

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 влияния адаптивной спектральной нормализации на качество**  
**генеративных моделей и стабильности их обучения**

**Выполнил:**

студент группы БПМИ201  
Амрин Айдар

\_\_\_\_\_  
(подпись)

\_\_\_\_\_  
(дата)

**Принял руководитель проекта:**

Ратников Федор Дмитриевич  
Ведущий научный сотрудник  
Факультета компьютерных наук НИУ ВШЭ

\_\_\_\_\_  
(подпись)

\_\_\_\_\_  
(дата)

Москва 2023

# Содержание

<b>1</b>	<b>Аннотация</b>	<b>3</b>
<b>2</b>	<b>Введение</b>	<b>3</b>
<b>3</b>	<b>Обзор литературы</b>	<b>4</b>
3.1	GAN . . . . .	4
3.2	Нестабильность GAN-ов . . . . .	5
3.3	WGAN . . . . .	7
3.4	WGAN-GP . . . . .	8
3.5	Спектральная нормализация . . . . .	9
<b>4</b>	<b>Описание MPD детектора и TPC трекера</b>	<b>10</b>
<b>5</b>	<b>Данные</b>	<b>12</b>
<b>6</b>	<b>Бейзлайн</b>	<b>14</b>
6.1	Валидация результатов . . . . .	15
<b>7</b>	<b>План</b>	<b>15</b>
	<b>Список литературы</b>	<b>16</b>
<b>A</b>	<b>Архитектура бейзлайна</b>	<b>18</b>
<b>B</b>	<b>Графики метрик</b>	<b>18</b>

# 1 Аннотация

В последние годы генеративно состязательные сети (англ. generative adversarial nets, GAN) пользуются большой популярностью и применяются в самого разного рода задачах. Одним из таких случаев является симуляция различных экспериментов физики высоких энергий. Хорошим примером служит статья [10], предоставляющая новый подход к ускорению симуляции время-проекционной камеры (англ. Time Projection Chamber, TPC) в эксперименте MPD (англ. Multi Purpose Detector) на ускорительном комплексе NICA (англ. Nuclotron-based Ion Collider fAcility). Не смотря на успех данной публикации, предложенный авторами метод, как и многие GAN based модели, страдает от очень нестабильного обучения [2], что приводит к необходимости использования различных методов регуляризации, таких как Gradient Penalty [8] или Gradient Clipping. Однако, применение таких методов часто влечет за собой заметное ухудшение в качестве генерации модели в силу строгости ограничения. Эта работа направлена на решение данной проблемы путем реализации и сравнения более гибких методов, основанных на спектральной нормализации, которая позволяет адаптивно влиять на степень регуляризации.

## Ключевые слова

Спектральная нормализация, генеративно состязательные сети, константа Липшица, быстрая симуляция

# 2 Введение

Компьютерная симуляция экспериментов физики высоких энергий очень важная задача, которая имеет множество приложений, таких как оптимизация дизайна самих детекторов [4], выбор лучшей стратегии анализа данных [12] и другие. Как правило чем точнее симуляция, тем точнее и сами финальные результаты эксперимента. Обычно симуляция эксперимента занимает очень много времени, и сопоставимо с настоящим экспериментом и полным процессингом данных [1]. А потому, более быстрые методы симуляций являются очень востребованными для нынешних и будущих физических задач.

В связи с нарастающим успехом использования глубинного обучения в физике высоких энергий, для быстрой симуляции стали использоваться GAN и VAE (англ. variational auto encoders) модели [5]. Эти генеративные модели позволяют выучивать распределение

данных, и достаточно быстро генерировать семплы из этого распределения. Работа предшествующая данной, [10] как раз использует GAN модель для симуляции ТРС трека в эксперименте с MPD, и показывает удовлетворительные результаты. Однако, этот метод, как и большинство GAN моделей в принципе, имеет одно узкое место – нестабильное обучение. Для решения данной проблемы часто используют различного рода регуляризации, такие как Gradient Clipping и Gradient Penalty [8]. Подобные методы регуляризаций накладывают очень строгое ограничение на итоговую модель, что существенно ухудшает качество генерации. Данная работа нацелена на улучшение метода описанного в [10], путем реализации более гибких методов, основанных на так называемой спектральной нормализации [11], которая позволяет адаптивно влиять на степень регуляризации нашей GAN модели, что в итоге даст прирост в качестве генерации без накладных вычислений.

## 3 Обзор литературы

### 3.1 GAN

Генеративно состязательные сети (англ. GAN) [7] – это класс моделей глубокого обучения обрешившие значительную популярность в последние года за счет своего умения генерировать реалистичные синтетические данные. GAN-ы состоят из двух нейронных сетей – генератора и дискриминатора – которые обучаются вместе в игровой схеме для генерации синтетических данных, таких как изображения, текст или звук.

Во время обучения генератор  $G$  (англ. Generator) принимает на вход случайный шум и генерирует синтетические данные, в то время как дискриминатор  $D$  (англ. Discriminator) пытается отличить реальные данные от фальшивых. Две сети обучаются поочередно, генератор пытается создавать данные, которые могут обмануть дискриминатор, а дискриминатор улучшает свою способность отличать реальные данные от синтетических. Конечная цель обучения GAN – достичь состояния, в котором сгенерированные данные неотличимы от реальных.

Обе модели  $G$  и  $D$  являются нейронными сетями, а потому их веса оптимизируются с помощью обратного распространения ошибки (англ. back propagation): скармливая  $D$  реальные данные и сгенерированные с помощью  $G$ , мы обучаем  $D$  отличать их, и с помощью пробрасывания градиентов  $D$  к  $G$  мы тренируем  $G$  генерировать более реалистичные данные. Более формально это можно записать в виде следующей минимаксной оптимизационной

задачи:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}} [\log D(x)] + \mathbb{E}_{\hat{x} \sim p_g} [\log(1 - D(\hat{x}))], \quad (1)$$

где  $V(D, G)$  – обозначение для оптимизируемой функции (value function),  $p_{data}$  – распределение настоящих данных, а  $p_g$  – распределение сгенерированных данных.

