МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Ярославский государственный университет им. П.Г. Демидова» Кафедра математического анализа

Сдано на кафедру

«5» июня 2025 г. Заведующий кафедрой

д. ф.-м. н.

Невский М.В.

Выпускная квалификационная работа

## Восстановление треков заряженных частиц по данным электромагнитного калориметра

направление подготовки

01.03.02 Прикладная математика и информатика

Научный руководитель

Алексеев В.В.

«5» июня 2025 г.

Студент группы ПМИ-43БО

Нехаенко П.А.

«5» июня 2025 г.

Ярославль, 2025

# Реферат

Дипломная работа изложена на 31 странице, включает введение, четыре гла- вы, заключение и приложение. В тексте приведено 13 рисунков, 2 таблицы и 9 ис- пользованных источников.

**Ключевые слова:** трековая реконструкция, стриповый калориметр, анти- протоны, эксперимент PAMELA, преобразование Хафа, метод проекционной релак- сации, оптимизация.

Работа посвящена задаче восстановления трёхмерной топологии взаимодей- ствия заряжённых частиц в кремниево-вольфрамовом стриповом калориметре экс- перимента PAMELA. Построена параметрическая модель прямолинейных и лома- ных траекторий, разработан алгоритм их инициализации с помощью преобразова- ния Хафа и реализована глобальная оптимизация, совмещающая геометрию треков с распределением энерговыделений. Для последующего определения энергетического профиля сформулирована выпуклая квадратичная задача, решаемая методом про- екционной релаксации.

На симулированных данных Geant4 комбинированный подход повысил метри- ку IoU с 0.45 до 0.53, Dice — с 0.62 до 0.69 и снизил Energy-WEMD на 34 %. Для ре- альных событий PAMELA достигнуто среднее покрытие IoU 24.7 % при уменьшении проекционной невязки на 33 %. Ограничения метода связаны с ростом вычислитель- ных затрат при числе треков свыше восьми и чувствительностью к несовместимым проекциям. Намечены расширения: регуляризация TV, стохастический поиск MCMC и применение 3D U-Net для локализации области интереса.

# Содержание

[Введение](#_bookmark1) 3

[Эксперимент PAMELA](#_bookmark2) 3

[Калориметр аппарата PAMELA](#_bookmark3) 3

1. [Постановка задачи](#_bookmark6) 6
   1. [Исходные данные](#_bookmark7) 6
   2. [Восстановление траекторий частиц](#_bookmark8) 6
   3. [Восстановление значений энерговыделений](#_bookmark10) 7
2. [Алгоритм восстановления траекторий](#_bookmark11) 9
   1. [Идентификация антипротонов в эксперименте PAMELA](#_bookmark12) 9
   2. [Восстановление трека антипротонов в калориметре](#_bookmark13) 9
      1. [Преобразование Хафа](#_bookmark14) 10
      2. [Методика восстановления треков частиц и античастиц на основе](#_bookmark17) [преобразования Хафа](#_bookmark17) 10
   3. [Обобщённая схема алгоритма реконструкции](#_bookmark18) 14
3. [Алгоритм восстановления энергий вдоль трека](#_bookmark19) 16
   1. [Формулировка задачи восстановления энергетического распределения](#_bookmark20) . 16
   2. [Метод проекционной релаксации и численное решение](#_bookmark21) 17
4. [Результаты](#_bookmark22) 18
   1. [Результаты на модельных данных (MC)](#_bookmark23) 18
   2. [Результаты на данных PAMELA](#_bookmark26) 19

[Заключение](#_bookmark30) 21

[Приложение](#_bookmark31) 22

[Список литературы](#_bookmark0) 31

# Введение

Антипротоны — это частицы антиматерии, которые в малом количестве при- сутствуют в галактических космических лучах (ГКЛ). Считается, что основным ме- ханизмом их генерации в Галактике являются взаимодействия высокоэнергичных космических лучей с межзвездным веществом, известные как механизм вторично- го рождения антипротонов [[1].](#_bookmark0) Эксперименты по регистрации антипротонов в кос- мических лучах проводятся с 1970-х годов, начиная с аэростатов и продолжая на искусственных спутниках Земли. Наиболее современными экспериментами являют- ся PAMELA [[2]](#_bookmark0) и AMS-02 [[3].](#_bookmark0) В данной работе используются данные эксперимента PAMELA, а также данные, полученные в результате моделирования электромагнит- ного калориметра, который является составной частью аппарата PAMELA, в среде моделирования Geant4 [[4].](#_bookmark0)

Один из способов регистрации антипротонов низких энергий (до 400 МэВ) за- ключается в исследовании топологии аннигиляции частицы в позиционно-чувствительном стриповом калориметре [[5].](#_bookmark0) Сложность заключается в том, что стриповый калори- метр предоставляет возможность измерять энерговыделения в двух проекциях, но

не дает объемную картину взаимодействия частицы с веществом калориметра.

## Эксперимент PAMELA

Аппарат **PAMELA** (Payload for Antimatter–Matter Exploration and Light– nuclei Astrophysics, рис. [1)](#_bookmark3) предназначен для исследования космического излучения с акцентом на компоненте антиматерии [[2].](#_bookmark0) Данный аппарат был установлен в гер- моблоке спутника *«Ресурс-ДК1»*, и осуществлял работу в 2006–2016 гг. Одной из важных составляющих аппарата является электромагнитный вольфрам-кремниевый калориметр, данные которого анализируются в дипломной работе.



Рис. 1: Компоновка спутникового комплекса PAMELA.

## Калориметр аппарата PAMELA

Калориметр аппарата PAMELA (рис. [2)](#_bookmark4) состоит из 44 однослойных кремни- евых сенсорных плоскостей, чередующихся с 22 вольфрамовыми плоскостями (тол- щина каждого слоя составляет 0,26 см). Кремниевые плоскости состоят из 3 × 3 = 9 кремниевых детекторов, каждый из которых разделён на 32 считывающих стрипа (полосы) с шагом 2,4 мм [[5].](#_bookmark0)

Большинство частиц, попадающих в калориметр, при взаимодействии с его веществом инициируют возникновение вторичных частиц, передавая им часть сво- ей энергии. Взаимодействие может быть электромагнитным, либо сильным (адрон- ным), в котором происходит взаимодействие частицы с ядром вещества-поглотителя (в данном случае, вольфрама). Картину, при которой происходит каскад взаимодей- ствий: вторичные частицы порождают новые, и т.д., называют *ливнем* (соответствен- но, электромагнитным или адронным).

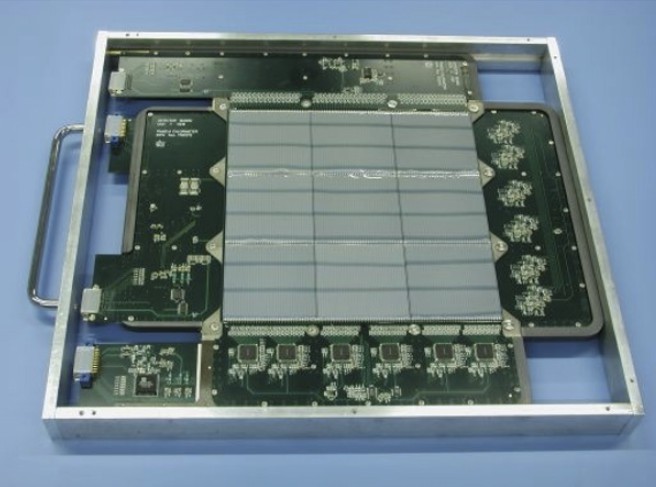


Рис. 2: Электромагнитный калориметр PAMELA.

При этом, для антипротонов низких (до 400 МэВ) энергий взаимодействие с веществом калориметра характеризуется типичной картиной аннигиляции: в точке взаимодействия порождаются 4-5 *π*-мезонов, причём направления разлёта порож- дённых частиц равновероятны. Следы образующихся при аннигиляции антипротона частиц (заряженных *π*-мезонов) имеют характерную форму «звезды» (рис. [3).](#_bookmark5) Та- кая топология взаимодействия отлична от типичной картины развития ливня, при котором порождённые частицы более вероятно полетят в направлении, близком к направлению первичной частицы. Это позволяет в задаче разделения электронов и антипротонов использовать дескрипторы, связанные с топологией взаимодействия. Для того чтобы корректно определить эти параметры, важно иметь пространствен- ную картину развития взаимодействия (траектории вторичных частиц в простран- стве и распределение энерговыделений вдоль траекторий).

