Московский государственный университет

имени М.В. Ломоносова

Химический факультет

Кафедра физической химии

Лаборатория строения конденсированных систем

Хайбрахманов Артур Ильнурович

Применение методов искусственного интеллекта для решения задач рентгенодифракционных исследований кристаллов

Дипломная работа

Научный руководитель:

д. х. н., профессор Лысенко К. С.

к. х. н., с. н. с. Дмитриенко А. О.

Москва, 2025

Оглавление

Введение

Был предложен подход, который может позволить ab initio решить проблему фаз в рентгенодифракционных исследованиях. Создан генератор синтетических дифракционных данных, который может быть использован для решения прикладных задач с использованием машинного обучения. Представлен автоматизированный контейнер, позволяющий проводить воспроизводимые эксперименты по решению задачи в рамках предложенного подхода. Были разработаны модели FFT_UNet и XRD_Transformer, учитывающие физическую специфику задачи, проведен сравнительный анализ и интерпретация их работы на реальных данных.

Решение проблемы фаз является важной задачей рентгеноструктурного анализа, особенно актуальной для белковой кристаллографии ввиду отсутствия ab initio решений в этой области. Методы машинного обучения способны преодолеть данную задачу. Предлагается увеличивать разрешение дифракционной картины, предсказывая моделью машинного обучения дальние отражения по ближним, что позволит решить проблему фаз ab initio для биомолекул. В работе разработан генератор синтетических рентгенодифракционных данных, который может быть использован для решения прикладных задач методами ИИ, и пайплайн для проведения воспроизводимых экспериментов для решения задачи в рамках предложенной методологии. Также были разработаны модели FFT_UNet и XRD_Transformer, подходящие под специфику задачи; проведены их сравнительный анализ и интерпретация их работы с помощью GradCAM и связей внимания. Было показано и обосновано, что методы глубокого обучения способны считывать кристаллографические связи и законы в рентгенодифракционных данных, но их точности численного восстановления амплитуд структурных факторов не хватает для решения проблемы фаз в рамках предложенной методологии.

1 Обзор литературы

1.1 Рентгеновская кристаллография

Рентгеновская кристаллография является важнейшим инструментом определения трехмерных структур кристаллов. Этот метод позволяет получить бесценные сведения об атомных и молекулярных структурах кристаллов, что крайне важно для понимания свойств и функций материалов в различных областях, включая химию, биологию и материаловедение. Рентгеноструктурный анализ основан на упругом рассеянии монохроматического рентгеновского излучения на трехмерной регулярной решетке атомов твердого вещества, что приводит к интерференции рентгеновских лучей.

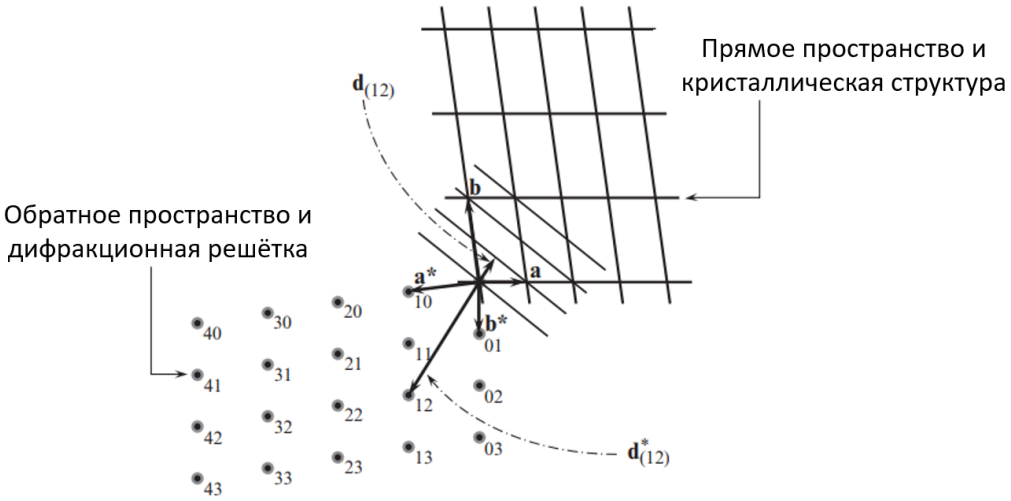

Рисунок 1.1. Связь прямого и обратного пространства [1]

Дифракция в кристалле описывается как отражения от семейств параллельных кристаллографических плоскостей элементарной ячейки кристалла (рис. 1.1). Каждая точка дифракционной картины задается набором из индексов Миллера h, k, l. Угол отражения (рассеяния) Θ для дифракционного максимума с межплоскостным расстоянием d_{hkl}, полученного при рассеянии рентгеновского излучения с длиной волны λ, определяется из закона Вульфа–Брэгга (уравнение 1.1). Также вводится обратное пространство, где каждая точка задаёт семейства кристаллографических плоскостей структуры, а значит, и отражения. Вместо индексов Миллера можно ввести вектор обратного пространства, задающий отражение $\overrightarrow{H} = (h, k, l)$. Таким образом, дифракционная картина кристалла является трансформацией упорядоченной атомной структуры в обратное пространство.

$$2\mathrm{d_{hkl}}\sin\Theta = \lambda, \tag{1.1}$$

Любое отражение также имеет математическое описание его интенсивности, известное как структурный фактор $F(\overrightarrow{H})$, зависящий от расположения r_j и коэффициентов рассеяния f_j всех атомов в элементарной ячейке кристалла:

$$\mathrm{F}(\overrightarrow{H}) = |\mathrm{F}(\overrightarrow{H})|\exp(\mathrm{i}\phi) = \sum_{\mathrm{j}=1}^{\mathrm{N}} \mathrm{F_j}(\overrightarrow{H}) = \sum_{\mathrm{j}=1}^{\mathrm{N}} \mathrm{f_j}\exp(2\pi\mathrm{i}(\overrightarrow{H}, \overrightarrow{r_j}))), \tag{1.2}$$

где $(\overrightarrow{H}, \overrightarrow{r_j}) = hx_j + ky_j + lz_j$, F_j — структурный фактор каждого из N симметрично независимых атомов в кристаллической ячейке, в котором закодирована информация об амплитуде f_j и фазе $\phi_j = 2\pi(\overrightarrow{H}, \overrightarrow{r_j})$ рассеяной этим атомом волны.

1.2 Проблема фаз

Структурный фактор, как следует из определения (уравнение 1.2), является комплексной величиной, которая описывает вклад дифракции всех атомов кристаллической решетки в отражение. Стоит отметить, что оно было получено в следующем приближении:

- Элементарная ячейка поделена на N атомов в точках $\overrightarrow{r_j}$.
- Каждый из атомов рассеивает волну с амплитудой f_j и фазой $\phi_j = 2\pi(\overrightarrow{H}, \overrightarrow{r_j})$.

Перейдем к более общему описанию — заменим атомы на маленькие параллелепипеды, внутри которых находятся электроны и рассеивают излучение. Тогда предыдущее приближение превращается в следующее:

- Элементарная ячейка поделена на N маленьких параллелепипедов, объем каждого ΔV и позиция $\overrightarrow{r_j}$.
- Каждый из параллелепипедов рассеивает волну с амплитудой $\rho(\overrightarrow{r_j})\Delta V$ и фазой $\phi_j = 2\pi(\overrightarrow{H} \cdot \overrightarrow{r_j})$, где $\rho(\overrightarrow{r_j})$ — электронная плотность внутри параллелепипеда.

Тогда выражение 1.2 можно записать следующим образом, устремив объемы параллелепипедов к нулю:

$$\mathrm{F(h,k,l)} = \sum_{\mathrm{j}=1}^{\mathrm{N}} \rho(\overrightarrow{r_j})\Delta\mathrm{V_j}\exp(2\pi\mathrm{i}(\overrightarrow{H} \cdot \overrightarrow{r_j})) = \int_{\mathrm{V}} \rho(\overrightarrow{r})\exp(2\pi\mathrm{i}(\overrightarrow{H} \cdot \overrightarrow{r_j}))\mathrm{dV}, \tag{1.3}$$

где интегрирование ведется по всему объему элементарной ячейки.

Можно заметить, что уравнение 1.3 является обратным преобразованием Фурье, переводящее электронную плотность в структурные факторы. Тогда можно записать прямое преобразование Фурье:

$$\rho(\overrightarrow{r}) = \int \mathrm{F}(\overrightarrow{H}) \exp(-2\pi \mathrm{i}(\overrightarrow{H} \cdot \overrightarrow{r}))\mathrm{dV}^* = \mathrm{V}^* \sum_{\overrightarrow{H}} \mathrm{F}(\overrightarrow{H}) \exp(-2\pi \mathrm{i}(\overrightarrow{H} \cdot \overrightarrow{r_j})), \qquad (1.4)$$

где V^* — объем элементарной обратной ячейки.

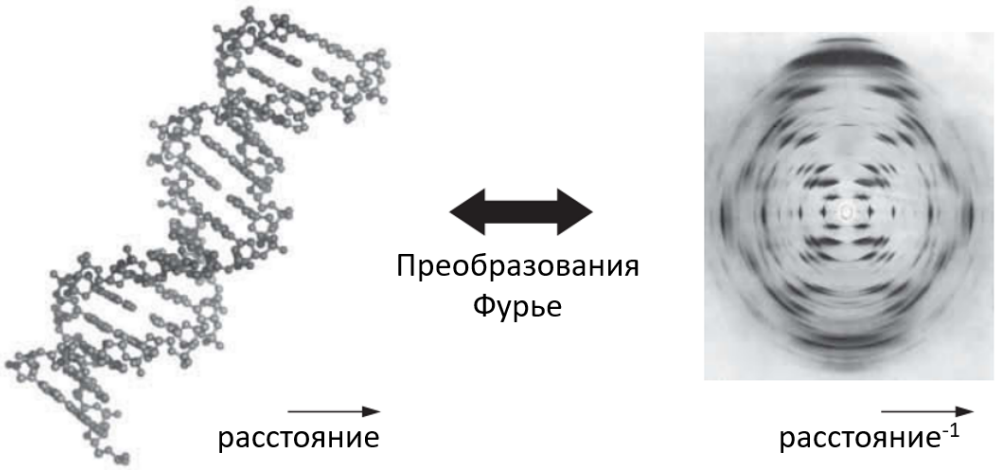

Рисунок 1.2. Схема связи электронной плотности и структурных факторов [1]

Если амплитуда и фазы всех дифрагированных лучей были бы зарегистрированы, можно было бы рассчитать распределение электронной плотности в кристалле, применив преобразвование Фурье к дифракционной картине (рис. 1.2). Однако в ходе рентгенодифракционного эксперимента регистрируются лишь интенсивность излучения, информация о фазах теряется, и для получения кристаллической структуры требуется решить так называемую "проблему фаз" или "фазовая проблема".

Таким образом, в идеальном случае, при сохранении полной информации о фазах в ходе эксперимента, структура кристалла могла бы быть непосредственно восстановлена из измеренных амплитуд методом обратного преобразования Фурье. Поскольку же фазовая составляющая оказывается недоступной при рентгеновском рассеянии, восстановление электронной плотности становится принципиально затруднённым. Эта утрата фазовой информации, известная как фазовая проблема, представляет собой краеугольный камень рентгеновской кристаллографии, и в последующих разделах будут рассмотрены основные методы её решения.

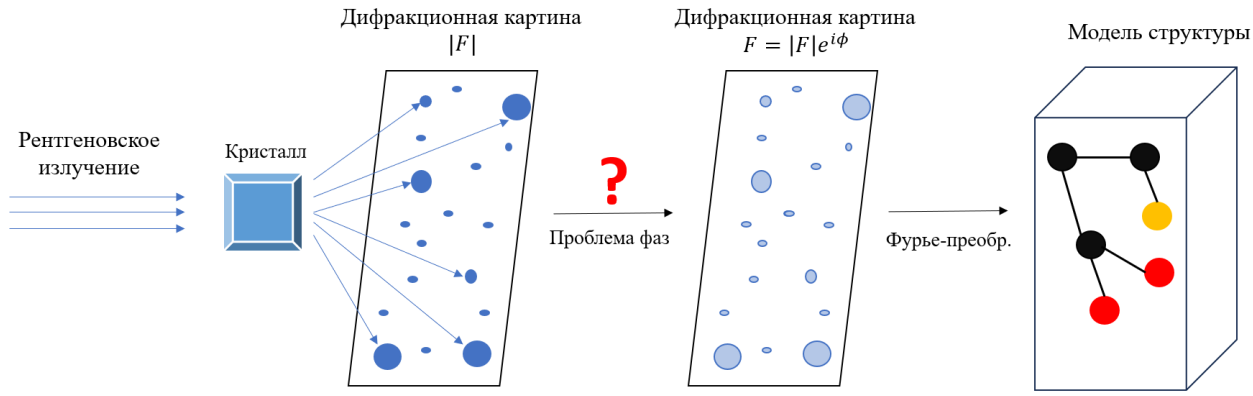

Рисунок 1.3. Схема определения кристаллической структуры

1.3 Методы решения фазовой проблемы

1.3.1 Прямые методы

В работе [2] впервые было предложено решение проблемы фаз для центросимметричной группы $P\bar{1}$, которое требует только знание амплитуд структурных факторов и химического состава кристалла. Данный подход, основанный на вероятностном подходе и предположении, что в распределение амплитуд структурного фактора уже заложена информация о фазах, в дальнейшем был распространен на все центросимметричные группы, а затем адаптирован для нецентросимметричных структур. В дальнейшем этот метод и подобные были названы прямыми, их объединяют статистические взаимосвязи между двумя, тремя и четырьмя сильными отражениями — структурные инварианты и полуинварианты [3].

Для того, чтобы избавиться от явной зависимости структурного фактора от угла рассеяния, авторы заменяют реальный кристалл с электронной плотностью $\rho(\overrightarrow{r})$ на идеальный, элементарная ячейка которого состоит из дискретных неподвижных точечных атомов, которые расположены в максимумах электронной плотности. Тогда структурный фактор $F(\overrightarrow{H}) = |F(\overrightarrow{H})| \exp(i\phi(\overrightarrow{H}))$ следует заменить на нормализованный структурный фактор $E(\overrightarrow{H}) = |E(\overrightarrow{H})| \exp(i\phi(\overrightarrow{H}))$ [4]:

$$|\mathrm{E}|^2 = \frac{|\mathrm{F}|^2}{< |\mathrm{F}|^2 >} \tag{1.5}$$

В уравнении 1.5 параметр $< |F|^2 >$ есть математическое ожидание (среднее) квадрата амплитуды структурного фактора, для расчета которого нужна *a priori* информация. Существует множество способов вычисления нормализованных структурных факторов,

которые исходят из количества доступных данных, приведём некоторые из них [5]:

1. Отсутствие структурной информации: позиции атомов приняты случайными величинами. Пусть ϵ — некоторый параметр, зависящий от группы симметрии, тогда:

$$< |\mathrm{F}|^2 >= \epsilon \sum_{\mathrm{j}=1}^{\mathrm{N}} \mathrm{f}_{\mathrm{j}}^2 \tag{1.6}$$

2. Известны M групп по M_i атомов в каждой, конфигурация которых известна, но неизвестны ориентация и позиция самих групп. Некоторое количество межатомных расстояний $r_{j_1 j_2}$ известно, тогда для отражения $\overrightarrow{H}$:

$$< |\mathrm{F}|^2 >= \epsilon \left[\sum_{\mathrm{j}=1}^{\mathrm{N}} \mathrm{f}_{\mathrm{j}}^2 + \sum_{\mathrm{i}=1}^{\mathrm{M}} \sum_{\mathrm{j}_1 \neq \mathrm{j}_2 = 1}^{\mathrm{M_i}} \mathrm{f}_{\mathrm{j}_1} \mathrm{f}_{\mathrm{j}_2} \frac{\sin(2\pi |\overrightarrow{H}| \mathrm{r}_{\mathrm{j}_1 \mathrm{j}_2})}{2\pi |\overrightarrow{H}| \mathrm{r}_{\mathrm{j}_1 \mathrm{j}_2}} \right] \tag{1.7}$$

3. Известны M групп по M_i атомов в каждой, конфигурация и ориентация которых известны, но неизвестна позиция самих групп. Некоторое количество межатомных расстояний $r_{j_1 j_2}$ зафиксированы, тогда для отражения $\overrightarrow{H}$:

$$< |\mathrm{F}|^2 >= \epsilon \left[\sum_{\mathrm{j}=1}^{\mathrm{N}} \mathrm{f}_{\mathrm{j}}^2 + \sum_{\mathrm{i}=1}^{\mathrm{M}} \sum_{\mathrm{j}_1 \neq \mathrm{j}_2 = 1}^{\mathrm{M_i}} \mathrm{f}_{\mathrm{j}_1} \mathrm{f}_{\mathrm{j}_2} \exp 2\pi \mathrm{i} (\overrightarrow{H}, \overrightarrow{r_{j_1 j_2}}) \right] \tag{1.8}$$

4. Известны M групп атомов и их позиция, тогда:

$$< |\mathrm{F}|^2 >= |\mathrm{F_M}|^2 + \epsilon \sum_{\mathrm{i}=1}^{\mathrm{Q}} \mathrm{f}_{\mathrm{i}}^2, \tag{1.9}$$

где F_M — структурный фактор известной подструктуры,, Q — количетсво неизвестных атомов.

Однако чтобы рассчитать нормализованные структурные факторы по уравнению 1.5 требуется ещё два условия — величины должны быть в абсолютной шкале, а наблюдаемые амплитуды являются относительными величинами. Также в формулах для $< |F|^2 >$ выше никак не учтено тепловое движение атомов. Оба обстоятельства можно преодолеть с использованием графика Вилсона [6], согласно которому наблюдаемые данные разделяются на несколько промежутков по переменной $s = \left(\frac{sin\theta}{\lambda}\right)^2$, в каждом из которых вычисляется средняя интенсивность $< I_{obs} >=< |F_{obs}|^2 >$. Для каждого

промежутка можно вычислить $K < I >$, где K — параметр, который необходим для перевода интенсивности рентгеновского излучения в абсолютные величины:

$$\mathrm{K} < \mathrm{I} >=< |\mathrm{F_{obs}}|^2 > \exp(-2\mathrm{Bs}^2), \tag{1.10}$$

где B — термический параметр, $< F_{obs} >$ вычисляется по уравнениям 1.6, 1.7, 1.8, 1.9.