## 3.2 Нестабильность GAN-ов

Несмотря на большой успех в генерации реалистичных данных, обучение GAN моделей является очень деликатной задачей. В 2017 году вышла статья [2], содержащая в себе важные теоретические выкладки объясняющие нестабильную динамику обучения GAN-ов. Авторы выделяют следующие основные проблемы, которые часто всплывают при обучении GAN моделей:

- **Несходимость:** параметры модели сильно колеблются, дестабилизируются и никогда не сходятся;
- **Mode collapse:** генератор учиться генерировать лишь ограниченное разнообразие семплов (т.е. генерирует похожие друг на друга данные);
- **Затухающие градиенты:** дискриминатор слишком успешно классифицирует настоящие данные от синтетических, что приводит к затуханию градиента генератора и в итоге плохому обучению;
- Имбаланс в обучении генератора и дискриминатора, что приводит к переобучению;
- Большая чувствительность к изменению гиперпараметров.

Из теоремы об идеальном дискриминаторе (теорема 2.2. (The perfect discriminator theorem) из [2]) получаем, что если два распределения (настоящих данных и сгенерированных) имеют носители<sup>1</sup> (англ. support) принадлежащих компактному топологическому многообразию<sup>2</sup> (англ. closed manifold) и они не идеально совпадают<sup>3</sup>, то существует идеальный дискриминатор  $D^*(x)$  (т.е. отличающий реальные данные от синтетических с точностью 1)

---

<sup>1</sup>Носителем функции  $f : A \rightarrow B$  называется множество  $\{x \in A : f(x) \neq 0\}$ .

<sup>2</sup> $n$ -мерное топологическое многообразие – это хаусдорфово топологическое пространство со счётной базой, в котором каждая точка имеет окрестность, гомеоморфную открытому подмножеству замкнутого полупространства в  $\mathbb{R}^n$

<sup>3</sup>Мы говорим, что два многообразия  $\mathcal{M}$  и  $\mathcal{P}$  идеально совпадают (англ. perfectly align), если существует такой  $x \in \mathcal{M} \cap \mathcal{P}$ , что  $\mathcal{M}$  и  $\mathcal{P}$  не пересекаются поперечно (англ. transversally intersect) в точке  $x$ .

и  $\nabla_x D^*(x) = 0$ . Казалось бы это должно быть хорошей новостью, поскольку чем лучше дискриминатор тем лучше будет стараться генератор, чтобы его обмануть, однако все устроено намного сложнее.

Давайте посмотрим на то, что происходит когда мы передаем градиенты из дискриминатора в генератор. Рассмотрим предложенную в [7] целевую функцию для обучения генератора:

$$L_G = \mathbb{E}_{\hat{x} \sim p_g} [\log(1 - D(\hat{x}))] \quad (2)$$

Согласно теореме о затухающих градиентах генератора (теорема 2.4. (Vanishing gradients on the generator) из [2]): если выполнены условия теоремы об идеальном дискриминаторе,  $\|D - D^*\| < \varepsilon$ , и  $\mathbb{E}_{z \sim p_{noise}} [\|J(G(z))\|_2^2] \leq M^2$ , то

$$\|\nabla L_G\|_2 = \|\nabla \mathbb{E}_{\hat{x} \sim p_g} [\log(1 - D(\hat{x}))]\|_2 < M \frac{\varepsilon}{1 - \varepsilon}. \quad (3)$$

Из 3 следует, что (следствие 2.1. из [2])

$$\lim_{\|D - D^*\| \rightarrow 0} \nabla L_G = \lim_{\|D - D^*\| \rightarrow 0} \nabla \mathbb{E}_{\hat{x} \sim p_g} [\log(1 - D(\hat{x}))] = 0. \quad (4)$$

Это следствие показывает, что когда дискриминатор становится лучше (ближе к идеальному  $D^*$ ), градиенты генератора начинают затухать. Авторы также пишут, что даже если поменять целевую функцию генератора на

$$\bar{L}_G = -\mathbb{E}_{\hat{x} \sim p_g} [\log(D(\hat{x}))], \quad (5)$$

описанную в первоначальной статье о GAN-ах [7], все равно наблюдается та же самая проблема с затухающими градиентами. Теорема о нестабильности обновления генератора (теорема 2.6. (Instability of generator gradient updates) из [2]) говорит о том, что имея целевую функцию  $\bar{L}_G$ , при выполнении условий теоремы об идеальном дискриминаторе и имея  $D$  такой, что  $D^* - D = \varepsilon$  и  $\nabla_x D^* - \nabla_x D$  — белые шумы<sup>4</sup>, то оказывается что каждая координата

$$\bar{L}_G = -\mathbb{E}_{\hat{x} \sim p_g} [\log(D(\hat{x}))] \quad (6)$$

имеет центрированное распределение Коши с бесконечной дисперсией. Из центрированно-

---

<sup>4</sup>Центрированный гауссовский процесс индексированный по  $x$  и независимый для каждого  $x$ .

сти распределения следует, что при ограничении обновлений мы получаем, что ожидаемое обновление генератора будет нулевым. А бесконечная дисперсия как раз таки и объясняет нестабильность в обучении генератора, что сильно замедляет сходимость.

### 3.3 WGAN

Как было указано ранее, предложенные в статье [7] целевые функции (2, 6) имеют проблемы нестабильного обучения и затухания градиентов. Именно для решения этих проблем были придуманы WGAN [3] (Wasserstein GAN), тип генеративно состязательных нейронных сетей минимизирующие аппроксимацию Earth Mover's distance (EMD), или другими словами расстояние Вассерштейна (Wasserstein's distance), вместо дивергенции Йенсена-Шеннона (Jensen-Shannon, JS), используемой в ванильном GAN. Расстояние Вассерштейна определяется следующим образом:

$$W(p_{data}, p_g) = \inf_{\gamma \in \Pi(p_{data}, p_g)} \mathbb{E}_{(x,y) \sim \gamma} [\|x - y\|], \quad (7)$$

где  $\Pi(p_{data}, p_g)$  обозначает совокупность всех распределений  $\gamma(x, y)$  с маргинальными распределениями  $p_{data}$  и  $p_g$  соответственно. Интуитивно,  $\gamma(x, y)$  – количество "массы" которую требуется транспортировать из  $x$  в  $y$ , чтобы получить из распределения  $p_{data}$  распределение  $p_g$ . Таким образом, EMD – это "цена" самой оптимальной транспортировки.

