		
Всего треков	43 649	293 634		
Всего хитов (включая истинные и ложные)	565 257	3 0 5 9 0 8 6		
Доля фейковых хитов	77%	77 %		
Доля истинных хитов	23 %	23 %		

Таблица 3. Характеристики наборов данных, использованных для обучения и валидации процесса обучения нейросетей

В ходе обучения моделей нейронных сетей были использованы параметры, которые представлены в таблице 4.

	RDGraphNet	TrackNETv2.1	
Оптимизатор	Adam [Kingma, Ba, 2014]		
Функция ошибки	torch.nn.functional.binary_cross_entropy		
Learning rate	0.0007	0.001	
Перевзвешивание ложных	$W_{\text{false}} = w_{\text{false}} \cdot 0.555$	$W_{\text{false}} = w_{\text{false}} \cdot 0.625$	
и истинных меток	$W_{\rm true} = w_{\rm true} \cdot 3$	$W_{\rm true} = w_{\rm true} \cdot 2.5$	

		U U		~
Таблица 4 Характеристики	гиперпараметров	неиросетеи	при их	ооучении
ruomigu n'rupukrepnermin	1 miepmapane ipob	mempoceren	mpn nn	ooy remm

В таблице 5 представлены результаты тестирования обученных моделей TrackNETv2. В качестве метрик оценки качества используются два показателя:

- полнота (recall): N<sup>rec</sup><sub>true</sub>/N<sub>MC</sub>, где N<sup>rec</sup><sub>true</sub> число верно реконструированных треков (истинных треков) и N<sub>MC</sub> число смоделированных (Монте-Карло) треков;
- точность (precision):  $N_{\text{true}}^{\text{rec}}/N^{\text{rec}}$ , где  $N^{\text{rec}}$  полное число реконструированных треков (включая фейковые треки).

Трек всегда состоит из трех хитов и считается верно реконструированным, если все найденные хиты происходят от одного Монте-Карло-трека. С точки зрения поиска треков понятие полноты совпадает с понятием эффективности распознавания треков, а точность — с чистотой распознавания треков. Вероятность, что найденный трек является фейковым, дается выражением 1 - p, где p — точность (precision).

	D		~	~		
	Результаты	тестирования	ооученных	молепеи на	тестовых	ланных
таолица с.	105,51010101	reempoballin	ooy iominin	поделен на	reerobbin	данным

	RDGraphNet	TrackNETv2.1
Полнота (recall)	0.9804	0.9414
Точность (precision)	0.8650	0.7711

Скорость обработки для двух моделей — RDGraphNet и TrackNETv2.1 — измерялась на разных устройствах, с использованием разных библиотек глубокого обучения и разной степенью оптимизации. Алгоритм модели RDGraphNet лучше распараллелен, а также переведен на язык C++, в то время как модель TrackNETv2.1 еще находится в стадии разработки. Поэтому на данной стадии некорректно сравнивать результаты двух моделей между собой, и мы приводим их только для полноты изложения. Результаты представлены в таблице 6.

Скорость обработки модели RDGraphNet в Python 3.7 на CPU измерялась на процессоре Intel Core i7-7700K @ 4.20GHz. Для C++ версии измерения проводились на процессоре Intel Xeon E3-1230 @ 3.50GHz. Для измерений на GPU использовалась видеокарта NVIDIA GeForce GTX 1080 Ti.

Таблица 6. Скорость обработки (количество событий, обрабатываемых за одну секунду) для каждой из моделей. Условия тестирования описаны в тексте статьи

	RDGraphNet	TrackNETv2.1
Preprocessing (CPU, Python, single-threaded)	19.97	1.5
Preprocessing (CPU, C++, multithreaded)	570	—
Inference (GPU)	260	85

Скорость обработки для модели TrackNETv2.1 измерялась на процессоре Intel Xeon CPU @ 2.20 GHz и на GPU Nvidia Tesla T4.

