

### Аннотация

Состязательные мосты Шрёдингера на задаче трансляции домена  
*Ксенофонт Григорий Сергеевич*

В магистерской диссертации рассматривается задача трансляции домена. Целью исследования является разработка нового метода решения задачи мостов Шрёдингера, который находит одноэтапное отображение, а также справляется с данными большой размерностью. Так в работе предложен новый подход, который находит решения задачи мостов Шрёдингера с помощью состязательного обучения, таким образом объединяя преимущества состязательных генеративных сетей (GAN) и мостов Шрёдингера. В данной работе проводятся эксперименты предложенного подхода на различных наборах данных, включая 2D данные и EMNIST. Полученные результаты показывают, что предложенный метод удовлетворяет всем поставленным требованиям и предлагает новый фреймворк для решения задачи мостов Шрёдингера.

### Abstract

Adversarial Schrödinger bridges on domain translation problem

The master's thesis examines the domain translation problem. The goal of the research is to develop a new method for solving the Schrödinger bridge problem, which finds a one-stage mapping and also copes with high-dimensional data. Thus, the paper proposes a new approach that finds solutions to the Schrödinger bridge problem using adversarial learning, thus combining the advantages of adversarial generative networks (GANs) and Schrödinger bridges. In this work, experiments of the proposed approach are carried out on various datasets, including 2D data and EMNIST. The results obtained show that the proposed method satisfies all the stated requirements and offers a new framework for solving Schrödinger bridges problem.

## Содержание

<b>1</b>	<b>Введение</b>	<b>3</b>
<b>2</b>	<b>Задача Мостов Шрёдингера</b>	<b>6</b>
2.1	Динамическая постановка задачи Мостов Шрёдингера . . . . .	6
2.2	Статическая постановка мостов Шрёдингера . . . . .	7
2.3	Система Шрёдингера . . . . .	7
2.4	Динамические полумосты Шрёдингера . . . . .	8
2.4.1	Статические полумосты Шрёдингера . . . . .	9
2.5	Методы решения мостов Шрёдингера . . . . .	9
<b>3</b>	<b>Обзор существующих решений</b>	<b>11</b>
3.1	Data Driven Schrödinger Bridge . . . . .	11
3.2	Light Schrödinger Bridge . . . . .	12
3.3	Diffusion Schrödinger Bridge . . . . .	12
3.4	Iterative Proportional Maximum Likelihood (IPML) . . . . .	13
3.5	Unpaired Neural Schrödinger Bridge . . . . .	13
<b>4</b>	<b>Постановка задачи</b>	<b>14</b>
<b>5</b>	<b>Исследование и построение решения задачи</b>	<b>15</b>
5.1	Состязательные генеративные модели (Generative-adversarial networks, GAN) и их обобщение f-GAN . . . . .	15
5.2	Состязательные мосты Шрёдингера . . . . .	17
<b>6</b>	<b>Реализация метода</b>	<b>19</b>
6.1	Выбор языка программирования . . . . .	19
6.1.1	Python . . . . .	19
6.1.2	C++ . . . . .	19
6.1.3	R . . . . .	19
6.1.4	MATLAB . . . . .	19
6.1.5	Java . . . . .	20
6.1.6	Сравнение и выбор . . . . .	20
6.2	Выбор фреймворка глубокого обучения . . . . .	20
6.2.1	TensorFlow . . . . .	20
6.2.2	Keras . . . . .	20
6.2.3	PyTorch . . . . .	21
6.2.4	JAX . . . . .	21
6.2.5	Сравнение и выбор . . . . .	21
6.3	Эксперименты . . . . .	22
6.3.1	2D данные . . . . .	22
6.3.2	EMNIST . . . . .	23
<b>7</b>	<b>Заключение</b>	<b>25</b>



Рис. 1: Два домена: синтетические и реальные виды города

## 1 Введение

Одно из активно развивающихся направлений в сфере машинного обучения является трансляция доменов. Данная задача состоит в изучении отображения  $T$ , который переносит элементы одного домена в другой. Например, трансляция синтетических изображения с камеры видео-регистратора в реальные (рисунок 1).

Сильным развитием данной задачи послужила, частная формулировка трансляции домена, а именно генеративное моделирование. Пусть дан наблюдаемый набор данных  $\mathcal{D}$ , элементы  $d$  которого принадлежат базовому распределению  $d \sim p_{data}$ . Цель любой генеративной модели состоит в том, чтобы аппроксимировать это распределение данных при наличии доступа к набору данных  $\mathcal{D}$ . После обучения, такая модель позволяет генерировать новые объекты  $\hat{d} \notin \mathcal{D}$ .

Так, например, состязательные сети (GAN), предложенные Гудфуллоу в статье [1] стали основой одного из значимых методов в задаче трансляции домена. В данном подходе обучается две модели генератор и дискриминатор, первая из которых пытается сгенерировать реалистичные данные, а вторая пытается определить, являются ли данные реальными или сгенерированными. Разработанный подход позволяет обучать неявным образом генеративные распределения, используя только сэмплы из реального распределения данных  $p_{data}$ . Методом, который обобщает GAN на задачу трансляции доменов, является CycleGAN, предложенный в [2]. Данный метод обучает два GAN, которые отображают данные в элементы противоположного набора.

Следующий класс генеративных моделей, адаптированных на задачу трансляции домена, стали диффузионные модели [3]. Они полагаются на итеративную процедуру, при которой распределение данных  $p_{data}$  сначала искажается с помощью процесса прямого зашумления, сходящегося к нормальному распределению. Затем изучается обратный во времени процесс, который восстанавливает данные из гауссовского распределения обратно в распределение данных, с использованием score matching подходов.

Для обобщения данной задачи важную роль сыграло осознание ограничения в виде фиксированного нормального распределения, которое является не самым оптимальным выбором скрытого пространства. Например, для задачи восстановления зашумленного изображения, имея начальное изображение, имеет смысл восстанавливать начиная с данного изображения. Однако, это требует изменения целевого распределения прямого процесса, что часто неразрешимо и приводит к численным трудностям.

В результате был предложен метод, решающий данную проблему, а именно Bridge matching [?]. Аналогично с диффузионными моделями данный метод, строит два стохастических процесса, прямой и обратный, между двумя априорными распределениями, таким образом обобщая задачу генеративного моделирования на задачу трансляции доменов.

Однако вышеуказанные подходы обладают рядом недостатков. Во-первых, для Bridge matching требуются специализированные наборы данных, объекты которых являются пары из двух заданных конечных распределений, то есть во время обучения оба элемента пары должны соответствовать одному объекту, например, изображение представленное в двух видах: зашумленное, высокого качества. Во-вторых, в данных методах никак не задается оптимальность отображений. В статье показывается, что данные методы ориентированы на небольшие преобразования. Это не является проблемой, когда распределения близки друг к другу, но в обратной ситуации, это не позволяет уловить более сложную семантику распределений. Таким образом, возникает потребность внедрить информацию о данных, что требует для каждой задачи адаптировать целевую функцию.

Решением этих проблем является введение дополнительных критериев оптимальности отображений. Такими требованиями обладают мосты Шрёдингера [4], [5]. Задача Мостов Шрёдингера находит стохастический процесс (рисунок 2), ограниченный двумя заданными конечными распределениями, который наиболее похож с винеревским процессом с точки зрения дивергенции Кульбака — Лейблера. Такая формулировка задачи, позволяет находить отображения между двумя **непарными** наборами данных, а также наиболее оптимальным образом, с точки зрения квадрата расстояния между элементами двух распределений  $\frac{\|x-y\|^2}{2}$ . Кроме того, данная задача позволяет вычислять вероятность этой стохастической процесса, что позволяет нам сравнивать два набора данных, то есть распределения, что может быть полезно для проверки гипотез и семантического сходства.

Также, несомненная актуальность исследований в сфере мостов Шрёдингера подтверждается количеством статей, публикуемых в данное время. Так по запросу сайта arXiv<sup>1</sup> количество статей за последние 12 месяцев составило 56, когда год назад за такой же промежуток времени результат оказался в два раза меньше и составил 23 статьи.

Из-за родства диффузионных генеративных моделей с мостами Шрёдингера было предложено несколько подходов [6], [7], [8], [9]. Несмотря на качество полученных результатов подходы предложенные в этих статьях являются вычислительно-дорогостоящими из-за этого возникает потребность в использовании других методов построения мостов Шрёдингера. Одним из примеров решения данной проблемы является подход описанный в следующей статье [10]. Однако, несмотря на значительную скорость обучения, данный подход плохо справляется с данными большой размерности.

Благодаря исследованиям состязательных генеративных сетей [11], а именно пониманию, что целевая функция GAN может быть обобщена до f-семейства дивергенций, в данной работе предлагается новый подход к обучению мостов Шрёдингера, который наследует преимущества состязательных генеративных сетей и решает вышеперечисленные проблемы.

