# The Schrцdinger Bridge Problem in GAN

A Preprint

Gregory S. Ksenofontov
Department of Intelligent Systems
Moscow Institute of Physics and Technology (National Research University)
1 Kerchenskaya st., Moscow, 117303, Russian Federation
ksenofontov.gs@phystech.edu

## Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Keywords Schrцdinger Bridge Problem · GAN · Unsupervised Domain Adaptation

## 1 Introduction

Let it is given two sets $X$ and $Y$, which contain, for example, pictures of two domains: for real and fake views from cars (Figure 1), respectively. We now want translate pictures from one to another domain. This task is called unsupervised domain adaptation and is solved using different approaches: Show and describe approaches. One another way to solve unsupervised domain adaptation problem is using Schrцdinger Bridges.

Overall, the Schrцdinger bridge provides us with a theoretically-grounded mechanism for mapping between two distribution, using stochastic processes. Furthermore, it gives the probability of this stochastic evolution, thus allowing us to compare two datasets/distributions which can be useful for hypothesis testing and semantic similarity.

## 2 What is Schrцdinger Bridge Problem?

The Schrцdinger Bridge Problem arises from a question: how to bound a prior stochastic process? The answer is given by the large deviations theory and particularly by Sanov's theorem.

Firstly, the problem requires understanding of path measure. Path measure is probability measure associated to the stochastic process. Let it is given i.i.d. paths (samples) from path measure $\mathbb{W}$ associated to a prior Wiener process, i.e. $\{x_i(t)\}_{i=1}^{N} \sim \mathbb{W}$, where $x_i(t) \in \mathcal{C}([0,1], \mathbb{R}^d), \forall i = \overline{1, N}$. The empirical path measure of the obtained paths is given by:

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

Now we want to bound associated process with $\pi_0(x)$ and $\pi_1(y)$, because otherwise having desired densities stochastic process can not translate from $\pi_0(x)$ to $\pi_1(y)$, i.e. given the transitional density $p^{\mathbb{W}}(y|x)$ of such

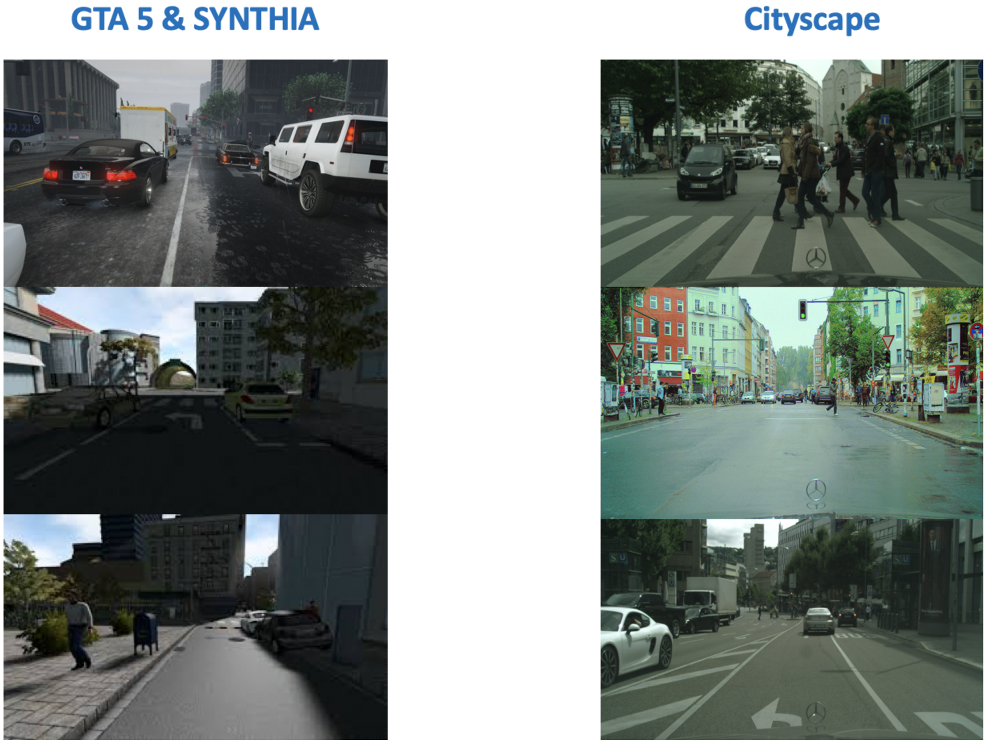

Figure 1: Two domains: Synthetic and Real city views

process we get:

$$\pi_1(y) \neq \int p^{\mathbb{W}}(y|x)\pi_0(x)dx,$$

So, we want the path measure $\hat{\mathbb{W}}$ be in a set of all measures $\mathcal{D}(\pi_0, \pi_1)$ that are bounded between $\pi_0(x)$ and $\pi_1(y)$, i.e.:

$$P\left(\hat{\mathbb{W}} \in \mathcal{D}(\pi_0, \pi_1)\right) = 1 \tag{1}$$

The Sanov's theorem gives asymptotic expression for such probability:

$$P\left(\hat{\mathbb{W}} \in \mathcal{D}(\pi_0, \pi_1)\right) \xrightarrow{N\to\infty} \exp\left(-N \inf_{\mathbb{Q}\in\mathcal{D}(\pi_0,\pi_1)} KL(\mathbb{Q}||\mathbb{W})\right) \tag{2}$$

Thus, the power of exponent must be zero or particularly KL-divergence between $\mathbb{Q}$ and prior $\mathbb{W}$ path measures must be zero to satisfy Equation 1. Now we can formulate two approaches to problem setting of Schrцdinger Bridge, dynamic and static.

### 2.1 Dynamic Schrцdinger Bridge Problem

The dynamic problem statement comes from Sanov's theorem 2 and it finds a path measure $\mathbb{Q}$ that is bounded by given distributions $\pi_0(x)$ and $\pi_1(y)$ and is closest in information-theoretic sense to prior path measure $\mathbb{W}^\gamma$ of Wiener process with drift coefficient $\sqrt{\gamma}$:

$$\hat{\mathbb{Q}} = \arg \min_{\mathbb{Q}\in\mathcal{D}(\pi_0,\pi_1)} D_{KL}(\mathbb{Q}||\mathbb{W}^\gamma) \tag{3}$$

### 2.2 Static Schrцdinger Bridge Problem

Let's decompose KL-divergence in 3. Firstly, lets reexpress RN-derivative conditioning on x(0) = $x$ and x(1) = $y$:

$$\frac{d\mathbb{Q}}{d\mathbb{W}^\gamma} = \frac{q(x,y)}{p^{\mathbb{W}^\gamma}(x,y)} \frac{d\mathbb{Q}_{(0,1)}}{d\mathbb{W}^\gamma_{(0,1)}}(\cdot|x,y)$$

Passing this expression to KL-divergence in 3 we get:

$$KL(\mathbb{Q}||\mathbb{W}^\gamma) = \mathbb{E}_\mathbb{Q}\left[\log\frac{d\mathbb{Q}}{d\mathbb{W}^\gamma}\right] = \mathbb{E}_\mathbb{Q}\left[\log\frac{q(x,y)}{p^{\mathbb{W}^\gamma}(x,y)}\frac{d\mathbb{Q}_{(0,1)}}{d\mathbb{W}^\gamma_{(0,1)}}(\cdot|x,y)\right] =$$

$$= \int \log\frac{q(x,y)}{p^{\mathbb{W}^\gamma}(x,y)}d\mathbb{Q} + \int \log\frac{d\mathbb{Q}_{(0,1)}}{d\mathbb{W}^\gamma_{(0,1)}}(\cdot|x,y)d\mathbb{Q} =$$

$$= \int q(x,y)\log\frac{q(x,y)}{p^{\mathbb{W}^\gamma}(x,y)}dxdy + \int \log q(x,y)\frac{d\mathbb{Q}_{(0,1)}}{d\mathbb{W}^\gamma_{(0,1)}}(\cdot|x,y)d\mathbb{Q}_{(0,1)} =$$

$$= KL\left(q(x,y)||p^{\mathbb{W}^\gamma}(x,y)\right) + \mathbb{E}_{q(x,y)}\left[KL\left(\mathbb{Q}_{(0,1)}(\cdot|x,y)||\mathbb{W}^\gamma_{(0,1)}(\cdot|x,y)\right)\right]$$