## Заключение

- Нейросетевая модель RDGraphNet, разработанная на основе реверсного орграфа для эксперимента BM@N с фиксированной мишенью, успешно адаптирована для цилиндрического детектора CGEM коллайдерного эксперимента BESIII. Обучение на модельных данных и последующее тестирование показали обнадеживающие результаты, полнота (recall): 98% и точность (precision) 86%.
- 2. Нейросетевая «локальная» модель TrackNETv2, разработанная для эксперимента BM@N, успешно адаптирована для цилиндрического детектора CGEM (BESIII). Полученная программа TrackNETv2.1 отлажена, обучена на модельных данных и протестирована. Однако наличие в CGEM-детекторе всего трех детектирующих слоев и большого количества ложных хитов привело к тому, что при достигнутой высокой полноте трекинга в 99 % точность составила всего несколько процентов. Модель находит много ложных треков. Это потребовало разработки дополнительных слоев нейросети, реализующих нейроклассификатор для отсева ложных треков, применение которого в отладочном режиме уже дало повышение точности до 77 %.
- 3. Успешно выполнена конвертация программы RDGraphNet, написанной на интерпретируемом языке Python, в C++. Это дало ускорение процесса трекинга для внутреннего детектора BESIII в 30 раз при той же эффективности. Также выполняется параллельная реализация модели TrackNETv2.1 на C++. Предполагается оптимизация программ с точки зрения потребления памяти и скорости работы при применении графических процессоров.
- 4. Несмотря на то что подход GNN демонстрирует многообещающие результаты, следует отметить, что он явно избыточен для детектора с всего тремя детектирующими слоями. GNN может стать гораздо более эффективным для трековых детекторов с более сложной структурой, как, например, детекторы MPD [Golovatyuk et al., 2016] и SPD [Savin et al., 2016] в проекте NICA.

## Список литературы (References)

- Никитин В. А., Ососков Г. А. Автоматизация измерений и обработки данных физического эксперимента (монография). М.: Изд. МГУ, 1986. 185 с. Nikitin V. A., Ososkov G. A. Avtomatizatsiya izmerenii i obrabotki dannykh fizicheskogo eksperimenta [Automation of measurements and data processing of a physical experiment]. — Moscow: Izd. MGU, 1986. — 185 p. (in Russian).
- Ососков Г. А. и др. Использование нейронных сетей для улучшения интерпретации эксперимента EXCHARM // Математическое моделирование. 1999. Т. 11, № 10. С. 116–126. Ososkov G. A. et al. Ispol'zovanie neironnykh setei dlya uluchsheniya interpretatsii eksperimenta EXCHARM [Using neural networks to improve interpretation of the EXCHARM experiment]. — Mathematical Models and Computer Simulations. — 1999. — Vol. 11, No. 10. — Р. 116–126 (in Russian).
- *Aad G. et al.* The ATLAS experiment at the CERN large hadron collider // Jinst. 2008. Vol. 3. P. S08003.
- *Ablikim M. et al. [BESIII Collaboration]* Design and Construction of the BESIII Detector // Nucl. Instrum. Meth. A 614. 2010. P. 345–399.
- *Abt I., Kisel I., Masciocchi S., Emelyanov D.* CATS: a cellular automaton for tracking in silicon for the HERA-B vertex detector // Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment. 2002. Vol. 489, Nos. 1–3. P. 389–405.