Чтобы найти параметры B и K, уравнение 1.10 логарифмируют, строят линейный график по уравнению 1.11 в координатах $(s^2, \ln \frac{<I>}{<|F_{obs}|^2>})$ и получают параметры B и K после аппроксимации.

$$\ln \frac{< \mathrm{I} >}{< |\mathrm{F_{obs}}|^2 >} = -\ln \mathrm{K} - 2\mathrm{Bs}^2 \tag{1.11}$$

Таким образом, после построения графика Вилсона и нахождения нужных параметров, нормализованные структурные факторы вычисляются по итоговой формуле:

$$|\mathrm{E}|^2 = \frac{\mathrm{KI_{obs}}}{< |\mathrm{F_{obs}}|^2 > \exp(-2\mathrm{Bs}^2)} \tag{1.12}$$

Также известны [5] плотности распределения полученных нормализованных структурных факторов, которые различаются в зависимости от симметрии кристаллической структуры (уравнения 1.13, 1.14), их вид представлен на рис. 1.4. Нетрудно показать, что первый момент $|E|$ независимо от структуры равен 1, а математическое ожидание величины $(E^2 - 1)^2$, которое можно рассчитать по уравнению 1.15, для центросимметричных кристаллов больше (2.0), чем для нецентросимметричных (1.0), что можно использовать как критерий центросимметричности структуры.

$$\text{Центросимметричная: } \mathrm{P}(|\mathrm{E}|) = \sqrt{\frac{2}{\pi}} \exp(-\frac{\mathrm{E}^2}{2}) \tag{1.13}$$

$$\text{Нецентросимметричная: } \mathrm{P}(|\mathrm{E}|) = 2|\mathrm{E}| \exp(-|\mathrm{E}|^2) \tag{1.14}$$

$$< (\mathrm{E}^2 - 1)^2 >= \int_0^\infty \mathrm{P(E)}(\mathrm{E}^2 - 1)^2 \mathrm{dE} \tag{1.15}$$

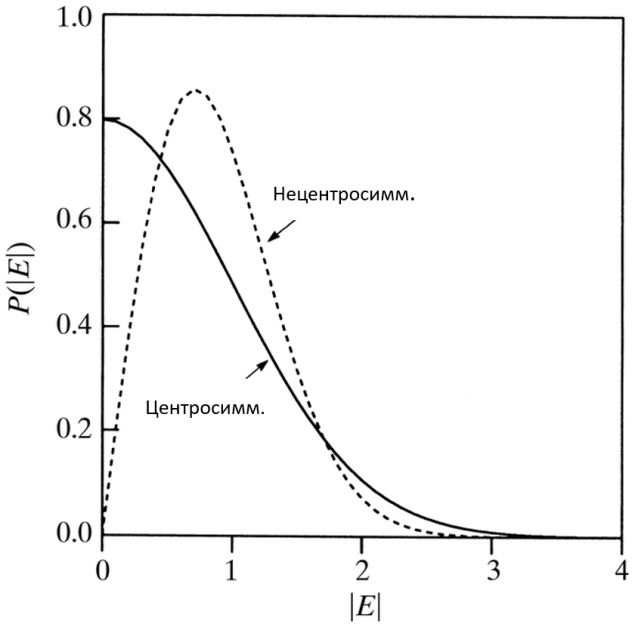

Рисунок 1.4. Плотность распределения $|E|$ для центросимметричных и нецентросимметричных кристаллов [5]

Прямые методы также основаны на структурных инвариантах — комбинации фаз рентгеновских отражений, сумма значений которых не меняется при сдвиге начала координат. Приведём некоторых из них. В работах [7], [8] была показана следующая взаимосвязь между высокоинтенсивными отражениями:

$$\mathrm{s}(\mathrm{h}+\mathrm{h}',\mathrm{k}+\mathrm{k}',\mathrm{l}+\mathrm{l}') \approx \mathrm{s}(\mathrm{h},\mathrm{k},\mathrm{l})\mathrm{s}(\mathrm{h}',\mathrm{k}',\mathrm{l}') \tag{1.16}$$

$$\phi(\mathrm{h}+\mathrm{h}',\mathrm{k}+\mathrm{k}',\mathrm{l}+\mathrm{l}') \approx \phi(\mathrm{h},\mathrm{k},\mathrm{l}) + \phi(\mathrm{h}',\mathrm{k}',\mathrm{l}') \tag{1.17}$$

Из закона Фриделя следует, что:

$$\phi(-\mathrm{h}-\mathrm{h}',-\mathrm{k}-\mathrm{k}',-\mathrm{l}-\mathrm{l}') = -\phi(\mathrm{h}+\mathrm{h}',\mathrm{k}+\mathrm{h}',\mathrm{l}+\mathrm{l}') \tag{1.18}$$

Тогда, сложив уравнения 1.18 и 1.17 мы получим выражение, которое получило название триплетное отношение:

$$\phi(\mathrm{h},\mathrm{k},\mathrm{l}) + \phi(\mathrm{h}',\mathrm{k}',\mathrm{l}') + \phi(-\mathrm{h}-\mathrm{h}',-\mathrm{k}-\mathrm{k}',-\mathrm{l}-\mathrm{l}') \approx 0 \tag{1.19}$$

Из данного уравнения и закона Фриделя так же нетрудно получить выражение для квартета отражений:

$$\phi(\mathrm{h},\mathrm{k},\mathrm{l}) + \phi(\mathrm{h}',\mathrm{k}',\mathrm{l}') + \phi(\mathrm{h}'',\mathrm{k}'',\mathrm{l}'') + \phi(-\mathrm{h}-\mathrm{h}'-\mathrm{h}'',-\mathrm{k}-\mathrm{k}'-\mathrm{k}'',-\mathrm{l}-\mathrm{l}'-\mathrm{l}'') \approx 0 \quad (1.20)$$

Также нельзя не упомянуть важное соотношение, которое является ключевым для прямых методов, а также нашло применение в других способах решения фазовой проблемы. Формула тангенсов — равенство, позволяющее рассчитать фазу отражения [9]:

$$\tan\phi(\overrightarrow{H}) = \frac{\sum\limits_{\overrightarrow{K}} |\mathrm{E}(\overrightarrow{K})\mathrm{E}(\overrightarrow{H}-\overrightarrow{K})| \sin(\phi(\overrightarrow{K}) + \phi(\overrightarrow{H}-\overrightarrow{K}))}{\sum\limits_{\overrightarrow{K}} |\mathrm{E}(\overrightarrow{K})\mathrm{E}(\overrightarrow{H}-\overrightarrow{K})| \cos(\phi(\overrightarrow{K}) + \phi(\overrightarrow{H}-\overrightarrow{K}))} \quad (1.21)$$

На сегодняшний день прямые методы являются одним из наиболее используемых и популярных подходов для решения проблемы фаз в рентгеновской кристаллографии, незаменимыми для исследования малых и средних молекул. Основываясь на анализе статистических соотношений между нормализованными структурными факторами, эти методы позволяют оценить вероятные значения фаз с помощью структурных инвариантов, что делает возможным восстановление электронной плотности без дополнительных экспериментальных данных. Подробности расчётов и имплементации метода достаточно громоздки и достойны отдельного изучения [4].

Прямые методы непрерывно совершенствуются: современные алгоритмы интегрируют классическую тангенсную формулу для начальной оценки фаз с передовыми техниками модификации электронной плотности, осуществляемыми непосредственно в прямом пространстве [10]. Такой гибридный подход позволяет не только получить более надёжные фазы благодаря доказанной статистической основе тангенс-формулы, но и существенно повысить качество и скорость сходимости решения за счёт итеративного улучшения электронной карты в реальном пространстве. В результате объединения этих методологических парадигм удаётся достичь значительного повышения эффективности и надёжности решения фазовой проблемы даже для сложных кристаллических структур.

1.3.2 Метод Паттерсона

Функция Паттерсона $P(\overrightarrow{u})$ представляет собой автокорреляционную функцию (свёртку функции с собой) электронной плотности [1]:

$$\mathrm{P}(\overrightarrow{u}) = \int_{\mathrm{V}} \rho(\overrightarrow{r})\rho(\overrightarrow{r}+\overrightarrow{u})\mathrm{d}\overrightarrow{r}, \quad (1.22)$$

где $\overrightarrow{u}$ — вектор координат ячейки Паттерсона, которая совпадает по размерам с обычной, V — объем кристаллической решетки в прямом пространстве.

Из свойств автокорреляционной функции прямо следует, что функция Паттерсона $P(\overrightarrow{u})$ принимает большие значения тогда и только тогда, когда электронная плотность принимает ненулевое значения в точке $\overrightarrow{r} = (x, y, z)$ и точке $\overrightarrow{r} + \overrightarrow{u} = (x+u, y+v, z+w)$, то есть в этих точках расположены атомы. Пики функции Паттерсона достигаются в таких положениях ячейки Паттерсона, которые отвечает межатомным векторам кристалла.

Из экспериментальных данных функция Паттерсона рассчитывается следующим образом:

$$\mathrm{P}(\overrightarrow{u}) = \frac{1}{\mathrm{V}} \sum_{\overrightarrow{H}} |F|^2(\overrightarrow{H}) \exp(-2\pi i(\overrightarrow{H}, \overrightarrow{u})) \tag{1.23}$$

Также можно выделить следующие свойства функции Паттерсона:

- Функция Паттерсона всегда чётная, поскольку для каждой пары атомов существует пара межатомных векторов: $P(\overrightarrow{u}) = P(\overrightarrow{-u})$.
- Карта Паттерсона (графическое представление функции) обладает той же симметрией, что и группа Лауэ кристалла.
- Карта Паттерсона всегда имеет большой пик в начале координат — явно следует из определения.
- Максимумы функции широкие и размазанные благодаря перекрыванию электронной плотности атомов.

Проанализируем количество максимумов функции Паттерсона [11]. Пусть в элементарной ячейке N атомов с атомными факторами рассеяния f_j. Из определения структурного фактора 1.2 можно получить следующее выражение для $|F|^2$:

$$|\mathrm{F}(\overrightarrow{H})|^2 = \mathrm{F}(\overrightarrow{H})\mathrm{F}^*(\overrightarrow{H}) = \left[\sum_{\mathrm{j}=1}^{\mathrm{N}} \mathrm{f_j} \exp(2\pi \mathrm{i}(\overrightarrow{H}, \overrightarrow{r_j}))\right] \left[\sum_{\mathrm{j}=1}^{\mathrm{N}} \mathrm{f_j} \exp(-2\pi \mathrm{i}(\overrightarrow{H}, \overrightarrow{r_j}))\right] \tag{1.24}$$

$$|\mathrm{F}(\overrightarrow{H})|^2 = \sum_{\mathrm{j}=1}^{\mathrm{N}} \mathrm{f_j^2} + \sum_{\mathrm{j}=1}^{\mathrm{N}} \sum_{\mathrm{i}=1,\mathrm{i}\neq\mathrm{j}}^{\mathrm{N}} \mathrm{f_i f_j} \exp(2\pi \mathrm{i}(\overrightarrow{H}, \overrightarrow{r_j} - \overrightarrow{r_i})) \tag{1.25}$$

Таким образом, объединив уравнения 1.25 и 1.23, получаем, что функция Паттерсона является суммой N^2 атомных взаимодействий, из которых N в начале координат с

весами f_j^2 и $N(N-1)$ попарных взаимодействий, пропорциональных $f_i f_j$. Таким образом, в карте Паттерсона $N(N-1)$ максимумов, которые зависят от межатомных векторов и типов атомов в ячейке. Нетрудно показать, что интенсивность этих пиков пропорциональна атомным номерам соответствующих атомов Z_i (уравнение 1.26, m — фактор мультиплетности). Поскольку функция Паттерсона является четной, для описания уникальных попарных взаимодействий достаточно $N(N-1)/2$ значений, поэтому часто используемым для представления данных является верхнетреугольная матрица.

$$\mathrm{P}(\overrightarrow{u_{ij}}) = \frac{\mathrm{mZ_iZ_j}}{\sum\limits_{\mathrm{j=1}}^{\mathrm{N}} \mathrm{f_j^2}} \tag{1.26}$$

Поскольку N максимумов карты Паттерсона, отвечающим свёртке электронной плотности каждого атома с самим собой, являются не очень информативными, от них можно избавиться, изменив структурные факторы перед расчётом функции Паттерсона по уравнению 1.27. Аналогично, если положения каких-то атомов заранее точно известны, пики, отвечающим их межатомным векторам, можно убрать благодаря модификации $|F(\overrightarrow{H})|^2$ в уравнении 1.28, где r_1, r_2 — позиции известных атомов 1 и 2, σ_1, σ_2 — их термические коэффициенты.

$$|\mathrm{F_{mod}}(\overrightarrow{H})|^2 = |\mathrm{F}(\overrightarrow{H})|^2 - \sum_{\mathrm{j=1}}^{\mathrm{N}} \mathrm{f_j^2} \tag{1.27}$$

$$|\mathrm{F_{mod}}(\overrightarrow{H})|^2 = |\mathrm{F}(\overrightarrow{H})|^2 - \sum_{\mathrm{j=1}}^{\mathrm{N}} \mathrm{f_j^2}\sigma_\mathrm{j}^2 - 2\mathrm{f_1 f_2}\sigma_1\sigma_2 \cos(2\pi(\overrightarrow{H}, \overrightarrow{r_1} - \overrightarrow{r_2})) \tag{1.28}$$

Для борьбы с шириной пиков, которые могут сильно повлиять на корректность определение фаз дифракционных отражений, используют процедуру утончения карты Паттерсона. Для этого вместо структурных факторов и их интенсивностей $|F|^2$ используют нормализованные структурные факторы $|E|^2$. Поскольку интенсивность нормализованных структурных факторов не так сильно падает с увеличением угла рассеяния благодаря поправочному множителю, полученный набор получен как бы от атомов меньшего размера, в результате чего максимумы Паттерсона также будут более узкие.

Карту Паттерсона можно представить суммой нескольких копий исходной структуры с разными весами [12], которые различаются тем, какой атом находится в начале паттерсоновских координат (u, v, w) (рис. 1.5). Для простых структур

низкомолекулярных соединений возможно расшифровать карту, сопоставив каждому максимуму межатомный вектор. Однако количество пиков функции Паттерсона для структуры из N атомов растёт по квадратичному закону, что делает невозможным прямую интерпретацию без дополнительных операций над картой.

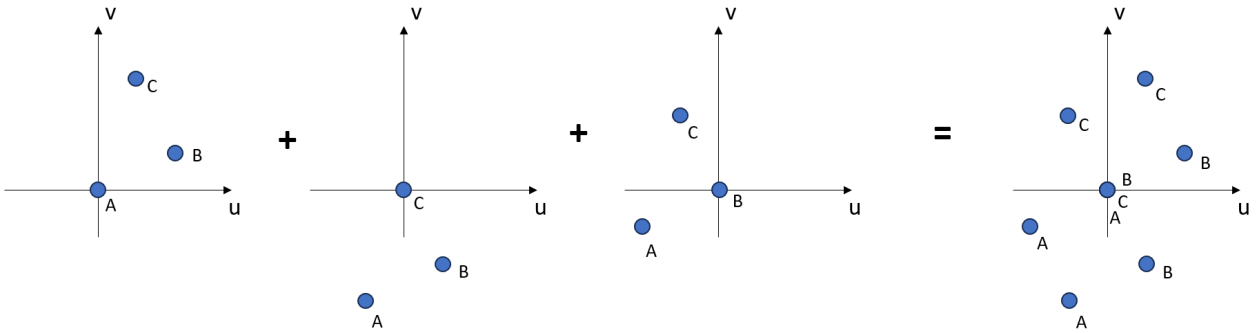

Рисунок 1.5. Схема карты Паттерсона для простейшей структуры из 3 атомов

С развитием компьютерных технологий во второй половине 20 века произошел расцвет методов, основанных на функции Паттерсона. Так, особенно полезным для решения фазовой проблемы оказался метод суперпозиции карт Паттерсона [13]. Было показано, что суперпозиция исходной карты Паттерсона со смещенной на какой-то трансляционный вектор может привести к меньшему набору межатомных векторов (рис. 1.6), поскольку точки, соответствующие образу исходной структуры, будут всегда повторяться. Суперпозиция представляет собой, например, пересечение или сумму наборов векторов (см. далее). Через множество применений данной процедуры можно получить исходную структуру.

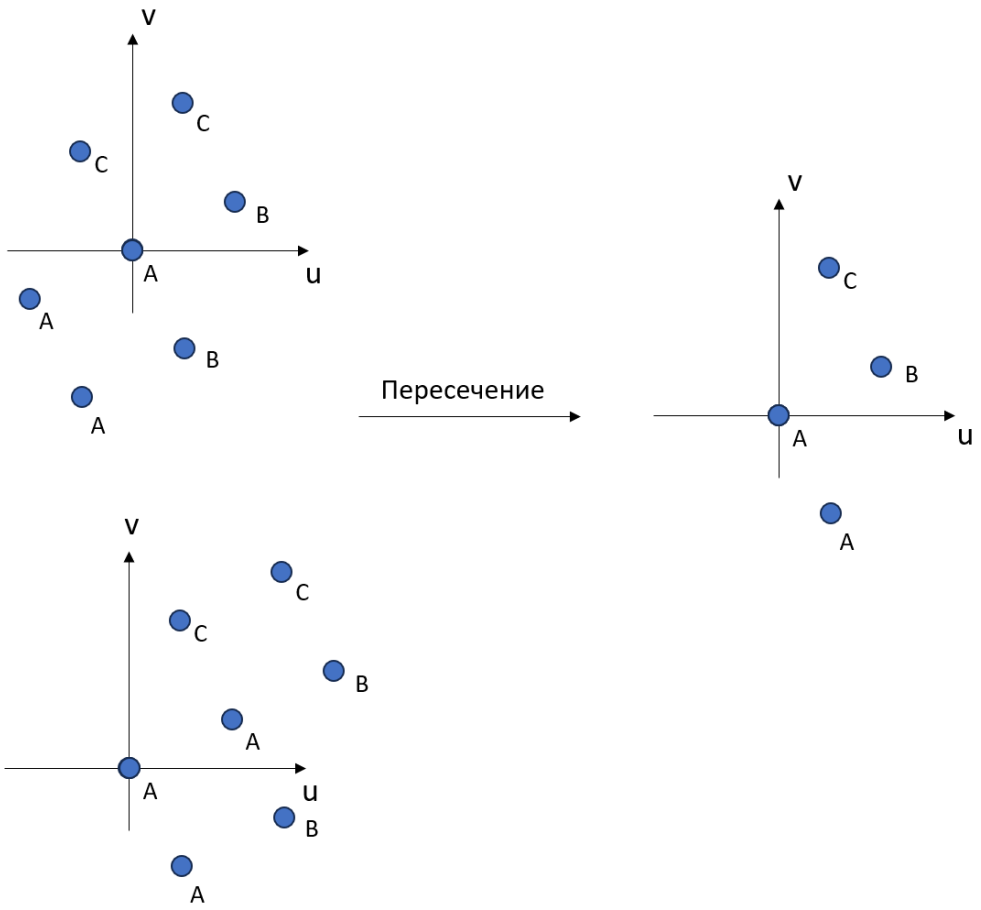

Рисунок 1.6. Суперпозиция (пересечение) исходной и смещенной на вектор $\overrightarrow{AB}$ карт Паттерсона

Есть три основных подхода, которые используются для выделения структуры молекулы из наложенных карт Паттерсона — с помощью функций суммы, произведения и минимума суперпозиции [11]. Пусть есть N карт Паттерсона, каждая из которых смещена на вектор $\overrightarrow{u_i}$. Функция суммы (уравнение 1.29) основывается на том, что она будет иметь наибольшие значения в точках пересечения всех карт, что будет отвечать расположениям атомов.

$$\mathrm{S}(\overrightarrow{r}) = \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{P}(\overrightarrow{r} + \overrightarrow{u_i}) = \sum_{\overrightarrow{H}} \left[|F(\overrightarrow{H})|^2 \exp(2\pi i(\overrightarrow{H}, \overrightarrow{r})) \left(\sum_{j=1}^{N} \exp(2\pi i(\overrightarrow{H}, \overrightarrow{u_i})) \right) \right] \qquad (1.29)$$

Функция произведения суперпозиции (уравнение 1.30) является более сильной по сравнению с суммой в том смысле, что все точки, в которых нулевое значение хотя бы у одной из карт, будут обнулены.

$$\mathrm{Pr}(\overrightarrow{r}) = \prod_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{P}(\overrightarrow{r} + \overrightarrow{u_i}) \tag{1.30}$$

Если при наложении двух карт в точке ненулевая плотность, то суммарное значение функции Паттерсона будет больше, чем от одного изображения структуры. Если взять минимум от накладывающихся карт, то при удачной суперпозиции останется лишь плотность от одной копии структуры [14]. Так, в SHELXS-96 [15] имплементирован вариант суперпозиции карт Паттерсона с функцией минимума, которая берётся от двух копий утонченной функции Паттерсона, смещенных на векторы $\overrightarrow{u}$ и $\overrightarrow{-u}$.