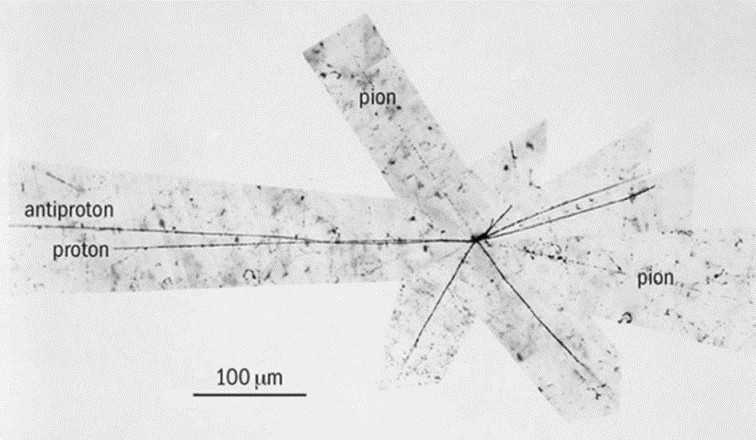


Рис. 3: Аннигиляция антипротона, наблюдавшаяся на ускорителе Беватроне в Ка- лифорнийском университете в Беркли в 1955 году с помощью фотоэмульсии. Анти- протон входит слева. Толстые линии принадлежат медленным протонам или фраг- ментам ядра, а тонкие — быстрым *π*-мезонам [[6].](#_bookmark0)

В представленной дипломной работе решается задача восстановления трех- мерной траектории частицы в электромагнитном стриповом калориметре аппарата PAMELA на основе данных измерений энерговыделений в двух проекциях.

Работа состоит из **введения**, **четырех глав**, **заключения** и **приложения**. Во **введении** описывается эксперимент PAMELA, в частности, электромаг- нитный калориметр аппарата PAMELA. Даётся описание механизма взаимодействия частиц с веществом калориметра; приводится общая характеристика работы, обос-

новывается важность и актуальность поставленной задачи.

В **первой главе** определяется набор исходных данных и формулируется за- дача реконструкции трека.

Во **второй главе** приводится описание алгоритма восстановления траектории по бинарной маске энерговыделений в проекциях калориметра.

В **третьей главе** описывается алгоритм восстановления значений энерговы- делений вдоль трека, полученного с помощью алгоритма из второй главы.

В **четвертой главе** приводятся результаты применения алгоритмов к дан- ным моделирования и экспериментальным данным, а также оценка точности работы алгоритмов.

В **заключении** подводятся итоги и намечаются дальнейшие шаги исследова-

ния.

В **приложении** представлен код реализации алгоритмов на языке Python.

**Список литературы** включает ??? наименований.

# Постановка задачи

## Исходные данные

Каждое событие прохождения заряженной частицы через калориметр харак- теризуется двумя матрицами отклика прибора с неотрицательными значениями

XZ ∈ R96×22*,* YZ ∈ R96×22*.* (1)

Строка матрицы с номером *z* соответствует набору энерговыделений, считанных в вольфрамовом слое с номером *z* кремниевым детектором, стрипы которого ориенти- рованы параллельно оси X (для матрицы YZ), либо оси Y (для матрицы XZ).

Для данных моделирования в среде Geant4 для каждого события известна следующая информация о каждом событии.

* Точка влёта первичной частицы (*x*start*, y*start).
* Углы влёта (зенитный и азимутальный) первичной частицы (*θ*start*, φ*start).
* Координаты пересечения каждой плоскости первичной частицей, энерговыде- ления в данных точках.
* Точка взаимодействия первичной частицы (*x*int*, y*int*, z*int).
* Количество порождённых частиц *N* .
* Типы вторичных частиц и углы (*θ*i*, φ*i), *i* = 1*, . . . , N* , задающие направления их разлёта.

Некоторые из вышеперечисленных параметров именованные, поскольку они будут использоваться в дальнейшем для определения упрощённой модели взаимо- действия частицы с калориметром.

Моделирование каскадов взаимодействий с тремя и более уровнями не прово- дилось, т. к. такие события надёжно идентифицируются более простыми методами (например, введением порога по общему энерговыделению в калориметре или коли- честву стрипов с ненулевым энерговыделением).

Далее, калориметр представляется матрицей *C* ∈ R96×96×22, в ячейке матрицы записывается энерговыделение в соответствующем объёме калориметра.

## Восстановление траекторий частиц

Первая задача заключается в восстановлении топологической картины взаи- модействия первичной частицы в калориметре. Её описание включает в себя траекто- рию первичной частицы, точку взаимодействия и траектории порождённых частиц. Для решения этой задачи требуется в т.ч. описать параметрическую модель взаимодействия. Сложность построения модели заключается в поиске «баланса»

между реалистичностью модели и сложностью (числом параметров).

Аналитическая постановка задачи следующая. Нужно описать детерминиро- ванную модель *M* взаимодействия первичной частицы

*M* : *ν* → {0*,* 1}96×96×22*, ν* ∈ P*,* (2)

где P — пространство параметров модели, *ν* — вектор параметров. Модель должна по набору параметров возвращать подмножество трёхмерных объёмов калориметра, через которые прошла частица.

Для реализации модели *M* при фиксированном наборе параметров *ν* опреде- лим проекции *M* x(*ν*)*, M* y(*ν*) ∈ {0*,* 1}96×22 следующим образом:

*M* x(*ν*)ik = sign

96

j=1

"Σ

*M* (*ν*)ijk *>* 0#

*, i* = 1*, . . .* 96*, k* = 1*, . . . ,* 22*.* (3)

*M* y(*ν*)jk = sign

96

i=1

"Σ

*M* (*ν*)ijk *>* 0#

*, j* = 1*, . . .* 96*, k* = 1*, . . . ,* 22*.* (4)

Т.е. если при фиксированных координатах *i, k* хотя бы одна из ячеек *M* (*ν*)ijk, *j* = 1*, . . . ,* 96 принимает значение 1, то *M* x(*ν*)ik = 1, иначе *M* x(*ν*)ik = 0. Аналогично для проекции Y.

Пусть XZbin, YZbin — бинаризованные матрицы энерговыделений. Теперь вос- становление траектории частицы заключается в решении задачи минимизации

*µ*bin(*M* x(*ν*)*,* XZbin) + *µ*(*M* y(*ν*)*,* XZbin) −→ min*,* (5)

ν∈P

где *µ*bin — некоторая метрика (в нестрогом смысле) на пространстве 0-1 матриц, которую также нужно выбрать.

## Восстановление значений энерговыделений

Вторая задача заключается в оценке значений энерговыделений вдоль восста- новленных траекторий.

Пусть *ν*∗ ∈ P — вектор параметров, являющийся решением первой задачи, *M* = *M* (*ν*∗) ∈ {0*,* 1}96×96×22 — матрица, задающая траекторию частиц, участвующих во взаимодействии. Зададим множество матриц

M = {*A* ∈ R96×96×22 | sign *A*ijk ⩾ *M*ijk*,*

*i* = 1*, . . . ,* 96*, j* = 1*, . . . ,* 96*, k* = 1*, . . . ,* 22}*,* (6)

принимающих неотрицательные значения только в тех ячейках, в которых *M* при- нимает значение 1, а в остальных принимает значение 0.

Пусть матрицы проекций *A*x*, A*y ∈ R96×22 определены следующим образом.

96

Σ

x = *A*ijk *>* 0*, i* = 1*, . . .* 96*, k* = 1*, . . . ,* 22*.* (7)

*A*

ik

j=1

96

Σ

y = *A*ijk*, j* = 1*, . . .* 96*, k* = 1*, . . . ,* 22*.* (8)

*A*

jk

i=1

Будем искать оптимальное решение на множестве матриц M. Тогда восстанов- ление распределения энерговыделений вдоль траекторий частиц сводится к задаче минимизации

*µ*(*A*x*,* XZ) + *µ*(*A*y*,* YZ) −→ min*,* (9)

A∈M

где *µ* — метрика на пространстве матриц R96×96×22, которую также нужно выбрать. *Замечание.* Две перечисленные задачи можно сформулировать в виде одной задачи восстановления 96 × 96 × 22 ≈ 200000 значений матрицы энерговыделений. Численное решение задачи минимизации для модели с таким числом параметров

является вычислительно сложной задачей. Разложение исходной задачи в виде двух подзадач [(5](#_bookmark9)) и [(3.1)](#_bookmark20) значительно облегчает вычисления, т. к. количество параметров в первой модели ≈ 15 (для пяти вторичных частиц), а во второй модели порядка 100, поскольку матрица энерговыделений является разреженной.

