ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Образовательная программа «Прикладная математика и информатика»

Отчет об исследовательском проекте на тему:

Исследование возможности улучшения качества генеративных моделей путем использования суррогатных признаков

Выполнила:

студентка группы БПМИ202
Иванова Алеся Александровна

(подпись) (дата) 18.05.2023

Принял руководитель проекта:

Рогачев Александр Игоревич
Преподаватель
Департамента анализа данных и искусственного интеллекта

(подпись) (дата) 18.05.2023

Москва 2023

Содержание

Аннотация **3**

Ключевые слова **3**

1 Введение **4**

2 Обзор литературы **6**

3 Используемые данные и их предварительная обработка **8**

4 Анализ существующих решений **9**

4.1 Базовая модель 9

4.2 Оценка качества модели 10

5 Эксперименты **14**

5.1 Выбор приближаемой метрики и архитектуры регрессионной модели 14

5.2 Выбор данных для обучения регрессионной модели 15

5.3 Исследование и сравнение архитектур моделей 16

5.4 Изучение влияния гиперпараметров на качество генерации 21

6 Заключение **26**

7 Направления дальнейшей работы **26**

Список литературы **28**

Приложение А: Архитектуры моделей **29**

Аннотация

При решении задачи генерации объектов по заданным параметрам бывает необходимо, чтобы сгенерированные объекты обладали определенными свойствами. Иногда данные метрики могут быть не дифференцируемы, из-за чего их может быть невозможно использовать в функции потерь при обучении генеративной модели. Одним из решений данной проблемы является использование суррогатных признаков - моделей, приближающих значения метрик. В данной работе исследуется возможность применения суррогатных признаков для улучшения качества генерации модели, реконструирующей отклик камеры временной проекции в многоцелевом детекторе, который является частью установки ионного коллайдера на базе нуклотрона. В качестве базовой модели использована генеративно-состязательная модель, предложенная в статье [6]. При помощи использования суррогатных признаков удалось улучшить качество генерации на 39.07% по сравнению с базовой моделью.

Ключевые слова

Глубинное обучение, генеративно-состязательная сеть, суррогатные признаки, физика высоких энергий, многоцелевой детектор

1 Введение

Одним из применений машинного обучения является генерация объектов по заданным параметрам. Для решения этой задачи используются генеративно-состязательные сети (англ. Generative adversarial network, сокращенно GAN), которые состоят из комбинации двух нейронных сетей: генератора и дискриминатора [4]. Генератор предназначен для преобразования выборок из фиксированного известного распределения в объекты из целевого распределения, то есть распределения обучающих данных. Дискриминатору подаются на вход примеры из целевого распределения и выходных данных генератора, и он старается научиться отличать эти объекты. Цель дискриминатора состоит в том, чтобы получить метрику, которая максимизирует разделение между обучающими данными и сгенерированными объектами, в то время как задача генератора заключается в минимизации этого разделения.

При генерации бывает необходимо, чтобы воспроизводились определенные свойства, которыми обладают объекты по отдельности, т.е. которые могут быть вычисленные на уровне каждого примера. Не всегда данные метрики дифференцируемы, поэтому бывает невозможно использовать их в качестве функций потерь напрямую. Если же подавать их дискриминатору напрямую, то генератору может быть трудно воспроизводить их, по той же самой причине. Одним из решений данной проблемы является использование суррогатных признаков. Суррогатные признаки - это модели, которые учатся предсказывать приближенные значения метрик. Выдаваемые ими значения являются дифференцируемыми, поэтому могут быть использованы в функции потерь основной модели. Суррогатные признаки можно обучать вместе с основной моделью. Их функция потерь зависит от того, насколько предсказания близки к настоящим значениям метрик, и может быть добавлена к функции потерь основной модели.

В экспериментах по физике высоких энергий часто крайне важно провести их реалистичное моделирование. Иногда это помогает оптимизировать конструкцию детектора или найти лучшие методы анализа данных, и обычно, чем точнее моделирование, тем точнее будет конечный экспериментальный результат. В данной работе исследуется возможность улучшения при помощи использования суррогатных признаков качества генерации GAN, реконструирующей отклик камеры временной проекции (англ. Time Projection Chamber, сокращённо TPC) в многоцелевом детекторе (англ. Multi Purpose Detector, сокращённо MPD), который является частью установки ионного коллайдера на базе нуклотрона (англ. Nuclotron-based Ion Collider fAcility, сокращённо NICA).

На Рисунке 1.1 изображено устройство MPD. MPD обнаруживает заряженные адро-

ны, электроны и фотоны. TPC является одним из самых внутренних “слоев” детектора и конструктивно представляет собой заполненный газом цилиндр с электрическим полем, которое заставляет ионизационные электроны дрейфовать к чувствительным пластинам, расположенным на торцевых крышках цилиндра. Поскольку скорость дрейфа электронов в поле известна, можно связать координаты пластины и время их отклика с положением электрона.

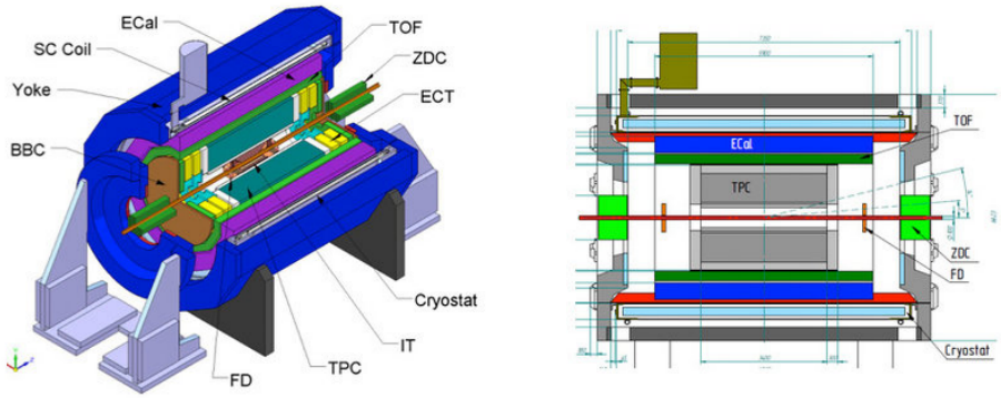
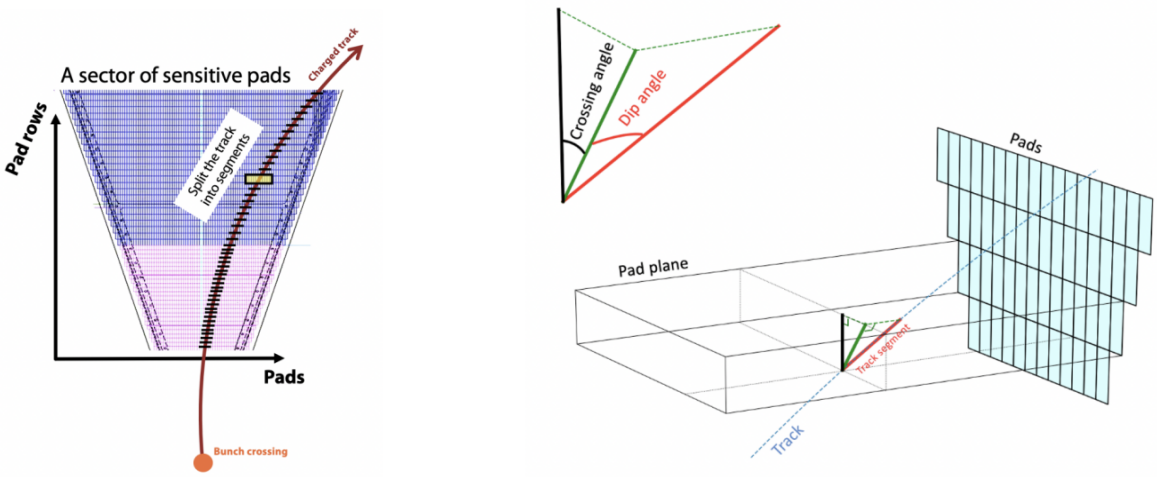


Рис. 1.1: Устройство MPD. TPC: камера временной проекции; TOF: детектор времени пролета; ECal: электромагнитный калориметр; FD: прямой детектор; BBC: счетчик лучей; ZDC: калориметр с нулевой температурой (иллюстрация из [6])

Чтобы упростить задачу имитации TPC, чувствительные пластины разбиваются на сегменты (отмечены желтым на Рисунке 1.2a), на которых фиксируется отклик во время движения электрона по соответствующему отрезку своей траектории. Отрезок траектории электрона может быть задан при помощи 3 координат и двух углов (см. Рисунок 1.2b), следовательно, есть 5 входных параметров для модели.



(a) : Проекция траектории электрона на сектор чувствительной пластины (иллюстрация из [6])

(b) : Траектория электрона внутри TPC (иллюстрация из [6])

Рис. 1.2

Отклик сегмента чувствительных пластин может быть представлен в виде матрицы действительных чисел или изображения, в котором первой координатой является время, а второй - номер пластины. В нашей задаче размеры матриц выбраны как (8, 16), то есть модель должна быть способна генерировать (8, 16) изображения (см. образец на Рисунке 1.3).

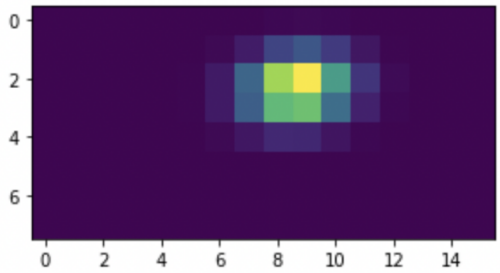


Рис. 1.3: : Образец отклика сегмента чувствительных пластин TPC (иллюстрация из [6])

В статье [6] предложено оценивать качество генерации путем вычисления моментов выходного изображения, а именно средних значений отклика по времени и по номеру пластины, их дисперсий, ковариации и суммы всех значений пикселей. Затем распределение каждого момента сравнивается с помощью χ^2, описанного в разделе 4.2, с распределением этого момента на изображении из реальных данных для того же набора из 5 входных признаков.

Наилучшего качества генерации на данный момент удалось достичь при помощи GAN, предложенной в статье [6]. В данной работе исследованы применения суррогатных признаков для улучшения результатов генерации. При помощи использования суррогатных признаков удалось улучшить качество генерации на 39.07% по сравнению с базовой моделью.

2 Обзор литературы

Генеративно-состязательные сети состоят из комбинации двух моделей, между которыми происходит минимаксная игра [4]. Генератор учится получать объекты, распределение которых близко к распределению обучающих данных. На вход генератору для каждого генерируемого объекта подается вектор латентных параметров, полученных из выбранного латентного распределения $\mathbb{P}_z$. Дискриминатор учится отличать реальные данные от сгенерированных. На вход дискриминатору подаются настоящие данные и данные, полученные при помощи генератора. Каждому входному объекту дискриминатор выдает вероятность, что этот объект был выбран из настоящих данных. Следующая функция определяет минимаксную игру генератора и дискриминатора:

$$\min_G \max_D L_{GAN}(D,G) = \mathbb{E}_{x\sim\mathbb{P}_r}[\log D(x)] + \mathbb{E}_{z\sim\mathbb{P}_z}[\log(1 - D(G(z)))], \tag{1}$$

где G - генератор, D - дискриминатор, $\mathbb{P}_r$ - распределение настоящих данных, $\mathbb{P}_z$ - латентное распределение.

Wasserstein Generative Adversarial Network (сокр. WGAN) является генеративно-состязательной моделью, на практике часто обладающей большей стабильностью обучения и лучшим качеством по сравнению со стандартной GAN [1]. В WGAN минимаксная игра выглядит следующим образом:

$$\min_G \max_D L_{WGAN}(D,G) = \mathbb{E}_{x\sim\mathbb{P}_r}[\log D(x)] - \mathbb{E}_{z\sim\mathbb{P}_z}[\log D(G(z))], \tag{2}$$

где G - генератор, D - дискриминатор, $\mathbb{P}_r$ - распределение настоящих данных, $\mathbb{P}_z$ - латентное распределение.

Одним из наиболее частых способов регуляризации WGAN является использование в функции потерь дискриминатора градиентного штрафа (англ. gradient penalty), который увеличивается при отклонении нормы градиента от 1 [5]. В этом случае используемая в функции потерь дискриминатора метрика Вассерштейна с градиентным штрафом определяется по формуле:

$$L = \mathbb{E}_{\widetilde{x}\sim\mathbb{P}_g}[D(\widetilde{x})] - \mathbb{E}_{x\sim\mathbb{P}_r}[D(x)] + \lambda\mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}(\|\nabla_{\hat{x}}D(\hat{x})\|^2 - 1)^2, \tag{3}$$

где $\mathbb{P}_r$ - распределение настоящих данных, $\mathbb{P}_g$ - распределение сгенерированных данных, $\mathbb{P}_{\hat{x}}$ - распределение данных, которые получаются как равномерный выбор точки вдоль прямой между парой точек, выбранных из распределения настоящих данных $\mathbb{P}_r$ и распределения сгенерированных данных $\mathbb{P}_g$.

Генеративно-состязательные сети широко используются в области физики высоких энергий для моделирования результатов экспериментов. Они зачастую обладают достаточно хорошим качеством генерации, при этом работают быстрее других методов глубинного обучения и реалистичного моделирования. Так, GAN были использованы в [9] для моделирования работы калориметра. Также GAN были применены в [3] для моделирования работы Черенковского детектора и в [7] для моделирования работы LHCb RICH детектора.