<sup>1</sup><https://arxiv.org/>

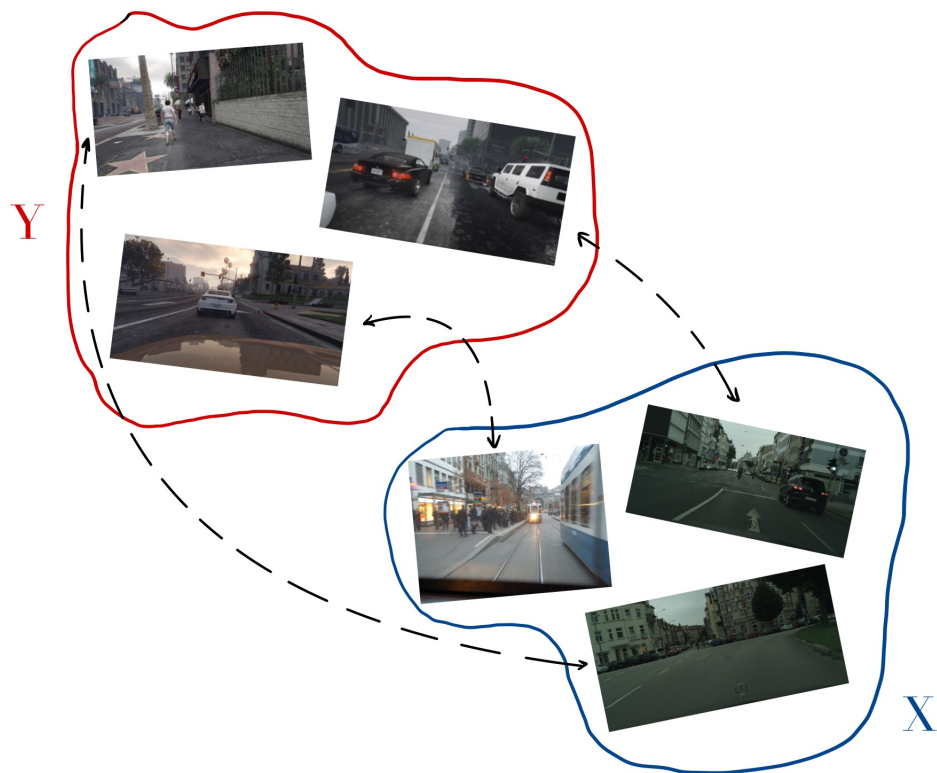


Рис. 2: Иллюстрация мостов Шрёдингера

Далее вводятся определения понятий, которые понадобятся в постановке задачи.

...

## 2 Задача Мостов Шрёдингера

Существует несколько эквивалентных задач мостов Шрёдингера. Для понимания проблематики актуальных решений и перед тем как перейти к постановке задачи и предложенному методу решения, потребуется рассмотреть три основные и их итеративные алгоритмы решения. Для этого, сначала потребуется ввести определение меры пути.

**Определение 2.1** (Мера пути). *Для процесса Ито вида*

$$dx(t) = b(t) + \sigma(t)d\beta(t),$$

*определенного на интервале  $[0, T]$ , с функцией сноса  $b(t)$ ,  $\sigma(t)$  – функция волатильности,  $\mathbb{P}$  называется мерой пути данного процесса с пространством исходов  $\Omega = C([0, T], \mathbb{R}^d)$ , если распределение  $\mathbb{P}$  описывает слабое решение данного СДУ.*

Иначе говоря, мера пути представляет собой меру вероятности, ассоциированную со случайным процессом, задаваемым СДУ. Например,  $\mathbb{W}^\gamma$  является мерой Винера и представляет вероятность траекторий винеровского процесса с волатильностью  $\sqrt{\gamma}$  то есть  $dx(t) = \sqrt{\gamma}d\beta(t)$ .

### 2.1 Динамическая постановка задачи Мостов Шрёдингера

Проблема моста Шрёдингера возникает из вопроса: как ограничить некоторый заданный случайный процесс? Ответ дает теория больших отклонений, в частности, теорема Санова [12].

**Теорема 2.1** (Санов). *Пусть заданы  $\{x_i(t)\}_{i=1}^N \sim \mathbb{W}^\gamma$  – независимые одинаково распределённые траектории из меры пути априорного винеровского процесса  $\mathbb{W}^\gamma$ , где  $x_i(t) \in C([0, T], \mathbb{R}^d), \forall i = \overline{1, N}$ . Также пусть дана эмпирическая мера  $\hat{\mathbb{W}}$  по заданным траекториям:*

$$\hat{\mathbb{W}}(A) = \frac{1}{N} \sum_{i=1}^N 1(x_i(t) \in A), A \in \mathcal{B}(\mathbb{R}^d)^{[0, T]},$$

*тогда вероятность, что  $\hat{\mathbb{W}}$  ограничена заданными распределениями  $\pi_0$  и  $\pi_T$  задана следующей асимптотической оценкой:*

$$P\left(\hat{\mathbb{W}} \in \mathcal{D}(\pi_0, \pi_T)\right) \xrightarrow{N \rightarrow \infty} \exp\left(-N \inf_{\mathbb{Q} \in \mathcal{D}(\pi_0, \pi_T)} D_{KL}(\mathbb{Q} \parallel \mathbb{W})\right), \quad (1)$$

*где  $\mathcal{D}(\pi_0, \pi_T)$  – множество всех мер путей ограниченных заданными распределениями  $\pi_0$  и  $\pi_T$ .*

Таким образом, для того чтобы ограничить  $\hat{\mathbb{W}}$  распределениями  $\pi_0$  и  $\pi_T$ , необходимо чтобы:

$$P\left(\hat{\mathbb{W}} \in \mathcal{D}(\pi_0, \pi_T)\right) = 1,$$

следовательно, чтобы степень экспоненты была нулевой. Для этого необходимо, чтобы дивергенция меры пути совпадали и Кульбака-Лейблера была нулевой.

Первой постановкой является динамическая. Она прямо следует из асимптотической оценки Санова 1 и находит такую меру пути  $\mathbb{Q}$ , ограниченную заданными распределениями  $\pi_0(x)$  и  $\pi_T(y)$ , что наиболее близка с точки зрения теории информации к априорной мере пути винеровского процесса  $\mathbb{W}^\gamma$  с волатильностью  $\sqrt{\gamma}$ :

$$\hat{\mathbb{Q}} = \arg \min_{\mathbb{Q} \in \mathcal{D}(\pi_0, \pi_T)} D_{KL}(\mathbb{Q} \parallel \mathbb{W}^\gamma) \quad (2)$$

## 2.2 Статическая постановка мостов Шрёдингера

Помимо динамической постановки также существует и статическая. Для ее вывод и доказательства эквивалентности рассмотрим декомпозицию дивергенции Кульбака-Лейблера в уравнении 2. Сначала выразим производную Радона-Никодима, условную по  $\mathbf{x}(0) = x$  и  $\mathbf{x}(T) = y$ :

$$\frac{d\mathbb{Q}}{d\mathbb{W}^\gamma} = \frac{q(x, y)}{p^{\mathbb{W}^\gamma}(x, y)} \frac{d\mathbb{Q}_{(0,T)}}{d\mathbb{W}_{(0,T)}^\gamma}(\cdot|x, y)$$

Подставляя это выражение в дивергенцию Кульбака-Лейблера в уравнении 2, получаем:

$$\begin{aligned} D_{KL}(\mathbb{Q}||\mathbb{W}^\gamma) &= \mathbb{E}_{\mathbb{Q}} \left[ \log \frac{d\mathbb{Q}}{d\mathbb{W}^\gamma} \right] = \mathbb{E}_{\mathbb{Q}} \left[ \log \frac{q(x, y)}{p^{\mathbb{W}^\gamma}(x, y)} \frac{d\mathbb{Q}_{(0,T)}}{d\mathbb{W}_{(0,T)}^\gamma}(\cdot|x, y) \right] = \\ &= \int \log \frac{q(x, y)}{p^{\mathbb{W}^\gamma}(x, y)} d\mathbb{Q} + \int \log \frac{d\mathbb{Q}_{(0,T)}}{d\mathbb{W}_{(0,T)}^\gamma}(\cdot|x, y) d\mathbb{Q} = \\ &= \int q(x, y) \log \frac{q(x, y)}{p^{\mathbb{W}^\gamma}(x, y)} dx dy + \int \log q(x, y) \frac{d\mathbb{Q}_{(0,T)}}{d\mathbb{W}_{(0,T)}^\gamma}(\cdot|x, y) d\mathbb{Q}_{(0,T)} = \\ &= D_{KL}(q(x, y)||p^{\mathbb{W}^\gamma}(x, y)) + \mathbb{E}_{q(x, y)} \left[ D_{KL}(\mathbb{Q}_{(0,T)}(\cdot|x, y)||\mathbb{W}_{(0,T)}^\gamma(\cdot|x, y)) \right] \end{aligned}$$

Видно, что второй член дивергенции Кульбака-Лейблера не учитывает маргинальные распределения, поэтому он не влияет на оптимизационные ограничения и может быть отброшен. Таким образом, не рассматривается динамика на интервале  $(0, 1)$  и для решения задачи важно только совместное распределение между крайними маргинальными распределениями. В результате получается статическая постановка задачи мостов Шрёдингера:

$$\begin{cases} \hat{q}(x, y) = \arg \min_{q(x, y)} D_{KL}(q(x, y)||p^{\mathbb{W}^\gamma}(x, y)), \\ \pi_0(x) = \int q(x, y) dy, \\ \pi_T(y) = \int q(x, y) dx, \end{cases} \quad (3)$$

где  $q(x, y)$  — это совместное распределение, которое является ближайшим к броуновскому движению при условии ограничений на маргинальные распределения (между плотностями вероятностей в момент времени 0 и T).