# Алгоритм восстановления траекторий

## Идентификация антипротонов в эксперименте PAMELA

Совокупность детекторов спектрометра позволяет надёжно идентифициро- вать частицы различными практически независимыми способами. Выделение анти- протонов в потоке космических лучей проводилось с использованием трековой систе- мы, которая измеряла энерговыделение и определяла знак заряда частиц по откло- нению в магнитном поле. Преимущество этого метода заключается в широком энер- гетическом диапазоне измерений (80 МэВ – 350 ГэВ). С другой стороны, в области низких энергий (до 400 МэВ) результаты могут быть проверены и дополнены неза- висимо: при помощи анализа топологии аннигиляции в позиционно- чувствительном стриповом калориметре и информации с детекторов время-пролётной (ВПС) систе- мы. Разрабатываемая методика позволит увеличить статистику в антипротонах низ- ких энергий за счет большего геометрического фактора калориметра, по сравнению с магнитным спектрометром, а также провести сравнение с данными, полученны- ми с помощью магнитно-трековой системы. Для создания методики идентификации низкоэнергетических антипротонов с помощью позиционно-чувствительного калори- метра эксперимента PAMELA нужно:

* Восстановить трек антипротонов в калориметре.
* Используя трек и энерговыделение в калориметре, создать критерии отбора для идентификации антипротонов на фоне протонов, преобладающих частиц в КЛ и *π*− мезонов, которые в большом количестве рождаются в результате взаимодействия КЛ с контейнером и другими элементами PAMELA.

## Восстановление трека антипротонов в калориметре

Идентификация антипротонов в эксперименте PAMELA возможна двумя неза- висимыми способами:

1. используя отклонение в магнитно-трековой системе и время пролета частицы через прибор;
2. анализируя топологию взаимодействия с веществом позиционно- чувствитель- ного стрипового калориметра.

В области низких энергий антипротоны останавливаются в калориметре, те- ряя свою энергию по закону Брэгга вплоть до точки остановки, где происходит их аннигиляция с веществом прибора, сопровождающаяся большим энерговыделением и рождением вторичных частиц, как правило, -мезонов. Отклик калориметра на та- кое событие представляет собой цветное пиксельное изображение в 2-х проекциях XZ и YZ, где Z – вертикальная ось прибора, а цветом показывается энерговыделение в пикселях в качестве которых выступают отдельные стрипы. Пиксели выстраива- ются в последовательности, связанные с траекторией движения заряженных частиц или античастиц. Используя преобразование Хафа как метод анализа изображений для поиска на них прямых линий (треков), удалось построить метод восстановления треков антипротонов, аннигилировавших в калориметре спектрометра PAMELA, а

также идентификации их на фоне других частиц, присутствующих в космическом излучении.

### Преобразование Хафа

Преобразование Хафа (англ. Hough transform) предназначено для выделения прямых линий на цифровом изображении. В простейшем случае оно является линей- ным и основано на переходе в некоторое пространство параметров, где поиск прямых осуществляется тривиально. Рассмотрим параметрическое уравнение прямой:

*ρ* = *x* · cos *θ* + *y* · sin *θ,* (10)

где *ρ* — это длина радиус-вектора, ближайшей к началу координат точки на прямой (т. е. нормали к прямой, проведённой из начала координат), а *θ* — угол между этим вектором и осью абсцисс (рис. [4,](#_bookmark16) слева).

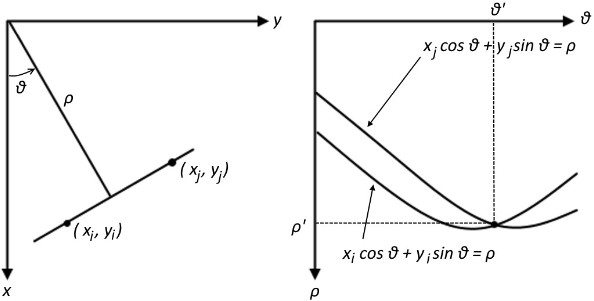


Рис. 4: Прямая в пространстве *X* - *Y* (слева), в пространстве параметров *ρ*−*θ* (справа)

Для применения преобразования Хафа к прямой на рисунке зафиксируем две точки (*x*i*, y*i) и (*x*j*, y*j) на этой прямой. Перейдём в пространство параметров (*θ, ρ*); при этом каждая из точек на исходном изображении переходит в синусоиду. Тогда

прямая на плоскости *X*−*Y* переходит в точку пересечения синусоид (*θ*′*, ρ*′) на плоско- сти *θ*−*ρ* (рис. [4,](#_bookmark16) справа). То есть с каждой прямой на исходном изображении можно связать точку с координатами (*θ, ρ*), которая является уникальной при условии, что *θ* ∈ [0*,* 2*π*] и *ρ >* 0 .

Нахождение точки пересечения синусоид позволяет определить уравнение пря-

мой [(10).](#_bookmark15)

### Методика восстановления треков частиц и античастиц на основе преобразования Хафа

Для создания методики проведено моделирование изотропных потоков анти- протонов в диапазоне жесткостей от 0.5 до 5 ГВ при помощи программного обеспе- чения, принятого в коллаборации PAMELA, и основанного на пакете Geant4 .

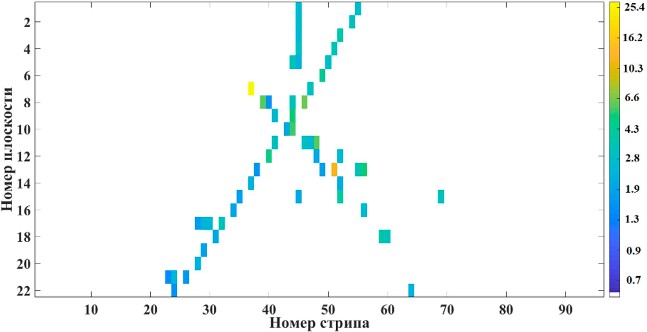


Рис. 5: Антипротон с жесткостью -0.7 ГВ в калориметре спектрометра PAMELA

Разработан алгоритм поиска треков частиц или античастиц, состоящий из сле- дующих пунктов:

* + - * преобразование изображения взаимодействующих в калориметре частиц в би- нарное изображение;
      * применение преобразования Хафа к полученному бинарному изображению и восстановление множества прямых линий;
      * классификация найденных линий на группы, соответствующие первичным и вторичным частицам;
      * восстановление и выбор трека, соответствующего первичной античастицы.

Покажем применение этого алгоритма на примере одного события – антипро- тона с жесткостью -0.7 ГВ. На рисунке 5 показано изображение его аннигиляции в калориметре: по оси *X* отложены номера плоскостей, а по оси *Y* – номера стрипов, цветом показано энерговыделение (чем ярче цвет, тем энерговыделение выше, справа на рисунке показана шкала в орч). На этом рисунке антипротон влетает вертикально сверху в апертуре спектрометра PAMELA, а другие 4 трека соответствуют продук- там его аннигиляции в веществе калориметра.

Преобразуем это изображение в бинарное, предполагая, что отсутствие сигна- ла - это логический ноль, а наличие энерговыделения - логическая единица (рис. 7). Применим преобразование Хафа, перейдя в пространство параметров (*θ, ρ*). Коли- чество пересечений синусоид, полученных в результате применения преобразования Хафа к рассматриваемому событию, приведено рисунке 7: оттенками серого показано количество пересечений в долях единицы; чем темнее, тем больше пересечений.

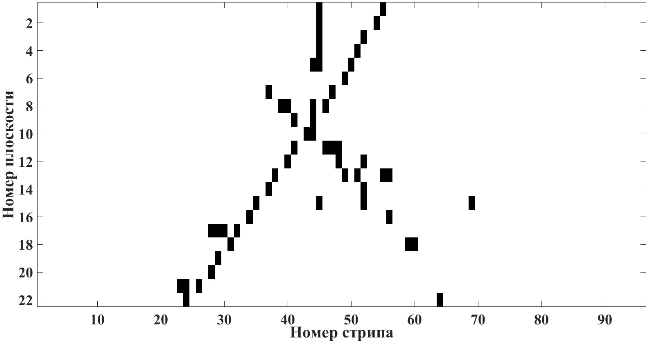


Рис. 6: Бинарное изображение взаимодействия антипротона в калориметре

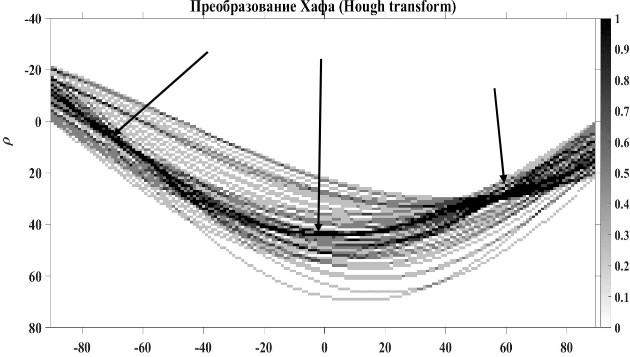


Рис. 7: Синусоиды, полученные в результате применения преобразования Хафа к рассматриваемому событию.