Использование GAN для быстрого моделирования детектора типа TPC было изучено в эксперименте ALICE [2]. В статье [6] описано применение GAN для симуляции откликов TCP в MPD детекторе. Возможность улучшения представленной в этой статье модели при

помощи добавления суррогатных признаков исследуется в данной работе.

Подход с использованием суррогатных признаков был ранее использован в различных генеративных моделях. В статье [10] описано использование суррогатных признаков при моделировании состояния земной коры. Суррогатным признаком в данной статье является модель, описывающая движение водных потоков. В статье [8] используются суррогатные модели для ускорения определения геофизических характеристик коллектора. В статье [11] при моделировании включения ветроэлектростанции используются суррогатные признаки, определяющие потоки воздуха.

3 Используемые данные и их предварительная обработка

Тестирование будет производиться на данных, используемых в статье [6]. Обучающий набор данных получен путем запуска подробного симулятора MPD, который основан на прохождении частиц через детектор и реалистичном моделировании откликов детектора.

В системе координат относительно данного сектора детектора отрезок траектории, пересекающий определенный сегмент чувствительных пластин, определяется четырьмя параметрами (см. Рисунок 1.2b):

- угол пересечения (англ. crossing angle) – угол между поперечной проекцией импульса частицы и нормалью к плоскости;
- угол падения (англ. dip angle) - угол между полным импульсом и его поперечной проекцией;
- длина дрейфа (англ. drift length) - расстояние от отрезка траектории до сегмента чувствительных пластин, отклик которого был зафиксирован, измеряется в количестве временных интервалов от начала движения по текущему отрезку траектории до фиксации отклика сегмента чувствительных пластин;
- координата чувствительной пластины (англ. pad coordinate) - координата проекции центра отрезка траектории на сегмент чувствительных пластин, единицей измерения является длина одной чувствительной пластины.

В используемом наборе данных угол пересечения лежит в диапазоне [-20; 20]°, угол падения лежит в диапазоне [-60, 60]°. Отклики чувствительных пластин извлекаются из 20-й строки сектора пластин детектора.

Всего в используемых данных содержится 20 000 примеров, которые были разделены на обучающие и валидационные подмножества в соотношении 75:25 соответственно.

Вклад в отклик детектора дает небольшой сегмент чувствительных пластин в небольшой период временных интервалов. Для каждого сегмента, чтобы использовать эту локализацию отклика, отклики сдвигаются на целые части от длины дрейфа и от координаты чувствительной пластины вдоль направлений времени и строки сегмента пластин детектора соответственно. После этого отклики во всем обучающем наборе укладываются в матрицу из 8 блоков по 16 временных интервалов, которая составляет целевое пространство. Ответы варьируются на несколько порядков, поэтому они масштабируются с помощью функции $log_{10}(x+1)$ для более плавного обучения. В силу физических соображений, достаточно использовать только дробную часть координаты чувствительной пластины, а не её полную координату (англ. pad coordinate fraction). Что касается длины дрейфа, как дробная часть, так и полное число вводятся в модель как два отдельных признака (англ. drift length fraction и drift length). Перед вводом в модель углы и длина дрейфа линейно масштабируются в отрезок [-1, 1].

4 Анализ существующих решений

4.1 Базовая модель

В качестве базовой модели была использована модель, предложенная в статье [6]. Архитектура базовой модели представленная на Рисунке 7.1.

Генератор имеет полносвязную архитектуру. На вход генератору для каждого примера подаётся 5 параметров, описанных в разделе 3, а также 32 латентных параметра, сгенерированных из стандартного нормального распределения. На выход для каждого примера из входных данных генератор выдает матрицу размера 8×16, описывающую отклик. Последний слой активации в генераторе использует специальную функцию активации:

$$f(x) = \begin{cases} \alpha T e^{x} & x \leq 0 \\ T(\alpha + (1-\alpha)\frac{x}{\delta}) & 0 \leq x \leq \delta \\ T - \delta + x & \delta \leq x. \end{cases} \tag{4}$$

Дискриминатор имеет сверточную архитектуру. На вход дискриминатору для каждого примера подаются матрица, полученная генератором, а также набор из 5 входных параметров. Входные данные перестраиваются под нужные размеры и конкатенируются. Далее идут

несколько групп слоев, состоящих из сверточного слоя, слоя активации ELU и слоя Dropout, зануляющего 2% элементов. Между некоторыми группами расположены Max Pooling слои. Сверточная часть дискриминатора выдает для каждого примера по 64 признака, которые конкатенируются с входными 5 признаками. Далее идут линейные слои, комбинирующие все полученные признаки и выдающие итоговые предсказания. В качестве функции потерь используется метрика Вассерштейна с градиентным штрафом, поскольку это приводит к наилучшему качеству сгенерированных данных.

Как генератор, так и дискриминатор обучались с использованием оптимизатора RMSProp со скоростью обучения, начинающейся с 0,0001 в начале процесса обучения и уменьшающейся в 0,999 раза после каждой эпохи. Выполнялось 8 шагов обновления дискриминатора за один шаг генератора. Размер батча составлял 96. Обучение модели длилось 10 000 эпох. Проектирование модели и обучение выполнялось с использованием фреймворка PyTorch v1.13. Примеры изображений, получающихся с использованием базовой модели, представлены на Рисунке 4.1.

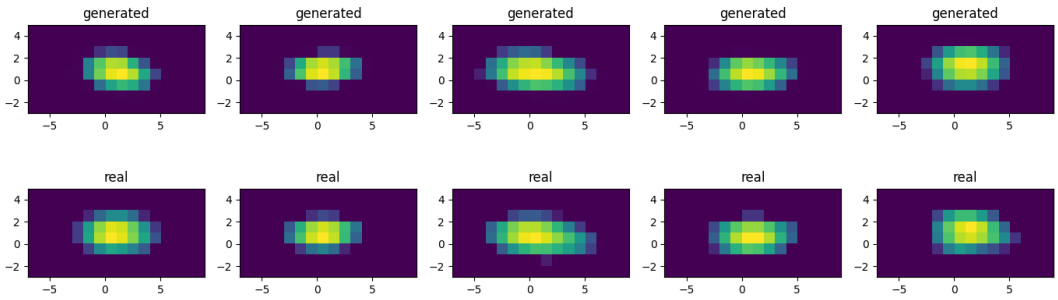


Рис. 4.1: Примеры изображений, выдаваемых генератором базовой модели (generated) и исходных изображений (real) для одинаковых наборов входных параметров.

4.2 Оценка качества модели

Для оценки качества генерации используется метод, предложенный в статье [6]. Вычисляются первые моменты выходного изображения, то есть средние значения отклика по времени и по номеру пластины (μ_t и μ_p), вторые моменты, то есть дисперсий распределений отклика по времени и номеру пластины (σ_t^2 и σ_p^2) и ковариации распределений (σ_{tp}). Далее для каждой из 5 указанных метрик и для каждого из 5 входных параметров вычисляется величина χ^2 следующим образом. Считаются наименьшее и наибольшее значение входного параметра, отрезок значений разбивается на $n_{bins} = 100$ отрезков. Для каждого значения входного параметра определяется, какому отрезку оно принадлежит. Далее внутри каждого

отрезка рассматривается распределение текущей метрики (μ_t, μ_p, σ_t^2, σ_p^2 или σ_{tp}), считается его матожидание и дисперсия, а также бутстрэп ошибка вычисления матожидания и дисперсии. Затем χ^2 вычисляется по следующим формулам:

$$\xi^{upper} = \hat{\xi} + \hat{\sigma}(\xi) \tag{5}$$

$$\xi^{lower} = \hat{\xi} - \hat{\sigma}(\xi) \tag{6}$$

$$err^2(\xi) = (err(\hat{\xi}))^2 + (err(\hat{\sigma}(\xi)))^2 \tag{7}$$

$$\chi^2(\xi) = \frac{(\xi_{gen}^{upper} - \xi_{real}^{upper})^2}{err^2(\xi_{gen}) + err^2(\xi_{real})} + \frac{(\xi_{gen}^{lower} - \xi_{real}^{lower})^2}{err^2(\xi_{gen}) + err^2(\xi_{real})}, \tag{8}$$

где $\hat{\xi}$ - среднее значение выбранной величины, $\hat{\sigma}(\xi)$ - среднеквадратичное отклонение, err - бутсрэп ошибка, ξ_{gen} и ξ_{real} - значения выбранной величины для сгенерированных и настоящих данных соответственно.

Далее для выбранной метрики складываются значения χ^2 по всем 5 входным параметрам. Также считается сумма χ^2 по всем метрикам и всем входным параметрам, данную величину будем называть суммарной χ^2, это будет основная метрика качества модели.

В Таблице 4.1 представлены значения χ^2 для каждой из метрик μ_t, μ_p, σ_t^2, σ_p^2 и σ_{tp}, а так же суммарная χ^2, которые получаются при применении базовой модели. Значения усреднены по эпохам с 9 000 по 10 000 для более стабильной оценки качества генерации.

Таблица 4.1: Метрики качества базовой модели.

χ^2 для μ_t	χ^2 для μ_p	χ^2 для σ_t^2	χ^2 для σ_p^2	χ^2 для σ_{tp}	суммарная χ^2
16.304 ±0.798	115.012 ±7.620	239.173 ±16.757	186.321 ±13.677	58.912 ±3.530	615.724 ±25.560

Изучим распределение на настоящих и сгенерированных данных каждой из метрик μ_t, μ_p, σ_t^2, σ_p^2 и σ_{tp}, а также амплитуды, равной сумме значений в матрице. Распределения представлены на Рисунке 4.2.

Видно, что у графиков для настоящих и сгенерированных данных есть достаточно большая площадь пересечения, что означает, что распределения метрик для настоящих и сгенерированы данных достаточно близки друг к другу.

Рассмотрим так же графики, изображенные на Рисунке 4.3. Они соответствуют величинам, частующим в подсчете метрики χ^2. По оси абсцисс на графиках отложены значения одного из входных признаков, значения разбиты на $n_{bins} = 100$ отрезков. По оси ординат отложены среднее значение выбранной метрики в каждом из отрезков, а также её верхнее и нижнее значения, обозначенные в формулах (5) (6) как $\hat{\xi}$, ξ_{upper}, ξ_{lower} соответственно.

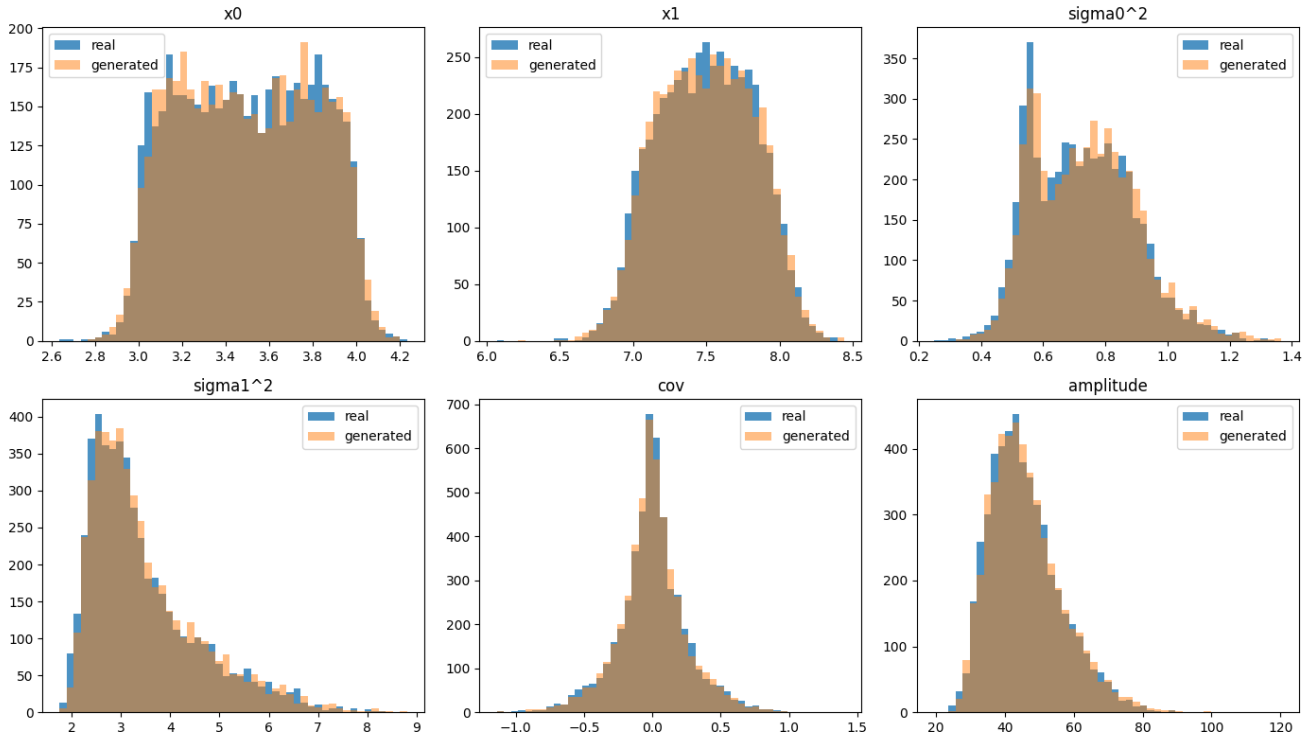


Рис. 4.2: Распределение на настоящих (real) и сгенерированных (generated) данных метрик μ_t (обозначено $x0$), μ_p (обозначено $x1$), σ_t^2 (обозначено $sigma0\hat{}2$), σ_p^2 (обозначено $sigma1\hat{}2$), σ_{tp} (обозначено cov) и амплитуды (обозначено $amplitude$) для базовой модели.