## 2.3 Система Шрёдингера

Третья формулировка требует рассмотрения Лагранжиана статической постановки 3:

$$\begin{aligned} L(q, \lambda, \mu) &= D_{KL}(q(x, y)||p^{\mathbb{W}^\gamma}(x, y)) + \int \lambda(x) \left( \int q(x, y) dy - \pi_0(x) \right) dx + \\ &+ \int \mu(y) \left( \int q(x, y) dx - \pi_T(y) \right) dy \end{aligned}$$

Предполагая, что  $p^{\mathbb{W}^\gamma}(x, y) = p_0^{\mathbb{W}^\gamma}(x) p^{\mathbb{W}^\gamma}(y|x)$ , где  $p_0^{\mathbb{W}^\gamma}(x)$  может быть любым, а т.к. априорный процесс является винеровским,  $p^{\mathbb{W}^\gamma}(y|x) = \mathcal{N}(y|x, \gamma I_d)$ , приравняем  $\frac{\partial L(q, \lambda, \mu)}{\partial q(x, y)}$  нулю и получим:

$$q^*(x, y) = \exp(\ln p^{\mathbb{W}^\gamma}(x, y) - \lambda(x) - 1) p^{\mathbb{W}^\gamma}(y|x) \exp(-\mu(y))$$

Теперь положив, что  $\hat{\phi}_0(x) = \exp(\ln p^{\mathbb{W}^\gamma}(x, y) - \lambda(x) - 1)$  и  $\phi_T(y) = \exp(-\mu(y))$  получаем:

$$q^*(x, y) = \hat{\phi}_0(x) p^{\mathbb{W}^\gamma}(y|x) \phi_T(y),$$

удовлетворяющее:

$$\hat{\phi}_0(x) \int \phi_T(y) p_W^\gamma(y|x) dy = \pi_0(x),$$

$$\phi_T(y) \int \hat{\phi}_0(x) p_W^\gamma(y|x) dx = \pi_T(y),$$

Теперь переобозначим термины с интегралами как:

$$\phi_0(x) = \int \phi_T(y) p_W^\gamma(y|x) dy,$$

$$\hat{\phi}_T(y) = \int \hat{\phi}_0(x) p_W^\gamma(y|x) dx.$$

Объединяя все вместе, следующая линейная функциональная система является системой Шрёдингера:

$$\begin{cases} \hat{\phi}_0(x) \phi_T(x) = \pi_0(x), \\ \hat{\phi}_T(x) \phi_T(y) = \pi_T(y). \end{cases} \quad (4)$$

## 2.4 Динамические полумосты Шрёдингера

Наиболее важной для этой работы является постановка задачи полумостов Шрёдингера. В отличии от оригинальной задачи мостов Шрёдингера, задача полумостов ограничивается только на одну сторону, то есть  $\mathcal{D}(\pi_0(x), \cdot)$  или  $\mathcal{D}(\cdot, \pi_T(y))$ .

Более формально задача прямого полумоста, то есть, ограниченного в момент времени 0, определяется так:

$$\mathbb{Q}^* = \arg \min_{\mathbb{Q} \in \mathcal{D}(\pi_0(x), \cdot)} D_{KL}(\mathbb{Q} || \mathbb{W}^\gamma) \quad (5)$$

а задача обратного определяется следующим образом:

$$\mathbb{P}^* = \arg \min_{\mathbb{P} \in \mathcal{D}(\cdot, \pi_T(y))} D_{KL}(\mathbb{P} || \mathbb{W}^\gamma) \quad (6)$$

В отличии от оригинальной постановки задачи мостов Шрёдингера, задачи обоих полумостов имеет следующее аналитическое решение:

$$\mathbb{Q}^*(A_0 \times A_{(0,1]}) = \int_{A_0 \times A_{(0,1]}} \frac{d\pi_0}{p_0^{\mathbb{W}^\gamma}}(x) d\mathbb{W}^\gamma$$

$$\mathbb{P}^*(A_{[0,T)} \times A_T) = \int_{A_{[0,T)} \times A_T} \frac{d\pi_T}{p_T^{\mathbb{W}^\gamma}}(y) d\mathbb{W}^\gamma$$

для прямого и обратного мостов соответственно.



### 2.4.1 Статические полумосты Шрёдингера

Аналогично и оригинальной постановке рассмотрим статическую формулировку полумостов. Прямой полумост задается следующим образом:

$$q^*(x, y) = \arg \min_{q(x, y) \in \mathcal{D}(\pi_0(x), \cdot)} D_{KL}(q(x, y) || p^{\mathbb{W}\gamma}(x, y)),$$

такое, что  $\pi_0(x) = \int q(x, y) dy$ ,

(7)

а обратный:

$$p^*(x, y) = \arg \min_{p(x, y) \in \mathcal{D}(\cdot, \pi_T(y))} D_{KL}(p(x, y) || p^{\mathbb{W}\gamma}(x, y)),$$

такое, что  $\pi_T(y) = \int q(x, y) dx$

(8)

Как и для динамической постановки статическая обладает аналитическим решением:

$$q(x, y)^* = p(x, y)^{\mathbb{W}\gamma} \frac{\pi_0(x)}{p^{\mathbb{W}\gamma}(x)}$$

$$p(x, y)^* = p(x, y)^{\mathbb{W}\gamma} \frac{\pi_T(y)}{p^{\mathbb{W}\gamma}(y)},$$

для прямого и обратного полумостов, соответственно.

Полумосты полезны не только из-за того, что обладают аналитическим решением, но и тем что позволяют легко справиться с ограничением на искомый мост, заменяя его на задачу с начальным значением. Также благодаря полумостам итеративно решается оригинальная задача мостов Шрёдингера.

## 2.5 Методы решения мостов Шрёдингера

Далее рассмотрим итеративные алгоритмы, о которых упоминалось ранее, для решения мостов Шрёдингера.

Алгоритм Фортета 1 является одним из старейших методов, доказавших свою сходимость для решения системы Шрёдингера. Этот алгоритм, впервые предложенный Фортетом в 1940 году, включает итеративный процесс, обеспечивающий выполнение маргинальных условий системы Шрёдингера 4 для распределений в моменты времени  $t = 0$  и  $t = 1$ .

---

#### Algorithm 1: Фортет

---

**Дано:**  $\pi_0(x)$ ,  $\pi_T(y)$ ,  $p(y|x)$

**Найти:**  $\hat{\phi}_0^{(i)}(x)$ ,  $\hat{\phi}_T^{(i)}(y)$

- 1 Инициализируем  $\phi_0^{(0)}(x)$  такое, что  $\phi_0^{(0)}(x) \ll \pi_0(x)$ ;
  - 2 Инициализируем  $i = 0$ ;
  - 3 **while** не сойдется **do**
  - 4      $\hat{\phi}_0^{(i)}(x) := \frac{\pi_0(x)}{\phi_0^{(i)}(x)}$ ;
  - 5      $\hat{\phi}_T^{(i)}(y) := \int p(y|x) \hat{\phi}_0^{(i)}(x) dx$ ;
  - 6      $\phi_T^{(i)}(y) := \frac{\pi_T(y)}{\hat{\phi}_T^{(i)}(y)}$ ;
  - 7      $\hat{\phi}_T^{(i+1)}(x) := \int p(y|x) \phi_T^{(i)}(y) dy$ ;
  - 8      $i := i + 1$ ;
-

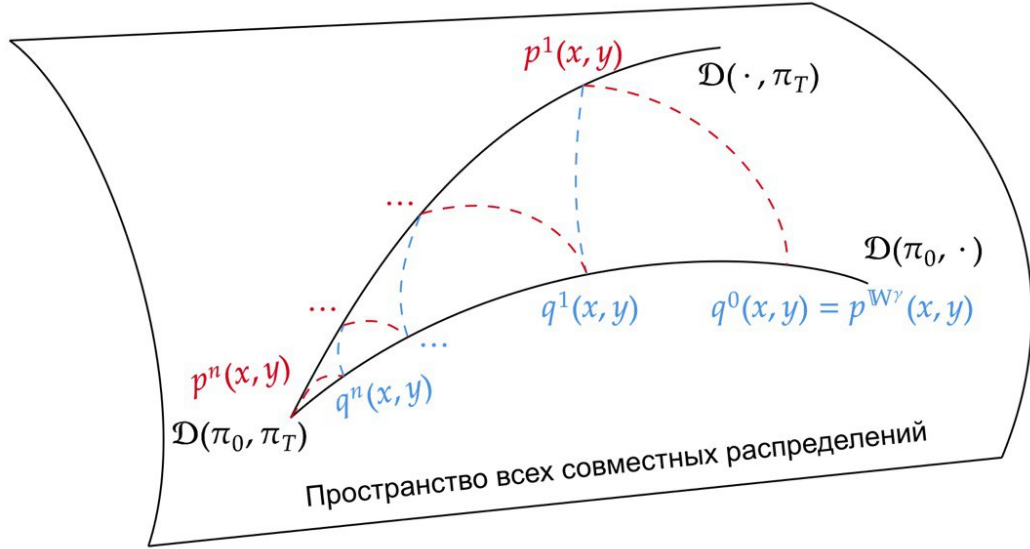


Рис. 3: Иллюстрация алгоритма IPF

Алгоритм Фортета фактически чередует выполнение маргинальных условий для моментов времени  $t = 0$  и  $t = T$ , пока не будет достигнута сходимость. В результате достигается решение, соответствующее заданным граничным условиям для обоих моментов времени.

Другим методом решения задачи мостов Шрёдингера является алгоритм Iterative Proportional Fitting (IPF). IPF – это метод, используемый для нахождения дискретное совместное распределение при заданных маргинальных, используя принципы максимальной энтропии или минимизации дискриминационной информации. Однако в контексте мостов Шрёдингера, а именно статической постановки задачи, в 1968 году Кульбаком [13] был разработан непрерывный вариант IPF, а сходимость его доказана Рупендорфом [14].