Самые темные области соответствуют наибольшему количеству пересечений синусоид, а значит линиям на исходном изображении. Все найденные линии показаны зелёным цветом на рисунке 8, большинство из них соответствуют углам около ±60° и 0°.

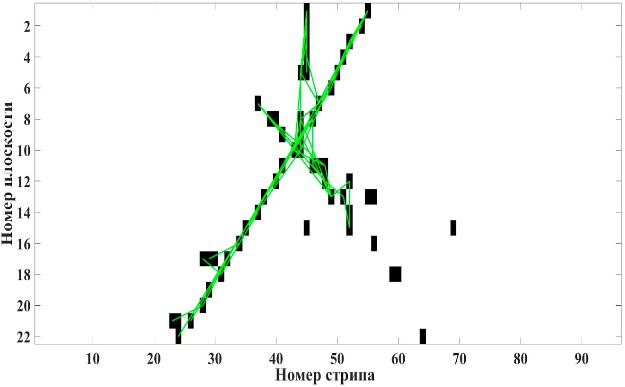


Рис. 8: Восстановленное изображение взаимодействия антипротона в калориметре

Для привязки найденных линий к различным частицам использован класси- фикатор k-средних с признаками:

* + - * угол наклона прямой
      * начальная точка прямой.

По ним классификатор распределяет линии, которые по своей топологии от- носятся к первичной и вторичным частицам. При этом треком первичной частицы считается трек, приходящий из основной апертуры прибора. На рисунке 9 красной линией показан восстановленный в калориметре трек, соответствующей треку пер- вичного антипротона. Энергию антипротона можно определить при помощи анализа длины пробега и энерговыделения до точки остановки (кривая Брэгга).

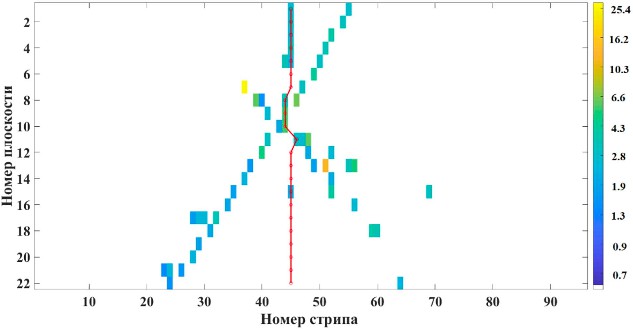


Рис. 9: Восстановленный в калориметре трек

Разработанная методика применена для восстановления треков антипрото- нов, что позволило идентифицировать эти античастицы низких энергий при помощи позиционно-чувствительного стрипового калориметра в эксперименте PAMELA.

## Обобщённая схема алгоритма реконструкции

Общая схема предлагаемого алгоритма реконструкции треков включает сле- дующие этапы:

1. Первичная обработка входных проекционных данных (*M*x*, M*y).
2. Выбор стратегии инициализации и задание начальных параметров модели.
3. Проведение глобальной оптимизации для получения приближённого решения.
4. Анализ полученного трека.

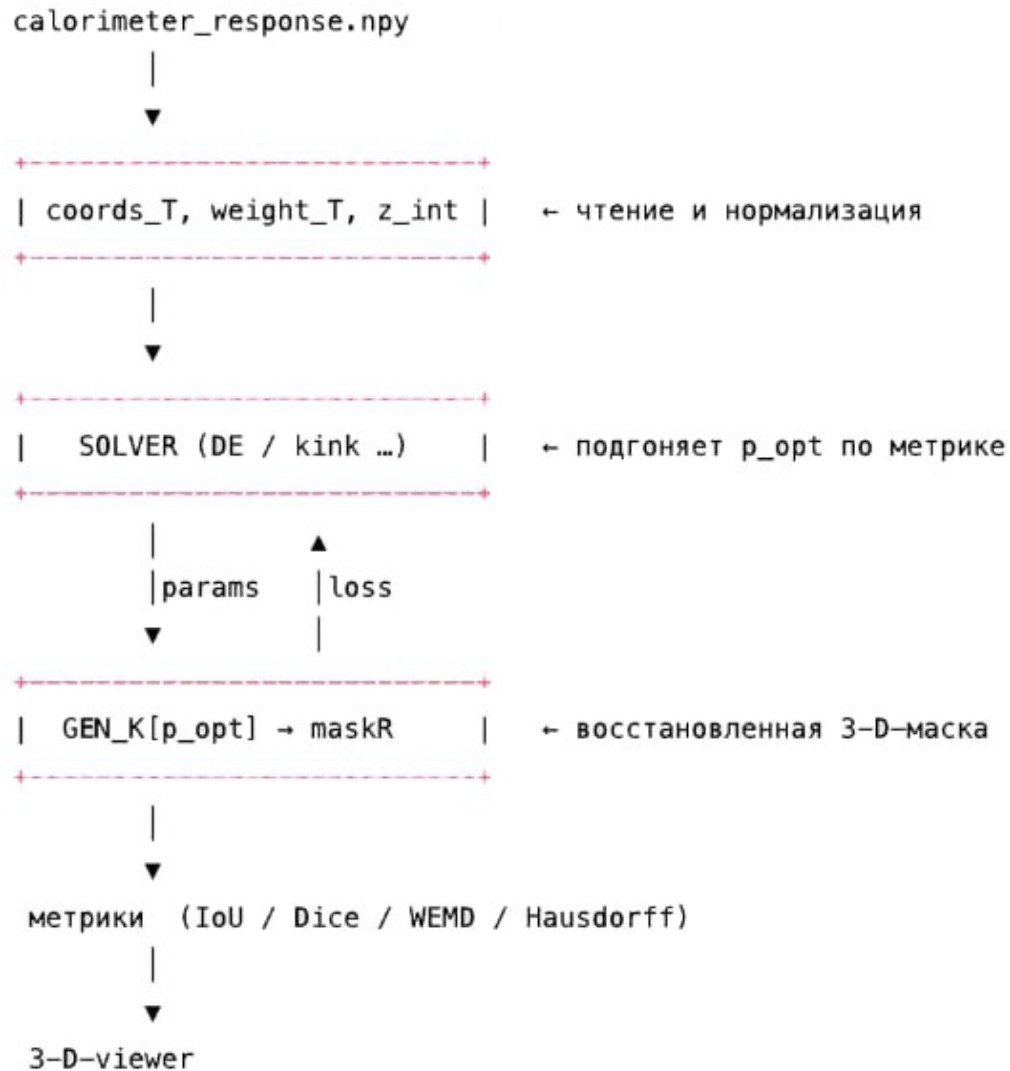


Рис. 10: Блок-схема алгоритма реконструкции треков.

# Алгоритм восстановления энергий вдоль трека

Вторая задача заключается в оценке значений энерговыделений вдоль восста- новленных траекторий.

## Формулировка задачи восстановления энергетического рас- пределения

Пусть *ν*∗ ∈ P — вектор параметров, являющийся решением первой задачи (вос- становление траекторий), и *M* = *M* (*ν*∗) ∈ {0*,* 1}96×96×22 — соответствующая бинарная матрица, задающая траектории частиц, участвующих во взаимодействии.

Определим множество матриц M:

M = {*A* ∈ R96×96×22 | *A*ijk ≥ 0*,* если *M*ijk = 1;

*A*ijk = 0*,* если *M*ijk = 0*,*

*i* = 1*, . . . ,* 96*, j* = 1*, . . . ,* 96*, k* = 1*, . . . ,* 22}*.*

Это множество включает все трёхмерные матрицы энерговыделений, которые при- нимают неотрицательные значения только в тех вокселях, где матрица *M* (рекон- струированный трек) имеет значение 1.

Проекции матрицы энерговыделений *A* ∈ M определяются как:

96

Σ

x = *A*ijk*, i* = 1*, . . . ,* 96*, k* = 1*, . . . ,* 22*.*

*A*

ik

j=1

96

Σ

y = *A*ijk*, j* = 1*, . . . ,* 96*, k* = 1*, . . . ,* 22*.*

*A*

jk

i=1

Эти проекции *A*x и *A*y представляют собой суммарные энерговыделения на соответ- ствующих 2D-проекциях реконструированного трека.