На основе данных графиков проанализируем, насколько хорошо для каждого из 5 входных параметров приближаются распределения метрик μ_t, μ_p, σ_t^2, σ_p^2 и σ_{tp}. Для первого входного параметра crossing angle распределения всех метрик для настоящих и сгенерированных данных очень близки. Для второго входного параметра dip angle распределения всех метрик, кроме σ_{tp}, достаточно близки, у σ_{tp} наблюдается расхождение значений для настоящих и сгенерированных данных. Для третьего входного параметра drift length наблюдаются расхождения распределений всех метрик, наиболее заметное расхождение у μ_p, σ_p^2 и σ_{tp}. Для четвертого входного параметра drift length fraction распределения метрик σ_t^2, σ_p^2 и σ_{tp} приближаются достаточно точно, в распределениях метрик μ_t, μ_p есть достаточное расхождение. Для пятого входного параметра pad coordinate fraction распределения всех метрик, кроме σ_p^2 и σ_{tp} приближаются хорошо, в распределениях метрик σ_p^2 и σ_{tp} есть расхождения на настоящих и сгенерированных данных.

Так как для метрики σ_{tp} наблюдаются расхождения в распределениях для настоящих и сгенерированных данных, то было решено использовать эту метрику для обучения суррогатного признака. При помощи приближения метрики суррогатным признаком в дискриминаторе будет с большим вкладом в итоговое предсказание учитываться значение данной метрики, что позволит улучшить воспроизводимость данной метрики на сгенерированных

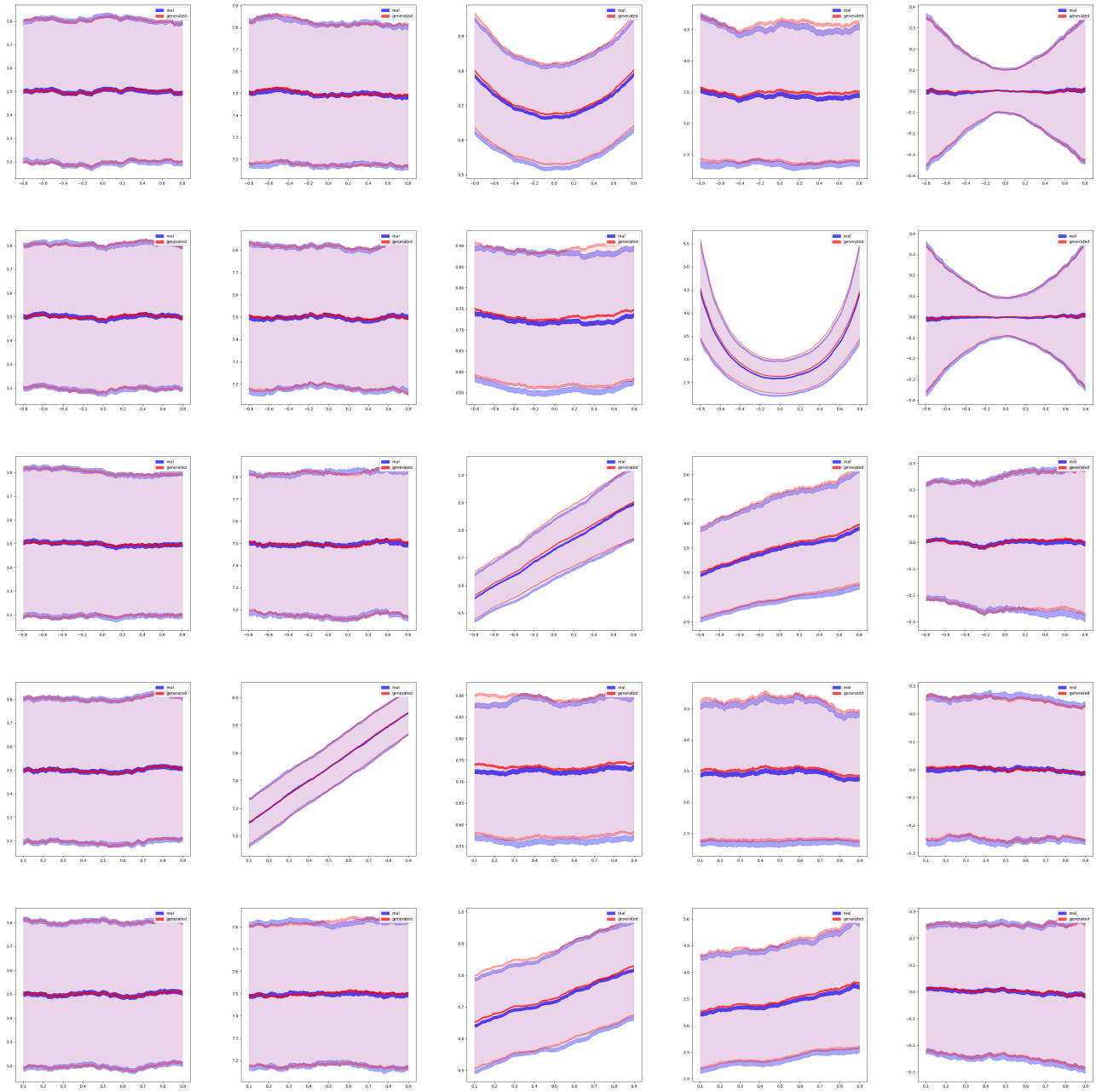


Рис. 4.3: Распределение на настоящих (real) и сгенерированных (generated) данных метрик μ_t, μ_p, σ_t^2, σ_p^2 и σ_{tp} (по строкам) в зависимости от входных параметров crossing angle, dip angle, drift length, drift length fraction, pad coordinate fraction (по столбцам) для базовой модели.

данных и улучшить общее качество генерации.

5 Эксперименты

5.1 Выбор приближаемой метрики и архитектуры регрессионной модели

Будем называть модель, вычисляющую суррогатный признак, регрессионной моделью (сокр. регрессор). Регрессионная модель на вход получает выход одного из слоёв дискриминатора GAN и выдает для каждого объекта число, приближающее значение выбранной метрики на данном объекте. Затем это значение добавляется к выходу одного из следующих слоев дискриминатора. При подсчете функции потерь считается сумма функции потерь дискриминатора, которая в данном случае выбрана как метрика Вассерштейна с градиентным штрафом, и среднеквадратичного отклонения значений, выдаваемых регрессором, от реальных значений метрики на каждом объекте.

В качестве приближаемой регрессионной моделью метрики было решено выбрать ковариацию распределений значений вдоль осей изображения (σ_{tp}). Для каждой исследуемой модели проанализировано качество на валидационной выборке. Метриками качества выступают χ^2, сравнивающая распределение выбранной метрики σ_{tp} реальных и сгенерированных объектов, и суммарная χ^2.

В качестве регрессионной модели было решено выбрать модель со сверточной архитектурой, так как дискриминатор базовой модели тоже имеет сверточную архитектуру. В рамках решаемой задачи у регрессионной модели не должно быть слишком много параметров, иначе есть большая вероятность переобучения. Переобучение может возникнуть, так как приближаемая метрика является относительно несложной функцией от объектов. Было решено выбрать сверточную архитектуру с 7 слоями. Первый слой является сверточным слоем с ядром размера 5 и нулевым паддингом, затем идут слой активации ELU и слой Dropout, зануляющий 2% элементов. Четвертый слой является сверточным слоем с ядром размера 4 и нулевым паддингом, затем идут слой активации ELU и слой Dropout, зануляющий 2% элементов. Седьмой слой является линейным, принимающим на вход по 16 параметров для каждого объекта и дающим на выход по 1 параметру для каждого объекта. Примеры архитектур дискриминатора моделей, использующих описанный регрессор, представлены на Рисунках 7.2, 7.3, 7.4, 7.5 и 7.6. В разделах 5.3 и 5.4 исследуется качество генерации при использовании данных моделей.

5.2 Выбор данных для обучения регрессионной модели

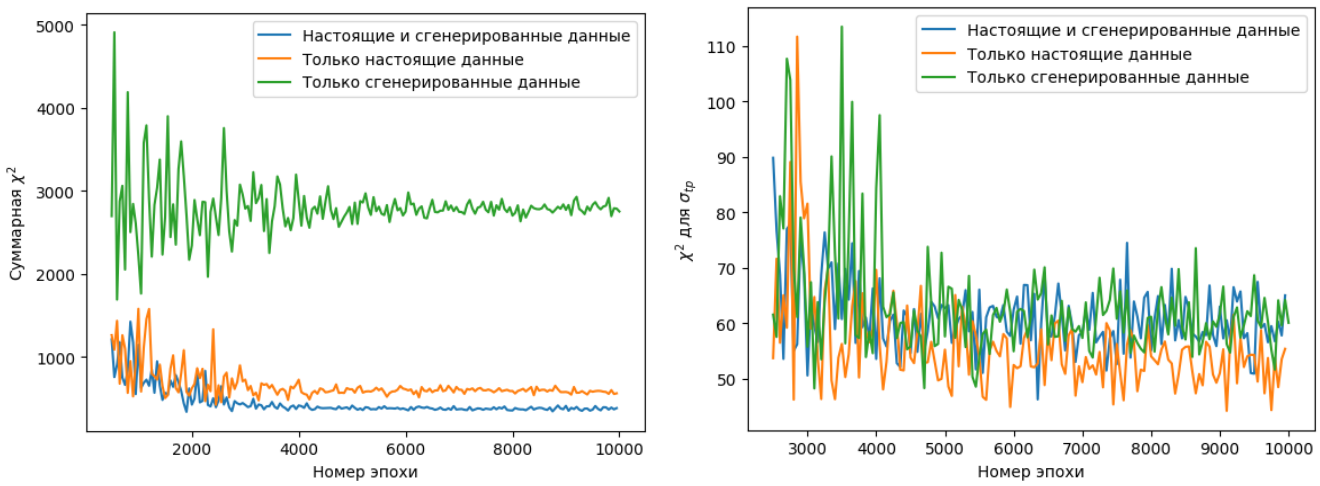
В качестве функции потерь регрессионной модели было использовано среднеквадратичное отклонение значений, выдаваемых регрессионной моделью, от настоящих значений метрики σ_{tp} на каждом объекте. В функции потерь может быть учтена среднеквадратичная ошибка только на настоящих данных, только на сгенерированных данных или на настоящих и на сгенерированных данных. От выбора данных для подсчета среднеквадратической ошибки может зависеть качество регрессионной модели.

Учёт ошибки на сгенерированных данных может приводить к худшему результату на начальных эпохах, когда распределение сгенерированных данных далеко от распределения настоящих данных. Это может также негативно отразиться на итоговых результатах обучения. С другой стороны, при вычислении среднеквадратической ошибки и на настоящих, и на сгенерированных данных, качество генерации может повыситься из-за использования большего объема данных. Поэтому было решено для одной из подробно исследуемых далее архитектур моделей проанализировать, как зависит качество генерации от данных, используемых для подсчета ошибки регрессора.

В данном эксперименте была использована модель, архитектура которой представлена на Рисунке 7.4. Данная модель в дальнейшем исследовании обозначена как Модель 3. Коэффициент λ_r, с которым среднеквадратичная ошибка предсказаний регрессионной модели входит в итоговую функцию потерь дискриминатора для данного эксперимента был взят равным 10^9. Исследование влияния гиперпараметров на качество генерации описано в разделе 5.4. Данная модель была обучена с подсчетом среднеквадратичной ошибки регрессора на настоящих, сгенерированных и на настоящих и сгенерированных данных.

Эксперимент показал (см. Рисунок 5.1), что значение суммарной метрики качества χ^2 хуже при использовании только ошибки регрессора на настоящих данных по сравнению с использование ошибки регрессора и на настоящих, и на сгенерированных данных. При использовании ошибки регрессора только на сгенерированных данных модель выдает качество гораздо хуже, чем при учете ошибки на настоящих данных или на настоящих и сгенерированных данных.

Таким образом, было решено в процессе обучения моделей учитывать среднеквадратичную ошибку регрессора и на настоящих, и на сгенерированных данных.



(a) Суммарная метрика χ^2

(b) Метрика χ^2 для σ_{tp}

Рис. 5.1: Значения метрик качества Модели 3 при $\lambda_r = 10^9$ в зависимости от номера эпохи при учёте в функции потерь ошибки регрессора на настоящих, сгенерированных и настоящих и сгенерированных данных.

5.3 Исследование и сравнение архитектур моделей

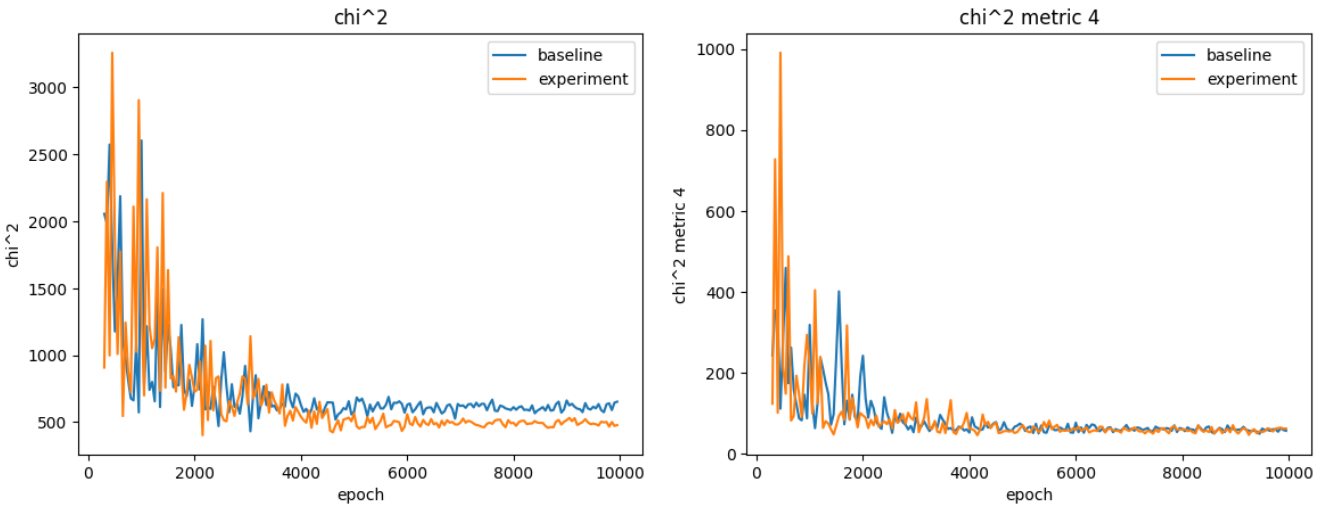
Было проанализировано качество генерации при подаче регрессионной модели на вход выходов различных слоев базового дискриминатора, а также при подаче выходов регрессионной модели в качестве входа для различных слоев дискриминатора.