---

**Algorithm 2: IPF**


---

**Дано:**  $\pi_0(x)$ ,  $\pi_T(y)$ ,  $p(y|x)$

**Найти:**  $q_i^*(x, y)$ ,  $p_i^*(x, y)$

- 1 Инициализируем  $p_T^{\mathbb{W}^\gamma}(y)$  такое, что  $p_T^{\mathbb{W}^\gamma}(y) \ll \pi_T(y)$ ;
  - 2 Инициализируем  $q_0^*(x, y) := p^{\mathbb{W}^\gamma}(x, y)$ ;
  - 3 Инициализируем  $i = 0$ ;
  - 4 **while** не сойдется **do**
  - 5      $p_i^*(x, y) = \arg \min_{p(x, y) \in \mathcal{D}(\cdot, \pi_T(y))} D_{KL}(p(x, y) || q_{i-1}^*(x, y))$ ;
  - 6      $q_i^*(x, y) = \arg \min_{q(x, y) \in \mathcal{D}(\pi_0(x), \cdot)} D_{KL}(q(x, y) || p_i^*(x, y))$ ;
  - 7      $i := i + 1$ ;
- 

Суть алгоритма Кульбака заключается в чередовании решения полумостов начиная с обратного. Однако, если шаги 5 и 6 переписать с помощью аналитического решения задачи полумостов, а также начать алгоритм с прямого прохода, данный алгоритм сводится к алгоритму Фортета. Иллюстрация IPF изображена на рисунке 3

Следующий алгоритм generalised IPF (g-IPF) обобщает IPF, предложенный Кульбаком, применяя его к динамической постановке задачи. Этот метод идентичен обычному IPF, но формулирует задачу в терминах меры пути.

---

**Algorithm 3: g-IPF**


---

**Дано:**  $\pi_0(x), \pi_T(y), \mathbb{W}^\gamma$

**Найти:**  $\mathbb{Q}_i^*(x, y), \mathbb{P}_i^*(x, y)$

- 1 Инициализируем  $\mathbb{Q}_0^*(x, y) := \mathbb{W}^\gamma$ ;
  - 2 Инициализируем  $i = 0$ ;
  - 3 **while** не сойдётся **do**
  - 4      $\mathbb{P}_i^*(x, y) = \arg \min_{\mathbb{P}(x, y) \in \mathcal{D}(\cdot, \pi_T(y))} D_{KL}(\mathbb{P}(x, y) || \mathbb{Q}_{i-1}^*(x, y));$
  - 5      $\mathbb{Q}_i^*(x, y) = \arg \min_{\mathbb{Q}(x, y) \in \mathcal{D}(\pi_0(x), \cdot)} D_{KL}(\mathbb{Q}(x, y) || \mathbb{P}_i^*(x, y));$
  - 6      $i := i + 1$ ;
- 

### 3 Обзор существующих решений

Для того чтобы сформулировать проблему и соответствующую постановку задачи, исследуемую в данной работе, в текущей главе будут рассмотрены существующие методы решения мостов Шрёдингера в контексте машинного обучения.

#### 3.1 Data Driven Schrödinger Bridge

Подход, предложенный в [15], является адаптацией алгоритма 1 к эмпирической постановке. Данный метод основан на том факте, что шаги 4 и 6 1:

$$\hat{\phi}_0^{(i)}(x) := \frac{\pi_0(x)}{\phi_0^{(i)}(x)},$$

$$\phi_T^{(i)}(y) := \frac{\pi_T(y)}{\hat{\phi}_T^{(i)}(y)}$$

могут быть эквивалентно сформулированы как проблема минимизации кросс-энтропии в пространстве распределений  $\mathcal{H}$ :

$$\hat{\phi}_0^{(i)}(x) = \arg \sup_{\phi_0(x) \in \mathcal{H}} \mathbb{E}_{\pi_0(x)} \left[ \ln \hat{\phi}_0(x) \phi_0^{(i)}(x) \right],$$

$$\phi_T^{(i)}(y) = \arg \sup_{\phi_T(y) \in \mathcal{H}} \mathbb{E}_{\pi_T(y)} \left[ \ln \phi_T(y) \hat{\phi}_T^{(i)}(y) \right].$$

Далее внедряя шаги 5 и 7 алгоритма 1 с помощью лагранжиана и параметризуя потенциалы с помощью смеси гауссиан, получают следующие два шага нового алгоритма:

$$\beta_i^* = \arg \max \hat{\beta} \frac{1}{M} \sum_s \ln \hat{\phi}_0(x_s; \hat{\beta}) \phi_0^{(i)}(x_s) - \int \hat{\phi}_0(x; \hat{\beta}) \phi_0^{(i)}(x) dx, \quad x_s \sim \pi_0(x),$$

и

$$\beta_i^* = \arg \max \beta \frac{1}{N} \sum_s \ln \phi_T(y_s; \beta) \hat{\phi}_T^{(i)}(y_s) - \int \phi_T(y; \beta) \hat{\phi}_T^{(i)}(y) dy, \quad y_s \sim \pi_T(y).$$

Далее декомпозируются ограничения введенные лагранжианом и оцениваются с помощью метода Монте Карло и выборки по значимости получают финальные оптимизационные задачи. Из-за проклятия размерности метода выборки по значимости, предложенный метод не масштабируются до данных с большой размерностью.

### 3.2 Light Schrödinger Bridge

Данный метод предложенный в [10] решает задачу мостов Шрёдингера в статической формулировке 3, декомпозируя совместное распределение с помощью потенциалов:

$$\pi_\theta(x_0, x_1) = \pi_0(x)p^{\mathbb{W}^\gamma}(y|x) = \pi_0(x)\frac{p^{\mathbb{W}^\gamma}(y|x)\phi_T(y)}{\hat{\phi}_0(x)}.$$

Аналогично с предыдущим методом авторы параметризуют потенциалы с помощью смеси гауссиан. Далее авторы замечают, что данная декомпозиция позволяет сформулировать целевую функцию, минимизация которой эквивалентна минимизации дивергенции Кульбака-Лейблера:

$$L(\theta) = \int_{\mathbb{R}^D} \log \hat{\phi}_0(x) \pi_0(x) dx - \int_{\mathbb{R}^D} \log \phi_T(y) \pi_T(y) dy.$$

Напрямую решая задачу мостов Шрёдингера (без IPF), авторам удалось значительно ускорить обучения моделей, однако такой метод не предназначен для работы с данными большой размерности.

### 3.3 Diffusion Schrödinger Bridge

Данный метод предложен Валентином Де Бортоли в статье [6]. Автор рассматривает динамическую постановку задачи 2 в дискретном виде, т.е. вместо меры пути используют совместное распределение в точках стохастического процесса:

$$q^*(x_0, \dots, x_T) = \arg \min_{q \in \mathcal{D}(\pi_0, \pi_T)} D_{KL}(q(x_0, \dots, x_T) || p^{\mathbb{W}^\gamma}(x_0, \dots, x_T)).$$

Авторы доказывают, что формулировка задачи мостов Шрёдингера через совместные вероятности эквивалентна динамической постановке мостов Шрёдингера, у которой ассоциированный стохастический процесс дискретизирован с помощью метода Эйлера-Маруямы. Используя данный факт, авторы [6] применяют алгоритм IPF, адаптированный к дискретной формулировке, для решения задачи мостов Шрёдингера. В результате основываясь на факте, что винеровский процесс является марковским, авторы формулируют минимизационные задачи IPF следующим образом:

$$\begin{aligned} p^{i+1}(x_{0:T}) &= \pi_0(x_0) \prod_{t=0}^{T-1} \left( \frac{q_{t|t+1}^i(x_t|x_{t+1}) q_{t+1}^i(x_{t+1})}{q_t^i(x_t)} \right), \\ q^i(x_{0:T}) &= \pi_T(x_T) \prod_{t=0}^{T-1} \left( \frac{p_{t+1|t}^i(x_{t+1}|x_t) p_t^i(x_t)}{p_{t+1}^i(x_{t+1})} \right). \end{aligned} \tag{9}$$

Так как такую динамику смоделировать невозможно, условные вероятности  $p_{t+1|t}^i(x_{t+1}|x_t)$  и  $q_{t|t+1}^i(x_t|x_{t+1})$  представляют с помощью нормального распределения:

$$\begin{aligned} p_{t+1|t}^i(x_{t+1}|x_t) p_t^i(x_t) &= \mathcal{N}(x_{t+1}; x_t + \gamma_{t+1} f_t^i(x_t), 2\gamma_{t+1} \mathbb{I}) \\ q_{t|t+1}^i(x_t|x_{t+1}) &= \mathcal{N}(x_t; x_{t+1} + \gamma_t b_{t+1}^i(x_{t+1}), 2\gamma_t \mathbb{I}). \end{aligned}$$

Далее аппроксимировав с помощью ряда Тейлора, для 9 авторы получают:

$$\begin{aligned} p_{t+1|t}^{i+1}(x_{t+1}|x_t) &\approx \mathcal{N}(x_{t+1}; x_t + \gamma_{t+1} f_t^{i+1}(x_t), 2\gamma_{t+1} I), \\ q_{t|t+1}^i(x_t|x_{t+1}) &\approx \mathcal{N}(x_t; x_{t+1} + \gamma_{t+1} b_{t+1}^i(x_{t+1}), 2\gamma_{t+1} I), \end{aligned}$$

где

$$\begin{aligned} f_t^{i+1}(x_t) &= -b_{t+1}^i(x_t) + 2\nabla \log q_t^i(x_t), \\ b_{t+1}^i(x_{t+1}) &= -f_t^i(x_{t+1}) + 2\nabla \log p_{t+1}^i(x_{t+1}). \end{aligned}$$

Затем логарифмы градиентов могут быть аппроксимированы с использованием score matching методов. Однако авторы, из соображений затраты памяти и вычислительной сложности аппроксимируют, среднее нормальных распределений 9, то есть функции дрейфа ассоциированных стохастических процессов.

Благодаря применению задачи мостов Шрёдингера, авторам удалось сократить требуемое число диффузионных шагов в сравнении с score-based моделями. Так, например, для задач генерации изображений CelebA и MNIST для score-based моделей потребовалось 100 диффузионных шагов, а для предложенного метода [6] 12.