Как оказалось, расстояние Вассерштейна является непрерывным<sup>5</sup> и почти всюду дифференцируемым, что делает ее хорошей функцией потерь для обучения нашей модели. На практике удобнее работать с другой формой записи Вассерштейновского расстояния, а именно с двойственностью Кантровича-Рубинштейна [13]:

$$W(p_{data}, p_g) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x \sim p_{data}} [f(x)] - \mathbb{E}_{x \sim p_g} [f(x)], \quad (8)$$

где супремум берется по всем 1-липшицевым<sup>6</sup> функциям  $f$ . Заметим, что если мы заменим 1-липшицевость на  $K$ -липшицевость, то получим слева выражения (8)  $K \cdot W(p_{data}, p_g)$ . Таким образом, если у нас есть параметризованное семейство функций  $\{f_w\}_{w \in \mathcal{W}}$ , такое что они все

<sup>5</sup>Непрерывным в том смысле, что если  $\theta_t \rightarrow \theta$ , то  $\mathbb{P}_{\theta_t}$  сходится к  $\mathbb{P}_{\theta}$  (по заданному расстоянию  $\rho$ , в нашем случае  $W(\cdot, \cdot)$ ).

<sup>6</sup>Отображение  $f$  метрического пространства  $(\mathcal{X}, \rho_{\mathcal{X}})$  в метрическое пространство  $(\mathcal{Y}, \rho_{\mathcal{Y}})$  называется  $L$ -липшицевым, если  $\rho_{\mathcal{Y}}(f(a), f(b)) \leq L \cdot \rho_{\mathcal{X}}(a, b)$  для любых  $a, b \in \mathcal{X}$ .

$K$ -липшицевы для некоторого  $K$ , то задача подсчета  $W(p_{data}, p_g)$  сводится к решению задачи

$$\max_{w \in \mathcal{W}} \mathbb{E}_{x \sim p_{data}}[f_w(x)] - \mathbb{E}_{x \sim p_g}[f_w(x)]. \quad (9)$$

Итак, вычислив (9) мы получим расстояние Вассерштейна в точности до умножения на константу. По теореме 3 из [3] у задачи (9) существует решение в виде функции  $f$  и

$$\nabla_{\theta} W(p_{data}, p_{g_{\theta}}) = -\mathbb{E}_{z \sim p_{noise}}[\nabla_{\theta} f(g_{\theta}(z))], \quad (10)$$

где генератор  $g$  – функция генератора, которая параметризована с помощью параметра  $\theta$  и принимает на вход случайный шум  $z \sim p_{noise}$  и  $g$  – непрерывна по  $\theta$ <sup>7</sup>. Теперь встает вопрос нахождения решения (9). Авторы статьи предлагают аппроксимировать эту функцию  $f$  (решение оптимизационной задачи (9) с помощью обучения нейронной сети параметризованной весами  $w$  лежащих в компакте  $\mathcal{W}$ . Несложно заметить, что если  $\mathcal{W}$  действительно компакт, то все функции  $f_w$  будут  $K$ -липшицевыми для некоторого  $K$ , которое будет зависеть не от конкретных весов  $w$ , а от  $\mathcal{W}$ , что приведет аппроксимирование решения (9) в точности до приемлимой константы. Таким образом, авторы предлагают метод в котором мы ограничиваем веса  $w$  в фиксированной коробке (weight clipping) после каждого шага градиентного спуска. В самой работе [3] пишется о том, что данный метод является очень плохим способом для ограничения условия липшицевости.

Преимуществом WGAN перед Vanilla GAN является то, что мы можем тренировать дискриминатор (аппроксимировать функцию  $f$ ) до полной сходимости. Это действительно так, поскольку чем больше мы тренируем дискриминатор, тем надежнее становятся градиенты расстояния Вассерштейна (10) (поскольку оно почти всюду дифференцируемо), и тем лучше обучается генератор. В секции 3.2. мы видели, что использование JS расстояния (в обычных GAN) приводит к тому, что при улучшении дискриминатора затухают градиенты генератора (что мешаем ему обучаться).

### 3.4 WGAN-GP

Как писалось в оригинальной статье представившей WGAN всему миру, Weight clipping является плохой регуляризацией для сохранения условия липшицевости. В 2017 году была опубликована новая работа [9] с описанием модели WGAN-GP (WGAN+Gradient Penalty),

---

<sup>7</sup>Это условие выполнено для обычных нейросетей, которые в свою очередь являются комбинаций аффинных преобразований и покомпонентно примененных нелинейных гладких липшицевых функций (такие как сигмоида, tanh, elu и т.д.)



которая представляет из себя WGAN использующий Gradient Penalty для сохранения условия липшицевости. Поскольку дифференцируемая функция  $f$  является 1-липшицевой тогда и только тогда, когда ее градиент почти везде имеет норму не более 1, авторы предлагают использовать следующую функцию потерь для обучения дискриминатора:

$$L = \underbrace{-\mathbb{E}_{x \sim p_{data}}[f(x)] + \mathbb{E}_{x \sim p_g}[f(x)]}_{\text{Original WGAN critic loss}} + \underbrace{\lambda \mathbb{E}_{z \sim p_{noise}}[(\|\nabla_z f(z)\|_2 - 1)^2]}_{\text{Gradient penalty}}, \quad (11)$$

где  $\lambda$  – гиперпараметр влияющий на степень регуляризации и ограничения условия липшицевости. Авторы статьи приводят экспериментальные сведения показывающие превосходство этого метода над Weight clipping.

### 3.5 Спектральная нормализация

Спектральная нормализация – это метод нормализации весов, который был впервые предложен в статье [11] для стабилизации обучения дискриминатора в GAN моделях. Авторы [11] вдохновившись работами [3, 9] так же ищут дискриминатор из множества  $K$ -липшицевых функций. Спектральная нормализация контролирует константу липшица функции  $f$  (дискриминатора) буквально ограничивая спектральную норму каждого из слоев  $g : h_{in} \rightarrow h_{out}$ . По определению, липшицева норма функции  $g$  записывается в виде

$$\|g\|_L = \sup_h \sigma(\nabla g(h)), \quad (12)$$

где  $\sigma(A)$  – спектральная норма матрицы  $A$  (или  $L_2$  матричная норма  $A$ ), которая определена следующим образом:

$$\sigma(A) := \max_{h \neq 0} \frac{\|Ah\|_2}{\|h\|_2} = \max_{\|h\|_2 \leq 1} \|Ah\|_2, \quad (13)$$

