Study of Diffusion Schrödinger Bridge model in Gaussian case

Gaëtan Ecrepont

December 2024

Abstract

Schrödinger Bridges (SB) generalize Optimal Transport by specifying not *where* but *how* to transport mass from one distribution to another, given a reference dynamic. Generative modeling can be achieved by finding SBs to go from p_{prior} to p_{data}, which amounts to solving a *dynamic* SB problem. De Bortoli et al. [3] introduced the Diffusion Schrödinger Bridge (DSB) model, a variational approach to the Iterative Proportional Fitting (IPF) algorithm to solve the *discrete dynamic* SB problem. DSB generalizes score-based generative modeling (SGM) introduced by Song et al. [10], and has stronger theoretical guarantees, in particular $p_T = p_{\text{prior}}$. This paper constitutes a theoretical and practical introduction to DSB. Our contribution is to explicit the closed-form solution of the *discrete dynamic* SB problem in the Gaussian case, and leverage this closed-form expression to assess the performance of the DSB model in various settings by varying the dimension and complexity of p_{data} and p_{prior}. In particular, we demonstrate that setting $L = 20$ DSB iterations as in the original paper amounts to under-training the DSB model.

Contents

1 Introduction

1.1 Score-Based Generative Modeling

Score-Based Generative Modeling (SGM) is a probabilistic approach to generative modeling which consists in sampling noise $\mathbf{x}_T \sim p_{\text{prior}}$ and gradually *denoising* it into $\mathbf{x}_0 \sim p_0$ such that $p_0 \simeq p_{\text{data}}$. SGM in fact derives from Denoising Diffusion Probabilistic Models (DDPM), introduced by Ho et al. in 2020 [5]. In DDPM, the data is progressively noised from p_{data} to $p_T \simeq p_{\text{prior}}$ from according to a Markov chain with Gaussian transitions, which we call the *forward process* or the *diffusion process*. The goal is to learn the *reverse process* to go from noise to data. Song et al. [10] showed that DDPMs were a discretization of Stochastic Differential Equations (SDEs). In this setting, the noising process can be seen as the solution of an Itô SDE $\mathrm{d}\mathbf{x} = f(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$, and thus we want to reverse this SDE in time to go from noise to data. The key is that the reverse SDE depends only on the time-dependent (Stein) score of the distribution $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ as $\mathrm{d}\mathbf{x} = [f(\mathbf{x}, t) - g^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x})]\mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}$ where $\bar{\mathbf{w}}$ is the reverse Brownian motion. The score is estimated using score matching techniques introduced by Vincent et al. [11]. Numerical SDE solvers can then be used from sampling, which amounts to performing ancestral sampling from $\mathbf{x}_T \sim p_{\text{prior}}$.

1.2 Diffusion Schrödinger Bridge

Although SGM provides state-of-the-art results, it is notoriously slow and computationally expensive. In particular, one must run the noising process long enough to have $p_T \simeq p_{\text{prior}}$ while maintaining a step size small enough to limit the time-discretization error of the SDE. In order words, SGM requires a large T and a small Δt, resulting in many steps N. To mitigate this issue, De Bortoli et al. generalized SGM through the Schrödinger Bridge (SB) problem by introducing the Diffusion Schrödinger Bridge (DSB) model [3]. Whereas SGM hinges on the *hypothesis* that $p_T \simeq p_{\text{prior}}$, in DSB we have $p_T = p_{\text{prior}}$ *by definition*. DSB can thus be applied with arbitrarily small values $T > 0$[1] which makes it faster than SGM for sampling as it requires less steps. In addition, SGM can be seen as first-order DSB and therefore DSB is a structurally superior method. Finally, DSB is more flexible than SGM since p_{data} isn't limited to a spherical Gaussian and can in fact be any available distribution. In particular, by setting $p_{\text{prior}} = p'_{\text{data}}$ one can perform dataset interpolation, demonstrating the usefulness of DSB for high-dimensional optimal transport between arbitrary data distributions.