1.3.3 Метод обратного заряда (charge–flipping)

В работе [16] был предложен простой, но эффективный алгоритм — метод обратного заряда (charge–flipping), использующий прямое и обратное пространство. Данный алгоритм основан на итеративном приближении фаз дифракционных максимумов, которые задаются случаяными в начале, к настоящим фазам, позволящим определить кристаллическую структуру по дифракционным данным.

Пусть в эксперименте был зарегистрирован набор дифракционных отражений, которые характеризуются амплитудами $F_{obs}(\overrightarrow{H})$. Незарегистрированные отражения в других точках обратного пространства принимаются за нулевые. Алгоритм начинается с инициализации фаз зарегистрированных отражений, которые выбираются случайным образом так, чтобы выполнялся закон Фриделя: $\phi(-\overrightarrow{H}) = -\phi(\overrightarrow{H})$, тогда получаем набор структурных факторов $F = F_{obs}(\overrightarrow{H}) \exp(i\phi(\overrightarrow{H}))$.

Запишем основные шаги алгоритма:

1. Для набора структурных факторов $F = F_{obs}(\overrightarrow{H})$ рассчитаем распределение электронной плотности $\rho(\overrightarrow{r})$, применив обратное преобразование Фурье.

2. Модифицируем электронную плотность $\rho(\overrightarrow{r})$ с помощью обращения заряда:

$$\rho(\overrightarrow{r}) \geq \delta : \mathrm{g} = \rho, \tag{1.31}$$

$$\rho(\overrightarrow{r}) < \delta : \mathrm{g} = -\rho, \tag{1.32}$$

где $\delta > 0$ — пороговое значение, параметр алгоритма.

3. Переходим в обратное пространство, применив преобразование Фурье к модифицированной электронной плотности $g(\overrightarrow{r})$. Получаем набор структурных факторов $G(\overrightarrow{H}) = |G(\overrightarrow{H})| \exp(i\psi(\overrightarrow{H}))$.

4. Ремодуляция: собираем новый набор структурных факторов F следующим образом: $F = F_{obs}(\overrightarrow{H}) \exp(i\psi(\overrightarrow{H}))$ — амплитуды равняются экспериментальным, фазы берутся из набора G, полученного в ходе обращения заряда. Вернуться к шагу 1.

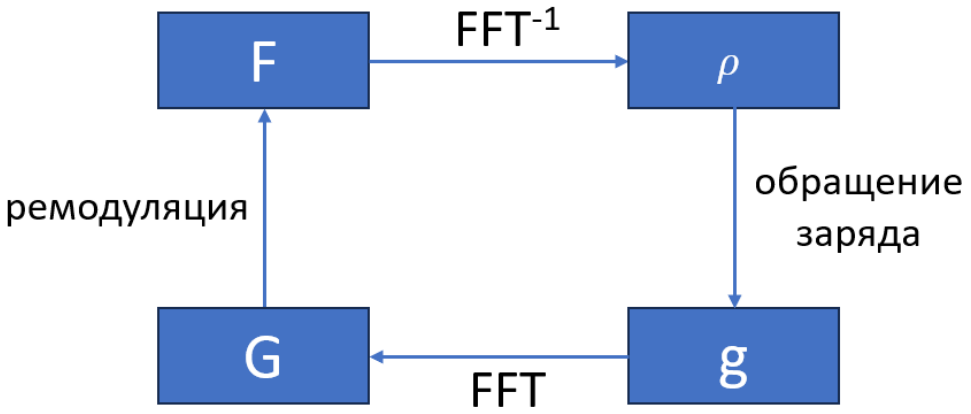

Рисунок 1.7. Схема итеративного цикла алгоритма charge–flipping

В итерационный процесс (рис. 1.7) не заложено условие окончания алгоритма, для мониторинга процесса можно следить за метриками, такими как R-фактор. Главными достоинствами метода являются его простота и отсутствие необходимости в дополнительных данных — метод позволил получить структуры множества соединений *ab initio* при наличии лишь экспериментальных дифракционных данных высокого разрешения, задавая лишь пороговое значение δ для электронной плотности.

В следующей работе [17] авторы модифицировали четвертый шаг алгоритма, явно используя отражения низкой интенсивности. Перед запуском алгоритма теперь дифракционные максимумы сортируются по наблюдаемой амплитуде и размечаются на две группы, к которым будут применяться разные преобразования в ходе вычислений — сильные и слабые отражения. Для сильных всё остается без изменений — фаза остаётся из шага 2, амплитуда структурного фактора берётся из экспериментальных данных. Фаза слабых отражений дополнительно смещается на $\Delta\phi$, а их амплитуда не изменяется на наблюдаемую. Это значит, что экспериментальные данные слабых отражений не используются в алгоритме, кроме начальной разметки на группы.

В модифицированном алгоритме charge–flipping появилось два дополнительных параметра: сдвиг фазы $\Delta\phi$ и доля отражений, которые можно считать слабыми, принятая

за 20%. Авторы показали, что оптимальное значение $\Delta\phi = \frac{\pi}{2}$ — поскольку при сдвиге на такую величину волны слабых отражений заменяются на ортогональные изначальным. Такое возмущение волн распространения слабых максимумов позволило на порядок увеличить долю успешных решений структур по сравнению с начальным алгоритмом.

Описанная модификация происходит в обратном пространстве, и в работе [18] авторы описывают подход к улучшению алгоритма с помощью изменения процедуры обработки функции электронной плотности. Charge–flipping в его классическом варианте можно рассматривать как алгоритм локального возмущения низких плотностей, то есть он никак не влияет на области с высокими значениями электронной плотности. Если в начале выполнения расчета будут получены некорректные высокие значения электронной плотности, то их практически невозможно исправить тривиальным обращением знака. В качестве аналога использования слабых отражений предлагается следующее улучшение в прямом пространстве: пусть в $(n+1)$–й цикл алгоритма:

$$\rho^{\mathrm{n}}(\overrightarrow{r}) < \delta : \mathrm{g}^{\mathrm{n+1}}(\overrightarrow{r}) = -\rho^{\mathrm{n}}(\overrightarrow{r}), \tag{1.33}$$

$$\rho^{\mathrm{n}}(\overrightarrow{r}) \geq \delta : \mathrm{g}^{\mathrm{n+1}} = \rho^{\mathrm{n}} + \beta(\rho^{\mathrm{n}} - \rho^{\mathrm{n-1}}) \tag{1.34}$$

где обычно $\beta \in [0.5, 1.0]$. Суть операции в следующем: измененная электронная плотность g^{n+1} будет вне интервала, сформированными ρ^{n-1}, ρ^{n}, со стороны последней. Процедура получила название 'flip–mem' и значительно улучшает классический алгоритм.

В этой же работе отмечено, что использование нормализованных структурных факторов $E(\overrightarrow{H}) = \frac{F_{obs}(\overrightarrow{H})}{[\sum_j f_j^2(\overrightarrow{H})]^{1/2}}$ вместо стандартных амплитуд позволяет увеличить скорость сходимости для больших структур. Кроме того, все модификации, которые были ранее упомянуты, подходят и для варианта с нормализованными амплитудами. Использование E уменьшает на порядок число итераций до сходимости по сравнению с F для всех вариантов алгоритма.

Метод обратного заряда позволяет в некоторых случаях решать макромолекулярные структуры *ab initio* [19]. Так, для таких структур, как лизоцим (2385 атомов, $d_{min} = 1.1$Å), алкогольдегидрогеназа(5866 атомов, $d_{min} = 1.0$Å), апамин (385 атомов, $d_{min} = 0.95$Å), фазы, рассчитанные алгоритмом, позволили корректно определить структуру. Авторы использовали метод в варианте с нормализованными структурными факторами $E(\overrightarrow{r})$, использованием слабых отражений (доля слабых отражений $\omega = 0.1$). Также авторы продемонстрировали, что charge–flipping является эффективным инструментов

для нахождения фаз для рентгенодифракционных данных низкого разрешения сложных структур с тяжелыми атомами, а такжюе с значимым анормальным рассеянием.

В работе [20] авторы представили свой вариант алгоритма обратного заряда с использованием нормализованных структурных факторов и новым способом возмущения, который применяет знание о формуле тангенсов, связывающей три отражения:

$$\tan\phi_{\mathrm{tf}}(\overrightarrow{H}) = \frac{\sum\limits_{\overrightarrow{K}} |\mathrm{E}(\overrightarrow{K})\mathrm{E}(\overrightarrow{H})\mathrm{E}(\overrightarrow{H}-\overrightarrow{K})| \sin(\phi(\overrightarrow{K})+\phi(\overrightarrow{H}-\overrightarrow{K}))}{\sum\limits_{\overrightarrow{K}} |\mathrm{E}(\overrightarrow{K})\mathrm{E}(\overrightarrow{H})\mathrm{E}(\overrightarrow{H}-\overrightarrow{K})| \cos(\phi(\overrightarrow{K})+\phi(\overrightarrow{H}-\overrightarrow{K}))} \tag{1.35}$$

Итоговый алгоритм (после случайной расстановки фаз для зарегистрированных отражений) выглядит следующим образом:

1. Обнулить 50% структурных факторов $E(\overrightarrow{H})$ с самыми низкими значениями амплитуд. Модуль остальных структурных факторов равен наблюдаемым.

2. Рассчитать электронную плотность $\rho(\overrightarrow{r})$ с помощью обратного преобразования Фурье.

3. Отнормировать электронную плотность, чтобы максимум функции был равен единице.

4. Определить граничное значение δ так, что 60% значений $\rho(\overrightarrow{r})$ лежат ниже порога.

5. Преобразовать плотность:

$$\rho < \delta : \mathrm{g} = -\rho \tag{1.36}$$

$$\rho \geq \delta : \mathrm{g} = \delta + (\rho - \delta)^{1/2} \tag{1.37}$$

6. Рассчитать набор структурных факторов G, применив преобразование Фурье к электронной плотности $g(\overrightarrow{r})$

7. Добавить к фазам с наибольшими значениями $E(\overrightarrow{H})$ долю разницы между фазами, полученной обратным зарядом ϕ_{cf} и рассчитанной по формуле тангенсов ϕ_{tf}:

$$\phi(\overrightarrow{H}) = \phi_{\mathrm{cf}}(\overrightarrow{H}) + \alpha(\overrightarrow{H})(\phi_{\mathrm{tf}}(\overrightarrow{H}) - \phi_{\mathrm{cf}}(\overrightarrow{H})), \tag{1.38}$$

где $\alpha(\overrightarrow{H})$ — параметр, рассчитываемый алгоритмом для каждого отражения, который отражает достоверность рассчитанных в ходе выполнения программы фаз. Вернуться к шагу 1.

Авторы продемонстрировали, что предложенный вариант метода позволяет решать за несколько минут структуры, которые ранее требовали сотни тысяч итераций. Использование формулы тангенсов для высокоинтенсивных отражений позволило повысить устойчивость и эффективность алгоритма, особенно при работе с низким разрешением данных (более 1Å), где классический вариант алгоритма чаще всего терпит неудачу.

В работе [?] был проведен сравнительный анализ метода обратного заряда с другими стандартными инструментами рутинного определения кристаллографических структур низкомолекулярных соединений. Для этого автор использовал charge–flipping в реализации программы SUPERFLIP [21] и традиционные прямые методы — SHELXS, SHELX86, SHELXD, SIR2004 [22]. Тестирование проводилось на наборе данных из 518 структур, включающих в себя органические, металлорганические и неорганические соединения. Метод обратного заряда показал эффективность, сравнимую с прямыми методами, в среднем процент успешного решения структур составляет более 92%. На рис. 1.8 представлены значения среднего R–фактора, достигаемого в ходе уточнения после успешного решения структур каждым методом. Также в статье показано, что charge–flipping является самым быстрым методом из рассматриваемых при той же эффективности решения. Таким образом, рассматриваемый алгоритм является полностью пригодным для определения низкомолекулярных структур в качестве рутинного метода.

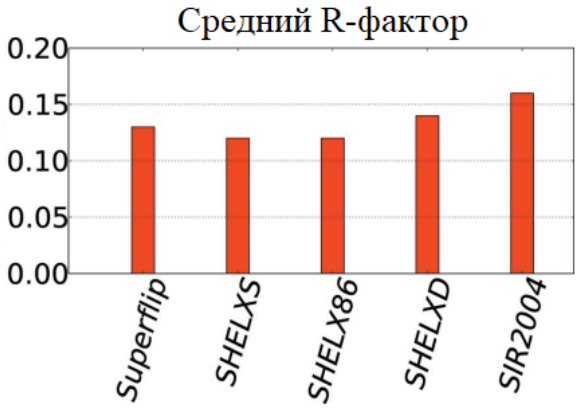

Рисунок 1.8. Средний R–фактор успешных решений различных методов решения

1.3.4 Метод VLD

Следующий рассматриваемый метод под названием Vive La Différence (VLD) является эффективным и универсальным методом для определения кристаллических структур, основанным на преобразовании электронной плотности и разностном Фурье–синтезе исходя из вероятностных и статических зависимостей между фазами и амплитудами [23]. Как и charge–flipping, метод является итеративным, но более комплексным, чем метод обратного заряда. Перед стартом расчёта фаз всем наблюдаемым амплитудам нормализованных структурных факторов E_{obs} присваиваются случайные фазы ϕ_m и рассчитывается электронная плотность кристаллической ячейки ρ_m. Здесь и далее величины, определяемые в ходе выполнения расчёта, назовём модельными. Алгоритм состоит из следующих шагов (схема представлена на рис. 1.9):

1. Модифицируем электронную плотность $\rho(\overrightarrow{r})$ следующим образом: обнуляем функцию во всех точках, кроме 2.5% с наибольшим значением плотности. По полученной измененной электронной плотности $g(\overrightarrow{r})$ рассчитываем модельные нормализованные структурные факторы E_m с помощью Фурье–преобразования.

2. Вычисляем разностную электронную плотность — синтез Фурье со следующими коэффициентами для каждого отражения:

$$\Delta \mathrm{E} = (\mathrm{mE_{obs}} - \mathrm{E_m}) \exp(\mathrm{i}\phi_\mathrm{m}) \tag{1.39}$$

   где m — коэффициент корреляции между модельными и реальными фазами.

3. Модифицируем полученную разностную плотность: обнуляем функцию во всех точках, кроме 4% с наибольшим значением по модулю. С помощью Фурье–преобразования рассчитываем разностные структурные факторы с амплитудами и фазами E_{diff}, ϕ_{diff}.

4. Используем полученные из разностной электронной плотности структурные факторы для расчёта коэффициента Фурье–синтеза (шаг 2), рассчитываем новую разностную плотность. Повторяем настоящий и предыдущий шаги β раз, итоговые параметры структурных факторов E'_{diff}, ϕ'_{diff}.

5. С помощью формулы тангенсов, которая была получена из вероятностного анализа и распределения фон Мизеса, можно рассчитать новые фазы ϕ_{calc}:

$$\tan(\theta) = \frac{\mathrm{E_m} \sin \phi_\mathrm{m} + \omega_\mathrm{diff} \mathrm{E_{diff}} \sin \phi_\mathrm{diff}}{\mathrm{E_m} \cos \phi_\mathrm{m} + \omega_\mathrm{diff} \mathrm{E_{diff}} \cos \phi_\mathrm{diff}} \tag{1.40}$$

где ω_m — параметр сходства модельной и реальной структур, отражающий среднюю корреляцию между моделью и структурой.

6. Экспериментальные E_{obs} и рассчитанные на предыдущем шаге фазы ϕ_{calc} используются для расчета электронной плотности, которая претерпевает γ циклов модификации, как в шаге 1.

7. Рассчитывается метрика $RESID$ — средняя ошибка между наблюдаемыми и рассчитанными амплитудами:

$$\mathrm{RESID} = \frac{\sum_{\vec{H}} |\mathrm{E_{obs}} - \mathrm{E_m}|}{\sum_{\vec{H}} \mathrm{E_{obs}}} \tag{1.41}$$

Если значение меньше 0.3 — можно считать, что решение достигнуто и цикл останавливается. Иначе — возвращение к шагу 1.