### 3.4 Iterative Proportional Maximum Likelihood (IPML)

В работе [8] авторы преобразуют алгоритм IPF в виде задачи максимума правдоподобия. Аналогично ранее описанной работе в данном методе используется тот факт, что мера пути в динамической постановке задачи мостов Шрёдингера, может быть параметризована с помощью функции сноса, соответствующего стохастического дифференциального уравнения. Однако в место того, чтобы оценивать функцию сноса с помощью score matching подхода, авторы данного метода оценивают, используя гауссовские процессы.

Также, в отличии от метода [15], где находится оценка маргинальных распределений с помощью метода максимума правдоподобия, авторы данной работы находят оценку условного распределения, то есть смещения. В такой постановке не требуется оценивать интеграл с помощью метода Монте Карло, что позволяет работать с данными большой размерности. Однако не смотря на это, для отображения элементов из одного набора данных в другой, моделирует стохастический процесс.

### 3.5 Unpaired Neural Schrödinger Bridge

Наиболее близкий метод к предложенному в данной работе является подход описанный в статье [9]. Авторы рассматривают динамическую задачу мостов Шрёдингера и аналогично предыдущим методам параметризуют условное распределение  $q(x_{t+1}|x_t)$ . Однако, в отличии от них данный метод не использует IPF для обучения, что сильно упрощает оптимизационную задачу.

Таким образом, несмотря на введение состязательного обучения, данный метод не решает проблему моделирования стохастического процесса.

## 4 Постановка задачи

В контексте машинного обучения, зачастую невозможно получить распределение данных в явной форме. Это вызывает необходимость рассмотрения альтернативных постановок задачи мостов Шрёдингера, в которых маргинальные распределения известны лишь посредством выборок, представленных в виде двух наборов данных  $\{x_i\}_{i=0}^N = X$  и  $\{y_j\}_{j=0}^M = Y$ .

Эмпирическими мостами Шрёдингера называется такая постановка задачи мостов Шрёдингера, в которой маргинальные распределения известны только через выборки (то есть их эмпирические распределения):

$$\hat{\pi}_0(x) = \frac{1}{N} \sum_{i=1}^N \delta(x - x_i), \quad \hat{\pi}_T(y) = \frac{1}{M} \sum_{j=1}^M \delta(y - y_j),$$

Элементы этих наборов принадлежат соответствующим распределениям, а именно  $x_0 = x \sim \pi_0(x)$  и  $x_T = y \sim \pi_T(y)$ .

Проанализировав существующие подходы, можно заключить, что большинство методов, таких как [6], [9], [7], [8], [16], основываются на моделировании дискретизированного стохастического процесса, представленного марковской цепочкой:

$$q(x_0, \dots, x_T) = \pi_1(x_T) \prod_{t=0}^{T-1} q_{t|t+1}(x_t | x_{t+1}).$$

В этих методах условное распределение  $q_{t|t+1}$  параметризуется с использованием нейронных сетей. Для преобразования одного набора данных в другой в этих методах итеративно сэмплируются элементы с использованием обученного  $q_{t|t+1}$ , таким образом моделируя стохастический процесс. В результате таких преобразований требуется несколько шагов для переноса элемента из одного набора в другой. Несмотря на значительный прогресс в сокращении числа необходимых шагов (например, [6] по сравнению с [3]), такие методы все еще являются вычислительно сложной задачей.

С другой стороны, методы, которые рассматривают задачу мостов Шрёдингера без моделирования процесса, такие как [10], [15], плохо справляются с данными большой размерности из-за необходимости явной параметризации и обучения условного распределения.

В связи с этим, возникает необходимость разработки подхода к решению задачи мостов Шрёдингера, который бы обладал следующими качествами:

- преобразование из одного распределения в другое должно осуществляться за один шаг;
- метод должен быть эффективным при работе с данными большой размерности.

Таким образом, цель данной работы состоит в предложении решения задачи мостов Шрёдингера, которое бы удовлетворяло указанным выше требованиям. В частности, предлагаемый подход должен обеспечивать одноступенчатое отображение распределений и быть пригодным для применения к данным большой размерности, что позволит преодолеть существующие ограничения современных методов и повысить их эффективность в практических применениях.

## 5 Исследование и построение решения задачи

Требуется разбить большую задачу, описанную в постановке, на более мелкие подзадачи. Процесс декомпозиции следует продолжать до тех пор, пока подзадачи не станут достаточно простыми для решения непосредственно. Это может быть достигнуто, например, путем проведения эксперимента, доказательства теоремы или поиска готового решения.

Исходя из требований поставленных ранее, данным условиям соответствуют состязательные генеративные модели. Во-первых, они хорошо работают с данными большой размерности, достигая передовые результаты в задачах генерации изображений [?], а во-вторых, генерация данных заключается в отображении элементов скрытого пространства в пространства данных, вызовом генератора, то есть за один шаг.

Для того чтобы связать состязательные генеративные модели с задачей мостов Шрёдингера, необходимо подробнее рассмотреть, что они из себя представляют.

### 5.1 Созязательные генеративные модели (Generative-adversarial networks, GAN) и их обобщение f-GAN

В статье Гудфеллоу [1] был представлен новый метод оценки генеративных моделей с использованием состязательного обучения.

Такой метод обучает генеративную модель  $G$  состязательно (генеративное распределение  $p_g$  изучается неявно), вводя дискриминативную модель  $D$ , которая пытается определить, является ли выборка реальной или сгенерированной:

$$\begin{aligned} \min_G \max_D V(G, D) &= \min_G \max_D \mathbb{E}_{x \sim p_{data}} [\log D(x)] - \mathbb{E}_{x \sim p_g} [1 - \log D(x)] = \\ &= \min_G \max_D \mathbb{E}_{x \sim p_{data}} [\log D(x)] - \mathbb{E}_{z \sim p(z)} [1 - \log D(G(z))], \end{aligned}$$

Важно отметить, что авторы доказывают, что выбор оптимального дискриминатора  $D^*$  аналогичен минимизации дивергенции Йенсена-Шеннона между  $p_{data}$  и  $p_g$ , то есть:

$$\begin{aligned} \min_G \max_D \mathbb{E}_{x \sim p_{data}} [\log D(x)] - \mathbb{E}_{x \sim p_g} [1 - \log D(x)] &= \\ = \min_G \mathbb{E}_{x \sim p_{data}} [\log D^*(x)] - \mathbb{E}_{x \sim p_g} [1 - \log D^*(x)] &= \min_G D_{JS}(p_{data} || p_g) - 2 \log 2, \quad (10) \end{aligned}$$

Опираясь на этот факт, в статье [11] расширяется теория состязательных сетей до более общего принципа, используя вариационное (двойственное) представление f-дивергенции.

Однако сначала нам нужно ввести понятие *сопряженной функции*.

**Определение 5.1** (Сопряженная функция). Пусть  $f : \text{dom}(f) \rightarrow \mathbb{R}$  – выпуклая функция, где  $\text{dom}(f) \subseteq \mathbb{R}$  – это интервал. Тогда выпуклая сопряженная функция  $f^*$  – это функция  $f^* : \text{dom}(f^*) \rightarrow \mathbb{R}$  определенная следующим образом:

$$f^*(y) = \sup_{x \in \text{dom}(f)} (yx - f(x)),$$

где  $\text{dom}(f^*) := \{y \in \mathbb{R} : f^*(y) < \infty\}$

Для формулировки вариационного представления также потребуются свойства выпуклой сопряженной функции.

**Свойство 5.1.** Выпуклая сопряженная функция  $f^*$  функции  $f$  удовлетворяет следующим свойствам:

1.  $f^*$  непрерывна в своей области определения;
2.  $f^*$  выпукла;
3.  $(f^*)^* = f$ .

Теперь, можно сформулировать теорему двойственности.

**Теорема 5.1** (Вариационное (двойственное) представление f-дивергенции). *Для любой f-дивергенции, имеетя:*

$$D_f(P||Q) = \sup_{g: \mathcal{X} \rightarrow \mathbb{R}} \mathbb{E}_P[g(X)] - \mathbb{E}_Q[f^*(g(X))], \quad (11)$$

где  $f^*$  – выпуклая сопряженная функция функции  $f$ , а супремум берется по всем функциям  $g$ , у которых оба математических ожидания конечны.

*Доказательство.*

$$D_f(P||Q) = \int_{\mathcal{X}} f\left(\frac{dP}{dQ}(x)\right) dQ(x) = \int_{\mathcal{X}} \sup_{y \in \text{dom}(f^*)} \left(y \frac{dP}{dQ}(x) - f^*(y)\right) dQ(x)$$

Теперь вместо того, чтобы брать супремум по  $y$ , мы можем заменить  $y$  супремумом по некоторой функции  $g: \mathcal{X} \rightarrow \mathbb{R}$ , поскольку  $y$  обычно зависит от  $x$ . Затем применяя неравенство Йенсена:

$$\begin{aligned} D_f(P||Q) &\geq \sup_{g: \mathcal{X} \rightarrow \mathbb{R}} \left( \int_{\mathcal{X}} g(x) dP(x) - \int_{\mathcal{X}} f^*(g(x)) dQ(x) \right) = \\ &= \sup_{g: \mathcal{X} \rightarrow \mathbb{R}} (\mathbb{E}_{x \sim P}[g(x)] - \mathbb{E}_{x \sim Q}[f^*(g(x))]) \end{aligned}$$

Вышеуказанная нижняя граница является жесткой и достигается при  $g(x) = f'\left(\frac{dP}{dQ}(x)\right)$  □

Дивергенция	$g_f$	$\text{dom}_{f^*}$	$f^*(x)$
Кульбака–Лейблера (КЛ)	$x$	$\mathbb{R}$	$\exp(x - 1)$
Обратная КЛ	$-\exp(-x)$	$\mathbb{R}_-$	$-1 - \log(-x)$
$\chi^2$ Пирсона	$x$	$\mathbb{R}$	$\frac{1}{4}x^2 + x$
Хеллингера	$1 - \exp(-x)$	$x < 1$	$\frac{x}{1-x}$
Йенсена–Шеннона	$\log(2) - \log(1 + \exp(-x))$	$x < \log(2)$	$-\log(2 - \exp(x))$
GAN	$-\log(1 + \exp(-x))$	$\mathbb{R}_-$	$-\log(1 - \exp(x))$

Таблица 1: Рекомендуемые функции активации последнего слоя и сопряженные функции для различных f-дивергенций.