что эквивалентно наибольшему сингулярному числу матрицы  $A$ . Таким образом, для линейного слоя  $g(h) = Wh$ , его липшицева норма равна

$$\|g\|_L = \sup_h \sigma(\nabla g(h)) = \sup_h \sigma(\nabla Wh) = \sup_h \sigma(W) = \sigma(W). \quad (14)$$

Рассмотрим нейросеть вида

$$f_\theta(x) = W^{N+1} a_N(W^N(a_{N-1}(W^{N-1}(\dots a_1(W^1 x) \dots))), \quad (15)$$

где  $\theta = \{W^1, W^2, \dots, W^{N+1}\}$  – обучаемые параметры модели,  $W^l \in \mathbb{R}^{d_l \times d_{l-1}}$ ,  $W^{N+1} \in \mathbb{R}^{1 \times d_N}$ , и  $a_l$  – функции активации (нелинейные покомпонентные функции). Тогда если липшицева норма функции активации  $\|a_l\|_L = 1$ <sup>8</sup>, то пользуясь неравенством  $\|g_1 \circ g_2\|_L \leq \|g_1\|_L \cdot \|g_2\|_L$  получаем, что

$$\begin{aligned} \|f\|_L &\leq \|(h_N \mapsto W^{N+1}h_N)\|_L \cdot \|a_N\|_L \cdot \|(h_{N-1} \mapsto W^N h_{N-1})\|_L \cdot \\ &\dots \cdot \|a_1\|_L \cdot \|(h_0 \mapsto W^1 h_0)\|_L = \prod_{l=1}^{N+1} \|(h_{l-1} \mapsto W^l h_{l-1})\|_L = \prod_{l=1}^{N+1} \sigma(W^l). \end{aligned} \quad (16)$$

Тогда отнормируем каждую матрицу весов  $W^l$  следующим образом:

$$W_{SN}^l := W^l / \sigma(W^l). \quad (17)$$

Несложно заметить, что тогда  $\sigma(W_{SN}^l) = 1$ , а потому в силу (16) получаем ограничение  $\|f\|_L \leq 1$ .

Получается, что спектральная нормализация основана на (17) нормировке матриц весов дискриминатора относительно спектральной (матричной  $L_2$ ) нормы. Авторы статьи предлагают считать спектральную норму матрицы весов  $W$ , или то же самое что и старшее сингулярное число  $W$ , с помощью алгоритма основанного на методе степенных итераций [6], предназначенный для быстрого поиска собственных значений матрицы. Основываясь на множестве экспериментов, авторы предлагают читателям новый метод регулирования условия липшецевости функции дискриминатора, и именно этот метод мы хотим исследовать в рамках нашей конкретной задачи симуляции ТРС трека в эксперименте MPD и сравнить с другими существующими решениями.

## 4 Описание MPD детектора и ТРС трека

MPD – детектор предназначенный для проведения экспериментов в области релятивистской ядерной физики, находящийся на ускорительном комплексе NICA – новый сверхпроводящий коллайдер протонов и тяжёлых ионов, строящийся с 2013 года на базе Лаборатории физики высоких энергий (ЛФВЭ) им. В. И. Векслера и А. М. Балдина Объединённого института ядерных исследований (ОИЯИ), в городе Дубна Московской области. Таким образом, MPD служит для детекции заряженных адронов, электронов и фотонов.

---

<sup>8</sup>Например, это верно для таких функций активаций как ReLU и leaky ReLU, и множество других популярных функций активаций являются  $K$ -липшицевыми для некоторого  $K$ .

TPC – самая главная компонента MPD, которая представляет из себя заполненный газом детектор цилиндрической формы (рис. 4.1, 4.2).

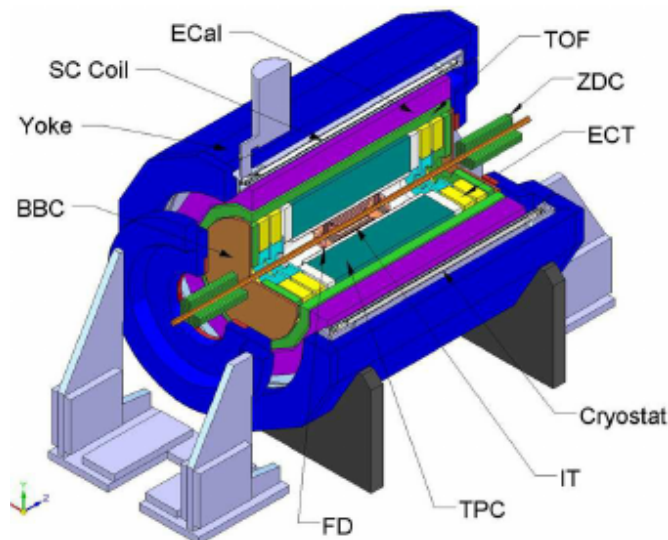


Рис. 4.1: Дизайн MPD. TPC: time projection chamber; TOF: time-of-flight detector; ECal: electro-magnetic calorimeter; FD: forward detector; BBC: beam-beam counter; ZDC: zero-degree calorimeter.

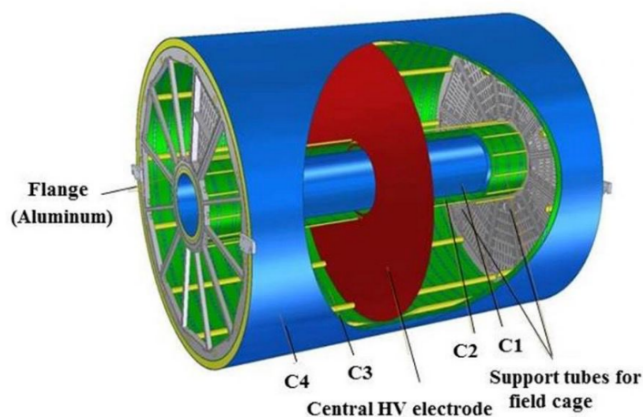


Рис. 4.2: Дизайн TPC.