Первой была обучена модель, архитектура дискриминатора которой представлена на Рисунке 7.2 (обозначим данную модель как Модель 1). На вход регрессора подаётся выход первого Max Pooling слоя дискриминатора, выход регрессора прибавляется к выходу одного из следующих Dropout слоев дискриминатора.

Суммарная метрика качества χ^2 улучшилась по сравнению с базовой моделью до 495.66 ± 18.07. Однако χ^2 для приближаемой метрики σ_{tp} осталась примерно на том же уровне и равна 58.495 ± 4.914 (см. Рисунок 5.2).

В следующем эксперименте было выбрано подавать выход регрессора как вход к линейной голове дискриминатора, то есть складывать его с выходом последнего Dropout слоя дискриминатора. Ожидается, что такая архитектура будет давать лучший результат, так как линейная голова комбинирует все извлеченные из данных признаки и делает предсказания на их основе. Архитектура дискриминатора модели представлена на Рисунке 7.3 (обозначим данную модель как Модель 2).

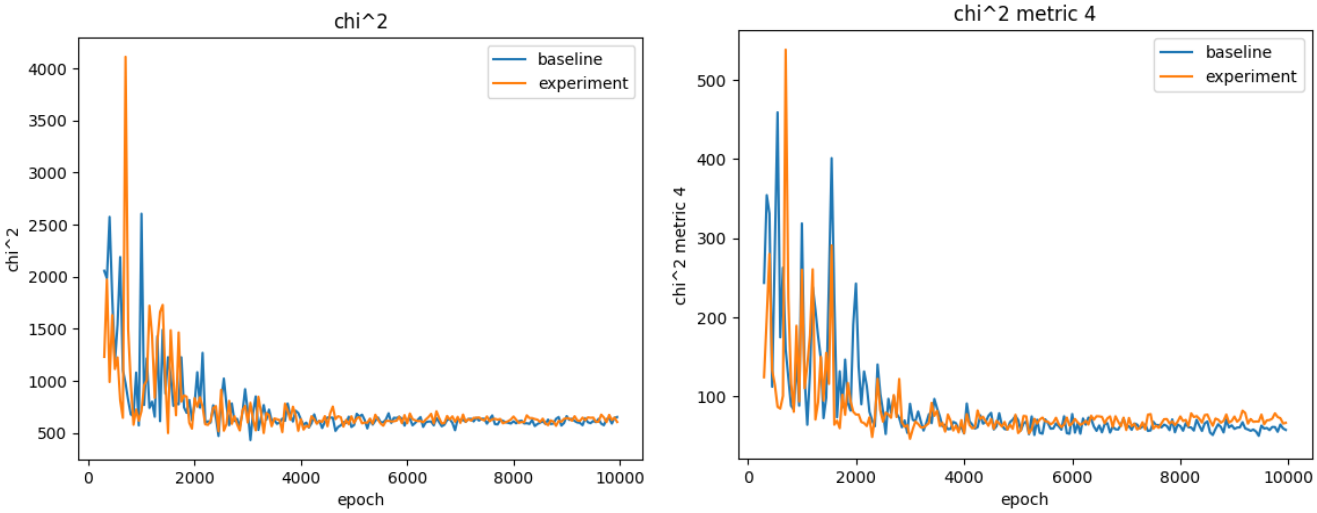
С помощью данной архитектуры не удалось улучшить значение суммарной метрики качества χ^2, она стала равна 634.25 ± 21.74 (см. Рисунок 5.3), и χ^2 для приближаемой метрики σ_{tp} стала равна 71.024 ± 4.762, что хуже, чем у базовой модели.



(a) Суммарная метрика χ^2

(b) Метрика χ^2 для σ_{tp}

Рис. 5.2: Значения метрик качества Модели 1 в зависимости от номера эпохи.



(a) Суммарная метрика χ^2

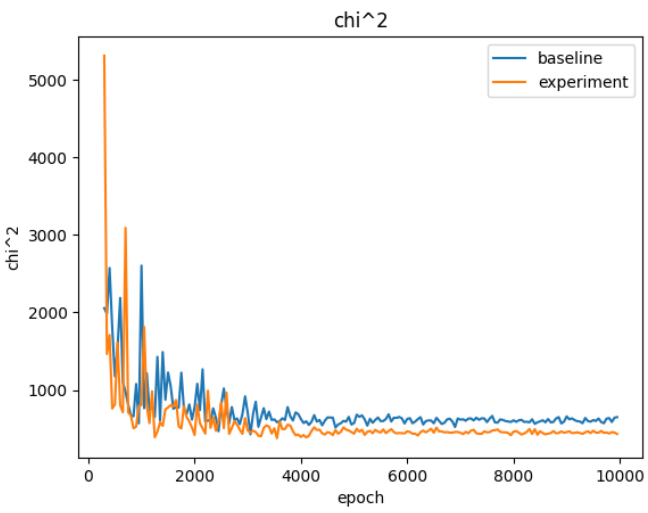
(b) Метрика χ^2 для σ_{tp}

Рис. 5.3: Значения метрик качества Модели 2 в зависимости от номера эпохи.

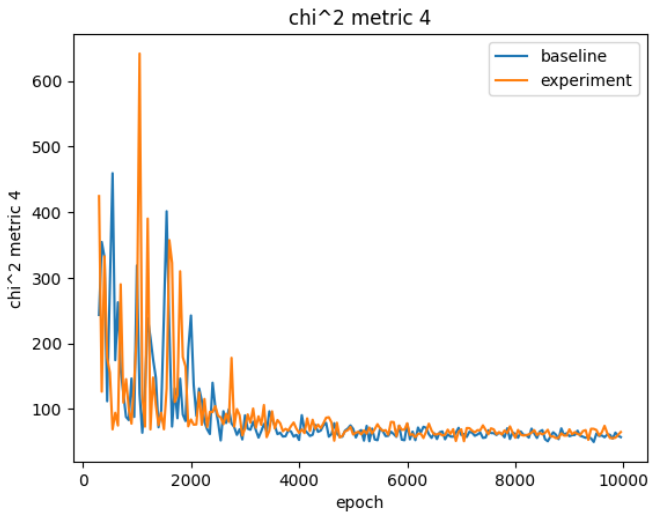
Далее были исследованы архитектуры, в которых на вход регрессионной модели подаются выходы различных слоев дискриминатора. В качестве следующего эксперимента была выбрана архитектура дискриминатора модели, представленная на Рисунке 7.4 (обозначим данную модель как Модель 3).

При помощи такой модели удалось добиться улучшения суммарной метрики качества χ^2 до 454.27 ± 11.32, χ^2 для отдельной метрики σ_{tp}, приближаемой регрессионной моделью, стала равна 63.421 ± 5.495, что немного хуже, чем у базовой модели (см. Рисунок 5.4).

Была также рассмотрена архитектура, представленная на Рисунке 7.5, где на вход регрессору подаётся выход более позднего слоя Dropout (обозначим данную модель как Модель 4).



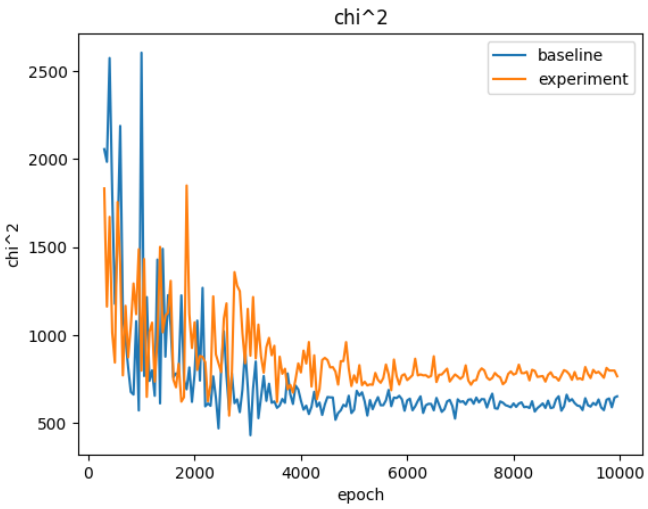
(a) Суммарная метрика χ^2



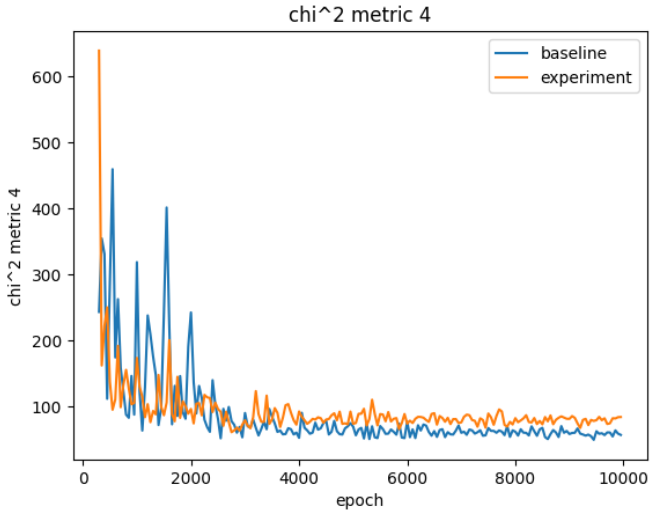
(b) Метрика χ^2 для σ_{tp}

Рис. 5.4: Значения метрик качества Модели 3 в зависимости от номера эпохи.

Использование данной архитектуры привело к ухудшению как χ^2 для рассматриваемой метрики σ_{tp} до 79.617 ± 4.604, так и суммарной χ^2 до 779.85 ± 22.07 (см. Рисунок 5.5).



(a) Суммарная метрика χ^2

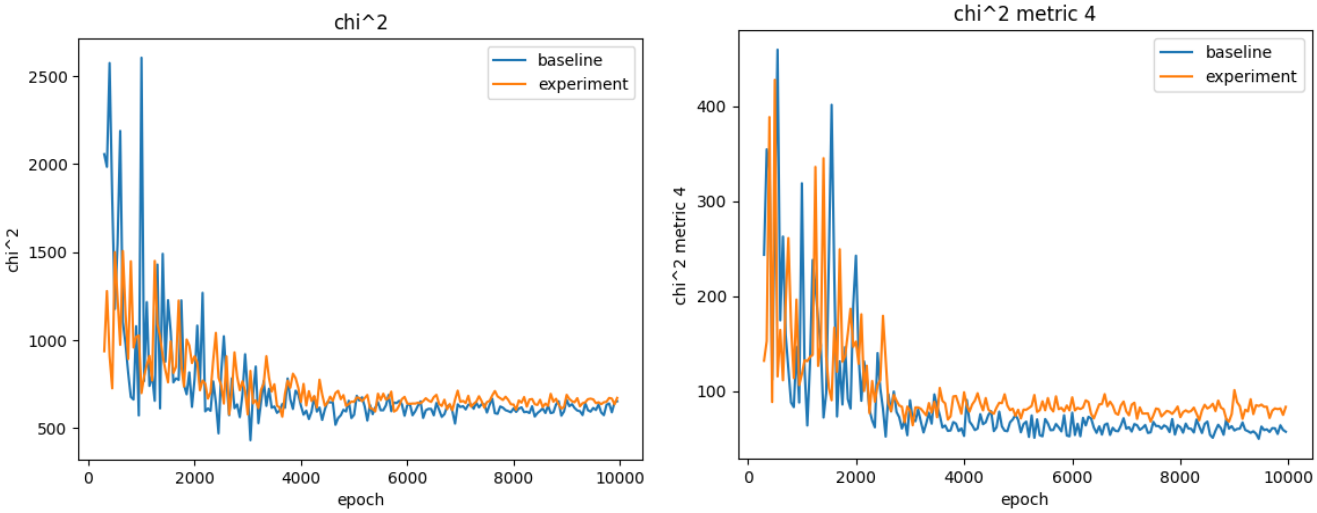


(b) Метрика χ^2 для σ_{tp}

Рис. 5.5: Значения метрик качества Модели 4 в зависимости от номера эпохи.

Было также решено протестировать модель, которая конкатенирует значения, выдаваемые регрессором, со значениями, выдаваемыми основными сверточными слоями дискриминатора. Ожидается, что такая модель будет работать хуже, так как тело дискриминатора выдаёт 64 признака, а регрессионная модель только 1 признак, и при применении линейного слоя этот признак будет вносить незначительный вклад в итоговое предсказание. Архитектура дискриминатора модели представлена на Рисунке 7.6 (обозначим данную модель как Модель 5).