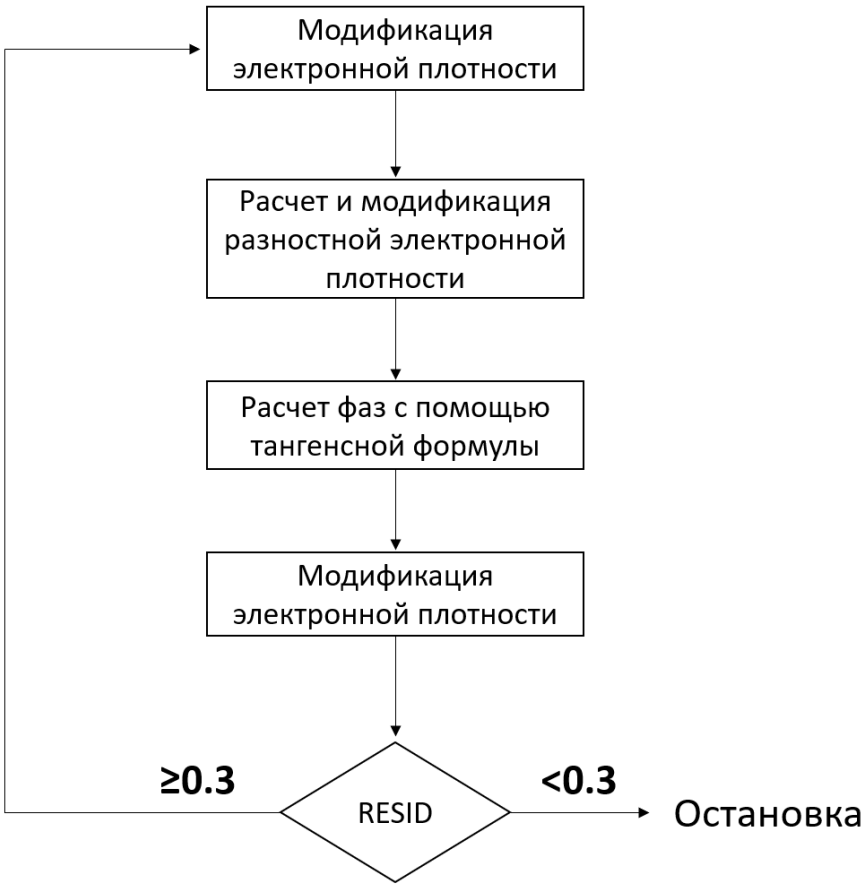

Рисунок 1.9. Схема итеративного цикла алгоритма VLD

Подробности процедур выбора настраиваемых параметров алгоритма и рассчета критических параметров, как ω_m и m, подробно описаны в оригинальной работе. Авторы протестировали предложенный метод на 33 низкомолекулярных структурах, разных по пространственной группе симметрии и наличию тяжелых атомов, из которых VLD успешно решил 30 структур за не более, чем 300 циклов. При дополнительных запусках с другими случайными фазами при инициализации модели привели к решению всего набора данных. Таким образом, показано, что метод является достаточно стабильным и обладает быстрой сходимостью (менее, чем 1 минута). Важно отметить, что в отличие от метода обратного заряда, метод решает структуру в правильной группе симметрии, а не P1.

В следующей работе [24] авторы развивают идеи первой работы и описывают существенные усовершенствования оригинального подхода к решению кристаллографической фазовой задачи. Ключевым усовершенствованием является внедрение процедуры RELAX, который позволяет автоматически переместить правильно ориентированную, но смещенную модель в корректное базисную позицию в пространственной группе. RELAX основана на наблюдении, что часто прямые методы определяет молекулярные фрагменты которые корректно ориентированы, но неправильно расположены. Использование процедуры существенно повысило процент успешно решённых структур, особенно в случае макромолекулярных соединений. Также в статье представлено значительное количество улучшений, связанные с улучшенной оценкой параметров качества модели, оптимизацией этапов модификаций электронной плотности (добавлена адаптивность модификации), а также дополнительные параметры качества решений для управления остановкой VLD. Тестирование на малых, средних молекулах и белках (разрешение до 1.2 Å) показало, что доработанный алгоритм решает структуры быстрее в среднем в 3–6 раз, определяет структуры белков за время, сравнимое с прямыми методами (SIR2011), а добавление RELAX позволило находить решения для крупных молекул именно благодаря этой процедуре.

В публикации [25] представлено множество дополнительных вариантов алгоритма VLD (4 протокола), которые ориентированы на решение проблемы фаз для среднемолекулярных и белковых соединений. Вариации метода различаются подходами к контролю метрик сходимости и внутренних параметров, а также способами комбинирования модели, разностной и обычной электронных плотностей (количество преобразований Фурье, использование или игнорирование экспериментальных данных, а также отказ от формулы тангенсов). В ходе оценки эффективности показано, что новые подходы уступают по успешности и скорости решения варианту с процедурой RELAX,

но могут быть в дальнейшем оптимизированы для *ab initio* решения макромолекулярных соединений.

1.3.5 Искусственный интеллект

Применение методов машинного обучения в рентгеновскодифракционных исследованиях — область, демонстрирующая стремительное развитие. Традиционно анализ дифракционных данных и решение обратной задачи в кристаллографии базировались на строго детерминированных алгоритмах, использующих априорные физические и химические знания о структуре вещества. Однако с ростом объёмов экспериментальных данных, увеличением доступной вычислительной мощности и успехами в смежных областях — таких как обработка изображений, анализ временных рядов и предсказание структур белков — возник интерес к использованию ИИ как инструмента для автоматизации и улучшения интерпретации дифракционных данных.

Первая публикация с решением фазовой проблемы методами глубокого обучения была опубликована почти год назад [26]. Для предсказания были выбраны центросимметричные структуры (группы симметрии $P2_1/c, C2/c, Pbca, Pnma, Pbcn, C2/m$), фазы дифракционных максимумов которых принимают два возможных значения — 0 и π [27]. Для обучения авторы сгенерировали 49 млн. синтетических структур, подавляющее большинство которых — органические и небольшая доля металлорганических. Нейронная сеть PhAI представляет собой бинарных классификатор из блоков трехмерных свёрток и многослойных перцептронов (рис. 1.10). Также в архитектуре сети представлен рециклинг фаз — данные прогоняются несколько раз через модель, чтобы добиться лучшего качества определения фаз отражений. Нейронная сеть использует только амплитуды структурных факторов, и достигает сопоставимого качества решения с методом обратного заряда. Нельзя не упомянуть, что модель способна решить фазовую проблему при разрешении рентгенодифракционных данных всего 2.0 Å, то есть используя лишь 10–20% объема данных, необходимых классическим методам. Авторы продемонстрировали высокую точность на экспериментальных данных, а также способность к предсказанию фаз для отсутствующих фаз (расширение фаз, phase extension). Таким образом, впервые был продемонстрирован потенциал машинного обучения для определения фаз. Недостатком модели является то, что она заточена под центросимметричные структуры. Исключительный интерес же представляют нецентросимметричные кристаллы, к

которым нельзя будет применить подход с бинарным классификатором, описанным в работе.

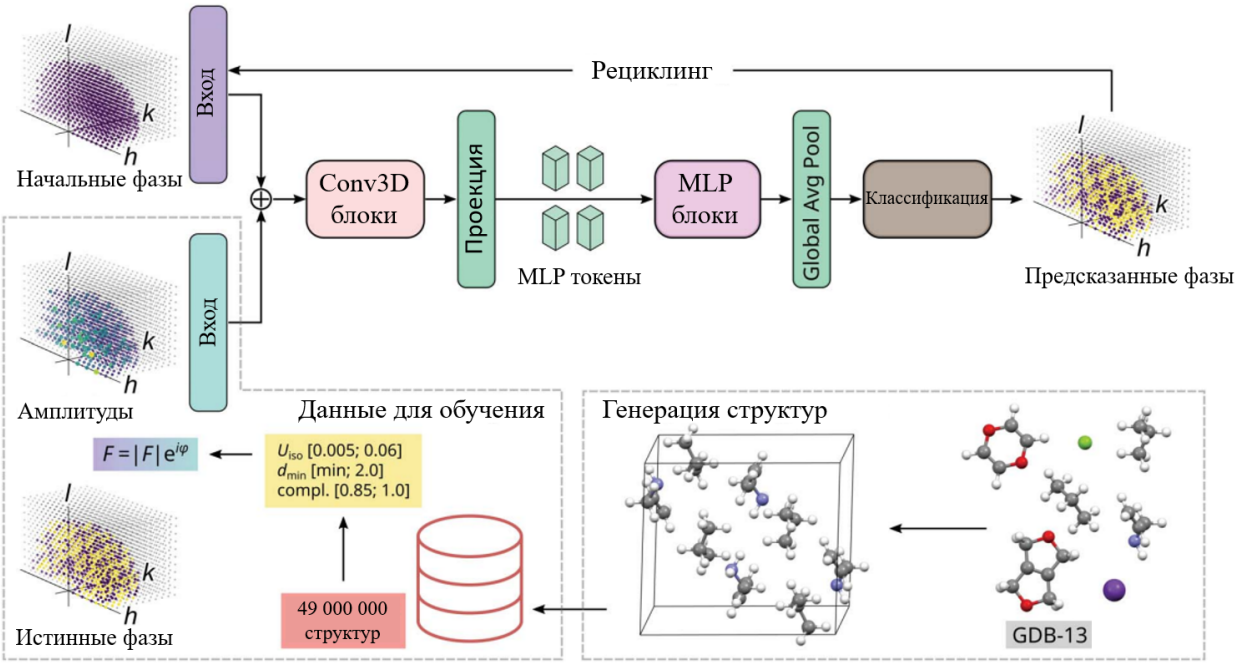

Рисунок 1.10. Схема нейронной сети PhAI для решения проблемы фаз [26]

Вдохновившись описанным исследованием, авторы [28] предлагают адаптацию его идеи для нецентросимметричных структур. В работе предложен метод phase seeding, который позволяет переформулировать задачу из регрессионной с определением непрерывных величин от 0 до 2π в задачу мультиклассовой классификации. Сначала весь диапазон фаз дискретизируется в ограниченный набор промежутков, в каждом из которых фазы отражений заменяют на одинаковое значение (рис. 1.11). Затем фаза каждого отражения случайным образом относится к одному из подмножеств. Авторы продемонстрировали на различных структурах, что для решения структур детерминированными методами достаточно будет предсказать с помощью методов искусственного интеллекта одно из дискретных значений фаз для каждого дифракционного максимума. Также предложено для решения фазовой проблемы методами ИИ использовать отражения с наибольшими значениями амплитуды нормализованного структурного фактора E.

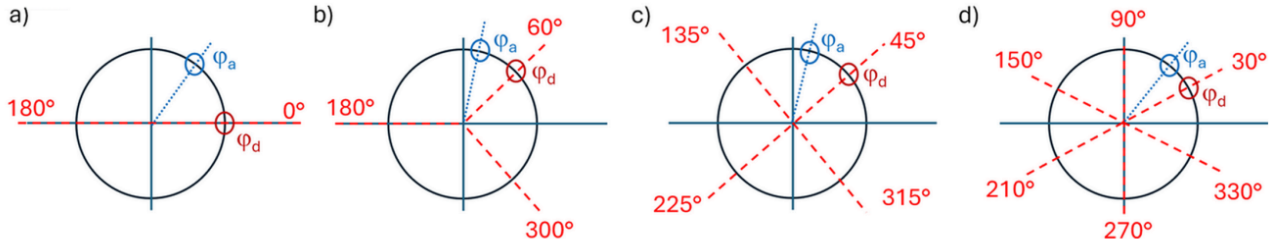

Рисунок 1.11. Дискретизация значений фаз: на 2 промежутка (a), на 3 промежутка (b), на 4 промежутка (c), на 6 промежутков (d). Истинное значение фазы — ϕ_a (синий круг), дискретизованное — ϕ_d (красный круг) [28]

Также были совершены попытки предсказать структуры белков напрямую из экспериментальных данных, минуя этап с определением фазовой информации. Так, была предложена модель RecCrysFormer, позволяющая напрямую предсказывать электронную плотность макромолекулярных соединений на основе карт Паттерсона и известной плотности отдельных белковых фрагментов [29]. Архитектура сети представляет собой трехмерные сверточные слои, переводящие карту Паттерсона и плотность в пространство признаков, которые затем подаются на вход в трансформер, выходные данные которого обрабатываются с помощью слоёв многослойного перцептрона и трехмерных свёрток для получения карт плотности всего белка. Примечательно, что в слое трансформера реализовано одностороннее внимание, поэтому только токены функции Паттерсона "смотрят" на токены фрагментов соединения. Авторы смогли добиться решения 93% тестовых структур, хотя и признают, что выбранные ими соединения меньше реальных белков.

Были найдены решения фазовой проблемы с помощью методов глубокого обучения в области физики, а именно в рамках метода когерентной безлинзовой микроскопии [30]. В обзорной статье выделены 3 подхода (рис. 1.12) — DL-pre-processing, в котором обученная модель повышает разрешение дифракционных данных, после чего проблема фаз решается классическими детерминированными методами; DL-in-processing, в рамках которого из экспериментальных интенсивности с помощью нейронной сети рассчитывают фазы напрямую; DL-post-processing, в котором уточняются зашумленные приближенные фазы, полученные из исходных интенсивностей. Несмотря на то, что для рентгеновской кристаллографии нельзя напрямую использовать уже готовые методы микроскопии, поскольку данные сильно различаются разрешением, эти идеи можно адаптировать. Так, существующее решение PhAI [26] можно отнести к DL-in-processing.

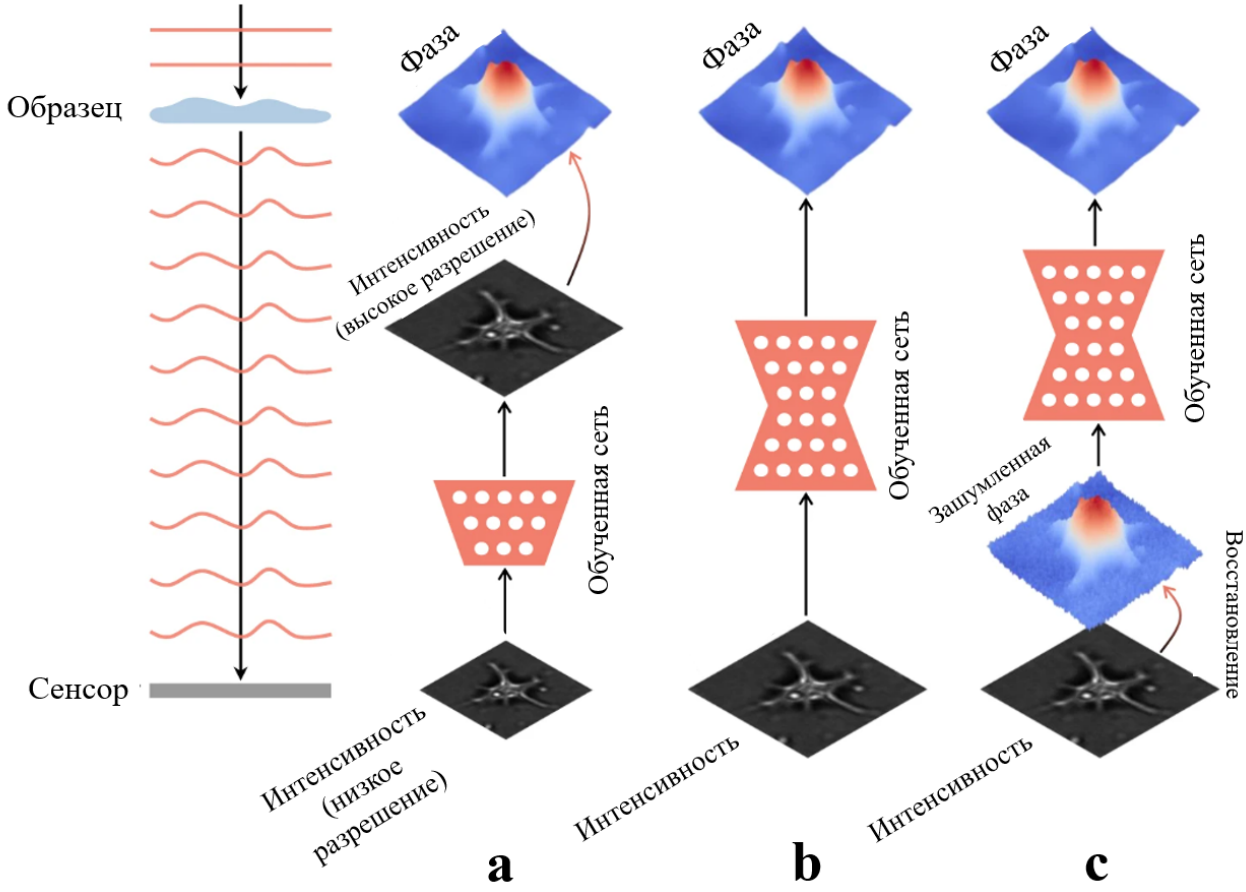

Рисунок 1.12. Подходы к восстановлению фаз в микроскопии: DL-pre-processing (a), DL-in-processing (b), DL-post-processing (c) [30]

1.4 Заключение

Проблема фаз является фундаментальной задачей рентгенодифракционных исследований кристаллов, без решения которой невозможно определять структуру кристаллического соединения. На сегодняшний день существует множество методов, позволяющих преодолеть отсутствие фазовой информации, но они имеют ряд ограничений, связанных с требованием к достаточно высокому разрешению дифракционных данных. Данное требование может не выполняться в рамках белковой кристаллографии, где фазовая проблема исключительно редко решается рутинными методами. Для определения структур макромолекулярных соединений требуется дополнительную информацию — знание о структуре белка с той же аминокислотной последовательностью или результаты рентгенодифракционных экспериментов той же структуры с добавлением тяжелых атомов. Поэтому решение проблемы фаз является особенно актуальной задачей для исследователей белковых структур.

Однако инструменты на основе методов глубокого обучения способны преодолеть

ограничения традиционных подходов, поскольку уже продемонстрирован их потенциал к определению кристаллических структур на основе ограниченных рентгенодифракционных данных с низким разрешением. По мере развития исследований в этой области, вероятно, алгоритмы искусственного интеллекта станут незаменимым инструментом в области кристаллографии, облегчая решение сложных структур.

Таким образом, целью данной работы является разработка метода решения проблемы фаз с помощью искусственного интеллекта.

В рамках данной цели были выделены следующие задачи:

1. Разработать программное обеспечение, позволяющее создать набор синтетических рентгенодифракционных данных для обучения нейронных сетей;
2. Разработать автоматизированный конвейер, с помощью которого можно проводить воспроизводимые численные эксперименты по обучению и тестированию моделей, а также решать проблему фаз;
3. Обучить подходящие модели глубокого обучения;
4. Произвести анализ полученных моделей на эффективность решения фазовой проблемы.

2 Методика решения

2.1 Подход

В работе было предложено предсказывать амплитуды структурных факторов дифракционных максимумов, которые нельзя получить из эксперимента, по известным из того же эксперимента. После предсказания достаточного количества отражений, разрешения данных должно хватить для определения фаз и расчета электронной плотности одним из рутинных ab initio методов, в качестве которого был выбран метод, реализованный в комплексе программ SHELXTL PLUS [22] (схема представлена на рис. 2.1). Разрешение — минимальное межплоскостное расстояние из набора дифракционных максимумов структуры [1].