Он оснащен электрическим полем, что способствует движению ионизированных электронов в сторону оснований цилиндра, каждое из которых оснащено камерой с катодными листами, регистрирующие соответствующие сигналы (рис. 4.2). Каждая из камер разделена на 12 одинаковых секторов (по  $30^\circ$ ). Каждый из секторов в свою очередь поделен на 53 ряда с пэдами (англ. pad) перпендикулярных радиальному направлению. Пэды имеют ширину 5мм и высоту 12мм (короткие) и 18мм (длинные) для внутренних (первых считая от центра окружности) 27 рядов и 26 внешних (последних) рядов, соответственно (рис. 4.3). Каждый из пэдов является неким чувствительным детектором, который регистрирует сигнал. Итого, TPC состоит из 95,232 чувствительных пэдов, записи которых собираются в 310 временных промежутках для каждого трека (траектории электрона).

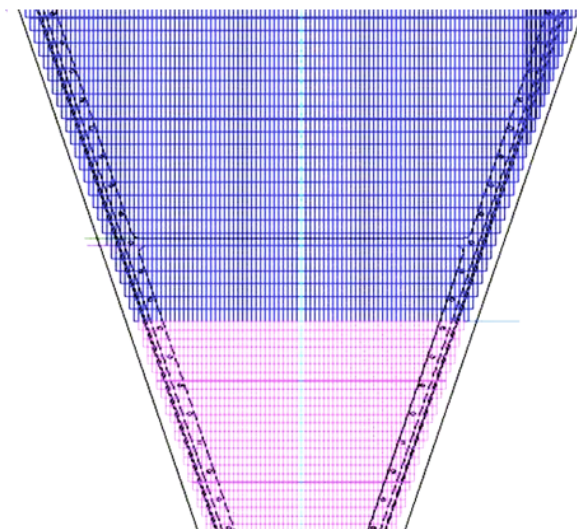


Рис. 4.3: Один из 12 секторов камеры. Синие ряды соответствуют длинным пэдам, а красные коротким.

## 5 Данные

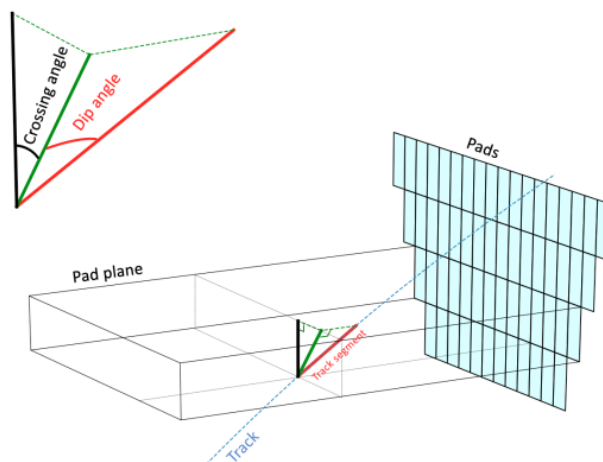


Рис. 5.1: Траектория электрона внутри ТРС.

Основная задача состоит в том, чтобы научиться генерировать сырые показания трека ТРС, т.е. генерировать показания каждого из 95,232 чувствительных пэдов для каждого из 310 временных промежутков в которых идет запись треков, итого около 30 млн. чисел для каждого из треков. Чтобы уменьшить размерность пространства наших целевых объектов (показаний ТРС), мы делим пэды на небольшие условно независимые подмножества, с условиями описанными ниже. Для того, чтобы снизить размерность мы делаем предположение

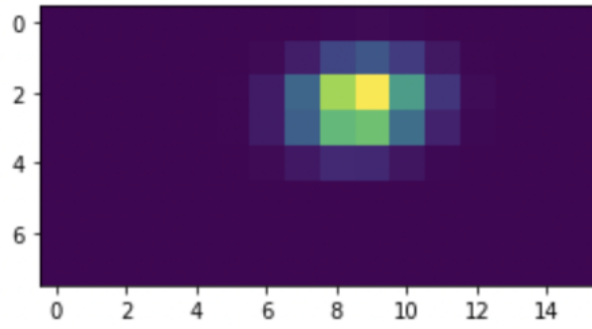


Рис. 5.2: Сэмпл отклика ТРС пэда размерности  $8 \times 16$ .

о том, что показатели трека в определенном ряду пэдов зависит лишь от маленьких сегментов траектории электрона, сформированных путем пересечения трека с соответствующей *пэд-плоскостью* (англ. *pad plane*). *Пэд-плоскостью* будем называть плоскость содержащую линии электрического поля направленных на данный ряд пэдов (рис. 5.1). Таким образом,

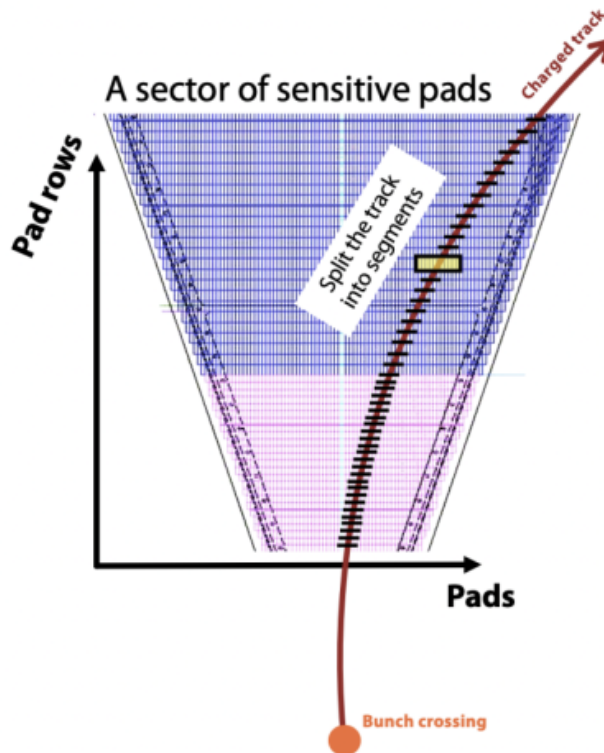


Рис. 5.3: Сегмент трека.

мы моделируем вклад от различных сегментов трека (отрезок трека ограниченный соответствующей пэд плоскостью, рис. 5.1, и рис. 5.3), и потом комбинируем их аддитивным способом (игнорируя то, что трек сам по себе может быть нелинейным). В работе [10] показано, что подобные опущения и предположения не сильно влияют на качество генерации. И наконец, мы пользуемся тем фактом, что показания датчиков от конкретного сегмента тре-

ка являются локализованными в пространстве и времени, а потому мы можем моделировать лишь небольшое количество пэдов и временных промежутков для данного сегмента трека. Таким образом для выбранного сектора ТРС трекера, в статье [10] сокращают пространство показаний детектора с  $95,232 \times 310$  до  $8 \times 16$  (смотреть рис. 5.2).