Таким образом авторы [11] формулируют следующую целевую функцию:

$$F(G, D) = \mathbb{E}_{x \sim p_{data}} [g_f(D(x))] + \mathbb{E}_{z \sim p(z)} [-f^*(g_f(D(G(z))))], \quad (12)$$

где  $G$  и  $D$  – генератор и дискриминатор, аналогично GAN, а  $g_f$  – функция активации, которая ограничивает выход дискриминатора на область определения функции  $f^*$ .

Так например, выбирая  $f^*(x) = -\log(1 - \exp(x))$  и  $g(x) = \log \frac{p(x)}{p(x)+q(x)}$  выводится целевая функция состязательных сетей 10. Примеры остальных дивергенций приведены в таблице 1.

Авторы упростили процедуру оптимизации седловой точки, которая изначально была введена в GAN и вывели целевые функции для всех f-дивергенций, самое важное, в с число, которых входит и дивергенция Кульбака–Лейблера, применяющаяся в задаче мостов Шрёдингера.



## 5.2 Состязательные мосты Шрёдингера

Вдохновившись идеей представления дивергенций в вариационном виде, в данной работе предлагается формулировка минимизационную задачу мостов Шрёдингера аналогичным способом. Для этого рассмотрим статическую постановку задачи мостов Шрёдингера 3.

Чтобы представить совместные распределения в алгоритме IPF (алгоритм 2) с помощью двойственного представления f-дивергенций (задача 11), вариационное представление рассматривается с учетом измеримого пространства  $(\mathcal{X} \times \mathcal{Y}, \Sigma)$  вместо  $(\mathcal{X}, \Sigma)$ . Таким образом, дивергенции в шагах алгоритма IPF (алгоритм 2) будут иметь следующий вид:

$$\begin{aligned} D_{KL}(p(x,y)||q_{i-1}^*(x,y)) &= \max_{g:\mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}} \mathbb{E}_{p(x,y)}[g(x,y)] - \mathbb{E}_{q_{i-1}^*(x,y)}[e^{(g(x,y)-1)}], \\ D_{KL}(q(x,y)||p_i^*(x,y)) &= \max_{v:\mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}} \mathbb{E}_{p_i^*(x,y)}[v(x,y)] - \mathbb{E}_{q_i^*(x,y)}[e^{(v(x,y)-1)}], \end{aligned} \quad (13)$$

обратный и прямой шаги, соответственно.

Аналогично f-GAN (уравнение 12) параметризуем вариационные функции  $g$  и  $v$  с помощью нейронных сетей  $D$  и  $T$ , соответственно, и получаем следующие минимизационные задачи:

$$\begin{aligned} \min_{p(x,y) \in \mathcal{D}(\cdot, \pi_T)} \max_D \mathbb{E}_{p(x,y)}[D(x,y)] - \mathbb{E}_{q_{i-1}^*(x,y)}[e^{(D(x,y)-1)}], \\ \min_{q(x,y) \in \mathcal{D}(\pi_0, \cdot)} \max_T \mathbb{E}_{q(x,y)}[T(x,y)] - \mathbb{E}_{p_i^*(x,y)}[e^{(T(x,y)-1)}], \end{aligned} \quad (14)$$

Благодаря вариационному представлению также возникает возможность избавиться от ограничений в минимизационной задаче, а именно следующих:  $p(x,y) \in \mathcal{D}(\cdot, \pi_T)$  и  $q(x,y) \in \mathcal{D}(\pi_0, \cdot)$ . Для этого представим совместные распределения  $p(x,y)$  и  $q(x,y)$  через условное и маргинальное распределения. Таким образом, целевая функция примет следующий вид:

$$\begin{aligned} \min_{p(x|y)} \max_D \mathbb{E}_{\pi_T(y)} \mathbb{E}_{p(x|y)}[D(x,y)] - \mathbb{E}_{\pi_0(x)} \mathbb{E}_{q_{i-1}^*(y|x)}[e^{(D(x,y)-1)}], \\ \min_{q(y|x)} \max_T \mathbb{E}_{\pi_0(x)} \mathbb{E}_{q(y|x)}[T(x,y)] - \mathbb{E}_{\pi_T(y)} \mathbb{E}_{p_i^*(x|y)}[e^{(T(x,y)-1)}] \end{aligned} \quad (15)$$

Для того, чтобы параметризовать условное распределение предлагается использовать аналогичный подход предложенный в статье CycleGAN [2]. Основная суть которого, заключается в том, что процесс генерации идет из объекта, на который мы обуславливаемся, а не из случайного распределения. Применяя эту идею, получим следующие целевые функции:

$$\begin{aligned} B(D, G) &= \mathbb{E}_{\pi_T(y)}[D(G(y), y)] - \mathbb{E}_{\pi_0(x)}[e^{(D(x, K(x))-1)}], \\ F(T, K) &= \mathbb{E}_{\pi_0(x)}[T(x, K(x))] - \mathbb{E}_{\pi_T(y)}[e^{(T(G(y), y)-1)}] \end{aligned} \quad (16)$$

Где  $G$  и  $K$  – генераторы, параметризованные нейронными сетями.

Для того, чтобы полноценно реализовать алгоритм IPF с помощью состязательного обучения необходимо проинициализировать  $q^0(x|y)$  с помощью условного распределения  $p^{\mathbb{W}^\gamma}(y|x) = \mathcal{N}(y|x, \gamma \mathbb{I})$ . Для этого на первом шаге обратного прохода будем сэмплировать из заданного априорного распределения, то есть:

$$\mathbb{E}_{\pi_T(y)}[D(G(y), y)] - \mathbb{E}_{\pi_0(x)} \mathbb{E}_{\mathcal{N}(y|x, \gamma \mathbb{I})}[e^{(D(x,y)-1)}]$$

---

**Algorithm 4:** Состязательный IPF

---

**Дано:**  $\pi_0(x)$ ,  $\pi_T(y)$ ,  $p(y|x)$

**Найти:**  $G^i, K^i$

```

1 Инициализируем  $i = 0$ ;
2 while не сойдется do
3    $G^i = \arg \min_G \max_D B(D, G)$ ;
4    $K^i = \arg \min_K \max_T F(T, K)$ ;
5    $i := i + 1$ ;
```

---

В итоге, получим новый состязательный алгоритм IPF (алгоритм 4).

Предложенный метод удовлетворяет всем поставленным качествам. Во-первых, отображение происходит за один шаг, так как для того чтобы отобразить данные из одного распределения в другое необходимо использовать один раз  $G$  или  $K$ . Во-вторых, в силу того, что метод наследует подход состязательного обучения, результирующая модель хорошо справляется с данными большой размерностью.

## 6 Реализация метода

Если в рамках работы писался какой-то код, здесь должно быть его описание: выбранный язык и библиотеки и мотивы выбора, архитектура, схема функционирования, теоретическая сложность алгоритма, характеристики функционирования (скорость/память).

Перед тем как сформулировать постановку экспериментов и продемонстрировать результаты, аргументируется выбор Python в качестве языка программирования, а PyTorch в качестве фреймворка машинного обучения, для проведения экспериментов.

### 6.1 Выбор языка программирования

Выбор Python в качестве языка программирования для глубокого обучения Глубокое обучение стало одной из ключевых областей в искусственном интеллекте, и выбор языка программирования играет решающую роль в успехе проектов. Рассмотрим основные языки программирования, используемые в этой области, и обоснуем, почему Python является наиболее предпочтительным.

#### 6.1.1 Python

Python является наиболее популярным языком для глубокого обучения благодаря своей простоте и широкому спектру библиотек, таких как TensorFlow, Keras и PyTorch. Простота синтаксиса Python позволяет сосредоточиться на алгоритмах и моделях, а не на технических деталях программирования. Кроме того, Python поддерживает объектно-ориентированное, процедурное и функциональное программирование, что делает его универсальным инструментом для различных задач глубокого обучения.

#### 6.1.2 C++

C++ славится своей скоростью и эффективностью, что делает его хорошим выбором для систем, требующих высокой производительности, таких как роботы и автономные транспортные средства. Однако сложность синтаксиса и время, необходимое на разработку и отладку программ, делают C++ менее привлекательным для исследователей и разработчиков, сосредоточенных на быстром прототипировании и экспериментировании.

#### 6.1.3 R

R был разработан специально для статистического анализа и визуализации данных, что делает его полезным инструментом для анализа данных и статистического моделирования. Однако для глубокого обучения R уступает Python в плане производительности и наличия специализированных библиотек. Кроме того, синтаксис R считается более сложным для изучения, чем у Python.

#### 6.1.4 MATLAB

MATLAB предоставляет мощные инструменты для работы с матрицами и числовыми вычислениями, что важно для некоторых аспектов глубокого обучения. Однако его использование ограничено высокими затратами на лицензии и меньшим сообществом пользователей. В то время как MATLAB обладает высокопроизводительными встроенными функциями, Python с его бесплатными библиотеками часто оказывается более экономически эффективным и гибким решением.