We can see that the second $KL$ divergence doesn't consider the marginal distributions, so it doesn't affect optimisation constraint. Thus, zeroing it we get static statement:

$$\begin{cases} \hat{q}(x,y) = \arg\min_{q(x,y)} KL(q(x,y)||p^{\mathbb{W}^\gamma}(x,y)), \\ \pi_0(x) = \int q(x,y)dy, \\ \pi_1(y) = \int q(x,y)dx \end{cases}$$

where $q(x,y)$ is a joint distribution which is closest to the Brownian-motion prior subject to marginal constraints (between PDF at time 0 and 1)

### 2.3 Schrцdinger System

Applying Lagrangian on static SBP we get

$$L(q,\lambda,\mu) = KL(q(x,y)||p^{\mathbb{W}^\gamma}(x,y)) + \int \lambda(x)\left(\int q(x,y)dy - \pi_0(x)\right)dx + \int \mu(y)\left(\int q(x,y)dy - \pi_1(y)\right)dy$$

, $p^{\mathbb{W}^\gamma}(x,y) = p_0^{\mathbb{W}^\gamma}(x)p^{\mathbb{W}^\gamma}(y|x)$, $p_0^{\mathbb{W}^\gamma}(x)$ , $p^{\mathbb{W}^\gamma}(y|x) = \mathcal{N}(y|x,\gamma I_d)$ ( ). $\frac{\partial L(q,\lambda,\mu)}{\partial q(x,y)}$

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

, $\hat{\phi}_0(x) = \exp\left(\ln p^{\mathbb{W}^\gamma}(x,y) - \lambda(x) - 1\right)$ $\phi_1(y) = \exp\left(-\mu(y)\right)$

$$q^*(x,y) = \hat{\phi}_0(x)p^{\mathbb{W}^\gamma}(y|x)\phi_1(y)$$

$x$ $y$

$$\pi_0(x) = \hat{\phi}_0(x)\phi_0(x)$$

$$\pi_1(y) = \phi_1(y)\hat{\phi}_1(y)$$

$\phi_0(x) = \int \phi_1(y)p^{\mathbb{W}^\gamma}(y|x)dy$, $\hat{\phi}_1(y) = \int \hat{\phi}_0(x)p^{\mathbb{W}^\gamma}(y|x)dx$

### 2.4 Schrцdinger Half Bridges

The half bridge problem consider only one of the boundaries constraint, i.e. $\mathcal{D}(\pi_0(x),\cdot)$ or $\mathcal{D}(\cdot,\pi_1(y))$.

#### 2.4.1 Dynamic Half Bridges

More formally forward half bridge is given by:

$$\mathbb{Q}^* = \arg\min_{\mathbb{Q}\in\mathcal{D}(\pi_0(x),\cdot)} KL\left(\mathbb{Q}||\mathbb{W}^\gamma\right) \tag{4}$$

and backward half bridge is given by:

$$\mathbb{P}^* = \arg\min_{\mathbb{P}\in\mathcal{D}(\cdot,\pi_1(y))} KL\left(\mathbb{P}||\mathbb{W}^\gamma\right) \tag{5}$$

And they admits following solutions, respectively

$$\mathbb{Q}^*(A_0 \times A_{(0,1]}) = \int_{A_0\times A_{(0,1]}} \frac{d\pi_0}{p_0^{\mathbb{W}^\gamma}}(x)d\mathbb{W}^\gamma \tag{6}$$

$$\mathbb{P}^*(A_{[0,1)} \times A_1) = \int_{A_{[0,1)}\times A_1} \frac{d\pi_1}{p_1^{\mathbb{W}^\gamma}}(y)d\mathbb{W}^\gamma \tag{7}$$