Поскольку, все пэды идентичны (для простоты считаем, что имеем дело лишь с короткими пэдами), мы тренируем модель на показаниях от маленького подмножества пэдов находящихся в одном ряду и просто переносим предсказания на все оставшиеся пэды. Так как мы считаем, что траектории электронов являются прямыми линиями, мы можем пользоваться геометрическими характеристиками для каждого сегмента трека, такие как: *crossing angle*, *dip angle*, *drift length* и *pad coordinate* (смотреть на рис. 5.1).

## 6 Бейзлайн

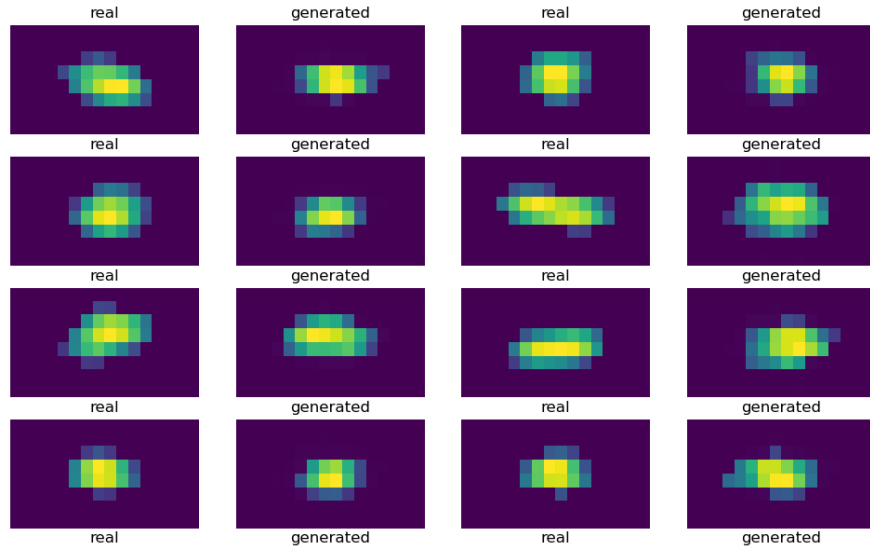


Рис. 6.1: Примеры генерации бейзлайна.

В качестве бейзлайна рассмотрим модель использованную в [10] и вместо Gradient Penalty будем использовать спектральную нормализацию поверх сверточных и линейных слоев (для ограничения липшицевой константы, [11]). В качестве функции потерь использовалось Вассерштейновское расстояние, и как говорилось ранее мы использовали спектральную нормализацию вместо Gradient Penalty в качестве стабилизатора обучения. И генератор, и дискриминатор обучались с помощью RMSprop оптимизатора со стартовыми **learning rate** = 0.0001 в начале обучения и с экспоненциальным расписанием умножался на 0.999 после каждой эпохи. Совершалось 8 обновлений дискриминатора на каждый шаг генератора (как упоминалось в конце секции 3.3, лучше добиться быстрой сходимости дескриминато-

ра, для более качественного обучения генератора). Размер батча составляет 32, количество эпох 3900. Подробную архитектуру бейзлайн модели можно увидеть в [A](#). Код обучения бейзлайна был написан с помощью фреймворка PyTorch и может быть найден в репозитории – <https://github.com/hzchet/tpc-sn-gen>. На рисунке 6.1 можно увидеть примеры сгенерированных данных.

## 6.1 Валидация результатов

Помимо визуальной валидации для более точной количественной оценки введем набор метрик, с помощью которых мы будем сравнивать сгенерированные данные с валидационными. Для каждого изображения отклика пэда мы будем вычислять моменты 1-го порядка, т.е. координаты центра масс распределения отклика, моменты 2-го порядка, т.е. квадрат ширины распределения отклика и ковариация координаты пэда и временной координаты. В итоге будем сравнивать распределение этих метрик для сгенерированных откликов и настоящих. Визуализации метрик доступны в [B](#).

## 7 План

Ниже предоставлен план дальнейшей работы:

- Прodelать исследовательский анализ над имеющимися данными, провести эксперименты с различными препроцессингами данных;
- Имплементировать метрики связанные с геометрическими характеристиками траекторий электрона для валидации качества генерации;
- Прodelать эксперименты с различными архитектурами и посмотреть как спектральная нормализация влияет на динамику обучения и качество генерации;
- Сделать детальный разбор того какой из подходов лучше для данной задачи: 1) Weight clipping; 2) Gradient penalty; 3) Спектральная нормализация;
- Поэкспериментировать с различными алгоритмами нахождения спектральной нормы (компонента спектральной нормализации);
- Прodelать анализ полученных результатов, составить итоговый отчет и презентацию;
- Защитить проект.