Задача восстановления распределения энерговыделений вдоль траекторий ча- стиц сводится к задаче минимизации:

*µ*(*A*x*,* XZ) + *µ*(*A*y*,* YZ) −→ min*,*

A∈M

где *µ* — метрика на пространстве матриц R96×22 (в данном случае, она применяется к проекциям 2D матриц, содержащих непрерывные значения). В вашем описании указано, что это сумма квадратов разностей:

min

V

Σ Σ

2

*V*i,j,k − XZ(*i, k*)

!

+ Σ Σ

2

*V*i,j,k − YZ(*i, j*)  *.*

! 

i,k j i,j k

Эта задача представляет собой выпуклую квадратичную программу, которая может быть решена с применением стандартных методов квадратичного программирова- ния.

## Метод проекционной релаксации и численное решение

Поскольку задача является выпуклой квадратичной программой, она может быть решена с применением стандартных методов квадратичного программирова- ния, таких как L-BFGS-B (при соответствующей формулировке) или алгоритмы из специализированных библиотек, как CVXOPT.

Для ускорения сходимости и стабилизации решения может быть использован метод *проекционной релаксации*. Этот метод является итерационным и состоит из следующих шагов:

1. **Инициализация:** Начальное распределение *V*i,j,k внутри реконструированно- го трека *M* (*ν*∗) инициализируется. Возможна равномерная инициализация или пропорциональная.
2. **Итерации проекций:** На каждой итерации:
   1. Текущая оценка *V*i,j,k проецируется на пространство, где её суммы по про- екциям соответствуют измеренным *XZ* и *Y Z*. Это достигается путём кор- ректировки значений *V*i,j,k с учётом расхождений в проекциях.
   2. Применяются ограничения: все значения *V*i,j,k обнуляются, если они отри- цательны, или если соответствующий воксель не принадлежит реконстру- ированному треку *M* (*ν*∗).

Процесс итераций продолжается до достижения сходимости, когда изменения значе- ний *V*i,j,k или целевой функции становятся пренебрежимо малыми.

*Замечание.* Как отмечается в постановке задачи, разделение исходной ком- плексной задачи на две подзадачи (восстановление геометрии, затем восстановление энергий) значительно облегчает вычислительную сложность. Количество парамет- ров в первой модели (геометрия) значительно меньше (≈ 15) по сравнению с прямым

восстановлением всей матрицы энерговыделений (96 × 96 × 22 ≈ 200000 значений),

которая хоть и является разреженной, но все равно требует решения большой систе-

мы.

# Результаты

## Результаты на модельных данных (MC)

Для оценки качества работы разработанного метода были использованы мо- дельные (симулированные) данные с известным распределением энергии. Сравне- ние восстановленного распределения *V* с эталонным *V* ⋆ позволяет оценить точность предложенного алгоритма.

Исходя из результатов, указанных в таблице [1,](#_bookmark24) можно заключить, что предло- женный подход позволяет производить более качественную реконструкцию в срав- нении с исключительно геометрическим методом.

Подход IoU Dice Energy-WEMD Proj MSE Только геометрическая реконструкция 0.45 0.62 1.25 0.08

Геометрическая + энергетическая рекон- 0.53 0.69 0.84 0.07

струкция

Таблица 1: Влияние точности восстановления распределения энергии на метрики качества реконструкции (усреднённые значения по выборке MC).

Результаты количественной оценки (Таблица [1)](#_bookmark24) подтверждают:

* Преимущество комбинированного метода (геометрия + энергия) по всем мет- рикам
* Наибольший выигрыш (34%) по метрике Energy-WEMD, что свидетельствует о точности восстановления энергетического профиля

Предложенный подход обладает рядом ограничений:

* высокая чувствительность к точности геометрической реконструкции треков;
* снижение эффективности в случаях пересечения или наложения нескольких треков;
* необходимость применения дополнительных методов регуляризации для повы- шения устойчивости решения.

Анализ синтетических данных (Рис. [11](#_bookmark25)) демонстрирует следующие ключевые особенности:

* Чёткое соответствие профиля энерговыделения вдоль оси *Z* ожидаемому рас- пределению Брэгга для заряженных частиц
* Наличие характерного пика в точке аннигиляции (*Z* ≈ 12*.*5), что соответствует модели взаимодействия антипротонов

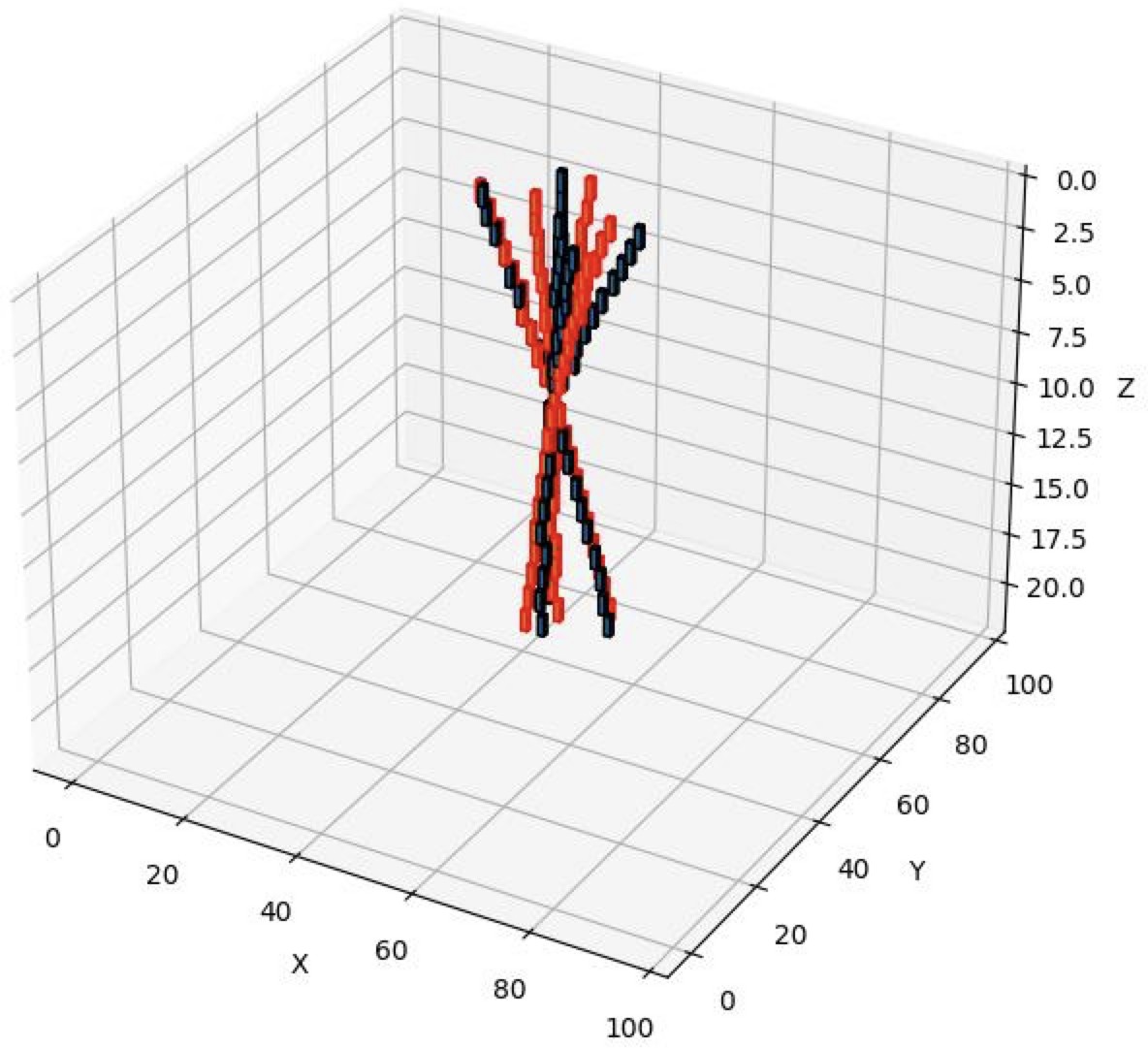


Рис. 11: Трёхмерная визуализация восстановленного события с выделенными трека- ми: востановленный (красный) и эталонный (черный)

## Результаты на данных PAMELA

Ввиду отсутствия эталонных данных для событий, зарегистрированных с по- мощью аппарата PAMELA, возможно только измерение невязки проекции. Исходя из данных, указанных в таблице [2,](#_bookmark27) можно заключить, что использование предложен- ного комбинированного метода дает более низкую невязку проекции по сравнению с исключительно геометрической реконструкцией.