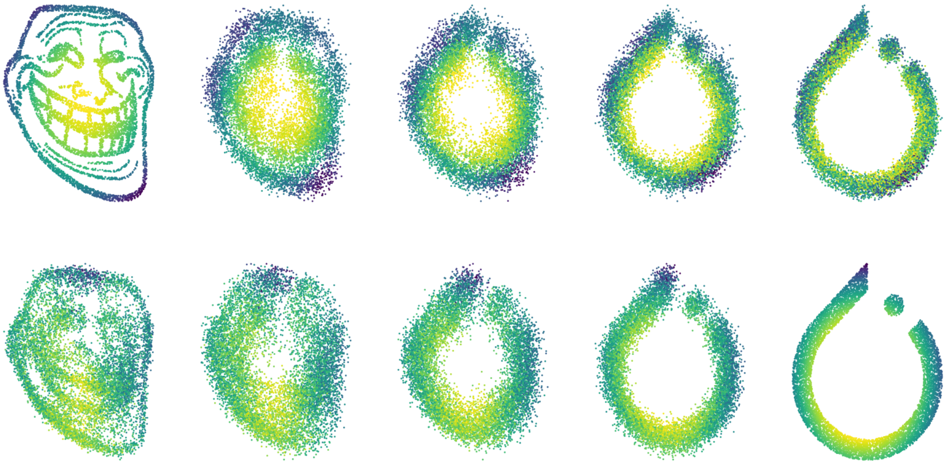

Figure 1: SB between two 2D distributions after $L = 20$ DSB iterations. The upper half is the forward process and the lower half is the reverse process. The x axis accounts for the time steps k.

1.3 Organization and contribution

The paper is organized as follows. In Section 2 we motivate and formulate the Schrödinger Bridge problem, in its *static* and *dynamic* forms. Section 3 presents the Iterative Proportional Fitting (IPF) algorithm which solves the discrete dynamic SB problem, and then describes the DSB which is a variational approach to IPF. Finally, in Section 4 we assess the performance of DSB in various Gaussian settings where the SB problem has a closed form solution. Our contribution is to leverage the closed-form solution of the SB problem in the Gaussian setting to compute the ground truth which can be utilized to measure the performance of DSB in various settings. We also detail the proofs of Proposition 3.2 and 3.1 which are at the core of DSB.

[1] When $T \to 0$ performance does decline a bit as the SB problem becomes hard to solve even for DSB.

2 The Schrödinger Bridge Problem

2.1 Motivation from statistical physics

The Schrödinger Bridge (SB) problem was introduced by Schrödinger in 1932 [9] and asks the following question.

Definition 2.1 (Intuitive formulation of SB)**.** *Let S be a particle system composed of a large number of independent particles following a Brownian motion. We observe $S \sim \nu_0$ at time 0 and $S \sim \nu_1$ at time T. What is the most likely dynamical behavior of S between 0 and T?*

In other words, the SB problem imposes an initial distribution ν_0, a final distribution ν_1 and a *reference dynamic* (here the Brownian motion i.e. diffusion according to the heat equation, but the SB problem can be generalized to any specified dynamics) and looks for the *distribution on paths* P which matches ν_0 at time 0 and ν_1 at time T while staying as close as possible to the reference dynamics between 0 and T. Note that by setting $\nu_0 = p_{\text{data}}$ and $\nu_1 = p_{\text{prior}}$ we recover generative modeling.

In the following two subsections, we rigorously formulate the modern SB problem in its *dynamic* and *static* forms. For a more thorough introduction to the Schrödinger Bridge problem, see e.g. [7].

2.2 Dynamic form

2.2.1 Notations

Let $\mathcal{X} = \mathbb{R}^n$ and $\Omega = C([0,1], \mathcal{X})$ the space of all continuous $\mathcal{X}$-valued paths on the unit time interval $[0,1]$. $\mathcal{X}$ is equipped with its Borel σ-field and Ω with the canonical σ-field $\sigma(X_t; 0 \leq t \leq 1)$ generated by the time projections $X_t : (\omega_s)_{0\leq s\leq 1} \in \Omega \mapsto \omega_t \in \mathcal{X}$.