Данная архитектура привела к ухудшению суммарной метрики качества χ^2 до 654.79± 15.70, а также к ухудшению χ^2 для метрики σ_{tp}, приближаемой регрессором, до 82.366±6.620 (см. Рисунок 5.6).



(a) Суммарная метрика χ^2

(b) Метрика χ^2 для σ_{tp}

Рис. 5.6: Значения метрик качества Модели 5 в зависимости от номера эпохи.

Значения метрик качества для различных рассмотренных архитектур представлены в Таблице 5.1. Значения усреднены по эпохам с 9 000 по 10 000 для более стабильной оценки.

Таблица 5.1: Метрики качества представленных моделей.

	суммарная χ^2	χ^2 для σ_{tp}
Базовая модель (Рисунок 7.1)	615.72 ±25.56	58.912 ±3.530
Модель 1 (Рисунок 7.2)	**495.66 ±18.07**	**58.495 ±4.914**
Модель 2 (Рисунок 7.3)	634.25 ±21.74	71.024 ±4.762
Модель 3 (Рисунок 7.4)	**454.27 ±11.32**	**63.421 ±5.495**
Модель 4 (Рисунок 7.5)	779.85 ±22.07	79.617 ±4.604
Модель 5 (Рисунок 7.6)	654.79 ±15.70	82.366 ±6.620

С помощью Модели 1 и Модели 3 удалось добиться лучшего качества генерации, чем было у базовой модели.

Исследуем качество приближения метрики σ_{tp} регрессионными моделями в каждом случае. Графики зависимости среднеквадратичной ошибки регрессионной модели от номера эпохи для валидационной выборки представлены на Рисунке 5.7.

По графикам видно, что у моделей с меньшей среднеквадратичная ошибкой регрессора лучше χ^2 для приближаемой метрики σ_{tp}. Наилучшие значение суммарной χ^2 также показывают Модель 1 и Модель 3, у которых наименьшая среднеквадратичная ошибка регрессора из представленных моделей.

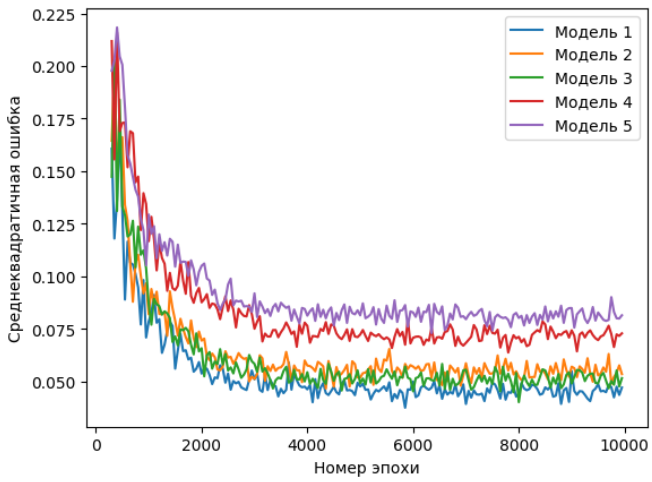
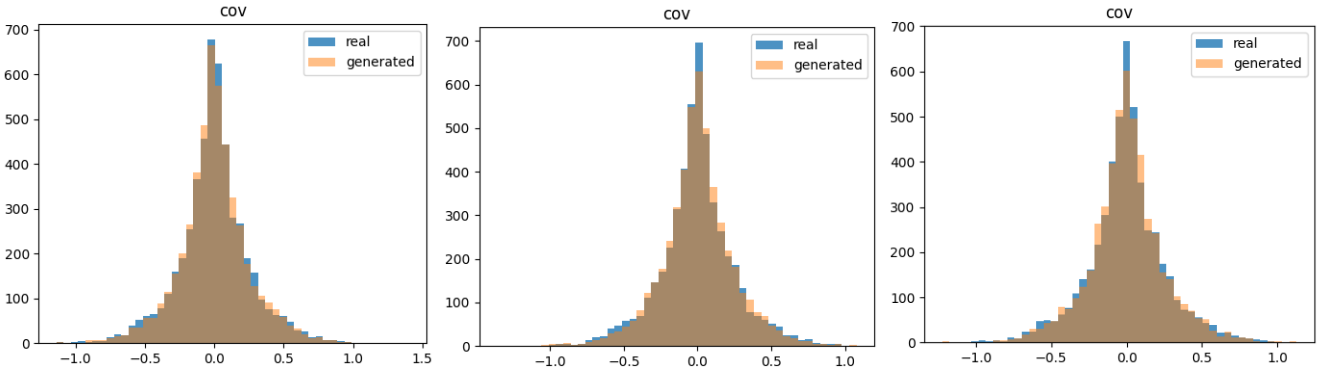


Рис. 5.7: График зависимости среднеквадратичной ошибки регрессора на валидационной выборке от номера эпохи для Модели 1, Модели 2, Модели 3, Модели 4, Модели 5.

Рассмотрим распределения метрики σ_{tp} на настоящих и сгенерированных этими моделями данных, изображенные на Рисунке 5.8. В распределениях не наблюдается явных различий, во всех случаях распределение на настоящих данных близко к распределению на сгенерированных данных.

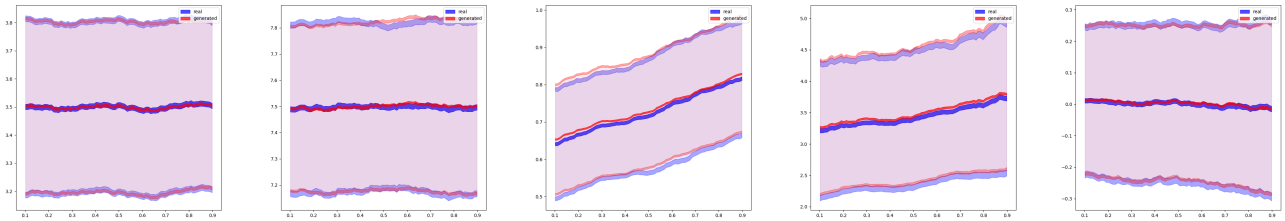


(a) Базовая модель (b) Модель 1 (c) Модель 3

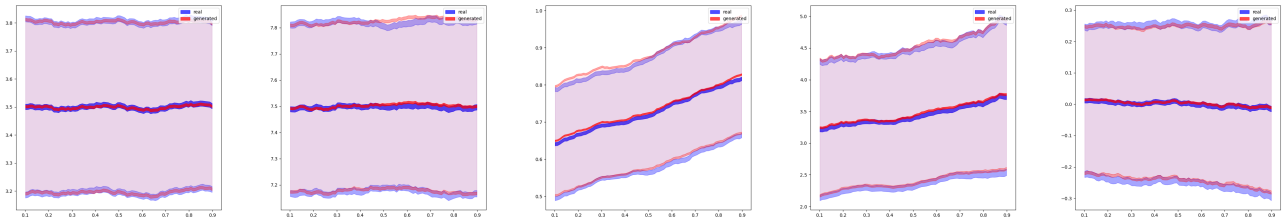
Рис. 5.8: Распределение на настоящих (real) и сгенерированных (generated) данных метрики σ_{tp} (обозначена *cov*) для Базовой модели, Модели 1 и Модели 3.

Рассмотрим графики распределение параметра σ_{tp} в зависимости от входных параметров, представленные на Рисунке 5.9. Для первого входного параметра crossing angle и для второго входного параметра dip angle распределения у базовой модели, Модели 1 и Модели 3 близки друг к другу. Для третьего входного параметра drift length у Модели 1 распределение на сгенерированных данных ближе к распределению на настоящих данных, чем у базовой модели, а у Модели 3 эти распределения еще ближе. В случае четвертого входного параметра drift length fraction у Модели 3 также получаются самые близкие распределения на сгенерированных и реальных данных, следующей по качеству идёт Модель 1. Для пятого

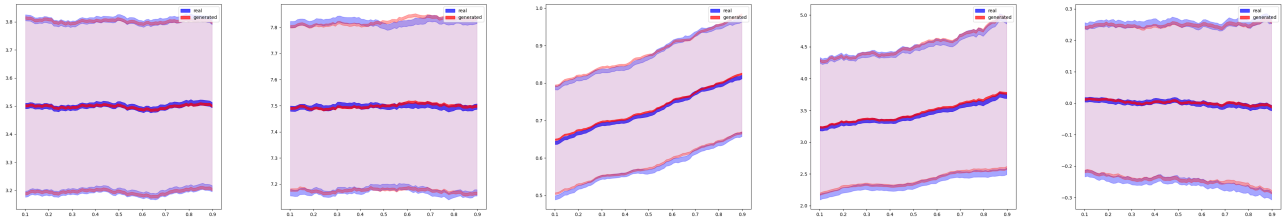
входного параметра pad coordinate fraction распределения всех трех рассмотренных моделей достаточно близки друг к другу.



(a) Базовая модель



(b) Модель 1



(c) Модель 3

Рис. 5.9: Распределение на настоящих (real) и сгенерированных (generated) данных метрики σ_{tp} в зависимости от входных параметров crossing angle, dip angle, drift length, drift length fraction, pad coordinate fraction (по столбцам) для Базовой модели, Модели 1 и Модели 3.

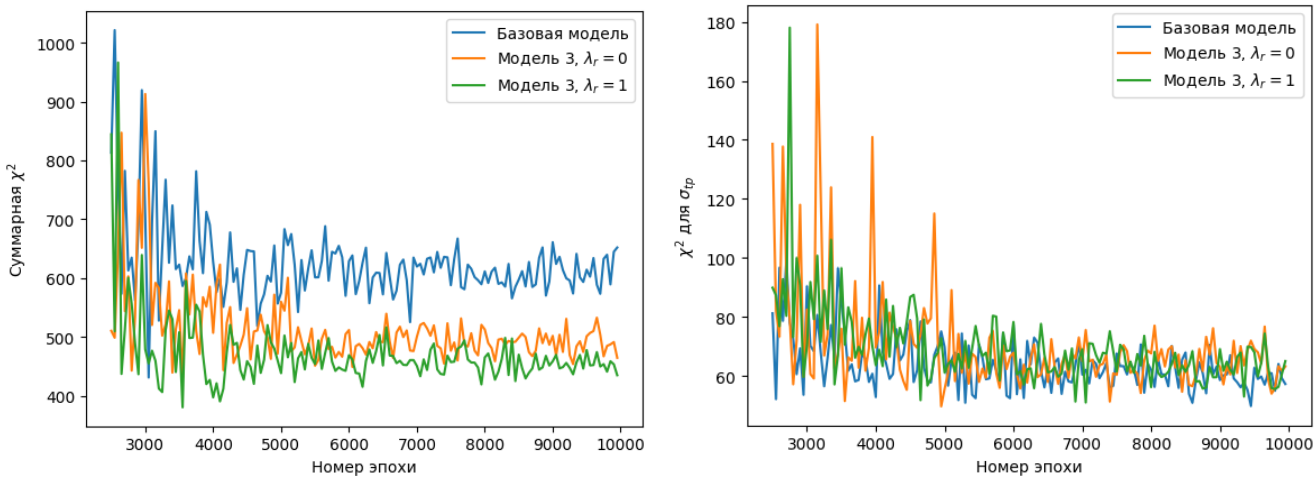
Таким образом, наилучшее качество из представленных моделей показала Модель 3, поэтому было решено использовать её архитектуру в дальнейших экспериментах.

5.4 Изучение влияния гиперпараметров на качество генерации

Наиболее важным гиперпараметром является коэффициент λ_r, с которым среднеквадратичная ошибка предсказаний регрессионной модели входит в итоговую функцию потерь. Были исследованы зависимость качества результатов модели в от коэффициента λ_r, а также зависимость качества приближения регрессионной моделью метрики σ_{tp} от коэффициента λ_r.

На улучшение качества генерации в сравнении с базовой моделью может влиять не только добавление среднеквадратичной ошибки регрессионной модели в функцию потерь, а также увеличение общего количества параметров модели. Поэтому было исследовано ка-

чество модели, имеющей такую же архитектуру, как Модель 3 (Рисунок 7.4), но в которой к функции потерь дискриминатора не добавляется среднеквадратичная ошибка регрессионной модели. Эксперимент показал, что качество модели действительно становится лучше, чем у базовой модели, из-за увеличения количества параметров. Однако качество модели с $\lambda_r = 1$ лучше, чем качество модели с $\lambda_r = 0$, то есть добавление среднеквадратичной ошибки регрессионной модели к функции потерь дискриминатора положительно сказывается на качестве генерации (см. Рисунок 5.10)



(a) Суммарная метрика χ^2

(b) Метрика χ^2 для σ_{tp}

Рис. 5.10: Значения метрик качества базовой модели и Модели 3 при $\lambda_r = 0$ и $\lambda_r = 1$ в зависимости от номера эпохи.