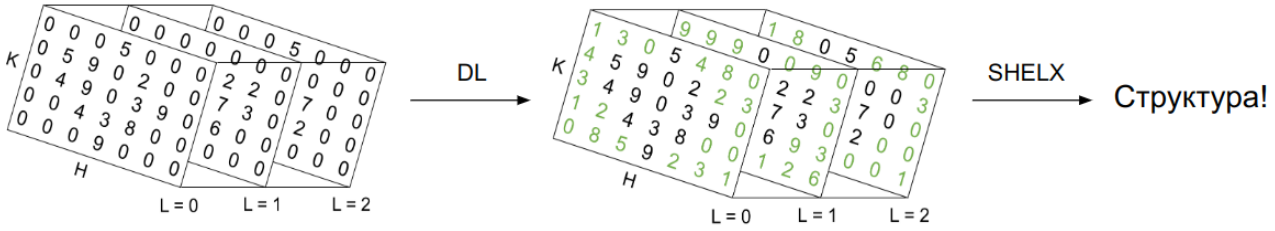

Рисунок 2.1. Схема решения проблемы фаз с помощью методов глубокого обучения (DL). Дифракционные картины (тензоры отражений $\mathbf{F}$) представлены в виде параллелепипедов

Дифракционные отражения являются точками обратного пространства, каждое из них можно однозначно описать индексами Миллера (h, k, l). Тогда дифракционную картину можно описать трехмерным тензором $\mathbf{F} \in \mathbb{R}^{H \times K \times L}$, в котором записаны амплитуды каждого отражения: $\mathbf{F}[h, k, l] = F(h, k, l)$. В точках, где не зарегистрированы дифракционные максимумы, в тензоре записаны нули. Таким образом, задача сводится к восстановлению трехмерного тензора. Также значения в каждом тензоре были отнормированы в диапазон 0–1. Получение результата (inference) моделей глубокого обучения должен выглядеть следующим образом: на вход подается тензор с рентгенодифракционными экспериментальными данными, на выходе должен быть тензор с амплитудами дополнительных отражений. В ходе обучения планируется научить модель восстанавливать тензор отражений по данным органических молекул.

В работе также была проведена обработка после получения результата моделью (постпроцессинг), не входящая в обучение и включающая в себя учёт систематических погасаний — "зануления" некоторых значений структурных факторов, что определяется симметрией структуры; в ходе обучения происходит явное восстановления части

тензора, которую не нужно предсказывать. Эффективность предсказания обученных моделей глубокого обучения проверялась на тестовой части синтетического датасета, а также тестовой части рентгенодифракционных данных моноклинных структур из CSD (Кембриджской Базы Структурных данных).

2.2 Рентгенодифракционные данные

Было разработано программное обеспечение, позволяющее генерировать случайные структуры и рассчитывать для них рентгенодифракционные данные (`github.com/blackwood168/xrd_simulator`). С помощью открытой библиотеки на языке Python CCTBX (Computational Crystallography Toolbox) [31] создаются кристаллические решетки, в которых случайным образом с учётом симметрии расставляются случайные атомы. Для получаемых синтетических структур реализованы расчёт дифракционной картины — индексов и структурных факторов отражений, вычисление порошковой дифрактограммы (реализована профильная функция Псевдо–Войдта [32] и осевая расходимость рентгеновского пучка, рис. 2.2), а также карты Паттерсона. Нужный расчет выбирает пользователь исходя из своей текущей задачи. Также генератор поддерживает последующий расчет данных рентгеновской порошковой дифракции. Созданное ПО может быть полезно для решения прикладных задач рентгенодифракционных исследований кристаллов с помощью методов машинного обучения.

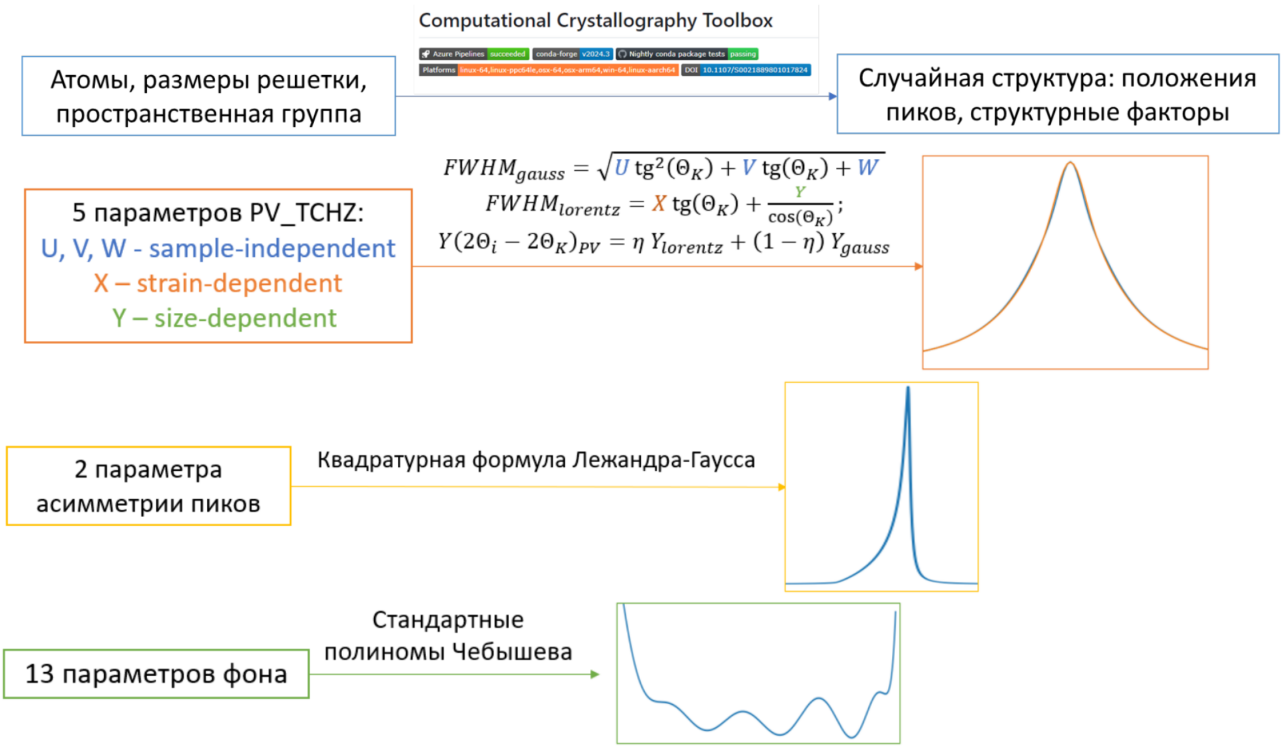

Рисунок 2.2. Схема генерации случайных дифрактограмм в разработанной программе генерации

Начальное обучение моделей глубокого обучения было решено проводить на синтетических структурах, которые были получены с помощью созданного генератора. Так как общее количество симметрично независимых отражений зависит от класса Лауэ, было решено сосредоточиться на моноклинных структурах, поскольку моноклинные группы симметрии являются одними из наиболее распространенных для белковых структур в базе данных белков (`rcsb.org/stats/distribution-space-group`). При генерации структур её основные параметры (группа симметрии, типы атомов, их количество) определяются случайным образом (случайное сэмплирование) из следующих значений:

- группы симметрии: $P2_1$, C2;
- атомы: C, N, O, Cl, Br;
- число симметрично независимых атомов в ячейке: 10–30;

При выполнении численных экспериментов со структурными факторами было решено работать с их амплитудами. Типичные распределения этих данных для сгенерированных структур представлены на рис. 2.3. Как можно заметить, распределение амплитуд более пологое, чем интенсивностей, поэтому именно амплитуды были выбраны для решения задачи.

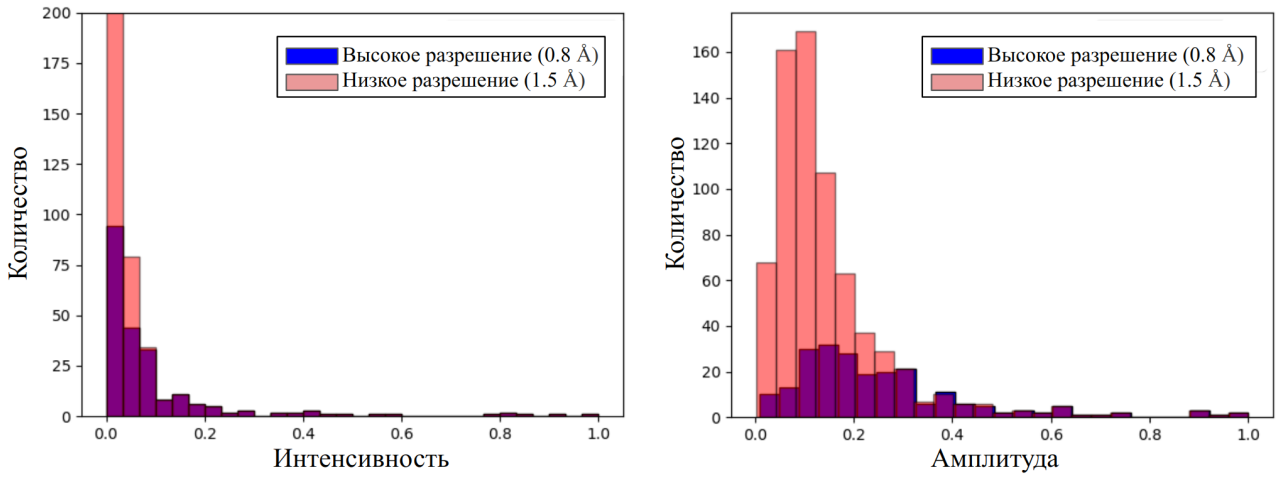

Рисунок 2.3. Типичные распределения интенсивностей (слева) и амплитуд (справа) дифракционной картины

600.000 полученных структур были разделены на наборы следующих размеров: 400.000, 100.000 и 100.000 для тренировочной, валидационной и тестовой выборок, соответственно. Из этих структур были собраны следующие наборы данных, с помощью которых будет проведено обучение моделей:

1. $D_1 = \left\{\mathbf{F}(\mathbf{1.5}\text{Å})_\mathbf{i}, \mathbf{F}(\mathbf{0.8}\text{Å})_\mathbf{i}\right\}_{i=1}^{n}$
2. $D_2 = \left\{\mathbf{F}(\mathbf{1.2}\text{Å})_\mathbf{i}, \mathbf{F}(\mathbf{1.0}\text{Å})_\mathbf{i}\right\}_{i=1}^{n}$
3. $D_3 = \left\{\mathbf{E}(\mathbf{1.2}\text{Å})_\mathbf{i}, \mathbf{E}(\mathbf{1.0}\text{Å})_\mathbf{i}\right\}_{i=1}^{n}$

Также в работе использовались реальные моноклинные молекулярные структуры малых молекул из Кембриджского Банка Структурных Данных [33], для которых были расчитаны дифракционные отражения. 10.000 структур использовались для дообучения моделей на реальных структурах, 2000 — были отложены для тестирования. На рис. 2.4 представлена зависимость среднего количества отражений от выбранного разрешения дифракционной картины. При повышении разрешения с 1.5 Åдо 0.8 Å(набор D_1) требуется с помощью нейронной сети увеличить число максимумов более чем в 5 раз, поэтому был также собран набор D_2, для которого потребуется расширить дифракционную картину в 1.7 раз.

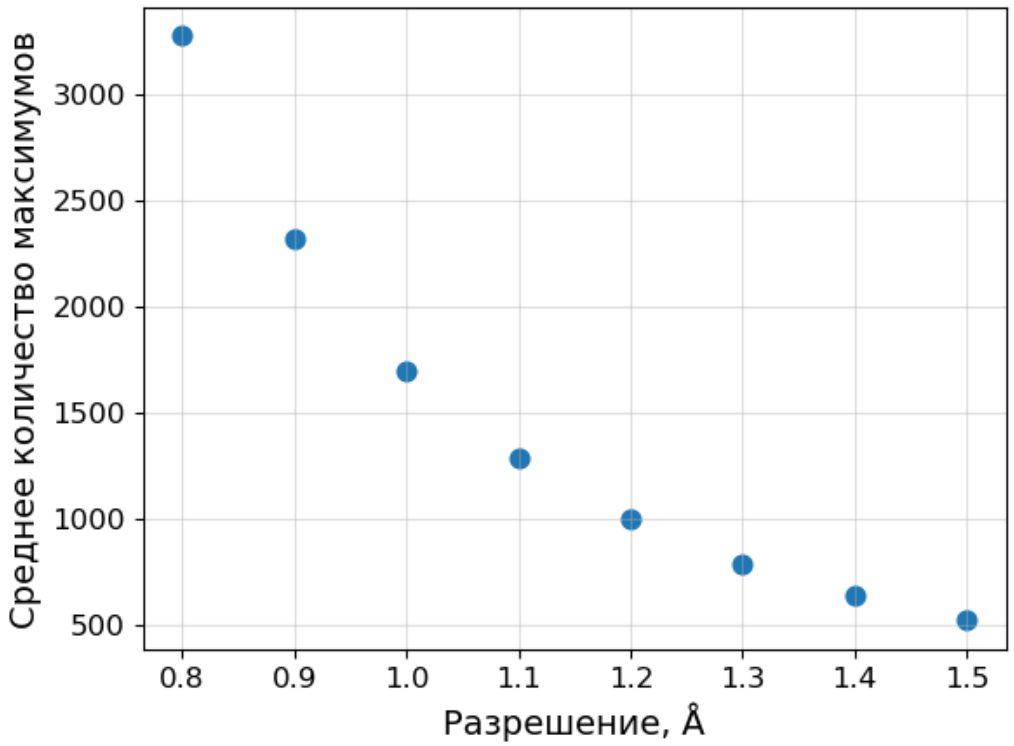

Рисунок 2.4. Зависимость среднего количества дифракционных отражений моноклинных ($P2_1$, C2) структур из CSD от разрешения

Рассчитанный набор амплитуд нормализованных структурных факторов D_3 использован из соображения, что нормализованные структурные факторы E лишены явной зависимости амплитуды от $\frac{\sin\theta}{\lambda}$, где Θ — угол отражения, λ — длина волны рентгеновского излучения. Это приводит к тому, что распределение $|E|$ менее смещено в сторону высокоинтенсивных максимумов по сравнению с $|F|$ (рис. 2.5), что делает данный набор данных более перспективным для решения с помощью машинного обучения, поскольку требуется предсказывать низкоинтенсивные отражения.

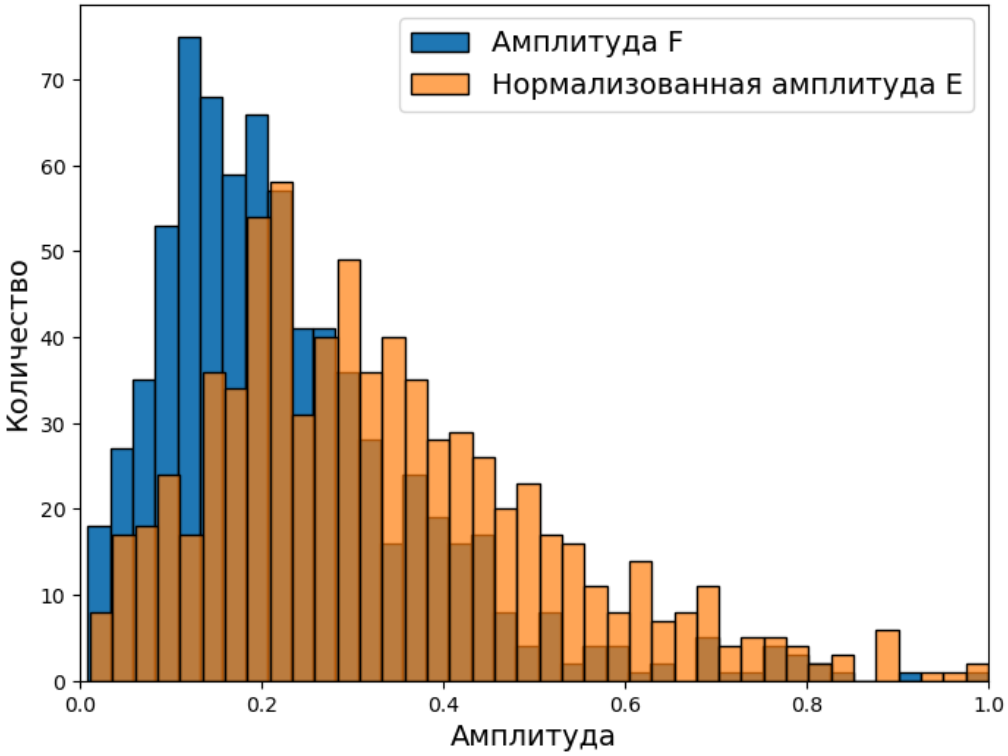

Рисунок 2.5. Типичные распределения амплитуд структурного фактора F и нормализованного структурного фактора E (разрешение 1.0 Å)

Размер полученных тензоров составляет (26, 18, 23) для набора данных D_1 и (23, 16, 21) для наборов D_2, D_3 — в тензоры таких размерностей помещаются все отражения для самой большой моноклинной структуры из синтетических данных (для разрешения 0.8 Åи 1.0 Å). Также значения в каждом тензоре были отнормированы в диапазон 0–1.

2.3 Модели машинного обучения

Обозначим модель машинного обучения как $g(\Theta, \mathbf{F})$, которая задаётся параметрами Θ и принимает на вход тензор рентгенодифракционных отражений $\mathbf{F}$. Модель рассчитывает дополненный тензор отражений $\mathbf{F^g_{high}} = g(\Theta, \mathbf{F_{low}})$, который должен быть максимально близок к реальной дифракционной картине высокого разрешения $\mathbf{F_{high}}$. Задача является регрессионной, процесс обучения на наборе данных, состоящего из пар $\{\mathbf{F_{low,i}}, \mathbf{F_{high,i}}\}_{i=1}^{n}$ стремится оптимизировать параметры Θ, минимизируя целевую функцию, в качестве которой выбрана среднеквадратичная ошибка (MSE):

$$\Theta^* = \arg\min_{\Theta} \left[\mathrm{MSE}(\Theta) := \frac{1}{\mathrm{n}} \sum_{\mathrm{i}=1}^{\mathrm{n}} \|\mathrm{g}(\Theta, \mathbf{F_{low}}) - \mathbf{F_{high}}\|_{\mathrm{F}}^2 \right] \tag{2.1}$$

Как уже было отмечено, в качестве функции потерь была выбрана среднеквадратичная ошибка, минимизация которой должна приводить к восстановлению тензора

рентгеновских отражений. В качестве метрики для оценки эффективности моделей также был использован R-фактор:

$$\mathrm{R} = \frac{\sum\limits_{\mathrm{h,k,l}} |\mathrm{F_{obs}} - \mathrm{F_{calc}}|}{\sum\limits_{\mathrm{h,k,l}} |\mathrm{F_{obs}}|}, \tag{2.2}$$

где $|F_{obs}|$ — экспериментальные структурные факторы, $|F_{calc}|$ — рассчитанные по модели структурные факторы. R-фактор является общепринятым стандартом в кристаллографическом сообществе для оценки качества структурных моделей. Нулевое значение R-фактора отвечает идеальному соответствию между данными модельной структуры и экспериментальными данными. В качестве экспериментальных структурных факторов в работе были использованы структурные факторы, рассчитанные по структуре, а F_{calc} — результат вычислений с помощью нейронных сетей.