Then, given any *path measure* $Q \in \mathrm{M}_+(\Omega)$ we define its time-marginals as the push-forward measures $Q_t = X_t \# Q \in \mathrm{M}_+(\mathcal{X})$ for $t \in [0,1]$. Thus for any $A \in \sigma(\mathcal{X}),\ Q_t(A) = Q(X_t \in A)$.

Intuitively, if Q describes the behavior of the random path $(X_t)_{0\leq t\leq 1}$ of a particle, then Q_t describes the random position X_t of this particle at time t. In particular, the concatenation of marginals $(Q_t)_{0\leq t\leq 1} \in \mathrm{M}_+(\Omega)^{[0,1]}$ contains less information than the joint distribution $Q \in \mathrm{M}_+(\Omega)$.

For any $(s,t) \in [0,1]$ we denote $Q_{st} = (X_s, X_t)\# Q \in \mathrm{M}_+(\mathcal{X}^2)$ such that for any $A \in \sigma(\mathcal{X}^2)$, $Q_{st}(A) = Q((X_s, X_t) \in A)$. We will be specifically interested in the endpoint marginal measure Q_{01} and we consequently define $Q^{x_0 x_1} = Q(\cdot | X_0 = x_0, X_1 = x_1) \in \mathrm{M}_+(\Omega)$ the *bridge* of Q between x_0 and x_1.

2.2.2 Why unbounded path measures

The reason why we consider unbounded path measures $Q \in \mathrm{M}_+(\Omega)$ instead of probability measures $P \in \mathrm{P}(\Omega)$ is that SB often considers the *Reversible Brownian Motion* (RBM) on $\mathcal{X}$ as the reference dynamic. The RBM diffuses according to the heat equation just like the regular Brownian motion, but its initial position X_0 is uniformly distributed on $\mathcal{X}$. In other words, if we denote $R \in \mathrm{M}_+(\Omega)$ the path measure of the RBM and R_0 its time-marginal at time 0, then R_0 is the Lebesgue measure on $\mathcal{X}$ i.e. $R_0(dx) = dx$. Considering *reversible* path measures like that of RBM as reference measures often simplifies computations, which is why they are preferred.

2.2.3 Statement of the dynamic SB problem

Definition 2.2 (Dynamic SB problem)**.** *Let $\nu_0, \nu_1 \in \mathrm{P}(\mathcal{X})$ be prescribed initial and final marginals and $R \in \mathrm{M}_+(\Omega)$ the reference dynamic. The associated Schrödinger problem is*

$$P^\star = \arg\min\{\mathrm{KL}(P|R) \mid P \in \mathrm{P}(\Omega),\ P_0 = \nu_0,\ P_1 = \nu_1\} \tag{S$_{\text{dyn}}$}$$

where KL *is the Kullback-Leibler (KL) divergence* $\mathrm{KL}(P|Q) = \int_{\mathcal{X}} P(x)\log(\frac{P(x)}{Q(x)})dx \in \mathbb{R} \cup \{+\infty\}$.

This is called the *dynamic* SB problem because we are optimizing over the space of *path probabilities* $\mathrm{P}(\Omega)$ i.e. we are describing *how* the mass goes from ν_0 to ν_1.

Note that since $\mathrm{KL}(\cdot|R)$ is strictly convex and the constraint set $\{P \in \mathrm{P}(\Omega),\ P_0 = \nu_0,\ P_1 = \nu_1\}$ is convex, (S$_{\text{dyn}}$) is a strictly convex problem and as such admits at most one solution $P^\star$.

2.3 Static form

Let us first state the SB problem in its *static* form. We will then explain the connection with the *dynamic* form.