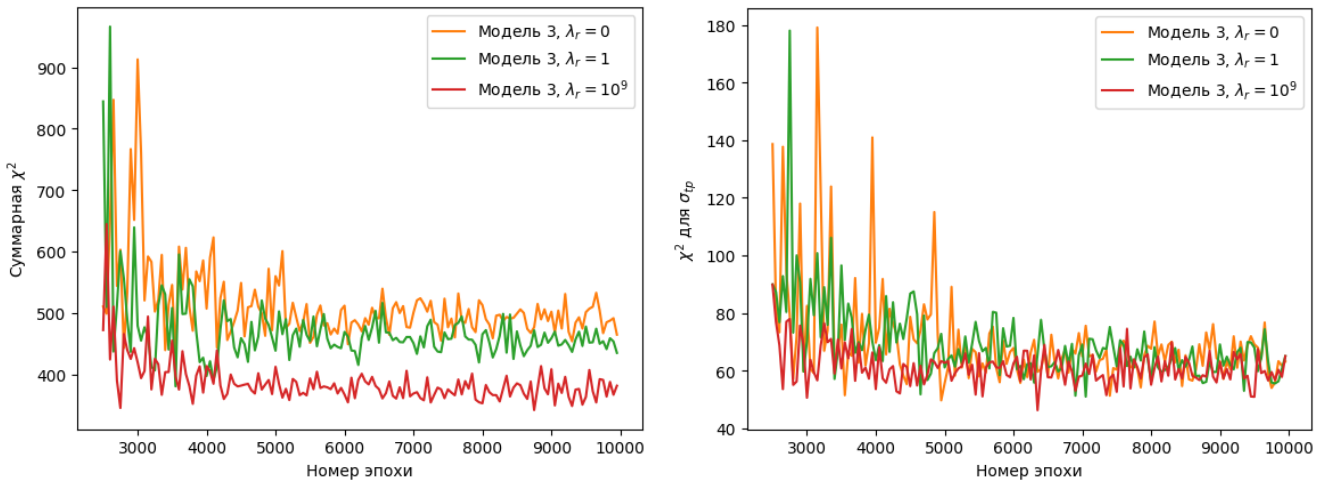
Было решено увеличить коэффициент λ_r, чтобы заставить модель сильнее фокусироваться на точном воспроизведении метрики σ_{tp}. Был проведён эксперимент с $\lambda_r = 10^9$, качество модели заметно улучшилось по сравнению с предыдущими экспериментами (см. Рисунок 5.11).

В Таблице 5.2 представлены значения суммарной χ^2 и значения χ^2 для σ_{tp} в зависимости от коэффициента λ_r. Значения усреднены по эпохам с 9 000 по 10 000 для большей стабильности результата.

Таблица 5.2: Метрики качества представленных моделей.

	суммарная χ^2	χ^2 для σ_{tp}
Базовая модель	615.72 ±25.56	58.912 ±3.530
Модель 3, $\lambda_r = 0$	490.30 ±20.94	64.304 ±5.729
Модель 3, $\lambda_r = 1$	454.27 ±11.32	63.421 ±5.495
Модель 3, $\lambda_r = 10^9$	**375.11 ±17.77**	**59.696 ±4.463**

На Рисунке 5.12 показано, как менялась во время обучения среднеквадратичная ошибка регрессионной модели при разных λ_r. Видно, что при $\lambda_r = 10^9$ модель ведёт себя более



(a) Суммарная метрика χ^2

(b) Метрика χ^2 для σ_{tp}

Рис. 5.11: Значения метрик качества Модели 3 при $\lambda_r = 0$, $\lambda_r = 1$ и $\lambda_r = 10^9$ в зависимости от номера эпохи.

стабильно во время обучения и метрика σ_{tp} приближается лучше, чем при $\lambda_r = 1$.

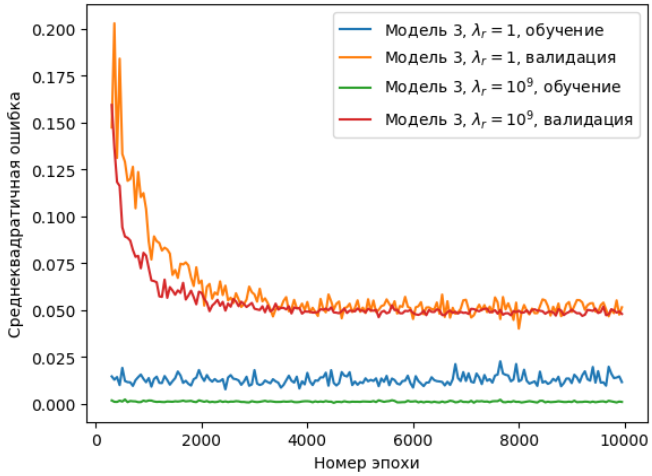
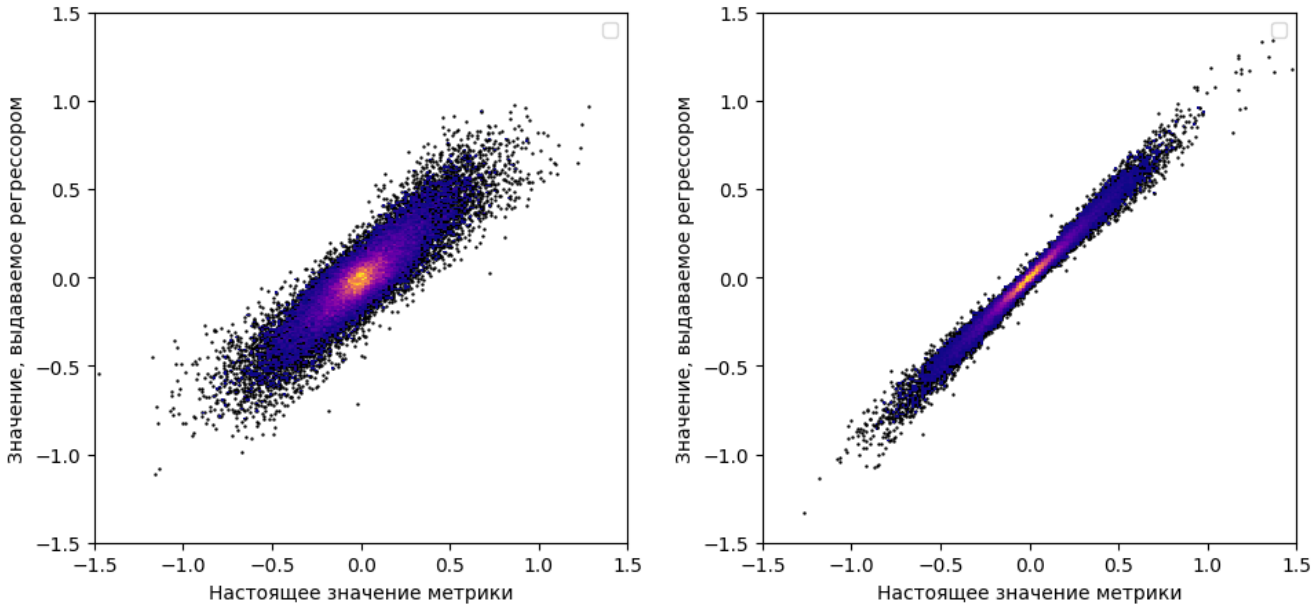


Рис. 5.12: График зависимости среднеквадратичной ошибки регрессора от номера эпохи для Модели 3 при $\lambda_r = 1$ и $\lambda_r = 10^9$.

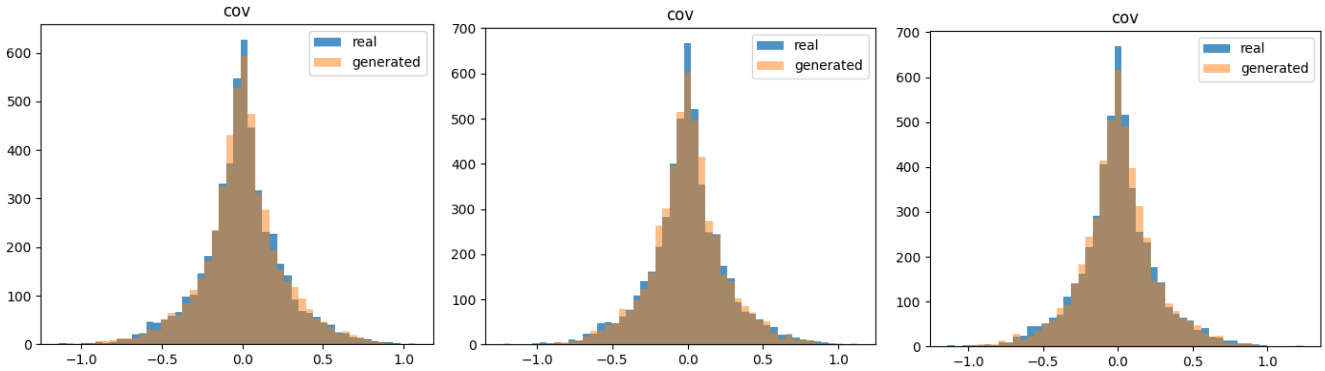
На Рисунке 5.13 показано, насколько точно приближается метрика σ_{tp} при помощи регрессионной модели. По оси абсцисс отмечены настоящие значения метрики, по оси ординат отмечены значения, выдаваемые регрессионной моделью. Для $\lambda_r = 10^9$ график получает более вытянутым и близким к диагональному, чем для $\lambda_r = 1$. Это означает, что регрессионная модель воспроизводит значение метрики лучше.



(a) Модель 3, $\lambda_r = 1$ (b) Модель 3, $\lambda_r = 10^9$

Рис. 5.13: Распределение настоящих значений метрики σ_{tp} и значений, выдаваемых регрессором для Модели 3 при $\lambda_r = 1$ и $\lambda_r = 10^9$.

Рассмотрим распределения значений метрики σ_{tp} на настоящих и сгенерированный данных, представленные на Рисунке 5.14. Видно, что при различных коэффициентах λ_r распределения на настоящих и сгенерированных данных имеют большую площадь пересечения, то есть достаточно близки друг к другу.

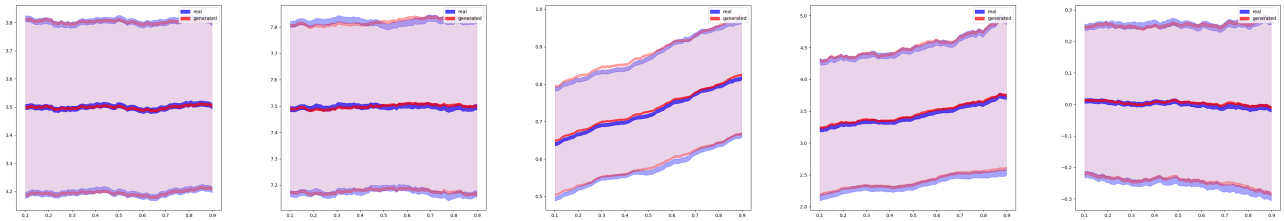


(a) Модель 3, $\lambda_r = 0$ (b) Модель 3, $\lambda_r = 1$ (c) Модель 3, $\lambda_r = 10^9$

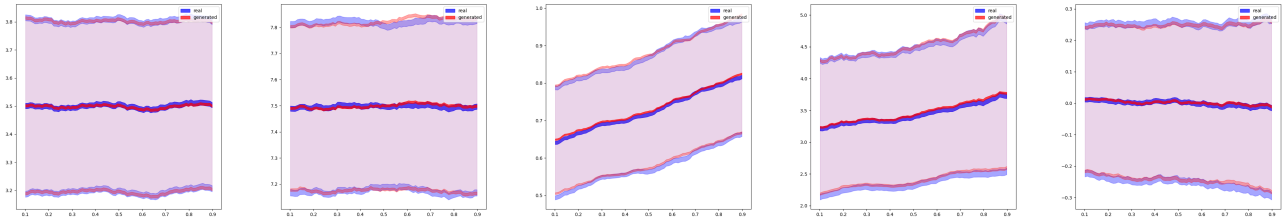
Рис. 5.14: Распределение на настоящих (real) и сгенерированных (generated) данных метрики σ_{tp} (обозначена *cov*) для Модели 3 при $\lambda_r = 0$, $\lambda_r = 1$ и $\lambda_r = 10^9$.

Рассмотрим графики распределение параметра σ_{tp} в зависимости от входных параметров, представленные на Рисунке 5.15. Для первого входного параметра crossing angle распределения метрики σ_{tp} на настоящих и сгенерированных данных достаточно близки для всех рассматриваемых моделей. Для второго входного параметра dip angle наблюдается сближение распределений на настоящих и сгенерированных данных с уменьшением параметра λ_r.

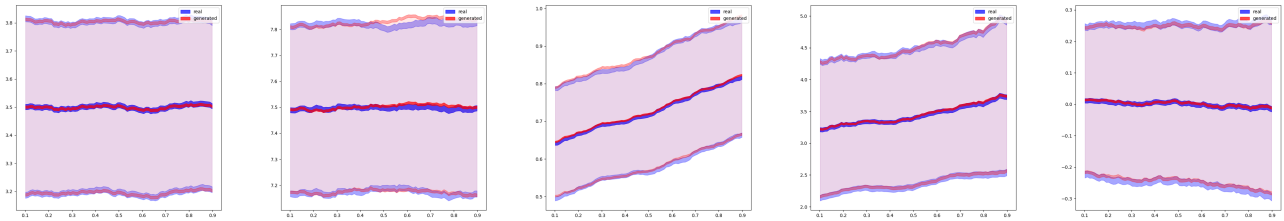
Для третьего входного параметра drift length и четвертого входного параметра drift length fraction наблюдается сближение распределений на настоящих и сгенерированных данных с увеличением параметра λ_r. Для пятого входного параметра pad coordinate fraction наиболее близки распределения на настоящих и сгенерированных данных при $\lambda_r = 10^9$, следующим по качеству приближения идет модель с $\lambda_r = 0$.



(a) Модель 3, $\lambda_r = 0$



(b) Модель 3, $\lambda_r = 1$



(c) Модель 3, $\lambda_r = 10^9$

Рис. 5.15: Распределение на настоящих (real) и сгенерированных (generated) данных метрики σ_{tp} в зависимости от входных параметров crossing angle, dip angle, drift length, drift length fraction, pad coordinate fraction (по столбцам) для Модели 3 при $\lambda_r = 0$, $\lambda_r = 1$ и $\lambda_r = 10^9$.

Таким образом, эксперименты показали, что параметр λ_r значительно влияет на качество генерации, поэтому при выборе оптимальной модели необходимо найти его наиболее подходящее значение.