## Список литературы

- [1] Johannes Albrecht и др. “A Roadmap for HEP Software and Computing RD for the 2020s”. B: *Computing and Software for Big Science* 3.1 (). URL: <https://doi.org/10.1007/s41781-018-0018-8>.
- [2] Martin Arjovsky и Leon Bottou. “Towards Principled Methods for Training Generative Adversarial Networks”. B: *International Conference on Learning Representations*. 2017. URL: [https://openreview.net/forum?id=Hk4\\_qw5xe](https://openreview.net/forum?id=Hk4_qw5xe).
- [3] Martin Arjovsky, Soumith Chintala и Léon Bottou. *Wasserstein GAN*. 2017. DOI: [10.48550/ARXIV.1701.07875](https://arxiv.org/abs/1701.07875). URL: <https://arxiv.org/abs/1701.07875>.
- [4] A Baranov, E Burnaev, D Derkach, A Filatov, N Klyuchnikov, O Lantwin, F Ratnikov, A Ustyuzhanin и A Zaitsev. “Optimising the Active Muon Shield for the SHiP Experiment at CERN”. B: *Journal of Physics: Conference Series* 934.1 (дек. 2017), с. 012050. DOI: [10.1088/1742-6596/934/1/012050](https://doi.org/10.1088/1742-6596/934/1/012050). URL: <https://dx.doi.org/10.1088/1742-6596/934/1/012050>.
- [5] Chapman, John, Cranmer, Kyle, Gadatsch, Stefan, Golling, Tobias, Ghosh, Aishik, Gray, Heather M., Lari, Tommaso, Pascuzzi, Vincent R., Raine, John A., Rousseau, David, Salamani, Dalila и Schaarschmidt, Jana. “Fast simulation methods in ATLAS: from classical to generative models”. B: *EPJ Web Conf.* 245 (2020), с. 02035. DOI: [10.1051/epjconf/202024502035](https://doi.org/10.1051/epjconf/202024502035). URL: <https://doi.org/10.1051/epjconf/202024502035>.
- [6] Gene H. Golub и Henk A. van der Vorst. “Eigenvalue computation in the 20th century”. B: *Journal of Computational and Applied Mathematics* 123.1 (2000). Numerical Analysis 2000. Vol. III: Linear Algebra, с. 35—65. ISSN: 0377-0427. DOI: [https://doi.org/10.1016/S0377-0427\(00\)00413-1](https://doi.org/10.1016/S0377-0427(00)00413-1). URL: <https://www.sciencedirect.com/science/article/pii/S0377042700004131>.
- [7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville и Yoshua Bengio. “Generative adversarial nets”. B: *Advances in neural information processing systems*. 2014, с. 2672—2680.
- [8] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin и Aaron Courville. *Improved Training of Wasserstein GANs*. 2017. DOI: [10.48550/ARXIV.1704.00028](https://arxiv.org/abs/1704.00028). URL: <https://arxiv.org/abs/1704.00028>.



- [9] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin и Aaron Courville. *Improved Training of Wasserstein GANs*. 2017. DOI: [10.48550/ARXIV.1704.00028](https://doi.org/10.48550/ARXIV.1704.00028). URL: <https://arxiv.org/abs/1704.00028>.
- [10] A. Maevskiy, F. Ratnikov, A. Zinchenko и V. Riabov. “Simulating the time projection chamber responses at the MPD detector using generative adversarial networks”. B: *The European Physical Journal C* 81.7 (2021). URL: <https://doi.org/10.1140/epjc/s10052-021-09366-4>.
- [11] Takeru Miyato, Toshiki Kataoka, Masanori Koyama и Yuichi Yoshida. “Spectral Normalization for Generative Adversarial Networks”. B: *International Conference on Learning Representations*. 2018. URL: <https://openreview.net/forum?id=B1QRgziT->.
- [12] “Performance of the MPD Detector in the Study of the Strangeness to Entropy Ratio in Heavy-Ion Collisions at the NICA Accelerator Complex”. B: *Physics of Particles and Nuclei Letters* 17.3 (2020), c. 358—369. URL: <https://doi.org/10.1134/S1547477120030085>.
- [13] Cedric Villani. “Optimal Transport: Old and New”. B: *Springer* (2009).

## А Архитектура бейзлайна

на рис. A.1 показана архитектура бейзлайн модели.

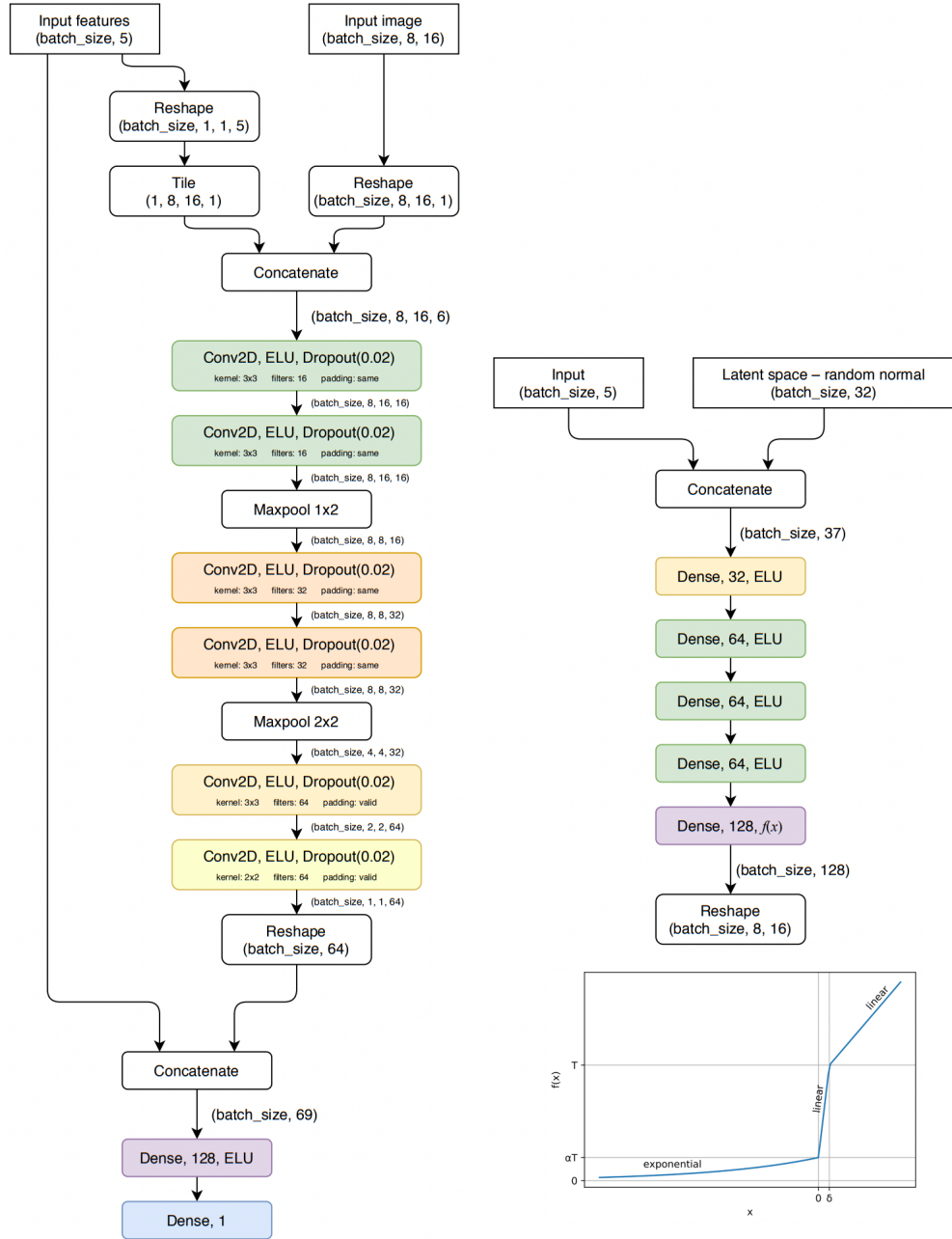


Рис. A.1: Архитектура бейзлайна, дискриминатор слева и генератор справа. Используется кастомная функция активации  $f(x)$  из статьи [10].