### 6.1.5 Java

Java известен своей масштабируемостью и надежностью, что делает его подходящим для построения крупномасштабной инфраструктуры ИИ. Однако более сложный синтаксис и необходимость в большем объеме кода по сравнению с Python затрудняют быстрое прототипирование моделей глубокого обучения. Кроме того, экосистема библиотек для глубокого обучения в Java не столь развита, как у Python.

### 6.1.6 Сравнение и выбор

Несмотря на преимущества каждого из рассмотренных языков программирования, Python выделяется своей простотой, универсальностью и богатой экосистемой библиотек для глубокого обучения. Эти факторы делают его наиболее подходящим выбором как для начинающих, так и для опытных разработчиков в области глубокого обучения. Python позволяет эффективно разрабатывать, тестировать и внедрять модели глубокого обучения, что подтверждается его широким использованием в академической среде и промышленности.

## 6.2 Выбор фреймворка глубокого обучения

Для проведения научного исследования в глубоком обучении требуются мощные и гибкие инструменты для построения и обучения нейронных сетей. Помимо этого, такой фреймворк должен обеспечивать простоту реализации задач для того, чтобы процесс проведения экспериментов не занимал долгое время. Для решения этого вопроса рассмотрим наиболее популярные библиотеки.

### 6.2.1 TensorFlow

TensorFlow, разработанный Google, является одним из наиболее популярных фреймворков для глубокого обучения. Он поддерживает как символическое, так и императивное программирование, что позволяет пользователям выбирать между статическим и динамическим построением графов вычислений. TensorFlow широко используется в промышленности благодаря его высокой производительности и возможности развертывания на различных платформах, включая мобильные устройства и серверы.

Основные преимущества TensorFlow:

- поддержка распределенных вычислений;
- интеграция с другими инструментами Google, такими как TensorBoard для визуализации и анализа проведенных экспериментов;
- широкая библиотека предварительно обученных моделей.

Однако, TensorFlow обладает значительной сложностью, особенно для начинающих исследователей, и требует большего объема кода для простых операций.

### 6.2.2 Keras

Keras, первоначально разработанный как высокоуровневая библиотека поверх Theano и TensorFlow, стремится упростить разработку моделей глубокого обучения. Он предоставляет удобный и интуитивно понятный интерфейс для создания нейронных сетей.

Основные преимущества Keras:

- простота использования и быстрота прототипирования, что полезно для проведения научных исследований;
- поддержка нескольких бэкендов, включая TensorFlow и Theano.

Тем не менее, ограниченные возможности по сравнению с низкоуровневыми фреймворками могут быть недостатком для сложных и специфических задач.

### 6.2.3 PyTorch

PyTorch, разработанный Facebook AI Research, представляет собой фреймворк для глубокого обучения, который стал популярным благодаря своей простоте и гибкости. PyTorch поддерживает динамическое построение графов вычислений, что делает его идеальным для научных исследований и экспериментов.

Основные преимущества PyTorch:

- интуитивно понятный и лаконичный синтаксис, напоминающий стандартный Python-код;
- поддержка динамических вычислений, что упрощает отладку и экспериментирование с моделями;
- широкая поддержка сообществом и наличие множества ресурсов для обучения и примеров.

### 6.2.4 JAX

JAX, разработанный Google Research, представляет собой мощный инструмент для численных вычислений и машинного обучения, который сочетает в себе возможности библиотек Autograd и XLA. В данном обзоре рассмотрим особенности и преимущества JAX, а также его применение и отличия от PyTorch.

JAX предлагает API, схожий с NumPy, но с возможностью выполнения вычислений на GPU и TPU, что значительно ускоряет процессы. Основные функции JAX включают:

Преимущества JAX:

- **производительность:** Благодаря компилятору XLA, JAX обеспечивает высокую производительность и оптимизацию вычислений на GPU и TPU;
- **гибкость:** Позволяет исследователям легко экспериментировать с новыми архитектурами моделей и алгоритмами.

### 6.2.5 Сравнение и выбор

Для научных экспериментов в области глубокого обучения ключевыми факторами являются простота использования, гибкость, поддержка сообществом и возможности для прототипирования. PyTorch выделяется среди других фреймворков благодаря:

- интуитивно понятному и читаемому коду, что снижает порог вхождения для начинающих исследователей;
- динамическому построению графов вычислений, что упрощает отладку и экспериментирование;
- активному сообществу и большому количеству образовательных ресурсов, что способствует быстрому освоению и применению фреймворка.

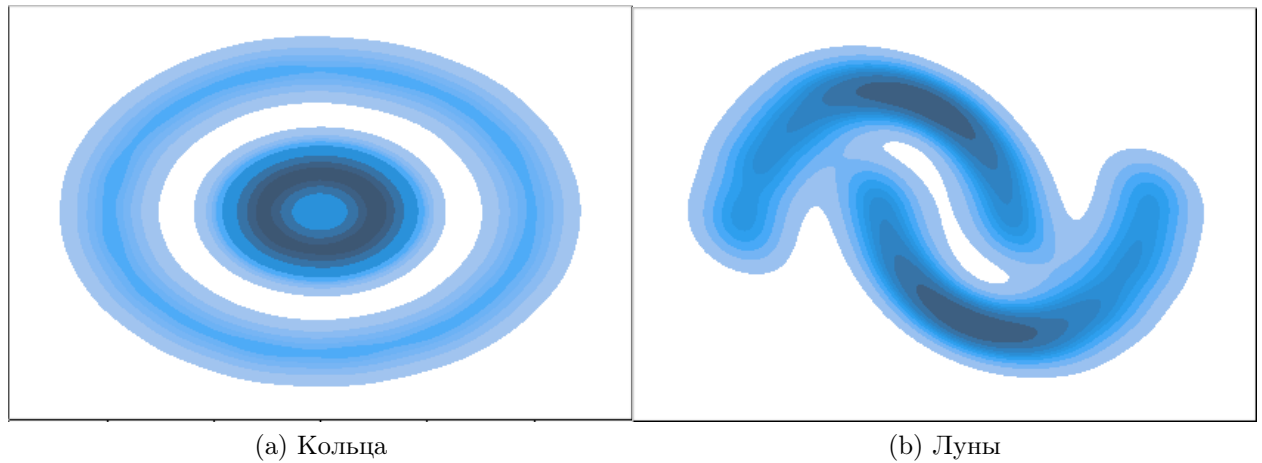


Рис. 4: Набор 2D данных

Исходя из рассмотренных факторов, PyTorch является наиболее подходящим инструментом для научных экспериментов в сфере глубокого обучения. Его простота, гибкость и поддержка сообществом делают его идеальным выбором для исследователей, стремящихся к быстрому прототипированию и адаптации моделей. Несмотря на сильные стороны TensorFlow и других фреймворков, PyTorch обеспечивает оптимальный баланс между функциональностью и удобством использования, что критически важно в научной работе.

### 6.3 Эксперименты

Для эмпирического подтверждения факта, что данный метод решения задачи мостов Шрёдингера удовлетворяет поставленным требованиям, рассматриваются два эксперимента:

- Эксперимент на 2D данных;
- Эксперимент на EMNIST.

Рассмотрим данные подробнее.

#### 6.3.1 2D данные

Перед тем как проверять на данных большой размерности, проводится эксперимент на 2D данных, который предназначен, во-первых, для санитарного теста работоспособности метода, а, во-вторых, для проверки способности отображения при различных параметрах  $\gamma$ .

В качестве данных был использованы наборы из библиотеки `sci-kit learn` кольца и луны (рисунок 4),  $X$  и  $Y$ , соответственно. Данный набор представляет собой набор пар  $(x, y)$  координат точек, которые формируют кольца или луны.

Для формирования набора лун были использованы следующие параметры:

- количество точек обучающей выборки: 5000;
- параметр стандартного отклонения шума, добавленного к данным: 0.05

Для колец выбраны следующие параметры:

- количество точек обучающей выборки: 5000;

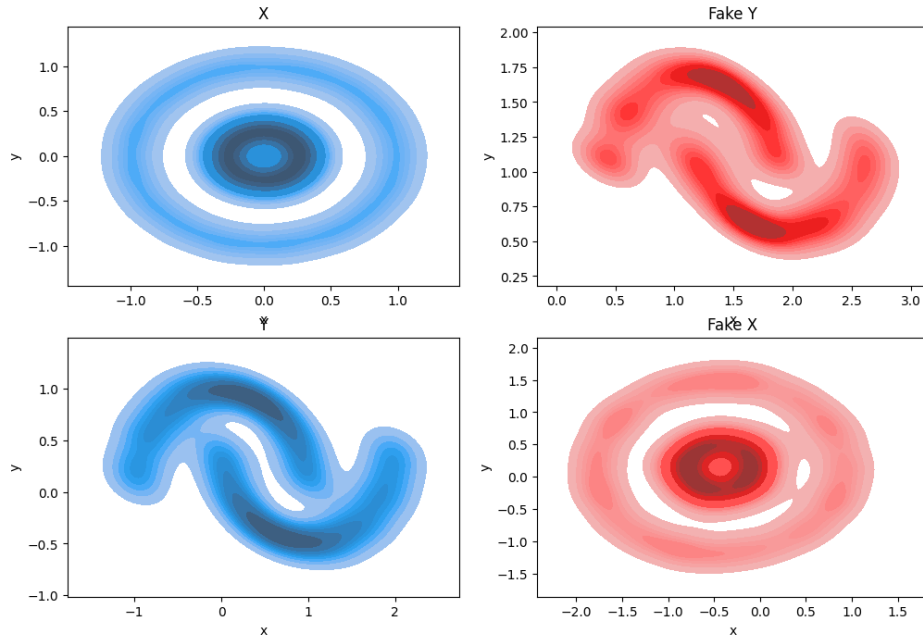


Рис. 5: Результаты 2D эксперимента

- параметр стандартного отклонения шума, добавленного к данным: 0.03
- расстояние от между кольцами: 0.3

Генератор параметризуется с помощью 3-х полно-связных слоев. Каждый слой имеет 256 нейронов. ReLU был выбран в качестве желаемой функции активации. На рисунке ?? продемонстрирована архитектура генератора.