Подход Proj MSE

Только геометрическая реконструкция 0.15

Геометрическая + энергетическая реконструкция 0.10

Таблица 2: Влияние точности восстановления распределения энергии на невязку про- екции реконструкции (усреднённые значения по выборке MC).

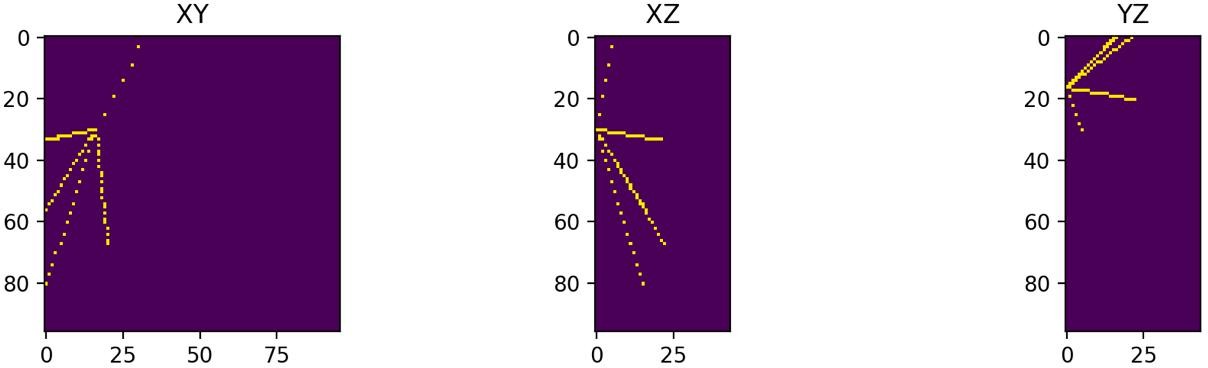


Рис. 12: Профиль распределения энерговыделения вдоль оси первичного трека.

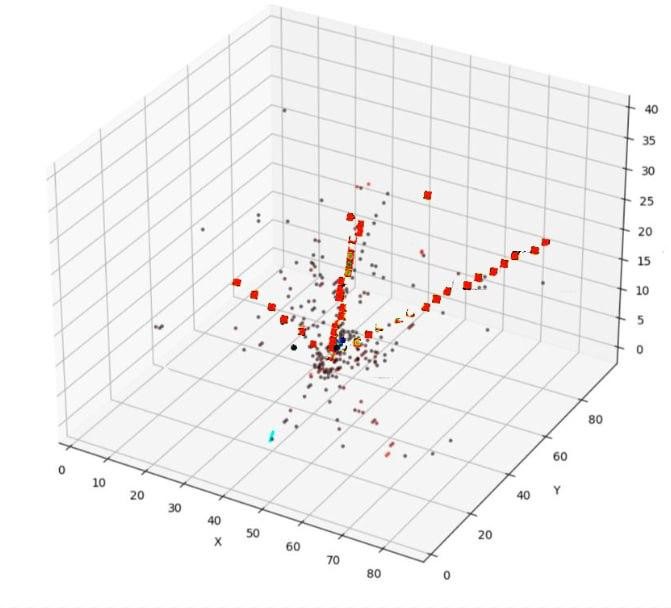


Рис. 13: Трёхмерная визуализация восстановленного события: восстановленный (красный) и энерговыделения (черный)

Анализ реальных данных (Рис. [12,](#_bookmark28) [13](#_bookmark29) ) выявили:

* Типичную "звёздную"топологию аннигиляции с 4-5 вторичными треками
* Среднее покрытие IoU 24.7% ± 3.2%, что обусловлено:
  1. Наложением треков в проекциях
  2. Шумовыми срабатываниями детектора Сравнение методов (Таблица [2)](#_bookmark27) показывает:
* Снижение проекционной невязки на 33% при использовании энергетической реконструкции
* Стабильное улучшение качества при увеличении числа треков (*N* ≤ 6)

# Заключение

Выпускная работа представила комбинированный метод восстановления трёх- мерной топологии взаимодействия заряжённых частиц в кремниево-вольфрамовом калориметре PAMELA, использующий сопряжённую оптимизацию геометрии тре- ков и распределения энерговыделений, заданных в двух ортогональных проекциях. Подход основан на параметрической модели прямолинейных и ломаных траекторий и решении выпуклой квадратичной задачи для распределения энергии, что позволи- ло совместить точность геометрической и энергетической реконструкции.

На симулированных данных Geant4 метод обеспечил рост IoU с 0.45 до 0.53, Dice — с 0.62 до 0.69 и снизил Energy-WEMD на 34% , при одновременном умень- шении проекционной ошибки MSE с 0.08 до 0.07, превзойдя чисто геометрический алгоритм . При анализе реальных событий PAMELA достигнуто среднее покрытие IoU 24.7 % ± 3.2 % и сокращение невязки проекций на 33 % . Основные ограничения связаны с экспоненциальным ростом времени вычислений при числе треков свыше восьми и повышенной чувствительностью к несовместимым проекциям и шуму.

Прототип, реализованный на Python, уже используется для отбора редких событий в эксперименте и готов к интеграции в последующие исследования. Даль- нейшая работа будет направлена на внедрение TV-регуляризации, стохастического поиска MCMC для многотрековых конфигураций и обучение 3D U-Net для лока- лизации области интереса, что должно повысить устойчивость и масштабируемость метода.

# Приложение

import numpy as np import scipy as sp import matplotlib

import matplotlib . pyplot as plt

from mpl\_toolkits . mplot3 d import Axes3 D import scipy . optimize as opt

import ot

from scipy . stats import uniform , truncnorm import time

from itertools import permutations

def plot\_3 d ( C):

fig = plt. figure ( figsize =(10 , 8))

ax = fig . add\_subplot (111 , projection =’3 d’) ax. set\_xlabel(’X’)

ax. set\_ylabel(’Y’) ax. set\_zlabel(’Z’) ax. set\_zlim (22 , 0)

ax. voxels (C, edgecolor=’k’)

def compare\_XY (X, Y):

fig = plt. figure ( figsize =(10 , 8))

ax = fig . add\_subplot (111 , projection =’3 d’) ax. set\_xlabel(’X’)

ax. set\_ylabel(’Y’) ax. set\_zlabel(’Z’) ax. set\_zlim (22 , 0)

ax. voxels (X, edgecolor=’k’) ax. voxels (Y, edgecolor=’r’)

def compare\_proj(X, Y):

fig , ax = plt. subplots (1 , 2)

ax [0]. matshow ( X. any ( axis =0). transpose () +

5 \* Y. any ( axis =0). transpose (), cmap =’ Greys ’) ax [1]. matshow ( X. any ( axis =1). transpose () +

5 \* Y. any ( axis =1). transpose (), cmap =’ Greys ’) ax [0]. set\_aspect (96 / 22)

ax [0]. set\_xlim (0 , 96)

ax [0]. set\_ylim (22 , 0)

ax [1]. set\_aspect (96 / 22)

ax [1]. set\_xlim (0 , 96)

ax [1]. set\_ylim (22 , 0)

def x\_proj( C):

return C. any ( axis =1)

def y\_proj( C):

return C. any ( axis =0)

def plot\_X\_projection ( C):

plt. matshow ( C. any ( axis =1). transpose (), cmap =’ Greys ’)

def plot\_Y\_projection ( C):

plt. matshow ( C. any ( axis =0). transpose (), cmap =’ Greys ’)

def plot\_projections ( C):

fig , ax = plt. subplots (1 , 2)

ax [0]. matshow ( C. any ( axis =1). transpose (), cmap =’ Greys ’) ax [1]. matshow ( C. any ( axis =0). transpose (), cmap =’ Greys ’) ax [0]. set\_aspect (96 / 22)

ax [0]. set\_xlim (0 , 96)

ax [0]. set\_ylim (22 , 0)

ax [1]. set\_aspect (96 / 22)

ax [1]. set\_xlim (0 , 96)

ax [1]. set\_ylim (22 , 0)

def check\_XY\_bounds (x, xmin =0 , xmax =95): return ( x >= xmin ) & ( x <= xmax )

def \_generate\_event ( startx , starty , start\_theta , start\_phi , zint , npart , theta\_part , phi\_part ):

startz = 0

maxz = 21

zint = int( zint)

l = ( np. tan ( start\_theta ) \* np. cos ( start\_phi), np. tan ( start\_theta ) \* np. sin ( start\_phi), 1)

lx = startx + l[0] \* np. arange (0 , zint + 1 , 1) ly = starty + l[1] \* np. arange (0 , zint + 1 , 1) lz = np. arange (0 , zint + 1 , 1 , dtype = int)

xint , yint = lx [-1], ly [-1]