Также в качестве метрики сравнения изображений рассматривался индекс структурного сходства SSIM [34], который зарекомендовал себя как хорошая метрика для оценки восстановления и увеличения разрешения изображений, однако эксперименты с ним в качестве добавки к функции потери привели к более низкому качеству восстановления модели с точки зрения R-фактора. Данный результат можно объяснить отсутствием локальной связанности наших данных.

В качестве базовой модели (baseline) была обучена модель UNet [35], адаптированная для трехмерных данных (рис. 2.6). Данная модель выбрана, потому что она хорошо себя проявляет в простейших задачах увеличения разрешения изображения за счет генерации недостающих пикселей на входных данных низкого разрешения (Super Resolution).

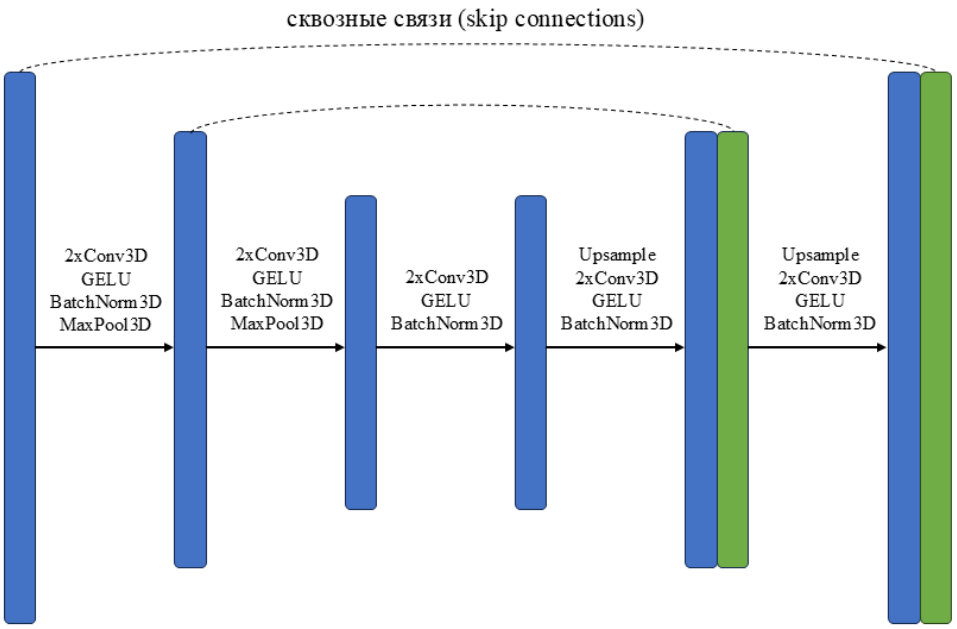

Рисунок 2.6. Схемы архитектуры модели UNet

Также была разработана и обучена модель на основе UNet с улучшенными слоями, содержащими Фурье-преобразование FFT_UNet (рис. 2.7). Данный подход был продемонстрирован [36] при работе с дифракционными данными и он является многообещающим и для нашей задачи, поскольку при переходе в прямое пространство наши данные являются локально связанными.

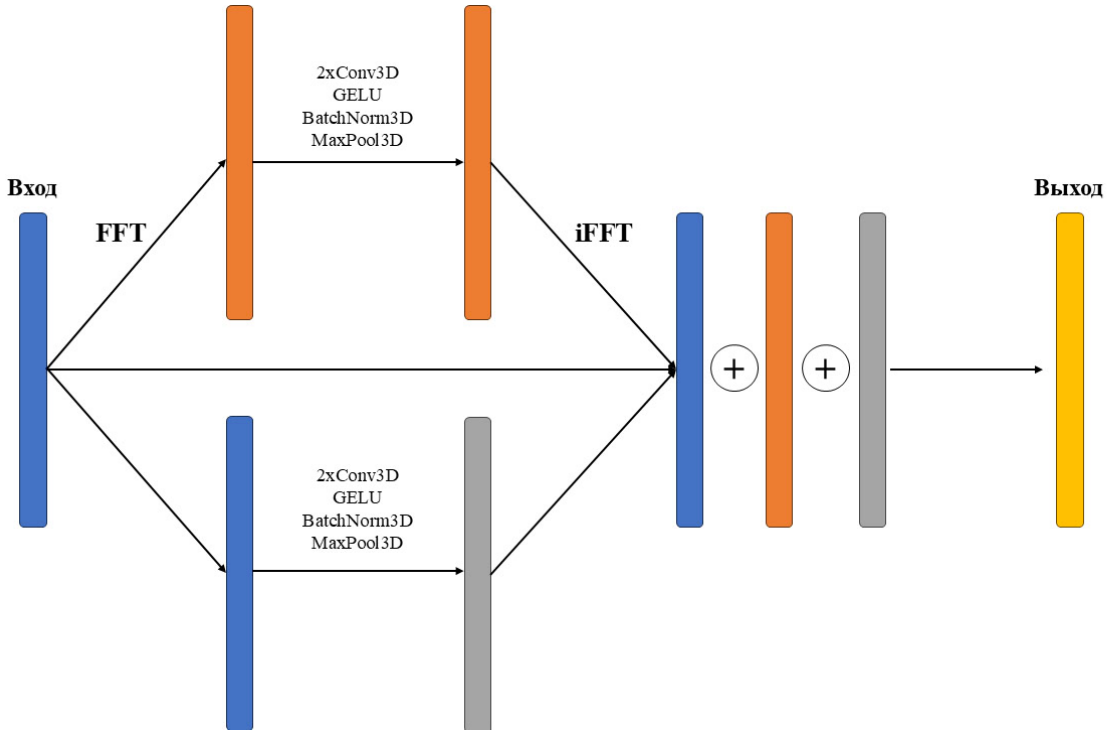

Рисунок 2.7. Схема слоёв с преобразованием Фурье

Поскольку тензоры отражений не являются локально связанными, актуально

использование механизма внимания. Он позволит модели находить связи между дальними отражениями. Так, был разработан трансформер XRD_Transformer для нашей задачи (рис. 2.8).

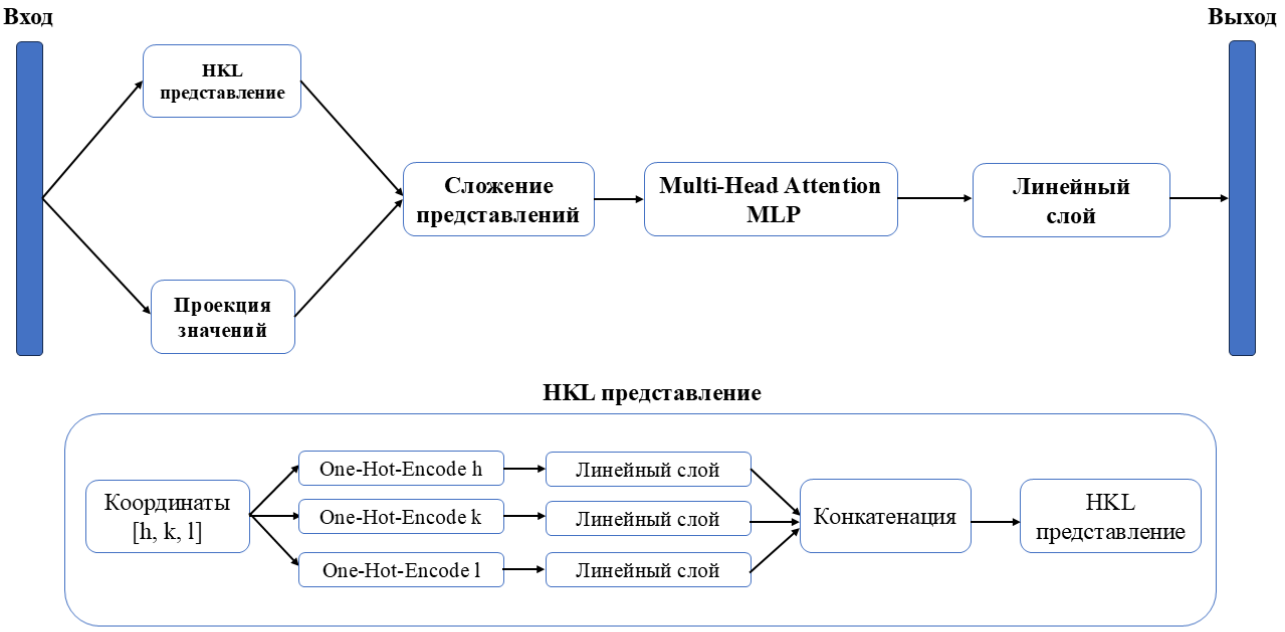

Рисунок 2.8. Схема архитектуры XRD_Transformer

В модели формируется единое векторное представление (embedding) из индексов Миллера и проекции значений амплитуды отражений, затем он проходит через 5 слоев трансформера, которые состоят из слоев многоголового внимания (Multi-Head Attention) и блоков многослойного перцептрона (MLP). Затем после нормализации (LayerNorm) и обратного проецирования в исходное пространство получается восстановленный тензор дифракционной картины.

Так как нам известны все возможные значения индексов Миллера (h, k, l) для структур выбранных размеров с заданным разрешением, HKL представления (рис. 2.8) формируется следующим образом: индексы кодируются с помощью унитарного кодирования (One-Hot Encoding), после чего каждый вектор с помощью обучаемого линейного слоя проецируется в вектор размерности $\frac{embed\_dim}{3}$ после конкатенации размерность полученного векторного представления составляет $embed\_dim$. Также в модели реализована возможность получения представления через полносвязный слой, который проецирует позицию (h, k, l) сразу в вектор, однако она не использовалась при обучении модели, так как первый способ является более физическим для нашей задачи, поскольку мы используем только симметрически независимые отражения.

3 Результаты и обсуждение

В рамках данной работы был разработан автоматизированный конвейер (пайплайн), позволяющий проводить воспроизводимые эксперименты по решению проблемы фаз с помощью предложенного подхода (`github.com/blackwood168/xrd_phase_ml`, рис. 3.1). В нем реализовано обучение и тестирование моделей, а также получение результатов (inference) на реальных массивах данных и структурах кристаллических соединений. В репозитории присутствуют маленькие наборы данных из сгенерированных и реальных структур малых органических молекул, также там представлены веса обученных в работе моделей. Воспроизводимость обучения обеспечивает фиксирование начальных значений генераторов случайных состояний.

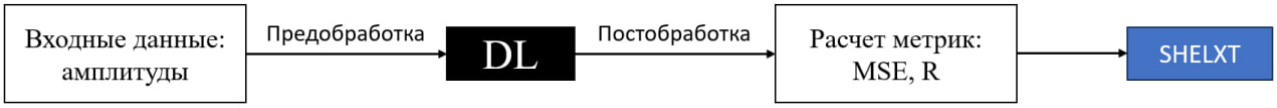

Рисунок 3.1. Схема разработанного автоматизированного контейнера, использующего модели глубокого обучения (DL) для предсказывания амплитуд структурных факторов

3.1 Решение для структурных факторов (разрешение 1.5Å)

3.1.1 Результаты обучения

Было проведено обучение на синтетических данных (набор D_1) и последующее дообучение на рентгенодифракционных данных реальных структур из Кембриджской Базы Структурных Данных (CSD). В ходе обучения модель повышала разрешение с 1.5 до 0.8 Å. Сравнение R–фактора для моделей до и после дообучения представлено в таблице 3.1. После дообучение на реальных данных точность предсказания моделей увеличивается минимум на треть, однако теряется точность на синтетических данных, кроме UNet с Фурье-преобразованием, который лишь прибавляет в точности на сгенерированных структурах после дообучения на реальных. Лучшую точность имеет модель UNet_FFT (R=0.336), от которой немного отстает трансформер. В сводной таблице 3.2 представлены значения метрик на реальных данных финальных моделей. Таким образом, UNet_FFT занимает меньше видеопамяти, работает быстрее и достигает лучшей метрики на тестовых реальных данных.

Таблица 3.1. Результаты обучения и эффективность дообучения моделей, повышающих разрешение с 1.5 до 0.8 Å(синт. — синтетические тестовые данные)

Модель	**Метрика**	**Синт.**	**CSD**
UNet	До, R	0.477	0.590
	После, R	0.632	0.393
	Δ, %	-32.6	33.4
FFT_UNet	До, R	0.726	0.646
	После, R	0.619	**0.336**
	Δ, %	14.7	48.0
XRD_Transformer	До, R	0.346	0.581
	После, R	0.615	0.358
	Δ, %	-77.9	38.4

Таблица 3.2. Значения метрик на тестовом реальном наборе данных моделей, повышающих разрешение с 1.5 до 0.8 Åпосле дообучения

Model / **Metric**	**UNet**	**FFT_UNet**	**XRD_Transformer**
MSE·10^{-3}	1,39	1,20	1,31
R	0,393	**0,336**	0,358

3.1.2 Определение структур

Модели демонстрируют достаточно большие значения R–фактора на реальных моноклинных структурах из Кембриджского Банка Структурных данных, качество восстановления тензора отражений даже после дообучения недостаточно для решения структуры с помощью программы SHELXT. На рис. 3.2 представлены характерные сечения тензора отражений для реальной структуры, где R-фактор восстановленного тензора с помощью FFT_UNet составляет 0.343, но SHELXT с помощью метода Паттерсона не смог определить кристаллическую структуру по таким данным.

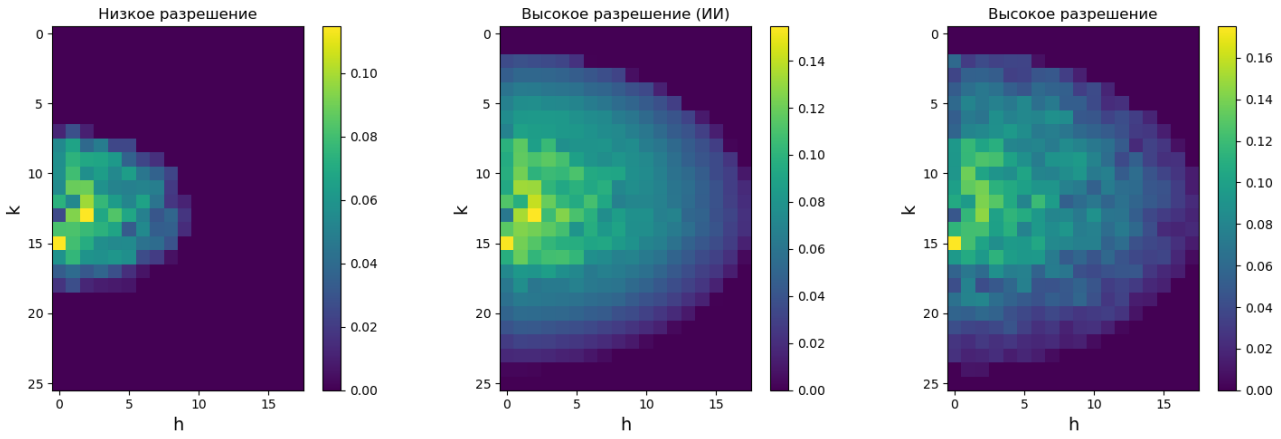

Рисунок 3.2. Типичное восстановление дифракционной картины реальной структуры, $R = 0.343$ (с 1.5 до 0.8 Å, усреднено по индексу l)

Можно заметить, что изображение восстановленного тензора отражений по сравнению с истинным размытое, что можно объяснить результатом минимизации среднеквадратичной ошибки. Поскольку дифракционная картина не является локально связанной, как изображения, соседние пиксели могут сильно различаться. Модели машинного обучения не могут точно предсказать значение амплитуды структурного фактора в каждом пикселе, но справляются с вычислением среднего значения. Стоит отметить, что несмотря на низкую точность решения поставленной задачи регрессии, модель достаточно точно определяет границы дифракционной картины, то есть какие отражения будут ненулевые в тензоре отражений более высокого разрешения.

3.2 Решение для структурных факторов (разрешение 1.2Å)

3.2.1 Результаты обучения

Было проведено обучение на синтетических данных (набор D_2) и последующее дообучение на реальных моделей, повышающих разрешение дифракционной картины с 1.2 до 1.0 Å. Сравнение R–фактора для моделей до и после дообучения представлено в таблице 3.3. После дообучение на настоящих органических структурах из CSD точность предсказания дифракционных максимумов для структур увеличивается больше всего для UNet с Фурье–преобразованием — более чем на треть. Лучшую точность показывает UNet_FFT (R=0.336), от которой немного отстает UNet. На этот раз трансформер показывает худший результат на реальных структурах. В сводной таблице 3.4 представлены значения метрик на реальных данных финальных моделей. Таким образом, UNet_FFT снова становится наиболее точной моделью машинного обучения для увеличения разрешения дифракционной картины. В дальнейшем для проблемы фаз будет использоваться именно эта модель.

Таблица 3.3. Результаты обучения и эффективность дообучения моделей, повышающих разрешение с 1.2 до 1.0 Å(синт. — синтетические тестовые данные)

Модель	**Метрика**	**Синт.**	**CSD**
UNet	До, R	0.119	0.255
	После, R	0.109	0.207
	Δ, %	8.4	18.8
FFT_UNet	До, R	0.107	0.234
	После, R	0.319	**0.152**
	Δ, %	-198.1	35.0
XRD_Transformer	До, R	0.236	0.337
	После, R	0.104	0.267
	Δ, %	55.9	26.2

Таблица 3.4. Значения метрик на тестовом реальном наборе данных моделей, повышающих разрешение с 1.2 до 1.0 Åпосле дообучения

Метрика \ Модель	**UNet**	**FFT_UNet**	**XRD_Transformer**
MSE·10^{-3}	0.684	0.489	0.199
R	0.207	**0.152**	0.267

3.2.2 Определение структур

Несмотря на низкие значения R–факторов, посчитанных при сравнении дифракционной картины, восстановленой нейронной сетью FFT_UNet, и данных, рассчитанных по реальным структурам (рис. 3.3), качества данных для определения структуры методом Паттерсона с помощью программного обеспечения SHELXT все еще не хватает. Так, лишь для 83% низкомолекулярных органических структур из CSD, выбранных в качестве тестовой выборки, метод нашёл какое-то подходящее решение, средний R–фактор по таким структурам равен 0.49. Структуры, чьё решение имеет R–фактор не более 0.25, составляют 10% всей тестовой выборки. Для сравнения — если попытаться решить структуры с помощью рентгенодифракционных данных, рассчитанных по структуре, с разрешением 1.0 Å, SHELXT выдаёт решение для 97% структур со средним R–фактором равным 0.13. Результаты представлены в таблице 3.5.