### 2.4.2 Static Half Bridges

Static forward half bridge is given by:

$$\begin{aligned} q^*(x,y) = \arg \min_{q(x,y) \in \mathcal{D}(\pi_0(x), \cdot)} KL\left(q(x,y) || p^{\mathbb{W}^\gamma}(x,y)\right) \\ s.t. \pi_0(x) = \int q(x,y) dy \end{aligned} \tag{8}$$

and backward half bridge is given by:

$$\begin{aligned} p^*(x,y) = \arg \min_{p(x,y) \in \mathcal{D}(\cdot, \pi_1(y))} KL\left(p(x,y) || p^{\mathbb{W}^\gamma}(x,y)\right) \\ s.t. \pi_1(y) = \int q(x,y) dx \end{aligned} \tag{9}$$

And they admits following solutions, respectively

$$q(x,y)^* = p(x,y)^{\mathbb{W}^\gamma} \frac{\pi_0(x)}{p^{\mathbb{W}^\gamma}(x)} \tag{10}$$

$$p(x,y)^* = p(x,y)^{\mathbb{W}^\gamma} \frac{\pi_1(y)}{p^{\mathbb{W}^\gamma}(y)} \tag{11}$$

Half bridges are a significantly easier problem than full bridges. Not only do they admit closed-form solutions in some sense, they also allow removing constraints by incorporating them as an initial value problem

### 2.5 Iteration Proportional Fitting (IPF)

One of the oldest algorithms for solving Schrцdinger Bridge Problem is Fortet's algorithm (1940), which is provided in Algorithm 1.

```
Algorithm 1: Fortet's Algorithm
Input: $\pi_0(x)$, $\pi_1(y)$, $p(y|x)$
Output: $\hat{\phi}_0^{(i)}(x)$, $\phi_1^{(i)}(y)$
Initialize $\phi_0^{(0)}(x)$ s.t. $\phi_0^{(0)}(x) << \pi_0(x)$;
Initialize $i = 0$;
while not converged do
    $\hat{\phi}_0^{(i)}(x) := \frac{\pi_0(x)}{\phi_0^{(i)}(x)}$;
    $\hat{\phi}_1^{(i)}(y) := \int p(y|x) \hat{\phi}_0^{(i)}(x) dx$;
    $\phi_1^{(i)}(y) := \frac{\pi_1(y)}{\hat{\phi}_1^{(i)}(y)}$;
    $\hat{\phi}_1^{(i+1)}(x) := \int p(y|x) \phi_1^{(i)}(y) dy$;
    $i := i + 1$;
```

, ipfp

The next algorithm was build on static formulation. The IPF was originally constructed for discrete formulation. First, continuous formulation was given by Kullback [1] and convergence was prooved by Ruschendorf [2]. The algorithm shown in Algorithm 2

## 3 Related works

### 3.1 Generative-adversarial networks (GAN)

Goodfellow et al. [3] introduced new method of estimating generative models using adversarial training. Such method trains generative model $G$ (generative distribution $p_g$ is learned implicitly) competitively by

Algorithm 2: Kullback's Algorithm

```
Input: $\pi_0(x)$, $\pi_1(y)$, $p(y|x)$
Output: $q_i^*(x, y), p_i^*(x, y)$
Initialize $p_1^{\mathbb{W}^\gamma}(y)$ s.t. $p_1^{\mathbb{W}^\gamma}(y) << \pi_1(y)$;
Initialize $q_0^*(x, y) := p^{\mathbb{W}^\gamma}(x, y)$;
Initialize $i = 0$;
while not converged do
    $p_i^*(x, y) = \arg\min_{p(x,y)\in\mathcal{D}(\cdot,\pi_1(y))} D_{KL}(p(x, y)||q_{i-1}^*(x, y))$;
    $q_i^*(x, y) = \arg\min_{q(x,y)\in\mathcal{D}(\pi_0(x),\cdot)} D_{KL}(q(x, y)||p_i^*(x, y))$;
    $i := i + 1$;
```

introducing discriminator model $D$, that tries to determine whether is a sample real or generated:

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

Also, authors proved that selecting optimal discriminator $D^*$ is similar to minimization of the Jensen-Shannon divergence between $p_{data}$ and $p_g$, i.e.:

$$\begin{aligned}\min_G \max_D \mathbb{E}_{x\sim p_{data}}[\log D(x)] - \mathbb{E}_{x\sim p_g}[1 - \log D(x)] = \\ \min_G \mathbb{E}_{x\sim p_{data}}[\log D^*(x)] - \mathbb{E}_{x\sim p_g}[1 - \log D^*(x)] = \min_G D_{JS}(p_{data}||p_g) - 2\log 2,\end{aligned} \tag{12}$$

### 3.2 f-GAN

Nowozin et al. [4] justifiably extends the theory of GANs [3] to more general principle using variational (dual) representation of f-divergence:

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

Selecting, for example, choosing $f^*(t) = -\log(1 - \exp(t))$ and $g(x) = \log \frac{p(x)}{p(x)+q(x)}$ yields 12.

Authors derived the GAN training objectives for all f-divergences and simplified the saddle-point optimization procedure that was originally introduced in GAN.

### 3.3 Diffusion Schrodinger Bridge Matching (DSBM)

Shi et al. [5] proposed new method of solving SBP in dynamic way using Iterative Markovian Fitting

### 3.4 Diffusion Schrodinger Bridge with Applications to Score-Based Generative Modeling

De Bortoli et al. [6] proposed using SBP in dynamic way to build generative model

## 4 Variational representation of KL-divergence

To calculate KL divergence between two joint distribution we can use variational (dual) representation of f-divergence. However, firstly we need to introduce conjugate function.

Definition 4.1 (Conjugate function). Let $f : dom(f) \to \mathbb{R}$ be a convex function, where $dom(f) \subseteq \mathbb{R}$ is an interval. Then the convex conjugate of $f$ is function $f : dom(f^*) \to \mathbb{R}$ defined as:

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

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

Now lets mention most important properties of convex conjugate

Property 4.1. The convex conjugate $f^*$ of $f$ satisfies next properties:

1. $f^*$ is continuous in its domain;
2. $f^*$ is convex;
3. $(f^*)^* = f$ – biconjucation

Now when we can show the duality theorem.

Theorem 4.1 (f-divergence variational (dual) representation). For any f-divergence, we have:

$$D_f(P||Q) = \sup_{g:\mathcal{X}\to\mathbb{R}} \mathbb{E}_P[g(X)] - \mathbb{E}_Q[f^*(g(X))], \tag{13}$$

where $f^*$ is the convex conjugate of $f$ and the supremum is taken over all functions $g$ for which both expectations are finite.

Proof.

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

Now instead of taking supremum over $y$, we can replace $y$ with supremum over some function $g : \mathcal{X} \to \mathbb{R}$, because $y$ is generally depends on $x$. Then applying Jensen inequality:

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

Above lower bound is tight and achieved at $g(x) = f'\left(\frac{dP}{dQ}(x)\right)$ □

# 5 Methods

## 5.1 Divergence disintegration

It can be easily shown that each step with KL-divergence in 2 could be replaced with following expression:

$$\begin{aligned} p_i^*(x,y) &= \arg\min_{p(x,y)\in\mathcal{D}(\cdot,\pi_1(y))} D_{KL}(p(x,y)||q_{i-1}^*(x,y)) = \\ &= \arg\min_{p(x,y)\in\mathcal{D}} \int_{\mathcal{X}\times\mathcal{Y}} p(x,y)\log\frac{p(y|x)p(x)}{q^{(i-1)}(y|x)\pi_0(x)}dxdy = \\ &= \arg\min_{p(x,y)\in\mathcal{D}} \int_{\mathcal{X}\times\mathcal{Y}} p(x,y)\log\frac{p(x)}{\pi_0(x)}dxdy + \int_{\mathcal{X}\times\mathcal{Y}} p(x,y)\log\frac{p(y|x)}{q^{(i-1)}(y|x)}dxdy = \\ &= \arg\min_{p(x,y)\in\mathcal{D}} D_{KL}(p(x)||\pi_0(x)) + \int_{\mathcal{X}\times\mathcal{Y}} p(x,y)\log\frac{p(y|x)}{q^{(i-1)}(y|x)}dxdy \end{aligned}$$