C = np. zeros ((96 , 96 , 22), dtype = int) track\_interrupted = False

if not ( check\_XY\_bounds ( xint) and check\_XY\_bounds ( yint )): idx = check\_XY\_bounds ( lx) & check\_XY\_bounds ( ly)

lx , ly , lz = lx[ idx ], ly[ idx ], lz[ idx ] track\_interrupted = True

lx\_int = np. round ( lx ). astype ( int) ly\_int = np. round ( ly ). astype ( int)

C[ lx\_int , ly\_int , lz] = 1

if not ( track\_interrupted ): lines = dict ()

direction = np. array ( theta\_part) < np. pi / 2

for line\_num in range ( npart ): newline = []

if direction [ line\_num ]: steps = maxz - zint + 1 newline = [

xint + np. tan ( theta\_part[ line\_num ]) \*

np. cos ( phi\_part[ line\_num ]) \* np. arange (0 , steps - 1 , 1), yint + np. tan ( theta\_part[ line\_num ]) \*

np. sin ( phi\_part[ line\_num ]) \* np. arange (0 , steps - 1 , 1), np. arange ( zint , maxz , 1 , dtype = int)

]

else :

steps = zint + 1 newline = [

xint - np. tan ( theta\_part[ line\_num ]) \*

np. cos ( phi\_part[ line\_num ]) \* np. arange (0 , steps , 1), yint - np. tan ( theta\_part[ line\_num ]) \*

np. sin ( phi\_part[ line\_num ]) \* np. arange (0 , steps , 1), np. arange ( zint , -1 , -1 , dtype = int)

]

idx = ( newline [0] >= 0) & ( newline [0] <= 95) & ( newline [1] >= 0) & ( newline [1] <= 95)

lines [ line\_num ] = [ newline [0][ idx ], newline [1][ idx ], newline [2][ idx ]]

for line\_num in range ( npart ):

C[ np. round ( lines [ line\_num ][0 ]). astype ( int), np. round ( lines [ line\_num ][1 ]). astype ( int), lines [ line\_num ][2]] = 1

return C

def \_generate\_N\_event ( params , N):

return \_generate\_event (\* params [0:5] , N, params [5: 5 + N],

params [5 + N: 5 + 2 \* N])

def generate\_random\_startx ( size = 100 ): loc , scale = 47.5 , 20.0

lower , upper = - loc / scale , loc / scale

samples = truncnorm . rvs ( lower , upper , loc =loc , scale = scale , size = size ) integers = np. round ( samples )

return integers

def generate\_random\_zint ( size = 100 ):

lower , upper = (0 - 10.5) / 4.0 , (21 - 10.5) / 4.0

samples = truncnorm . rvs ( lower , upper , loc =10.5 , scale =4.0 , size = size ) integers = np. round ( samples ). astype ( int)

return integers

def generate\_random\_phi\_angle ( size = 100 ):

samples = uniform . rvs (- np. pi , np. pi , size = size ) return samples

def generate\_random\_theta\_start\_angle ( size = 100 ): scale = 0.3

lower , upper = - np. pi / 3 / scale , np. pi / 3 / scale

samples = truncnorm . rvs ( lower , upper , loc =0 , scale = scale , size = size ) return np. abs ( samples )

def generate\_random\_theta\_int\_angle ( size = 100 ): samples = uniform . rvs (0 , np. pi , size = size ) return samples

def wasserstein\_distance ( mat1 , mat2 ): coords 1 = np. argwhere ( mat1 == 1) coords 2 = np. argwhere ( mat2 == 1)

if len ( coords1 ) == 0 or len ( coords 2 ) == 0: distance = np. inf

else :

cost\_matrix = ot. dist( coords1 , coords2 , metric =’ euclidean ’) weights 1 = np. ones ( len ( coords 1 )) / len ( coords1 )

weights 2 = np. ones ( len ( coords 2 )) / len ( coords2 ) distance = ot. emd2 ( weights1 , weights2 , cost\_matrix )

return distance

def hamming\_distance ( mat1 , mat2 ): return np. sum ( mat1 != mat2 )

def \_objective\_N ( params , to\_x , to\_y , N):

startx , starty , theta , phi , zint , N, theta\_part , phi\_part =

\* params [0:5] , N, params [5: 5 + N], params [5 + N: 5 + 2 \* N] E = \_generate\_event ( startx , starty , theta , phi , zint , N, theta\_part , phi\_part)

return wasserstein\_distance ( to\_x , x\_proj( E))

+ wasserstein\_distance ( to\_y , y\_proj( E))

event = test\_events [23]

X, Y = x\_proj( event), y\_proj( event)

start\_x , start\_y = np. argwhere ( X )[0][0] , np. argwhere ( Y )[0 ][0 ]

particle\_num = 7

start\_theta\_part = generate\_random\_theta\_int\_angle ( size = particle\_num ) start\_phi\_part = generate\_random\_phi\_angle ( size = particle\_num )

start\_result = opt. minimize ( \_objective\_N ,

x0 =[ start\_x , start\_y , 0.0 , 0.0 , 10.0 ,

\* start\_theta\_part , \* start\_phi\_part], args =( X, Y, particle\_num ),

bounds =[(0 , 95), (0 , 95), (0 , np. pi / 3),

(- np. pi , np. pi), (0 , 21),

\*[(0 , np. pi)] \* particle\_num ,

\*[(- np. pi , np. pi)] \* particle\_num ],

callback = lambda result: print(".", end =""), method =’ Nelder - Mead ’)

print(" Differential␣evolution ...")

def diff\_callback ( xk , convergence ):

current\_min = \_objective\_N ( xk , X, Y, particle\_num ) print( r"{0:.3 f} ␣/ ␣". format( current\_min ), end ="") return False

result = opt. differential\_evolution ( \_objective\_N , args =( X, Y, particle\_num ),

x0 = start\_result.x, init=’ sobol ’,

bounds =[(0 , 95), (0 , 95), (0 , np. pi / 3),

(- np. pi , np. pi), (0 , 21),

\*[(0 , np. pi)] \* particle\_num ,

\*[(- np. pi , np. pi)] \* particle\_num ], callback = diff\_callback ,

maxiter =2000 ,

tol=1 e -3)

x\_dir , y\_dir , z\_dir = spherical\_to\_cartesian ( np. ones ( particle\_num ),

theta\_angles , phi\_angles ) x\_dir , y\_dir , z\_dir = x\_dir / np. abs ( z\_dir), y\_dir

/ np. abs ( z\_dir), z\_dir / np. abs ( z\_dir)

fwd\_idx = z\_dir > 0

fwd\_list = np. where ( fwd\_idx )[0] fwd\_permutations = list( permutations ( fwd\_list )) print( fwd\_permutations )

bwd\_idx = z\_dir < 0

bwd\_list = np. where ( bwd\_idx )[0] bwd\_permutations = list( permutations ( bwd\_list )) result\_permutations\_events = []

counter = 0

for i in range ( len ( fwd\_permutations )):

for j in range ( len ( bwd\_permutations )): y\_dir\_new = np. zeros ( particle\_num )

for k in range ( len ( fwd\_permutations [ i])):

y\_dir\_new [ fwd\_list[ k]] = y\_dir[ fwd\_permutations [ i][ k]]

for k in range ( len ( bwd\_permutations [ j])):

y\_dir\_new [ bwd\_list[ k]] = y\_dir[ bwd\_permutations [ j][ k]]

r, theta\_angles\_new , phi\_angles\_new = cartesian\_to\_spherical ( x\_dir , y\_dir\_new , z\_dir) result\_permutations\_events += [

\_generate\_N\_event ( np. concatenate ([ result. x [:5], theta\_angles\_new , phi\_angles\_new ]), N= particle\_num )]

print( counter , \* result. x [:5], theta\_angles\_new , phi\_angles\_new ) counter += 1

for i in range ( len ( result\_permutations\_events )):

compare\_XY ( event , result\_permutations\_events [ i]) plt. savefig (’{0}. png ’. format( i))

plt. close ()