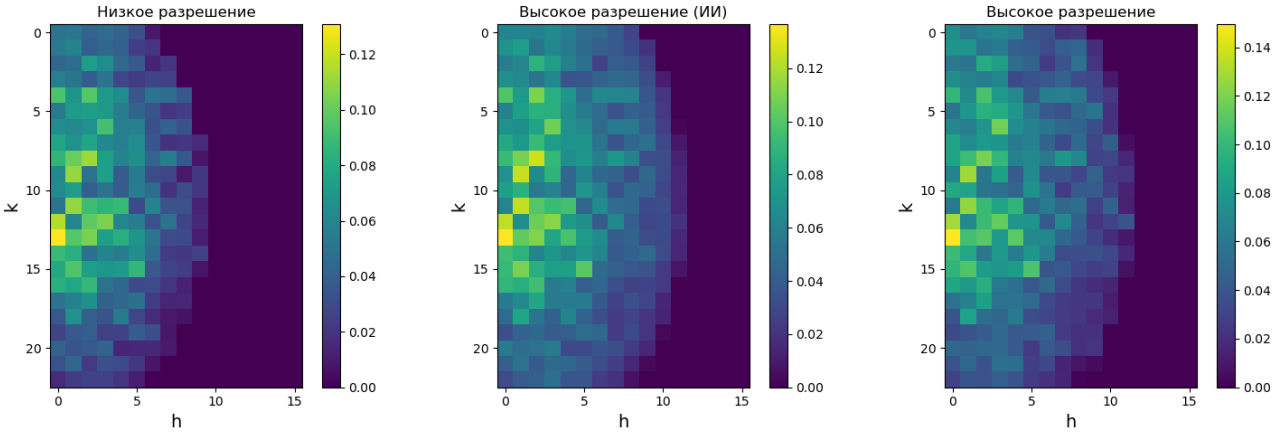

Рисунок 3.3. Типичное восстановление дифракционной картины реальной структуры, $R = 0.133$ (с 1.2 до 1.0 Å, усреднено по индексу l)

Таблица 3.5. Данные по эффективности определения решения по рентгенодифракционным данным с разрешением 1.0Å, полученных с помощью ИИ, и рассчитанным по структуре

Данные / **Показатель**	**Восстановленные (ИИ)**	**Рассчитанные**
Наличие решений, %	83	97
Наличие решений с $R \leq 0.25$, %	10	94
Средний R	0.49	0.13

3.3 Решение для нормализованных структурных факторов

3.3.1 Результаты обучения

3.3.2 Определение структур

3.4 Анализ моделей

Для более глубокого понимания трансформера был проведен анализ внутренней работы модели. Особый интерес представляют карты внимания (рис. 3.5), которые визуализируют, как модель распределяет свое внимание при обработке дифракционных данных тестовой реальной кристаллической структуры. Механизм внимания в трансформере позволяет модели определять, какие части входных данных наиболее важны для принятия решения в каждый момент времени. В первых слоях трансформера внимание распределено равномерно и рассеянно по всей структуре данных. Это соответствует этапу сбора общего контекста, когда модель пытается получить целостное представление о кристаллической структуре. В последующих слоях внимание становится более сфокусированным и локализованным, что указывает на то, что модель научилась выделять специфические взаимосвязи между различными частями данных. Важно отметить способность модели устанавливать связи между отражениями, находящимися на значительном расстоянии друг от друга в обратном пространстве. Это критически важно для решения проблемы фаз, так как часто ключевые взаимосвязи существуют между отражениями, которые не являются ближайшими соседями. Кроме того, наблюдаются характерные диагональные паттерны внимания, которые коррелируют с известными систематическими погасаниями в кристаллографии. Можно предположить, что модель глубокого обучения без явного указания из обучающей выборки выучила механизм погасаний.

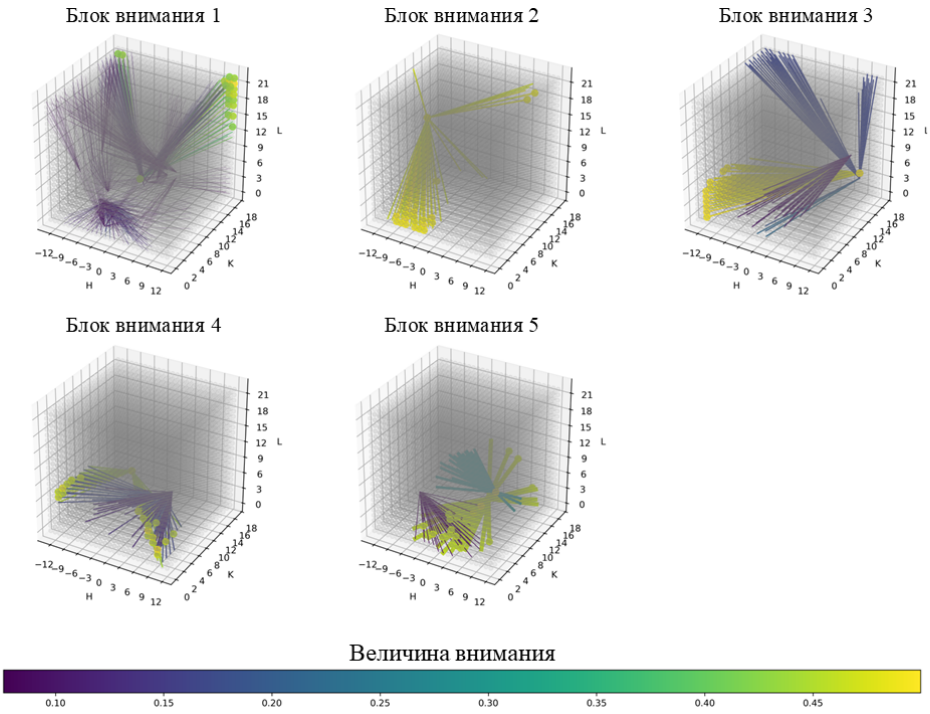

Рисунок 3.4. Связи внимания в блоках трансформера

Для количественного подтверждения этой гипотезы, было проанализировано распределение значений внимания для наиболее сильных связей в каждом блоке трансформера (рис. 3.5). Особый интерес представляет сравнение двух типов связей: между точками, соответствующим систематическим погасаниям и обычными отражениями. Можно заметить, что во время сбора общего контекста в первом блоке распределение значений внимания для связей с погасшими отражениями сопоставимо с таковым для обычных отражений. Это логично, поскольку на этом этапе модель еще не дифференцирует типы отражений, а просто собирает общую информацию о структуре. Однако уже во втором блоке наблюдается значительно уменьшение связей внимания, соединяющих погасания. Это указывает на то, что модель начинает осознавать, что эти отражения несут меньше полезной информации, требуемой для предсказывания дальних отражений. В третьем же блоке модель игнорирует точки обратного пространства, соответствующие систематическим погасаниям, и в выходном тензоре на соответствующих местах стоят нули.

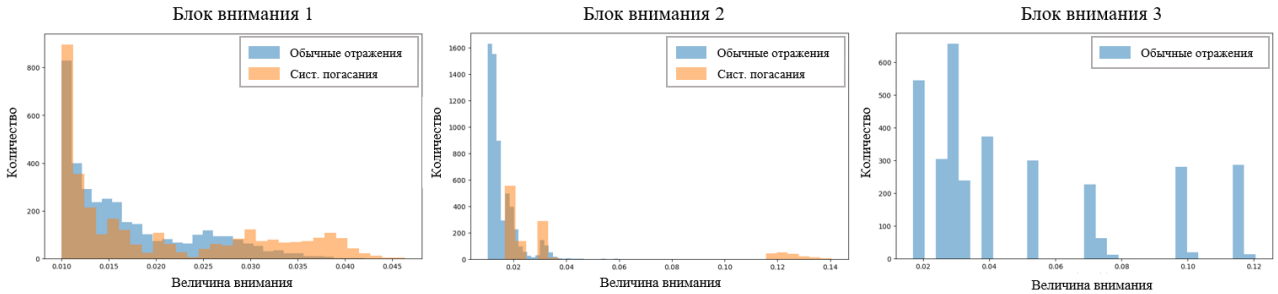

Рисунок 3.5. Распределение величин внимания в первых блоках трансформера

Такое поведение модели демонстрирует, что она не просто запомнила шаблоны из обучающей выборки, а научилась автоматически определять и игнорировать отражения, соответствующие систематическим погасаниям (если такие присутствуют). Трансформер действительно успешно выучил кристаллографическую закономерность, значит, данная архитектура может являться ключевой для дальнейших исследований применения методов глубокого обучения для решений кристаллографических задач.

Для понимания внутренней работы моделей на основе UNet, был проведен анализ первого блока обеих архитектур с помощью метода GradCAM (рис. 3.6) [37]. Этот метод позволяет визуализировать, какие области входных данных имеют наибольшее влияние для принятия решений моделью.

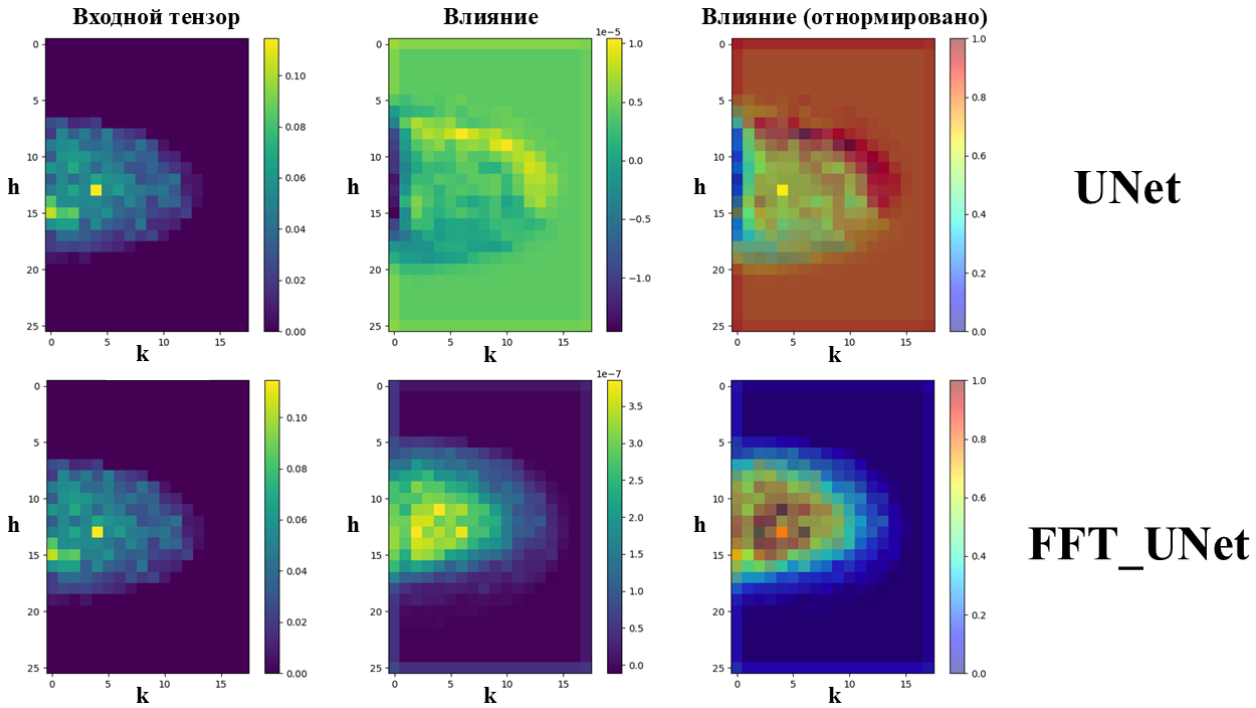

Рисунок 3.6. Тепловые карты влияния GradCAM (усреднено по индексу l)

Анализ тепловой карты влияния для модели UNet показал интересную особенность: модель концентрируется во всем пространстве за пределами изначальной дифракционной

картины низкого разрешения. Это может указывать на то, что модель UNet не полностью учитывает физические ограничения задачи и пытается извлечь информацию из областей, где она физически не может существовать.

В отличие от UNet, паттерн внимания для FFT_UNet выглядит более физически обоснованным. Значения влияния распространяются преимущественно на область входной дифракционной картины и немного выходят за её границы. Это поведение более физически обосновано, так как модель ищет информацию вблизи границ дифракционной картины, где могут находиться важные детали структуры.

Это различие в поведении моделей может объяснять, почему FFT_UNet показывает лучшие результаты в восстановлении дифракционной картины. Её способность более точно определять области, где может находиться полезная информация, и игнорировать физически нереалистичные области, делает её более эффективной в поиске правильного решения.

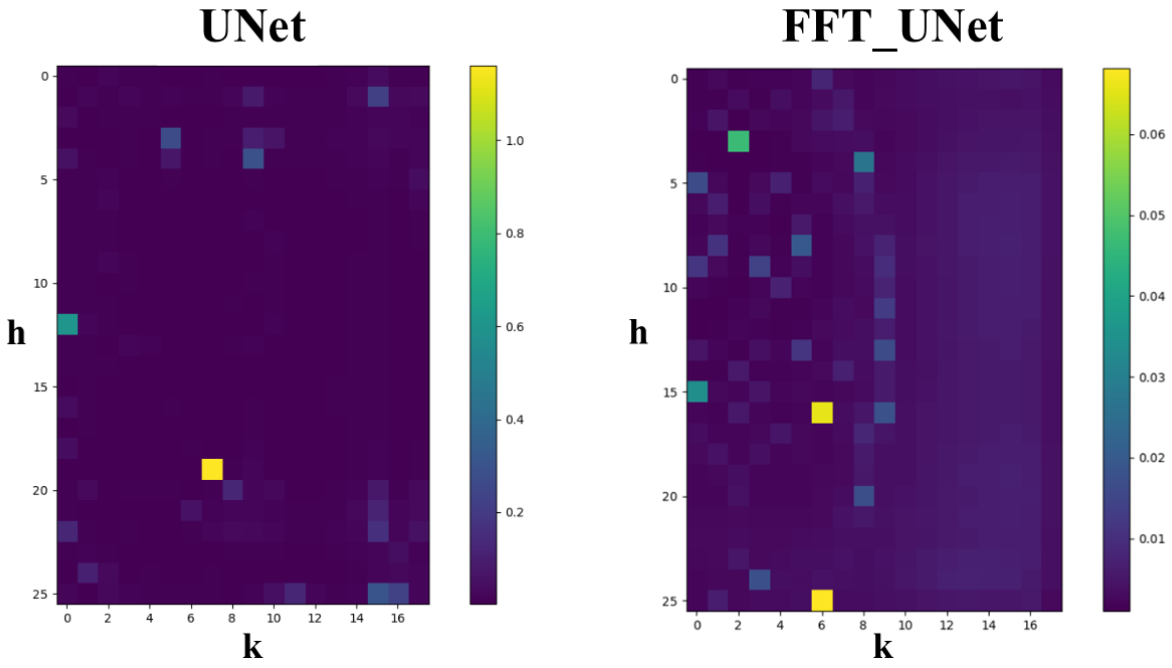

Рисунок 3.7. Тепловые карты чувствительности, усреднено по индексу l

Для оценки устойчивости моделей к шуму в экспериментальных данных был проведен анализ карт чувствительности (рис. 3.7). Карты чувствительности показывают, как сильно меняется выход модели при добавлении случайного шума к входным данным, что позволяет оценить устойчивость модели в различных областях обратного пространства. Базовая модель UNet демонстрирует неустойчивое поведение – выход модели может измениться при добавлении шума на очень большие значения. Модель с Фурье-

преобразованием FFT_UNet стабильно работает при наличии шума в данных. Высокая робастность в большей части пространства указывает на то, что модель научилась извлекать надежные признаки из данных и не переобучилась на конкретные значения амплитуд структурных факторов.

Анализ результатов показывает, что нейронная сеть успешно справляется с определением общих характеристик дифракционной картины. Она точно предсказывает границы дифракционной картины и корректно восстанавливает средние значения амплитуд структурных факторов по небольшим областям. Однако изменения амплитуд между соседними точками обратного пространства недостаточно четко выражены, что приводит к потере важных деталей в распределении амплитуд.

Это ограничение становится особенно критичным, если учесть фундаментальную особенность дифракционных отражений: каждое отражение содержит информацию о всей кристаллической структуре, и для успешного решения проблемы фаз необходимо чрезвычайно точное определение амплитуды в каждой точке. Хотя модели глубокого обучения выявляют некоторые кристаллографические закономерности, они не достигают необходимой точности в численном определении значений для каждой точки.

Таким образом, методы глубокого обучения демонстрируют значительный потенциал в распознавании кристаллографических закономерностей в обратном пространстве, но сталкиваются с фундаментальным ограничением при решении проблемы фаз. Это ограничение связано с необходимостью чрезвычайно точного численного восстановления амплитуд отражений, что требует более точного подхода к определению значений в каждой точке. Это наблюдение указывает на необходимость разработки новых архитектур или подходов, которые могли бы сочетать способность к распознаванию паттернов с более точным численным восстановлением.

4 Выводы

- Разработано программное обеспечение по генерации синтетических рентгенодифракционных данных, которое может быть использование для решения прикладных задач рентгеновской дифракции с помощью ИИ (`github.com/blackwood168/xrd_simulator`)
- Разработан единый пайплайн (`github.com/blackwood168/xrd_phase_ml`), позволяющий проводить воспроизводимые эксперименты по обучению, тестированию и inference задачи предсказывания дальних отражений по ближним для решения проблемы фаз
- Разработаны и обучены на синтетических и реальных структурах UNet в качестве baseline, FFT_UNet — кастомный UNet с Фурье-преобразованием в слоях, XRD_Transformer — модель на основе трансформер со специфичным эмбеддингом, вписывающимся в физику задачи
- Проведен анализ моделей, подтверждающий, что модели глубокого обучения выявляют кристаллографические закономерности и имеют потенциал решения прикладных задач рентгеновской дифракции
- У моделей не достает качества численного восстановления рентгеновских отражений для решения проблемы фаз, несмотря на улавливание рентгенодифракционных паттернов; приведено обоснование и предложен план изменения методологии для решения задачи

References

[1] Girolami G. X-ray Crystallography. — Mill Valley : University Science Books, 2016.