Now, the first term is KL-divergence between generative distribution $p(x)$ and true data distribution $\pi_0(x)$, so following Nowozin et al. [4] we can easily replace it with dual representation and treat to it as adversarial term of equation. Also, it should be mentioned that the KL-divergence is reversed.

The second term should be reviewed as well. Having two, backward and forward, steps it would be inconvenient to have four different conditional distributions appears in both steps, i.e. $p(x|y)$, $q(x|y)$ and $p(y|x)$, $q(y|x)$ (which should be trained). So let's revert one of the conditional using Bayes' theorem:

$$\begin{aligned} p_i^*(x,y) = \arg\min_{p(x,y)\in\mathcal{D}} \max_D \mathbb{E}_{x\sim p(x)}[-\exp(-D(x))] - \mathbb{E}_{x\sim\pi_0(x)}[D(x)-1] + \int_{\mathcal{X}\times\mathcal{Y}} p(x,y)\log\frac{p(y|x)}{q^{(i-1)}(x|y)}dxdy + \\ + \int_{\mathcal{X}\times\mathcal{Y}} p(x,y)\log q^*(y)dxdy - \int_{\mathcal{X}\times\mathcal{Y}} p(x,y)\log\pi_0(x)dxdy \end{aligned}$$

The third term could be written as expectation of KL-divergence between two conditionals by rewriting joint distribution $p(x,y) = \pi_1(y)p(x|y)$.

Marginalizing the fourth term $\int_{\mathcal{X}\times\mathcal{Y}} p(x,y)\log q^*(y)dxdy = \int_{\mathcal{Y}} \pi_1(y)\log q^*(y)dy$ it can bee seen that fourth term is constant and could be dropped in minimization problem.

Finally, joint distribution in the last term could be marginalize as well:

$$p_i^*(x,y) = \arg\min_{p(x,y)\in\mathcal{D}}\max_D \mathbb{E}_{x\sim p(x)}[-\exp(-D(x))] - \mathbb{E}_{x\sim\pi_0(x)}[D(x)-1] + \int_{\mathcal{Y}}\pi_1(y)\int_{\mathcal{X}} p(x|y)\log\frac{p(x|y)}{q^{(i-1)}(x|y)}dxdy- \\ -\int_{\mathcal{X}} p(x)\log p(x)dx = \\ = \arg\min_{p(x,y)\in\mathcal{D}}\max_D \mathbb{E}_{x\sim p(x)}[-\exp(-D(x))] - \mathbb{E}_{x\sim\pi_0(x)}[D(x)-1] + \mathbb{E}_{x\sim p(x)}[D_{KL}(p(y|x))||q^{(i-1)}(x|y))] - H(p(x),\pi_0(x))$$

Similarly we prove it for the forward step. So, this is the algorithm for

This approach has two problems. The first is having cross entropy term in the minimization task. The problem is a necessity to calculate a probability of real data distribution. The second problem absurd amount of model need to be trained, however only two are needed in inference.