Для параметризации дискриминатора используется почти та же архитектура. 3 полно-связных слоя со скрытым слоем размером 256, однако вместо ReLU функцией активации является LeakyReLU с коэффициентом угла 0.2. На вход подается конкатенация элементов из  $X$  и  $Y$ , а на выходе скаляр. На рисунке ?? продемонстрирована архитектура дискриминатора.

Что касается процесса оптимизации, информация следующая: используется AdamW в качестве алгоритма оптимизации с коэффициентом L2 регуляризации 0,01 для генератора и 0,5 для дискриминатора; скорости обучения были установлены  $1e - 6$  для генераторов и дискриминаторов; размер батча был выбран 1024. Генераторы обучались на протяжении 400 эпох, а количество шагов состязательного обучения составило 60.

Главным гиперпараметром задачи мостов Шрёдингера является  $\gamma$ . В монографии [17] показано, что некоторые методы работают при больших  $\gamma = 1000$ , а некоторые — при низких  $\gamma = 1$ . Меньшее значение  $\gamma$  гарантирует наиболее оптимальное отображение, и, соответственно, менее разнообразный вывод. В экспериментах были протестированы следующие параметры  $\gamma$ : 1 и 0.01.

Результаты демонстрируют, что метод отлично справляется с малыми параметрами  $\gamma$ . Также можно заметить, что при различных  $\gamma$ , действительно, отображение становится менее разнообразным, подтверждая, тот факт, что данный метод находит оптимальное отображение.

### 6.3.2 EMNIST

Целью данного эксперимента является подтверждение того факта, что метод работает на данных большой размерностью. Для этого был выбран набор данных EMNIST,

который представляет собой набор рукописных символов, полученных из специальной базы данных NIST 19 и преобразованных в формат изображения 28x28 пикселей и структуру набора данных, которая напрямую соответствует набору данных MNIST.

В качестве набора  $X$  был выбран поднабор EMNIST, а именно 145,600 латинских рукописных букв с 26 классами. В качестве  $Y$  был выбран классический MNIST, в котором 70,000 цифр. Классы обоих наборов были сбалансированы. Результаты данного эксперимента продемонстрированы на изображении ???. Таким образом, можно заключить, что данный метод справляется с данными большой рамерностью.



## 7 Заключение

В ходе выполнения данной магистерской диссертации были достигнуты следующие результаты:

1. Проведен обзор и анализ различных подходов к решению задачи мостов Шрёдингера, включая Data Driven Schrödinger Bridge, Light Schrödinger Bridge, Diffusion Schrödinger Bridge, Iterative Proportional Maximum Likelihood (IPML) и Unpaired Neural Schrödinger Bridge.
2. Разработан метод состязательных мостов Шрёдингера, объединяющий идеи генеративных состязательных сетей (GAN) и задачи мостов Шрёдингера.
3. Проанализированы инструменты для реализации, включая язык программирования Python и фреймворк глубокого обучения PyTorch, что было обосновано их гибкостью и поддержкой сообщества.
4. Проведен эксперимент на 2D данных с использованием наборов данных кольца и луны из библиотеки sci-kit learn. Параметры набора данных были выбраны таким образом, чтобы проверить работоспособность метода и его способность к отображению при различных параметрах  $\gamma$ , а также проверить оптимальность отображения, что подтверждает пригодность предложенного метода для решения задачи мостов Шрёдингера.
5. Проведен эксперимент на данных большой размерности с использованием набора данных EMNIST, который подтверждает способность метода работать с большими и сложными наборами данных.

В итоге, можно заключить, что задача, поставленная в данной магистерской диссертации, была решена в полной мере. Разработанный метод состязательных мостов Шрёдингера доказал свою эффективность и применимость для задач трансляции доменов. Проведенные эксперименты подтвердили теоретические выводы и продемонстрировали высокую эффективность предложенного подхода. В будущем данный метод может быть расширен и адаптирован для решения более широкого круга задач в области машинного обучения и анализа данных.

## Список литературы

- [1] Generative Adversarial Networks / Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza et al. — 2014.
- [2] Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks / Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros. — 2020.
- [3] *Ho, Jonathan*. Denoising Diffusion Probabilistic Models / Jonathan Ho, Ajay Jain, Pieter Abbeel. — 2020.
- [4] *Schrödinger, E.* Über die Umkehrung der Naturgesetze / E. Schrödinger. — Sitzungsberichte der Preussischen Akademie der Wissenschaften. Physikalisch-mathematische Klasse. Verlag der Akademie der Wissenschaften in Kommission bei Walter De Gruyter u. Company, 1931. <https://books.google.ru/books?id=FTYYtwAACAAJ>.
- [5] *Léonard, Christian*. A survey of the Schrödinger problem and some of its connections with optimal transport / Christian Léonard. — 2013.
- [6] Diffusion Schrödinger Bridge with Applications to Score-Based Generative Modeling / Valentin De Bortoli, James Thornton, Jeremy Heng, Arnaud Doucet // *arXiv preprint arXiv:2106.01357*. — 2021.
- [7] Diffusion Schrödinger Bridge Matching / Yuyang Shi, Valentin De Bortoli, Andrew Campbell, Arnaud Doucet. — 2023.
- [8] Solving Schrödinger Bridges via Maximum Likelihood / Francisco Vargas, Pierre Thodoroff, Austen Lamacraft, Neil Lawrence // *Entropy*. — 2021. — . — Vol. 23, no. 9. — P. 1134. <http://dx.doi.org/10.3390/e23091134>.
- [9] Unpaired Image-to-Image Translation via Neural Schrödinger Bridge / Beomsu Kim, Gihyun Kwon, Kwanyoung Kim, Jong Chul Ye. — 2024.
- [10] *Korotin, Alexander*. Light Schrödinger Bridge. — 2024.
- [11] *Nowozin, Sebastian*. f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization / Sebastian Nowozin, Botond Cseke, Ryota Tomioka. — 2016.
- [12] *Санов, И. Н.* О вероятности больших отклонений случайных величин / И. Н. Санов. — 1957. — Vol. 42(84).
- [13] *Kullback, S.* Probability Densities with Given Marginals / S. Kullback // *The Annals of Mathematical Statistics*. — 1968. — Vol. 39, no. 4. — Pp. 1236 – 1243. <https://doi.org/10.1214/aoms/1177698249>.
- [14] *Rüschendorf, Ludger*. Convergence of the iterative proportional fitting procedure / Ludger Rüschendorf // *Annals of Statistics*. — 1995. — Vol. 23. — Pp. 1160–1174. <https://api.semanticscholar.org/CorpusID:122665767>.
- [15] *Pavon, Michele*. The Data-Driven Schrödinger Bridge / Michele Pavon, Giulio Trigila, Esteban G. Tabak // *Communications on Pure and Applied Mathematics*. — 2021. — Vol. 74, no. 7. — Pp. 1545–1573. <https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.21975>.

- [16] Deep Generative Learning via Schrödinger Bridge / Gefei Wang, Yuling Jiao, Qian Xu et al. — 2021. — 18–24 Jul. — Vol. 139. — Pp. 10794–10804. <https://proceedings.mlr.press/v139/wang211.html>.
- [17] *Vargas, Francisco*. Machine-learning approaches for the empirical Schrödinger bridge problem / Francisco Vargas. — No. UCAM-CL-TR-958. 2021. — . <https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-958.pdf>.
- [18] *Mott-Smith, H.* The theory of collectors in gaseous discharges / H. Mott-Smith, I. Langmuir // *Phys. Rev.* — 1926. — Vol. 28.
- [19] *Морз, Р.* Бесстолкновительный PIC-метод / Р. Морз // Вычислительные методы в физике плазмы / Ed. by Б. Олдера, С. Фернбаха, М. Ротенберга. — М.: Мир, 1974.
- [20] *Киселёв, А. А.* Численное моделирование захвата ионов бесстолкновительной плазмы электрическим полем поглощающей сферы / А. А. Киселёв, Долгонос М. С., Красовский В. Л. // Девятая ежегодная конференция «Физика плазмы в Солнечной системе». — 2014.
- [21] Schrödinger Bridge Samplers / Espen Bernton, Jeremy Heng, Arnaud Doucet, Pierre E. Jacob. — 2019.
- [22] *Fortet, Robert*. Résolution d'un système d'équations de M. Schrödinger / Robert Fortet // *Journal de Mathématiques Pures et Appliquées*. — 1940. — Vol. 9e série, 19, no. 1-4. [http://www.numdam.org/item/JMPA\\_1940\\_9\\_19\\_1-4\\_83\\_0/](http://www.numdam.org/item/JMPA_1940_9_19_1-4_83_0/).
- [23] *Essid, Montacer*. Traversing the Schroedinger Bridge strait: Robert Fortet's marvelous proof redux / Montacer Essid, Michele Pavon. — 2018.
- [24] Augmented Bridge Matching / Valentin De Bortoli, Guan-Horng Liu, Tianrong Chen et al. — 2023.
- [25] Flow Matching for Generative Modeling / Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu et al. — 2023.