## В Графики метрик

Внизу можно увидеть графики для метрик описанных в секции 6.1

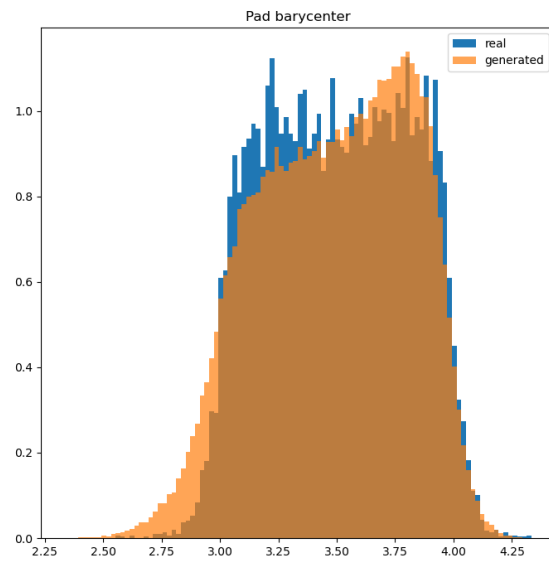


Рис. В.1: Распределение барицентов координат пэда для синтетических и реальных данных.

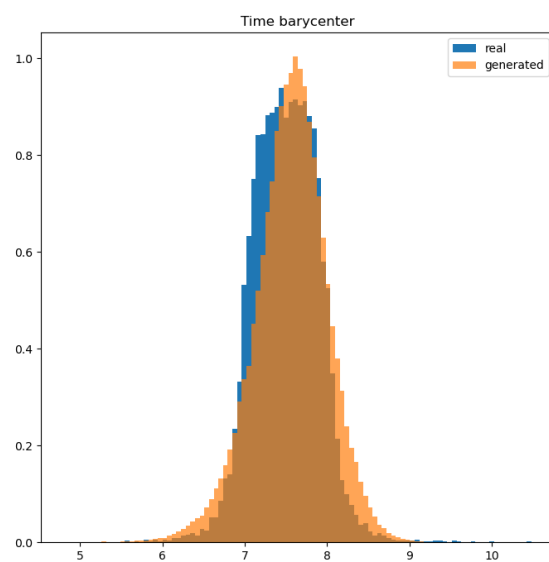


Рис. В.2: Распределение барицентов координат времени для синтетических и реальных данных.

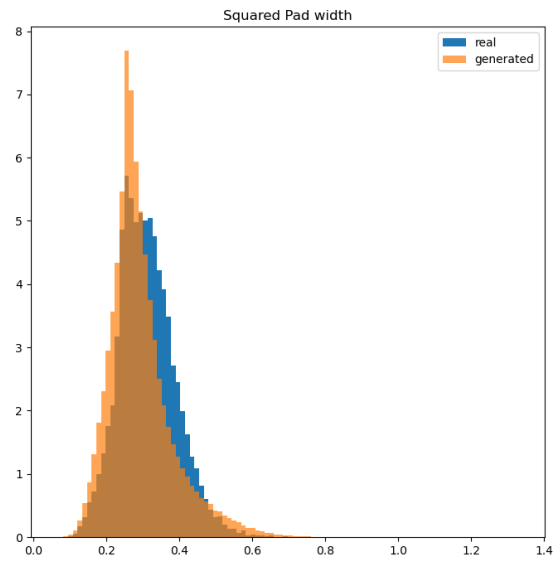


Рис. В.3: Распределение вторых моментов координат пэда для синтетических и реальных данных.

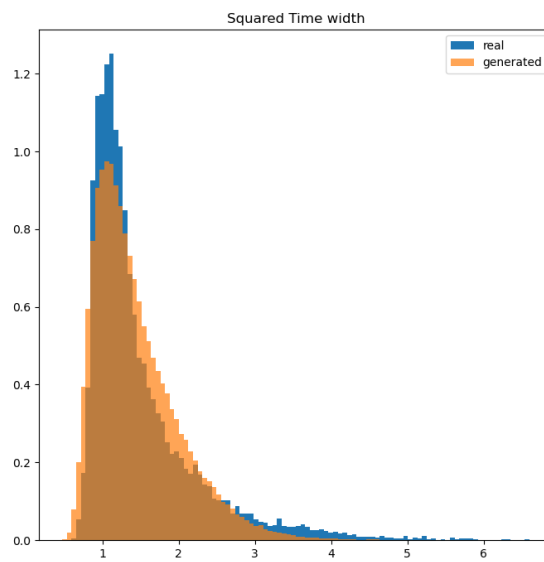


Рис. В.4: Распределение вторых моментов координат времени для синтетических и реальных данных.

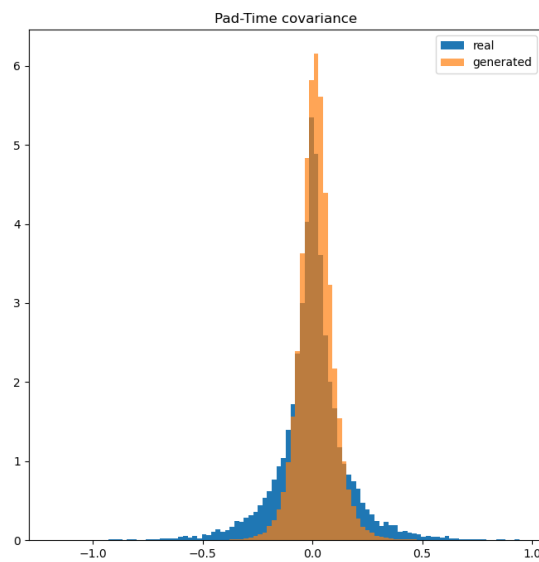


Рис. В.5: Распределение ковариаций координат пэда и временной координаты для синтетических и реальных данных.