- *Amrouche S. et al.* The Tracking Machine Learning challenge: Accuracy phase // arXiv:1904.06778 [hep-ex]. 2019.
- Asner D. M. et al. Physics at BES-III // Int. J. Mod. Phys. A 24. 2009. S1-794.
- Baranov D., Merts S., Ososkov G., Rogachevsky O. New Algorithm of Seed Finding for Track Reconstruction // EPJ Web of Conferences. – EDP Sciences, 2016. – Vol. 108. – P. 02012.
- Baranov D., Mitsyn S., Ososkov G., Goncharov P., Tsytrinov A. Novel approach to the particle track reconstruction based on deep learning methods // Selected Papers of the 26th International Symposium on Nuclear Electronics and Computing (NEC 2017), Budva, Montenegro, September 25–29, 2017. – CEUR Proceedings. – Vol. 2023. – P. 37–45.
- Baranov D., Ososkov G., Goncharov P., Tsytrinov A. Catch and Prolong: recurrent neural network for seeking track-candidates // The XXII International Scientific Conference of Young Scientists and Specialists (AYSS-2018). – EPJ Web of Conferences. – EDP Sciences, 2019. – Vol. 201. – P. 05001.
- BESIII Collaboration, BESIII Cylindrical GEM Inner Tracker CDR // Internal note, 2014.
- Cho K., Van Merriënboer B., Gulcehre C., Bahdanau D., Bougares F., Schwenk H., Bengio Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation // arXiv:1406.1078 [cs.SL]. – 2014.
- *Denby B. H.* Neural networks and cellular automata in experimental high energy physics // Comput. Phys. Commun. 1988. Vol. 49. P. 429–448.
- Denisenko I., Ososkov G. Primary processing of hits in cylindrical GEM tracker at the BESIII experiment // AIP Conf. Proc. 2163. 2019. P. 030002.
- Dugas C., Bengio Y., Belisle F., Nadeau C., Garcia R. Incorporating second-order functional knowledge for better option pricing // Advances in neural information processing systems. – 2001. – P. 472–478.
- Farrell S., Calafiura P., Mudigonda M., Prabhat, Anderson D., Vlimant J. R., Zheng S., Bendavid J., Spiropulu M., Cerati G., Gray L., Kowalkowski J., Spentzouris P., Tsaris A. Novel deep learning methods for track reconstruction // arXiv:1810.06111 [hep-ex]. – 2018.
- Frühwirth R. Application of Kalman filtering to track and vertex fitting // Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment. – 1987. – Vol. 262, Nos. 2–3. – P. 444–450.
- Golovatyuk V., Kekelidze V., Kolesnikov V., Rogachevsky O., Sorin A. The Multi-Purpose Detector (MPD) of the collider experiment // Eur. Phys. J. A52 (8). 2016. P. 212.
- Goncharov P., Ososkov G., Baranov D. Particle track reconstruction with the TrackNETv2 // AIP Conference Proceedings. AIP Publishing LLC, 2019. Vol. 2163, No. 1. P. 040003.
- Goncharov P., Shchavelev E., Ososkov G., Baranov D. BM@N Tracking with Novel Deep Learning Methods // EPJ Web of Conferences. EDP Sciences, 2020. Vol. 226. P. 03009.
- Goodfellow I., Bengio Y., Courville A. Softmax Units for Multinoulli Output Distributions // Deep Learning. MIT Press. 2016. P. 180–184.
- Gouache J.-C. Status report of the CERNLSD system // European Spiral Reader Symposium. 1972. P. 21–32. DOI: 10.5170/CERN-1972-016.21
- *Gyulassy M., Harlander M.* Elastic tracking and neural network algorithms for complex pattern recognition // Comput. Phys. Commun. 1991. Vol. 66. P. 31–46.

- *Hansroul M., Savard D., Jeremie H.* Fast circle fit with the conformal mapping method // Nucl. Instrum. Meth. A. 1988. Vol. 270. P. 498–501. CERN-DD-88-20.
- *Hopfield J. J.* Neural networks and physical systems with emergent collective computational abilities // Proc. Nat. Acad. Sci. 1982. Vol. 79. P. 2554–2558.
- Ju X. et al. Graph neural networks for particle reconstruction in high energy physics detectors // arXiv:2003.11603 [physics.ins-det]. 2020.
- Kingma D. P., Ba J. L. A method for stochastic optimization // arXiv:1412.6980 [cs.LG]. 2014.
- Lavezzi L. et al. The Cylindrical GEM Inner Tracker of the BESIII experiment: prototype test beam results // Journal of Instrumentation. 2017. Vol. 12, No. 07. P. C07038.
- *Mankel R.* A Concurrent track evolution algorithm for pattern recognition in the HERA-B main tracking system // Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment. 1997. Vol. 395, No. 2. P. 169–184.
- Peterson C. Track Finding With Neural Networks // Nucl. Instrum. Meth. A. 1989. Vol. 279. P. 537.
- Savin I., Efremov A., Pshekhonov D., Kovalenko A., Teryaev O., Shevchenko O., Nagajcev A., Guskov A., Kukhtin V., Toplilin N. Spin physics experiments at NICA-SPD with polarized proton and deuteron beams // Eur. Phys. J. A. – 2016. – Vol. 52 (8). – P. 215.
- Sauli F. The gas electron multiplier (GEM): Operating principles and applications // Nucl. Instrum. Meth. A. 2016. Vol. 805. P. 2.
- Shchavelev E., Goncharov P., Ososkov G., Baranov D. Tracking for BM@N GEM Detector on the Basis of Graph Neural Network // Proceedings of the 27th Symposium on Nuclear Electronics and Computing (NEC 2019), Budva, Montenegro. 2019. Vol. 2507. P. 280–284.
- Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting // Journal of machine learning research. – 2014. – Vol. 15, No. 1. – P. 1929–1958.
- *Trubnikov G. et al.* Project of the Nuclotron-based ion collider facility (NICA) at JINR // Proceedings of EPAC08, Genoa, Italy. 2008.
- *Xu B., Wang N., Chen T., Li M.* Empirical evaluation of rectified activations in convolutional network // arXiv:1505.00853 [cs.LG]. 2015.