[2] Hauptman H. and Karle J. Solution of the phase problem for space group ıt Pøverline1 // Acta Crystallographica. — 1954. — Vol. 7, no. 4. — P. 369–374.

[3] Hauptman H. The Direct Methods of X-ray Crystallography // Science. — 1986. — Vol. 233, no. 4760. — P. 178–183.

[4] Giacovazzo C. Direct Phasing in Crystallography: Fundamentals and Applications. — Oxford University Press, 1998.

[5] Giacovazzo C. International Tables for Crystallography. — 2 ed. — International Union of Crystallography, 2010. — Vol. B: Reciprocal space.

[6] Wilson A. J. C. Determination of Absolute from Relative X-Ray Intensity Data // Nature. — 1942. — Vol. 150, no. 3796. — P. 151–152.

[7] Sayre D. The squaring method: a new method for phase determination // Acta Crystallographica. — 1952. — Vol. 5, no. 1. — P. 60–65.

[8] Cochran W. A relation between the signs of structure factors // Acta Crystallographica. — 1952. — Vol. 5, no. 1. — P. 65–67.

[9] Karle J. and Karle I. L. The symbolic addition procedure for phase determination for centrosymmetric and non-centrosymmetric crystals // Acta Crystallographica. — 1966. — Vol. 21, no. 6. — P. 849–859.

[10] Burla M. C., Giacovazzo C., and Polidori G. A robust tangent procedure // Journal of Applied Crystallography. — 2013. — Vol. 46, no. 6. — P. 1592–1602.

[11] Rossmann M. G. and Arnold E. Patterson and molecular-replacement techniques // International Tables for Crystallography Volume B: Reciprocal space / ed. by Shmueli U. — Dordrecht : Springer Netherlands, 2001. — P. 235–263.

[12] Buerger M. J. Solution Functions for Solving Superposed Patterson Syntheses // Proceedings of the National Academy of Sciences. — 1953. — Vol. 39, no. 7. — P. 674–678.

[13] Hendrixson T. L. and Jacobson R. A. Locating symmetry elements in Patterson superposition maps // Zeitschrift für Kristallographie - Crystalline Materials. — 1997. — Vol. 212, no. 8. — P. 577–585.

[14] Pavelčík F., Kuchta L., and Sivý J. Patterson-oriented automatic structure determination. Utilizing Patterson peaks // Acta Crystallographica Section A. — 1992. — Vol. 48, no. 6. — P. 791–796.

[15] Sheldrick G. M. Patterson superposition and ab initio phasing // Macromolecular Crystallography Part A. — Academic Press, 1997. — Vol. 276 of Methods in Enzymology. — P. 628–641.

[16] Oszlányi G. and Süto A. Ab initio structure solution by charge flipping // Acta Crystallogr A. — 2004. — Vol. 60, no. Pt 2. — P. 134–141.

[17] Oszlányi G. and Süto A. ıt Ab initio structure solution by charge flipping. II. Use of weak reflections // Acta Crystallographica Section A. — 2005. — Vol. 61, no. 1. — P. 147–152.

[18] Oszlányi G. and Süto A. The charge flipping algorithm // Acta Crystallographica Section A. — 2008. — Vol. 64, no. 1. — P. 123–134.

[19] Dumas C. and van der Lee A. Macromolecular structure solution by charge flipping // Acta Crystallographica Section D. — 2008. — Aug. — Vol. 64, no. 8. — P. 864–873.

[20] Coelho A. A. A charge-flipping algorithm incorporating the tangent formula for solving difficult structures // Acta Crystallographica Section A. — 2007. — Vol. 63, no. 5. — P. 400–406.

[21] Palatinus L. and Chapuis G. ıt SUPERFLIP – a computer program for the solution of crystal structures by charge flipping in arbitrary dimensions // Journal of Applied Crystallography. — 2007. — Vol. 40, no. 4. — P. 786–790.

[22] Sheldrick G. M. *SHELXT*– Integrated space-group and crystal-structure determination // Acta Crystallogr A Found Adv. — 2015. — Vol. 71, no. 1. — P. 3–8.

[23] Burla M. C., Giacovazzo C., and Polidori G. From a random to the correct structure: the VLD algorithm // Journal of Applied Crystallography. — 2010. — Vol. 43, no. 4. — P. 825–836.

[24] Burla M. C., Carrozzini B., Cascarano G. L., Giacovazzo C., and Polidori G. Advances in the VLD algorithm // Journal of Applied Crystallography. — 2011. — Vol. 44, no. 6. — P. 1143–1151.

[25] Burla M. C., Giacovazzo C., and Polidori G. Phasing medium-size structures and proteins by the VLD algorithm // Journal of Applied Crystallography. — 2011. — Vol. 44, no. 1. — P. 193–199.

[26] Larsen A. S., Rekis T., and Madsen A. PhAI: A deep-learning approach to solve the crystallographic phase problem // Science. — 2024. — Vol. 385, no. 6708. — P. 522–528.

[27] Cowtan K. Phase problem in X-ray crystallography, and its solution // eLS. — 2003.

[28] Carrozzini B., De Caro L., Giannini C., Altomare A., and Caliandro R. The phase-seeding method for solving non-centrosymmetric crystal structures: a challenge for artificial intelligence // Acta Crystallographica Section A. — 2025. — Vol. 81, no. 3. — P. 188–201.

[29] Pan T., Dramko E., Miller M. D., Jr G. N. P., and Kyrillidis A. RecCrysFormer: Refined Protein Structural Prediction from 3D Patterson Maps via Recycling Training Runs. — 2025.

[30] Wang K., Song L., Wang C., Ren Z., Zhao G., Jiazhen D., Di J., Barbastathis G., Zhou R., Zhao J., and Lam E. On the use of deep learning for phase recovery // Light: Science & Applications. — 2024. — Vol. 13.

[31] Grosse-Kunstleve R. W., Sauter N. K., Moriarty N. W. and Adams P. D. The*Computational Crystallography Toolbox*: crystallographic algorithms in a reusable software framework // J Appl Crystallogr. — 2002. — Vol. 35, no. 1. — P. 126–136.

[32] David W. I. F. Powder diffraction peak shapes. Parameterization of the pseudo-Voigt as a Voigt function // Journal of Applied Crystallography. — 1986. — Vol. 19, no. 1. — P. 63–64.

[33] Groom C. R. and Allen F. H. The Cambridge Structural Database in Retrospect and Prospect // Angew Chem Int Ed. — 2014. — Vol. 53, no. 3. — P. 662–671.

[34] Zeng K. and Wang Z. 3D-SSIM for video quality assessment // 19th IEEE International Conference on Image Processing. — 2012. — P. 621–624.

[35] Ronneberger O., Fischer P., and Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation. — 2015.

[36] Yang Y., Lian Q., Zhang X., Zhang D., and Zhang H. HIONet: Deep priors based deep unfolded network for phase retrieval // Digital Signal Processing. — 2023. — Vol. 132. — P. 103797.

[37] Selvaraju R. R., Cogswell M., Das A., Vedantam R., Parikh D., and Batra D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization // International Journal of Computer Vision. — 2020. — Vol. 128, no. 2. — P. 336–359.

Приложение

Приложение 1. Общий вид скрипта для генерации структурных данных

```
SPACE_GROUPS = ['P21', 'C2']
ELEMENTS = ["C", "N", "O", "Cl", "Br"]
N_ATOMS_LIMS = (10, 30)
ATOM_VOLUME_START_WIDTH = (14, 8)
d_high = 1.0  # High resolution limit
d_low = 1.2   # Low resolution limit
n_atoms = sample_gen(range(*N_ATOMS_LIMS))

# Initialize structure generator
str_generator = GenBuilder(
    classname=core.CctbxStr.generate_packing,
    sg=sample_gen(SPACE_GROUPS),
    atoms=sample_gen(ELEMENTS, size=n_atoms),
    atom_volume=distr_gen(sts.uniform(*ATOM_VOLUME_START_WIDTH)),
    seed=utils.distr_gen(sts.randint(1, 2**32-1))
)
def runner(pattern):
    """Process a single crystal structure pattern.
    Args:
        pattern: Crystal structure pattern object
    Returns:
        dict: Contains structure parameters and intensity data
    """
    ### Здесь разный код в зависимости от задачи ###

# Processing parameters
CHANKS = 50
CHANK_SIZE = 10000
CORES = 12
# Process structures in chunks
for i in range(CHANKS):
    chank = it.islice(str_generator, CHANK_SIZE)
    pool = mp.Pool(CORES)
    with pool as p: results = p.map(runner, chank)
    directory = f'...'
    if not os.path.exists(directory): os.makedirs(directory)
    # Save results
    np.savez_compressed(f'{directory}/{i}',
        db=np.array(results)
    )
    pool.close()
```

Приложение 2. Функция runner для расчета интенсивностей случайных структур

```python
def runner(pattern):
    """Process a single crystal structure pattern.

    Args:
        pattern: Crystal structure pattern object

    Returns:
        dict: Contains structure parameters and intensity data
    """
    params = pattern.report_params()
    structure = pattern.structure

    # Calculate structure factors and intensities
    a_high = structure.structure_factors(d_min=d_high).f_calc().sort()
    I_high = a_high.as_intensity_array().data().as_numpy_array()
    ind_high = np.array(list(a_high.indices()))

    a_low = structure.structure_factors(d_min=d_low).f_calc().sort()
    ind_low = np.array(list(a_low.indices()))

    return {
        'structure_params': params,
        'ind_low': ind_low,
        'I_high': I_high,
        'ind_high': ind_high
    }
```

Приложение 3. Функция runner для расчета нормализованных структурных факторов случайных структур

```python
def runner(pattern):
    """Process a single crystal structure pattern.

    Args:
        pattern: Crystal structure pattern object

    Returns:
        dict: Contains structure parameters and intensity data
    """
    params = pattern.report_params()
    structure = pattern.structure

    # Calculate structure factors and intensities
    a_high = structure.structure_factors(d_min=d_high).f_calc().sort()
    elements_to_parse = ['N', 'O', 'C', 'Cl', 'Br']
    asu_content = {}
    for scatterer in structure.scatterers():
        for elem in elements_to_parse:
            if scatterer.label == elem or scatterer.label.startswith(elem):
                asu_content[elem] = asu_content.get(elem, 0) + 1
    I_high = structure.structure_factors(d_min=d_high).f_calc().sort().as_intensity_array()
    e_high = a_high.as_amplitude_array().normalised_amplitudes(asu_content).array().data()
    e_high = e_high.as_numpy_array()
    ind_high = np.array(list(a_high.indices()))

    a_low = structure.structure_factors(d_min=d_low).f_calc().sort()
    e_low = a_low.as_amplitude_array().normalised_amplitudes(asu_content).array().data()
    e_low = e_low.as_numpy_array()
    ind_low = np.array(list(a_low.indices()))

    return {
        'structure_params': params,
        'ind_low': ind_low,
        'I_high': I_high,
        'ind_high': ind_high,
        'e_high': e_high,
        'e_low': e_low
    }
```

Приложение 4. Функция runner для расчета карт Паттерсона случайных структур

```python
def runner(pattern):
    """Process a single crystal structure pattern.

    Args:
        pattern: Crystal structure pattern object

    Returns:
        dict: Contains Patterson maps, structure parameters and intensity data
    """
    params = pattern.report_params()
    structure = pattern.structure

    # Calculate structure factors
    I_high = structure.structure_factors(d_min=d_high).f_calc().sort().as_intensity_array()
    ind_high = np.array(list(I_high.indices()))
    I_low = structure.structure_factors(d_min=d_low).f_calc().sort().as_intensity_array()
    ind_low = np.array(list(I_low.indices()))

    patt_low = pu.calculate_patterson_fft(I_low.data().as_numpy_array(),
                            miller_indices = ind_low, map_shape = (12,12,12))
    patt_high = pu.calculate_patterson_fft(I_high.data().as_numpy_array(),
                            miller_indices = ind_high, map_shape = (24,24,24))
    assert patt_low.min() == 0 and patt_high.min() == 0
    assert patt_low.max() == 1 and patt_high.max() == 1

    # Prepare intensity and index data
    I_high = I_high.data().as_numpy_array()
    I_low = I_low.data().as_numpy_array()

    return {
        'patt_low': patt_low,
        'patt_high': patt_high,
        'structure_params': params,
        'ind_low': ind_low,
        'ind_high': ind_high
    }
```

Приложение 5. Общий вид скрипта для генерации случайных порошковых дифрактограмм

```
BKG_MAX_ORDER = 13
BKG_MIN_ORDER = 2
bkg_generator = ...
SPACE_GROUPS = ["P-1", "P21/c", "C2/c", "Pbca", "I41"]
ELEMENTS = ["C", "N", "O", "Cl"]
N_ATOMS_LIMS = (3, 30)
ATOM_VOLUME_START_WIDTH = (14, 8)
n_atoms = sample_gen(range(*N_ATOMS_LIMS))
str_generator = ...
GAUSS_STEPS = 0.01
symmetric_profile_generator = ...
profile_generator = ...
phase_generator = ...
GRID = np.linspace(3.0, 90.0, 4351)
CUKA1 = [[1.540596, 1]]
CUKA12 = [[1.540596, 2/3], [1.544493, 1/3]]
pattern_generator = GenBuilder(
    classname= core.Pattern,
    waves = sample_gen([CUKA1, CUKA12]),
    phases = ([phase] for phase in phase_generator),
    bkg = bkg_generator,
    scales = distr_gen(sts.uniform(100,20000), size = 1),
    bkg_range = (sorted(ii) for
                 ii in utils.distr_gen(sts.uniform(500, 7000), size = 2)))
def runner(pattern):
    return (pattern.report_params(), pattern.pattern(GRID))
CHANKS = 2
CHANK_SIZE = 50
CORES = 4
for i in range(CHANKS):
    chank = it.islice(pattern_generator, CHANK_SIZE)
    pool = mp.Pool(CORES)
    with pool as p:
        results = p.map(runner, chank)
    y = pd.DataFrame( [ y for y,_ in results  ]  )
    x = pd.DataFrame( [ x for _,x in results  ]  )
    y.to_csv(f'test_y_{i}.csv')
    x.to_csv(f'test_x_{i}.csv')
```

Приложение 6. Реализация асимметрии максимумов в рентгеновских порошковых дифрактограммах

```
class AxialCorrection:
    def __init__(self, profile, HL, SL, N_gauss_step):
        self.L = 1
        self.H = HL
        self.S = SL
        self.N_gauss_step = N_gauss_step
        self.profile = profile
    def h(self, phi, peak):
        return self.L*np.sqrt(np.cos(phi*np.pi/180)**2/np.cos(peak*np.pi/180)**2 - 1)
    def phi_min(self, peak):
        a = np.cos(peak*np.pi/180) * np.sqrt( ((self.H+self.S)/self.L)**2 + 1 )
        if a > 1 :
            return 0
        else:
            return 180/np.pi*np.arccos( a )
    def phi_infl(self, peak):
        a = np.cos(peak*np.pi/180)*np.sqrt( ((self.H-self.S)/self.L)**2 + 1 )
        if a > 1 :
            return 0
        else:
            return 180/np.pi*np.arccos(a)
    def W2(self, phis, peak):
        result = np.zeros(len(phis))
        cond1 = (self.phi_min(peak) <= phis) & (phis <= self.phi_infl(peak))
        result[cond1] = self.H + self.S - self.h(phis[cond1], peak)
        cond2 = (phis > self.phi_infl(peak)) & (phis <= peak)
        result[cond2] = 2 * min(self.H, self.S)
        return result
    def calc(self, Th2, peak):
        phmin = self.phi_min(peak)
        dd = np.abs(peak - phmin) / self.N_gauss_step
        N_gauss = np.ceil(dd).astype(int)
        if (N_gauss == 1):
            return self.profile.calc(Th2, peak)
        xn, wn = np.polynomial.legendre.leggauss(N_gauss)
        step = Th2[1] -Th2[0]
        deltan = (peak+phmin)/2 + (peak-phmin)*xn/2
        tmp_assy = np.zeros(len(Th2))
        arr1 = wn*self.W2(deltan, peak)/self.h(deltan, peak)/np.cos(deltan*np.pi/180)
        for dn in range(len(deltan)):
            tmp_assy += arr1[dn] * self.profile.calc(Th2, deltan[dn])
        tmp_assy = tmp_assy / np.sum(arr1)
        return(tmp_assy)
```

```
class PV_TCHZ:
    def __init__(self, parameters):
        self.U = parameters[0] / 1083  # Follow GSAS conventions
        self.V = parameters[1] / 1083  # Follow GSAS conventions
        self.W = parameters[2] / 1083  # Follow GSAS conventions
        self.X = parameters[3] / 100   # Follow GSAS conventions
        self.Y = parameters[4] / 100   # Follow GSAS conventions
        self.Z = parameters[5] / 100   # Follow GSAS conventions
    def fwhmL(self, peak):
        peak = peak / 180 * np.pi
        return (self.X * np.tan(peak/2) + self.Y/np.cos(peak/2))
    def fwhmG(self, peak):
        peak = peak / 180 * np.pi
        return np.sqrt(self.U * np.tan(peak/2) ** 2 +
                       self.V * np.tan(peak/2) +
                       self.W +
                       self.Z / np.cos(peak/2) ** 2)
    def lorenz(self, Th2, peak, l):
        return (2 / np.pi / l) / (1 + 4 * (Th2 - peak)**2 / l**2)
    def gauss(self, Th2, peak, g):
        return (2 * (np.log(2)/np.pi) ** 0.5 / g) *
         np.exp(-4 * np.log(2) * (Th2 - peak)**2 / g**2)
    def n_for_tchz(self, l, g):
        G = g ** 5 + 2.69269*g ** 4 * l + 2.42843 * g ** 3 * l ** 2 +
        4.47163 * g ** 2 * l ** 3
        G += 0.07842 * g * l ** 4 + l ** 5
        G = l / (G ** 0.2)
        n = 1.36603 * G - 0.47719 * G ** 2 + 0.11116 * G ** 3
        return n
    def tchz(self, Th2, peak, l, g, n):
        return n* self.lorenz(Th2, peak, l) + (1 - n)* self.gauss(Th2, peak, g)
    def calc(self, Th2, peak):
        wl = self.fwhmL(peak)
        wg = self.fwhmG(peak)
        n = self.n_for_tchz(wl, wg)
        return self.tchz(Th2, peak, wl, wg, n)
```