### 5.2 Dual representation in IPFP

The second method comes from idea of representing the KL-divergence in a dual form. However, instead of representing KL-divergence between marginals, we doing it for joint distributions. To consider joint distributions in the 2 we take 13 considering measurable space $(\mathcal{X}\times\mathcal{Y},\Sigma)$ instead of $(\mathcal{X},\Sigma)$, so the resulting variational representation for KL-divergence would take following form:

$$\min_{p(x,y)\in\mathcal{D}} D_{KL}(p(x,y)||q_{i-1}^*(x,y)) = \min_{p(x,y)\in\mathcal{D}}\max_{g:\mathcal{X}\times\mathcal{Y}\to\mathbb{R}} \mathbb{E}_{p(x,y)}[g(x,y)] - \mathbb{E}_{q_{i-1}^*(x,y)}\left[e^{(g(x,y)-1)}\right] \tag{14}$$

Now to constraint $p(x,y)$ and $q(x,y)$ on both marginals we use ancestral sampling in both expectations. Thus, we get:

$$\min_{p(x,y)\in\mathcal{D}} D_{KL}(p(x,y)||q_{i-1}^*(x,y)) = \min_{p(x,y)\in\mathcal{D}}\max_{g:\mathcal{X}\times\mathcal{Y}\to\mathbb{R}} \mathbb{E}_{\pi_1(y)}\mathbb{E}_{p(x|y)}[g(x,y)] - \mathbb{E}_{\pi_0(x)}\mathbb{E}_{q_{i-1}^*(y|x)}\left[e^{(g(x,y)-1)}\right] \tag{15}$$

Following GAN approach we can implicitly learn $p(x|y)$ and $q(y|x)$ by using generative model.

Also, it should be mentioned and each, backward and forward, steps the algorithm trains two different discriminators $g$ and $g'$.

### 5.3 Dual representation in Static SB

Following previous idea of representing KL divergence in dual form we can inherit such idea to Static SB:

$$\min_{q(x,y)\in\mathcal{D}} D_{KL}\left(q(x,y)||p^{\mathbb{W}^\gamma}(x,y)\right) = \min_{q(x,y)\in\mathcal{D}}\max_{g:\mathcal{X}\times\mathcal{Y}\to\mathbb{R}} \mathbb{E}_{p(x,y)}[g(x,y)] - \mathbb{E}_{p^{\mathbb{W}^\gamma}(x,y)}\left[e^{(g(x,y)-1)}\right], \\ \text{s.t. } \pi_0(x) = \int_{\mathcal{Y}} q(x,y)dy, \pi_1(y) = \int_{\mathcal{X}} q(x,y)dx$$

The next step is to follow dual representation of Static SB (The Schrodinger System) To be continued :)

## 6 Experiments

In this section we explore the performance of proposed methods on 1D and 2D toy datasets. We compare proposed methods with a baseline, a method when it is given two marginal distributions, and the KL-divergence between them can be explicitly measured.

Following Vargas [7] for each we divide tests in these groups:

- Unimodal to Unimodal Tests: This is one of the simplest tests and serves as the most basic sanity check for each method;

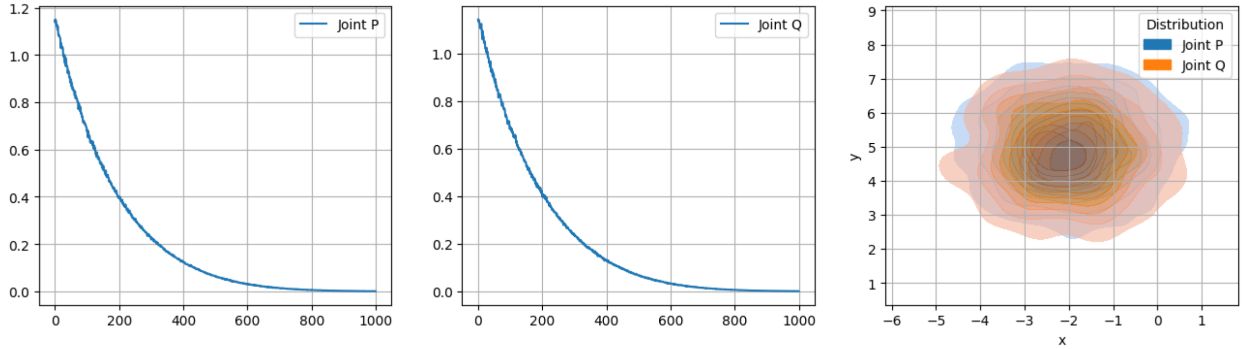

Figure 2: Training loss and joint distributions visualization

- Unimodal to Multimodal Tests: Due to possible mode collapse we evaluate the ability of a method to fit multimodal marginals.