EVENT\_ID = 1.0

evt = hits [ hits . event\_ID == EVENT\_ID ]

coords\_T , weight\_T = [], [] for \_, r in evt. iterrows ():

x, y, z = map ( int , ( r. index\_along\_x , r. index\_along\_y , r. layer )) if 0 <= x < 96 and 0 <= y < 96 and 0 <= z < 44:

coords\_T . append (( x, y, z)) weight\_T . append ( r. energy\_release )

coords\_T = np. array ( coords\_T , float) weight\_T = np. array ( weight\_T , float ); weight\_T /= weight\_T . sum ()

print(" hits :", len ( coords\_T ))

def wemd ( mask\_bool ):

P = np. argwhere ( mask\_bool)

if len ( P) == 0 or len ( coords\_T ) == 0: return 1 e6 a, b = weight\_T , np. ones ( len ( P)) / len ( P)

M = ot. dist( coords\_T , P) return ot. emd2 (a, b, M)

def iou ( m\_bool ):

tgt = np. zeros ((96 , 96 , 44), bool)

for x, y, z in coords\_T . astype ( int): tgt[x, y, z] = 1 inter = np. logical\_and ( tgt , m\_bool ). sum ()

union = np. logical\_or( tgt , m\_bool ). sum () return inter / union if union else 0

def dice ( m\_bool ):

tgt = np. zeros ((96 , 96 , 44), bool)

for x, y, z in coords\_T . astype ( int): tgt[x, y, z] = 1 inter = np. logical\_and ( tgt , m\_bool ). sum ()

return 2 \* inter / ( tgt. sum () + m\_bool. sum () + 1e -8)

def \_generate\_kink\_event ( startx , starty , th0 , ph0 , zint ,

k\_break , npart , \* angles ): mask = np. zeros ((96 , 96 , 44), np. uint8 )

def step ( x0 , y0 , th , ph , z0 , z1 ): z, x, y = z0 , x0 , y0

while z < z1 and 0 <= x < 96 and 0 <= y < 96 and z < 44: mask [ int( x), int( y), int( z)] = 1

x += np. tan ( th) \* np. cos ( ph)

y += np. tan ( th) \* np. sin ( ph)

z += 1

step ( startx , starty , th0 , ph0 , 0 , max ( int( zint) - 1 , 0))

ptr = 0

for \_ in range ( npart ):

th\_a , ph\_a , th\_b , ph\_b = angles [ ptr: ptr + 4]; ptr += 4

step ( startx , starty , th\_a , ph\_a , 0 , int( k\_break )) step ( startx , starty , th\_b , ph\_b , int( k\_break ), 44)

return mask

def make\_kink\_gen ( N): def g(\* p):

p = list( p)

args = p [:5] + [ p [5]] + [ N] + p [6:]

return \_generate\_kink\_event (\* args ) return g

GEN\_K = { n: make\_kink\_gen ( n) for n in (2 , 3 , 4)}

\_xy , \_tp = [(0 , 95), (0 , 95)], [(0 , np. pi), (- np. pi , np. pi)]

\_z , \_kb = [(5 , 35)], [(10 , 40)] BOUNDS\_K = {

2: \_xy + \_tp + \_z + \_kb + \_tp \* 4 , 3: \_xy + \_tp + \_z + \_kb + \_tp \* 6 , 4: \_xy + \_tp + \_z + \_kb + \_tp \* 8 ,

}

def make\_obj\_k ( N): def f( p):

p = list( p);

p [0] = int( round ( p [0 ]));

p [1] = int( round ( p [1]))

p [4] = int( round ( p [4 ]));

p [5] = int( round ( p [5])) try :

m = GEN\_K [ N ](\* p) > 0

except:

return 1 e6 return wemd ( m)

return f

def x 0 \_random\_kink ( N):

base = [ np. random . randint (96), np. random . randint (96),

np. random . rand () \* np. pi ,

np. random . uniform (- np. pi , np. pi), np. random . randint (5 , 35),

np. random . randint (10 , 40)] sec = [ np. random . rand () \* np. pi ,

np. random . uniform (- np. pi , np. pi)] \* 2 \* N return np. array ( base + sec )

def x 0 \_max E\_kink (N, df):

sl0 = df[ df. layer == 0]

idx = sl0 . energy\_release . idxmax ()

x0 , y0 = df. loc [ idx , [" index\_along\_x ", " index\_along\_y "]]

base = [ int( x0 ), int( y0 ),

np. pi / 4 , 0 ,

np. random . randint (5 , 35), np. random . randint (10 , 40)]

sec = [ np. random . rand () \* np. pi ,

np. random . uniform (- np. pi , np. pi)] \* 2 \* N return np. array ( base + sec )

def x 0 \_hough\_kink (N, df):

sl = df[ df. layer < 4][[" index\_along\_x ", " index\_along\_y "]]. values

x0 , y0 = sl. mean (0)

base = [ x0 , y0 , np. pi / 4 , 0 ,

np. random . randint (5 , 35), np. random . randint (10 , 40)]

sec = [ np. random . rand () \* np. pi ,

np. random . uniform (- np. pi , np. pi)] \* 2 \* N return np. array ( base + sec )

strategies = {

" random ": lambda N: x 0 \_random\_kink ( N), " maxE ": lambda N: x 0 \_max E\_kink (N, evt),

" hough ": lambda N: x 0 \_hough\_kink (N, evt)

}

N = 4

table = []

for name , sfn in strategies . items (): x0 = sfn ( N)

t0 = time . time ()

de = differential\_evolution ( make\_obj\_k ( N), BOUNDS\_K [ N], init=pop ,

popsize =20 , maxiter =100 , seed =42 , disp = False ) dt = time . time () - t0

m = GEN\_K [ N ](\* de. x) > 0

table . append ([ name , dt , iou ( m), dice ( m )])

def \_generate\_four\_event ( startx , starty , theta0 , phi0 , zint ,

theta1 , phi1 , theta2 , phi2 , theta3 , phi3 , theta4 , phi4 ):

return \_generate\_event(

startx , starty , theta0 , phi0 , zint , 4 ,

[ theta1 , theta2 , theta3 , theta 4 ], [ phi1 , phi2 , phi3 , phi4 ]

)

bounds\_four = [ (0 , 95),

(0 , 95),

(0 , np. pi),

(- np. pi , np. pi),

(0 , 44),

(0 , np. pi), (- np. pi , np. pi),

(0 , np. pi), (- np. pi , np. pi),

(0 , np. pi), (- np. pi , np. pi),

(0 , np. pi), (- np. pi , np. pi),

]

def \_objective\_four ( params , target\_mask ): params = list( params )

params [0] = int( round ( params [0]))

params [1] = int( round ( params [1])) params [4] = int( round ( params [4])) params = tuple ( params )

if len ( params ) != 13: return 1 e6

try :

gen\_mask = \_generate\_four\_event (\* params ) > 0 return wasserstein\_distance ( gen\_mask , target\_mask )

except Exception as e: print( e)

return 1 e6

# Список литературы

1. *Возомолов Э. А.* Антипротоны и дейтоны в галактических космических лучах: дис. доктора физико-математических наук // Физ.-техн. ин-т им. А. Ф. Иоффе РАН. — 2003.
2. *Picozza P. [и др.]* PAMELA – A payload for antimatter matter exploration and light-nuclei astrophysics // Astroparticle Physics. — 2007. — Т. 27, № 4. — С. 296—315. — ISSN 0927-6505.
3. *Aguilar M. [и др.]* Antiproton Flux, Antiproton-to-Proton Flux Ratio, and Properties of Elementary Particle Fluxes in Primary Cosmic Rays Measured with the Alpha Magnetic Spectrometer on the International Space Station // Phys. Rev. Lett. — 2016. — Т. 117, № 9. — С. 091103.
4. *Agostinelli S. [и др.]* Geant4–a simulation toolkit // Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment. — 2003. — Т. 506, № 3. — С. 250—303. — ISSN 0168-9002.
5. *Boezio M. [и др.]* The electron–hadron separation performance of the PAMELA electromagnetic calorimeter // Astroparticle Physics. — 2006. — Т. 26, № 2. — С. 111—118. — ISSN 0927-6505.
6. *Chamberlain O. [и др.]* Observation of antiprotons // Physical Review. — 1955. — Т. 100, № 3. — С. 947.
7. *Иванов В. К., Васин В. В., Танана В. П.* Теория линейных некорректных задач и ее приложения. — М.: Наука, 1978.
8. *Тихонов А. Н., Арсенин В. Я.* Методы решения некорректных задач. — М.: Наука, 1979.
9. *Поляк Б. Т.* Введение в оптимизацию. — М.: Наука, 1983.



Рис. 14: ПАМЕЛА