Для выбранной архитектуры дискриминатора с добавлениям регрессионной модели (представлена на Рисунке 7.4) наилучшее качество генерации было продемонстрировано при большом коэффициенте λ_r ($\lambda_r = 10^9$), с которым среднеквадратичная ошибка регрессора входит в функцию потерь. При использовании такого коэффициента удалось достичь наиболее точного приближения регрессионной моделью настоящих значений метрики σ_{tp} и более стабильного процесса обучения регрессионной модели, а также лучшего воспроизведения

значений метрики σ_{tp} на сгенерированных данных в зависимости от 5 входных параметров модели.

6 Заключение

В данной работе была исследована возможность применения суррогатных признаков для улучшения качества генерации модели, реконструирующей отклик камеры временной проекции в многоцелевом детекторе, который является частью установки ионного коллайдера на базе нуклотрона. В качестве базовой модели была использована генеративно-состязательная модель, предложенная в статье [6]. Была произведена оценка качества базовой модели и выбрана метрика σ_{tp} для приближения регрессионной моделью. Было исследовано, как влияет на качество генерации использование настоящих и сгенерированных данных при подсчете ошибки регрессионной модели. Было выяснено, что наилучшее качество генерации достигается при учете ошибки регрессионной модели как на настоящих, так и на сгенерированных данных. Были исследованы различные архитектуры дискриминатора генеративно-состязательной модели с добавлением регрессионной модели. Наилучшее качество генерации было показано моделью, имеющей архитектуру дискриминатора, представленную на Рисунке 7.4. При использовании данной модели удалось улучшить суммарную метрику качества χ^2 на 26.22% по сравнению с базовой моделью. Далее была исследована зависимость качества генерации от коэффициента λ_r, с которым ошибка регрессионной модели входит в функцию потерь дискриминатора. Наилучшего качества удалось добиться при использовании $\lambda_r = 10^9$. При таком значении параметра удалось улучшить суммарную метрику качества χ^2 на 39.07% по сравнению с базовой моделью.

7 Направления дальнейшей работы

В рамках дальнейшей работы возможна при использовании дополнительных ресурсов более точная и стабильная оценка качества предложенных моделей путем проведения повторных экспериментов и усреднения результатов. Также возможно проведение большего количества экспериментов для нахождения оптимальных значений гиперпараметров и более точной оценки их влияния на итоговое качество генерации. Может быть проведено дальнейшее исследование влияния качества регрессионной модели на качество GAN, а также влияние коэффициента, с которым функция потерь регрессора входит в общую функцию потерь дискриминатора, на качество генерации. Более того, возможно исследование способов

динамического подбора гиперпараметров во время обучения. Возможно также исследование качества генерации при использовании других архитектур базовой и регрессионной моделей, и при использовании других метрик в качестве суррогатных признаков. Кроме этого, возможен анализ других метрик качества генерации и влияния на них использования суррогатных признаков. Также можно исследовать применимость суррогатных признаков для других задач генерации.

Список литературы

[1] Martin Arjovsky, Soumith Chintala и Leon Bottou. “Wasserstein Generative Adversarial Networks”. В: *34th International Conference on Machine Learning* 70 (2017), с. 214—223.

[2] Kamil Deja1, Tomasz Trzcin´ski1 и Łukasz Graczykowski. “Generative models for fast cluster simulations in the TPC for the ALICE experiment”. В: *EPJ Web of Conferences* 214.06003 (2019).

[3] Denis Derkach и др. “Cherenkov Detectors Fast Simulation Using Neural Networks”. В: *arXiv preprint, arXiv:1903.11788, version 1* (2019).

[4] Ian J. Goodfellow и др. “Generative Adversarial Networks”. В: *arXiv preprint, arXiv:1406.2661, version 1* (2014).

[5] Ishaan Gulrajani и др. “Improved Training of Wasserstein GANs”. В: *arXiv preprint, arXiv:1704.00028, version 3* (2017).

[6] A. Maevskiy и др. “Simulating the Time Projection Chamber Responses at the MPD Detector Using Generative Adversarial Networks”. В: *The European Physical Journal C* 81.599 (2021).

[7] Artem Maevskiy и др. “Fast Data-Driven Simulation of Cherenkov Detectors Using Generative Adversarial Networks”. В: *arXiv preprint, arXiv:1905.11825, version 2* (2019).

[8] Saraiva Marcus и др. “Data-driven Full-waveform Inversion Surrogate using Conditional Generative Adversarial Networks”. В: *arXiv preprint, arXiv:2105.00100, version 1* (2021).

[9] Luke de Oliveira, Michela Paganini и Benjamin Nachman. “Learning Particle Physics by Example: Location-Aware Generative Adversarial Networks for Physics Synthesis”. В: *arXiv preprint, arXiv:1701.05927, version 2* (2017).

[10] Suihong Song и др. “GANSim-surrogate: Anintegrated framework for conditional geomodelling and uncertainty analysis”. В: *Earth arXiv preprint, version 1, doi:10.31223/X5N357* (2022).

[11] Jincheng Zhang и Xiaowei Zhao. “Wind Farm Wake Modeling Based on Deep Convolutional Conditional Generative Adversarial Network”. В: *Energy* 238, Part B (2022).

Приложение А: Архитектуры моделей

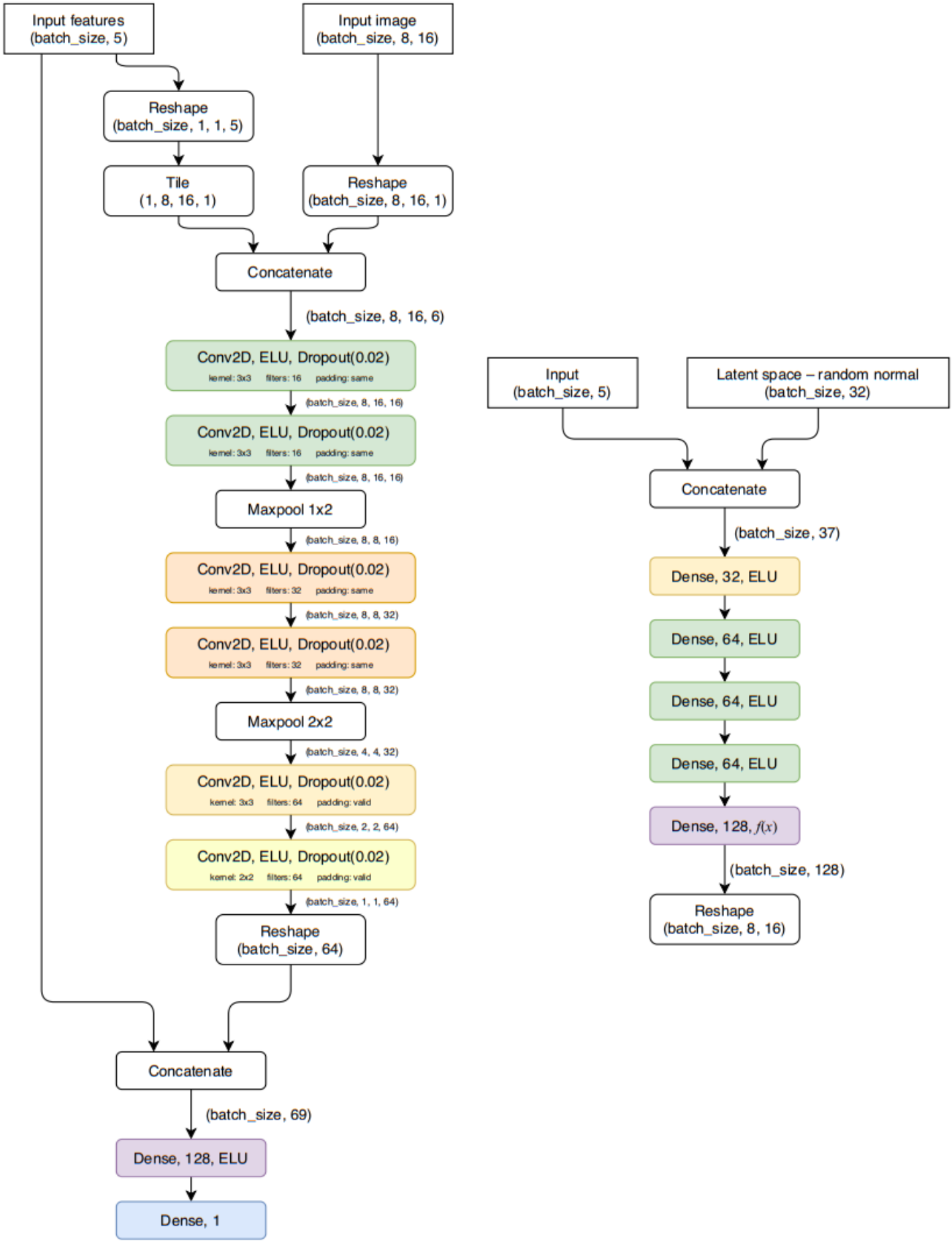


Рис. 7.1: Архитектура базовой модели: генератор (справа) и дискриминатор (слева) (иллюстрация из [6])

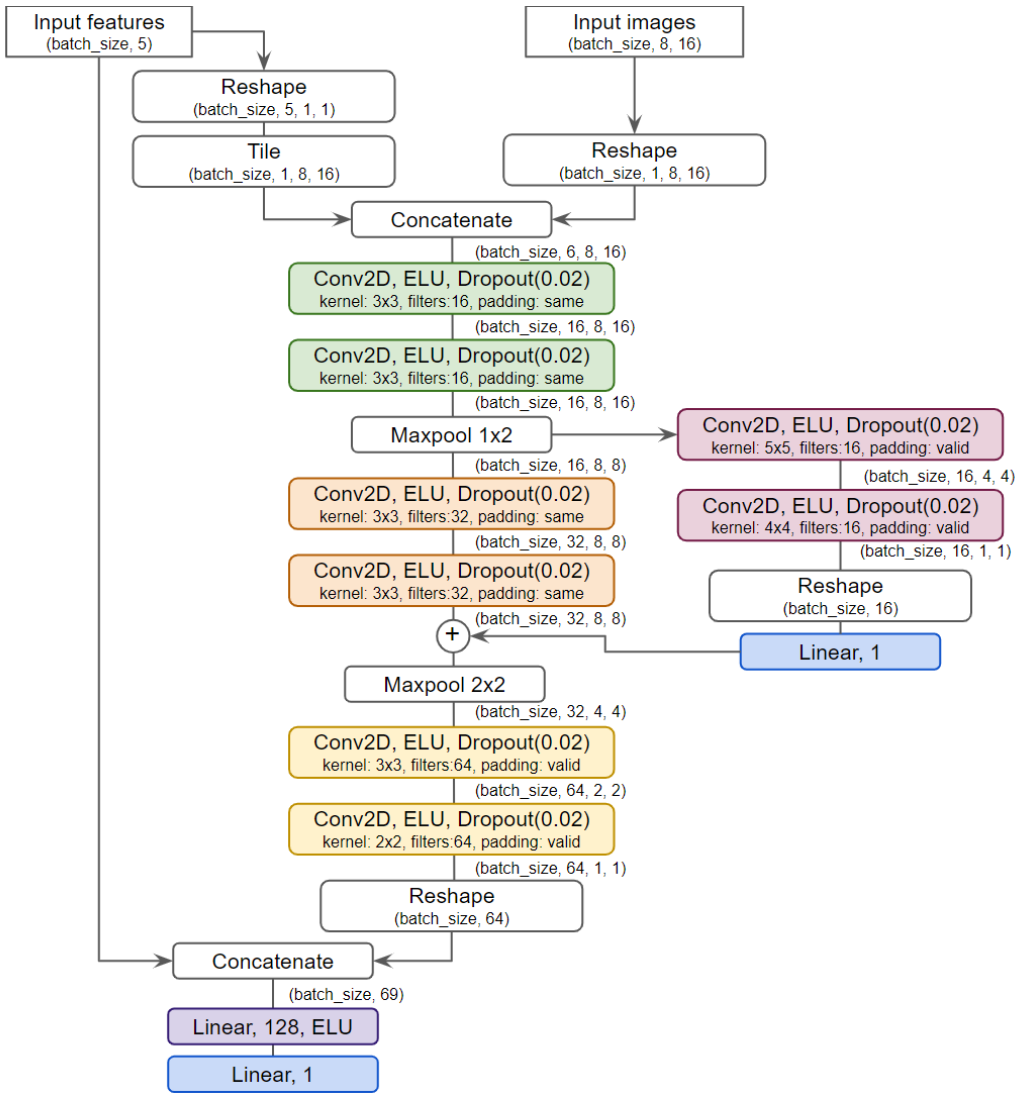


Рис. 7.2: Архитектура дискриминатора Модели 1.