### 6.1 Baseline

The baseline we explicitly estimate KL divergence as proposed in Algorithm 2. We fixate half of the parameters of multivariate normal distribution that are connected to marginals, particularly, fist half of the mean vector and half of diagonal of covariance matrix. The rest parameters are free to be fitted.

In this experiment, we explore the following generative process for the marginals:

$$\pi_0(x) = \mathcal{N}(-2; 1)$$
$$\pi_1(y) = \mathcal{N}(5; 1)$$

In 1000 epochs IPFP converges to its minimum. Overall, result is nearly perfect, wich could be seen from Figure 2. More precisely mean and variance are presented in Table 1

| Parameter | $x$ | $y$ |
|---|---|---|
| Mean (true) | $-2$ | 5 |
| Mean (trained) | $-1.9996$ | 4.8524 |
| Variance (true) | 1 | 1 |
| Variance (trained) | 1.0392 | 1.0396 |

Table 1: True and trained parameters of two joint distributions

Having made sure that the model could be easily trained using IPFP, let's check the other proposed methods.

### 6.2 Divergence disintegration

Now let's see the first introduced method with disintegration of joint KL-divergence.

### 6.3 Dual representation in IPFP

Following single-step gradient learning method from [4], we alternately train variational function and generator.

We parameterize generator with 3 fully connected layers (FCN). Each layer has 256 hidden neurons and length of output is twice larger than input, which is giving the ability to extract parameters of normal distribution, mean and variance. ReLU was chosen as desired function of activation.

Another challenge was initialisation of generator, so, that it matches $p^{\mathcal{W}^\gamma}(y|x) = \mathcal{N}(x, \gamma\mathbb{I})$. To do so all weights was set to identity matrix and second half of biases was filled with $\log\gamma$. The second generator $p(x|y)$ was initialized using Xavier uniform distribution.

To parameterize variational function we use almost the same architecture. 3 FCN with 256 hidden neurons, however we use LeakyReLU activation function with the angle of negative slope set to 0.1. The input is concatenated data vectors and output is scalar. Here we do not apply any initialization.

Regarding optimization process the information is following: we use Adam as optimization algorithm with weight decay 0.1 for generator and 0.5 for variational function, running averages coefficients set to $0.5, 0.999$ for all optimizers; learning rates were set $1e-3$ for generators and $1e-4$ for variational functions; batch size was chosen 4098 so that training models were able to get better understanding of original distribution.

Because of random nature of initialization, the optimized generation distributions were so far away in KL-divergence sense, that optimization led to gradients explosion. So, additional $L_2$ regularization played crucial role in optimization.

Choosing of $\gamma$ was another challenge. Vargas [7] showed that some methods works with large $\gamma = 1000$ and some with low $\gamma = 5$. In out experiments we find out that higher $\gamma$ led to gradient explosion, probably, only little part of trained distribution was covered by another and, consequently, KL-divergence was tremendous. So, our optimal $\gamma$ was set to 5.

## References

[1] S. Kullback. Probability densities with given marginals. The Annals of Mathematical Statistics, 39(4): 1236 – 1243, 1968. doi:10.1214/aoms/1177698249. URL https://doi.org/10.1214/aoms/1177698249.

[2] Ludger Rüschendorf. Convergence of the iterative proportional fitting procedure. Annals of Statistics, 23: 1160–1174, 1995. URL https://api.semanticscholar.org/CorpusID:122665767.

[3] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[4] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization, 2016.

[5] Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion schrцdinger bridge matching. 2023.

[6] Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrцdinger bridge with applications to score-based generative modeling. arXiv preprint arXiv:2106.01357, 2021.

[7] Francisco Vargas. Machine-learning approaches for the empirical schrцdinger bridge problem. Technical Report UCAM-CL-TR-958, University of Cambridge, Computer Laboratory, June 2021. URL https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-958.pdf.