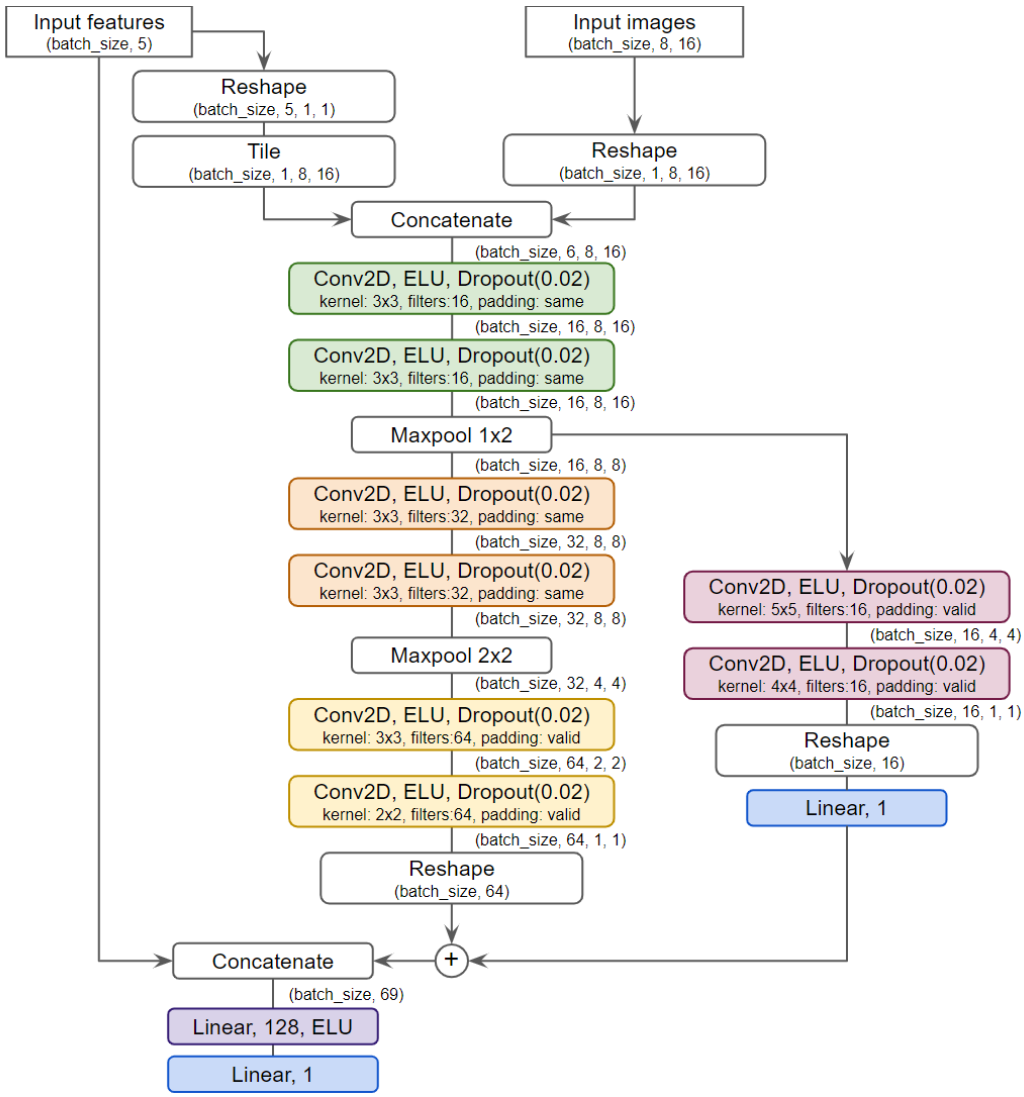


Рис. 7.3: Архитектура дискриминатора Модели 2.

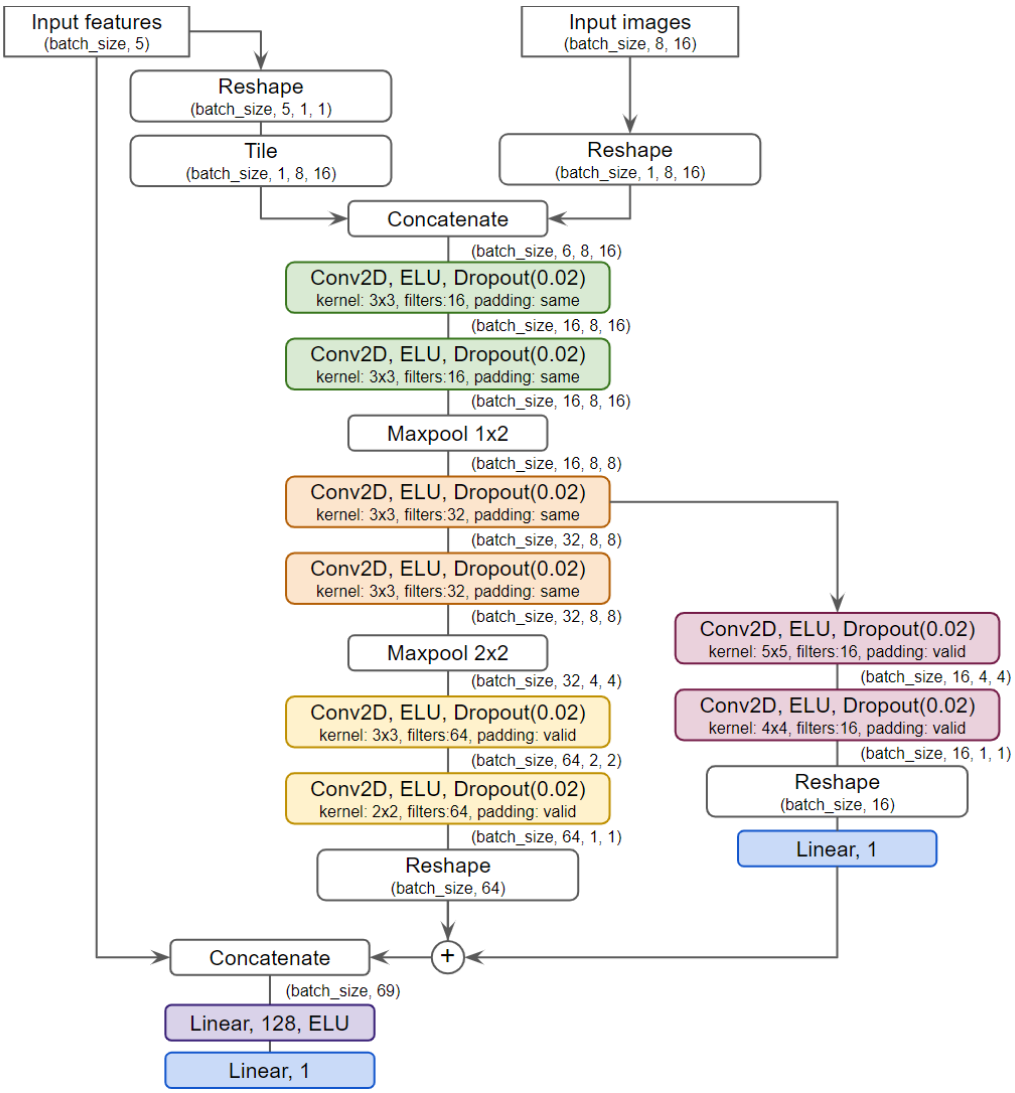


Рис. 7.4: Архитектура дискриминатора Модели 3.

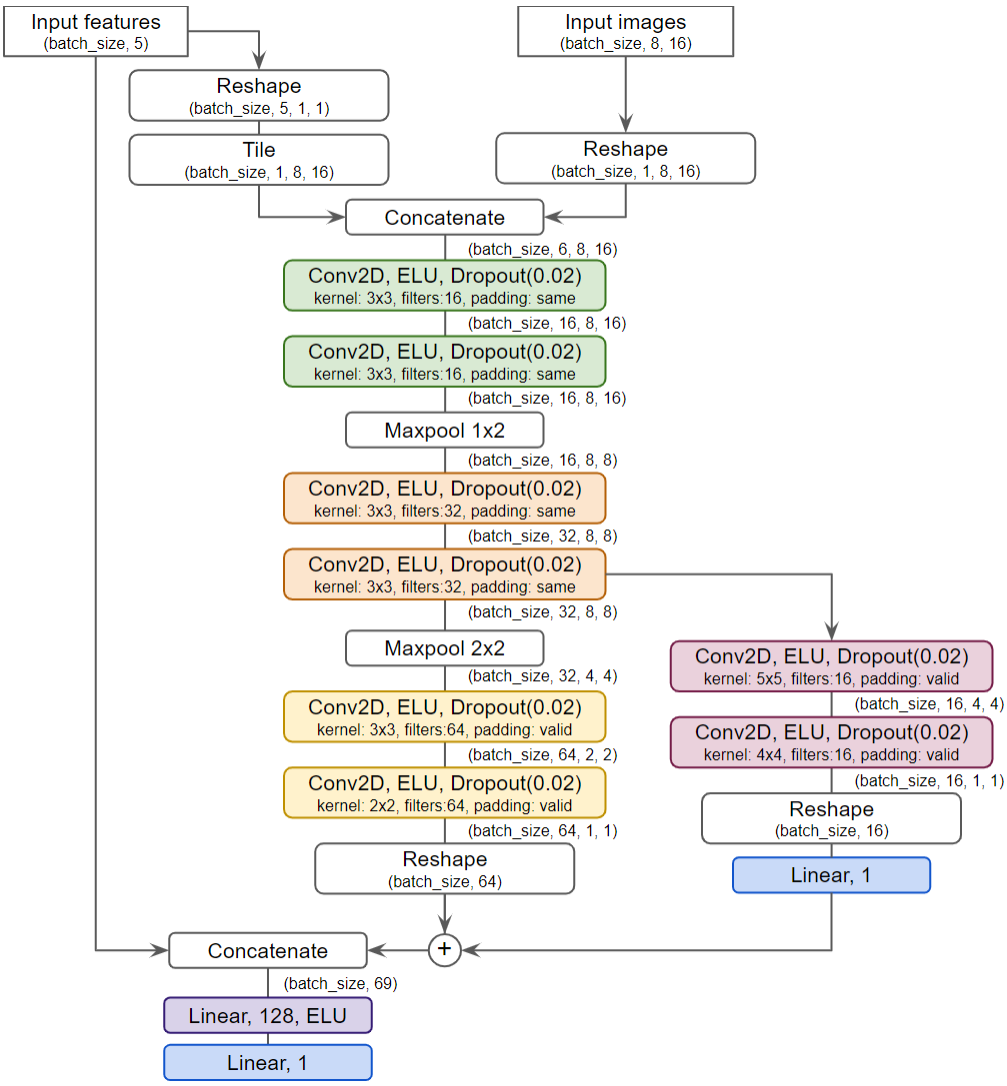


Рис. 7.5: Архитектура дискриминатора Модели 4.

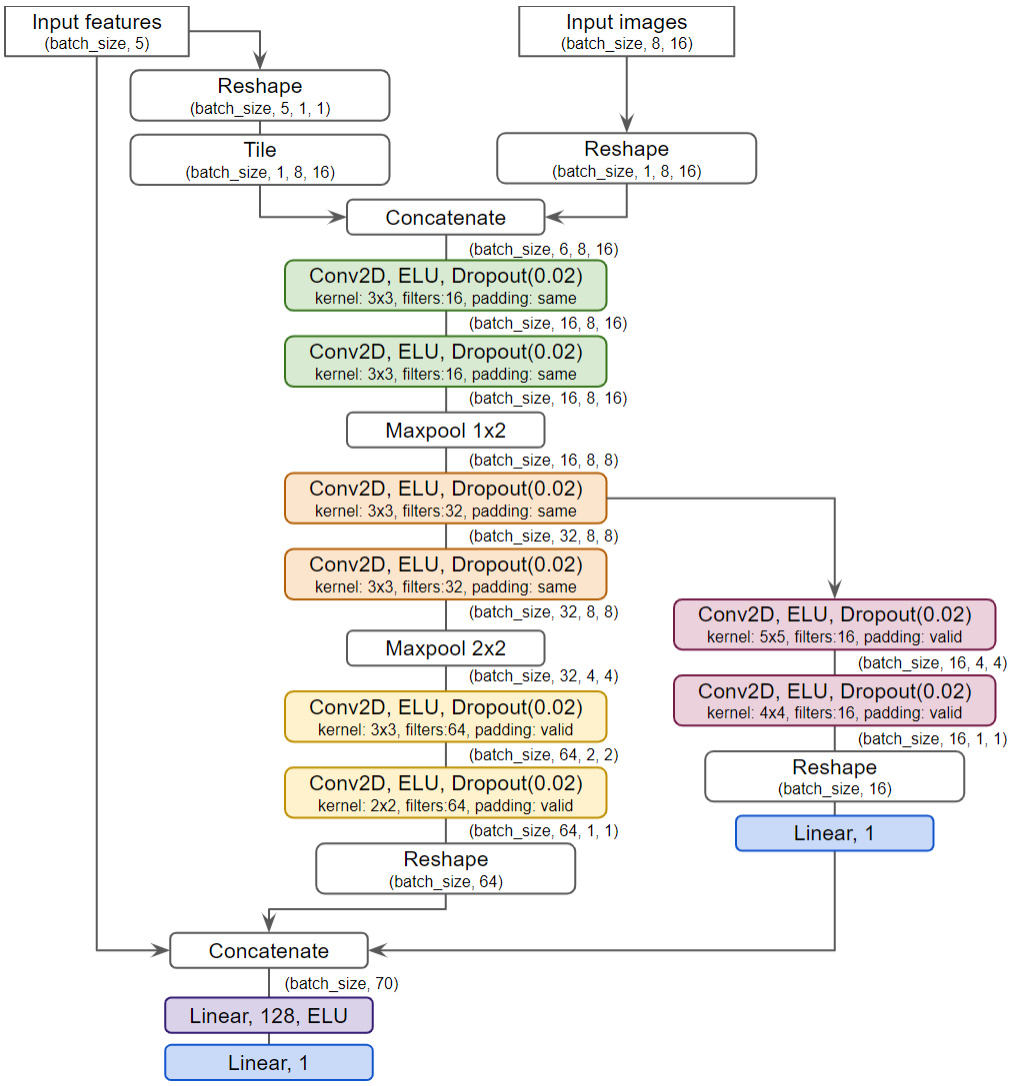


Рис. 7.6: Архитектура дискриминатора Модели 5.