2.3.1 Statement of the static SB problem

Definition 2.3 (Static SB problem). *Let $\nu_0, \nu_1 \in \mathrm{P}(\Omega)$ be prescribed initial and final marginals and $R_{01} \in \mathrm{M}_+(\mathcal{X}^2)$ the reference measure. The associated Schrödinger problem is*

$$P^{s,\star} = \arg\min\{\mathrm{KL}(P^s|R_{01}) \mid P^s \in \mathrm{P}(\mathcal{X}^2),\ P^s_0 = \nu_0,\ P^s_1 = \nu_1\} \tag{$\mathrm{S_{stat}}$}$$

This is called the *static* SB problem because we are optimizing over the space of *bridge endpoints* probabilities $\mathrm{M}_+(\mathcal{X}^2)$ i.e. we are only specifying *where* the mass goes when from ν_0 to ν_1.

($\mathrm{S_{stat}}$) is a strictly convex problem for the same reasons as ($\mathrm{S_{dyn}}$) and as such admits at most one solution $P^{s,\star}$.

Besides, one can rewrite $\mathrm{KL}(P^s|R_{01}) = \mathbb{E}_{(X_0,X_1)\sim P^s}[c(X_0,X_1)] - \mathrm{H}(P^s)$ where $c(x_0,x_1) = -\log R_{01}(x_0,x_1)$ and H is the entropy function. Thus ($\mathrm{S_{stat}}$) can be seen as an entropy-regularized optimal transport problem.

2.3.2 From dynamic to static SB

We now present two lemmas useful to link the *static* and *dynamic* SB problems. Proofs can be found in e.g. [8].

Lemma 2.1 (Disintegration formula). *Let Ω and Z be two Polish spaces equipped with their respective Borel σ-fields. For any measurable function $\phi : \Omega \to Z$ and any measure $Q \in \mathrm{M}_+(\Omega)$ we have the disintegration formula*

$$Q(\cdot) = \int_Z Q(\cdot|\phi = z)Q_\phi(dz) \tag{1}$$

where $Q_\phi = \phi\#Q$ and $z \in Z \mapsto Q(\cdot|\phi = z) \in \mathrm{P}(\Omega)$ is measurable.

Lemma 2.2 (Additive property of the KL divergence). *Let Ω and Z be two Polish spaces equipped with their respective Borel σ-fields. For any measurable function $\phi : \Omega \to Z$ and any measure $R \in \mathrm{M}_+(\Omega)$ and probability $P \in \mathrm{P}(\Omega)$ we have*

$$\mathrm{KL}(P|R) = \mathrm{KL}(P_\phi|R_\phi) + \int_Z \mathrm{KL}\big(P(\cdot|\phi = z)\big|R(\cdot|\phi = z)\big)P_\phi(dz) \tag{2}$$

which can be rewritten in expectation form as

$$\mathrm{KL}(P|R) = \mathrm{KL}(P_\phi|R_\phi) + \mathbb{E}_{P_\phi}\big[\mathrm{KL}\big(P_{|\phi=z}\big|R_{|\phi=z}\big)\big] \tag{3}$$

Theorem 2.1 (Equivalence of dynamic and static SB). *The Schrödinger problems* ($\mathrm{S_{dyn}}$) *and* ($\mathrm{S_{stat}}$) *admit respectively at most one solution $P^\star \in \mathrm{P}(\Omega)$ and $P^{s,\star} \in \mathrm{P}(\mathcal{X}^2)$.*

1. *If* ($\mathrm{S_{dyn}}$) *admits the solution $P^\star$, then $P^{s,\star} = P^\star_{01}$ is the solution of* ($\mathrm{S_{stat}}$).
2. *Conversely, if $P^{s,\star}$ solves* ($\mathrm{S_{stat}}$), *then* ($\mathrm{S_{dyn}}$) *admits the solution $P^\star(\cdot) = \int_{\mathcal{X}^2} R^{x_0x_1}(\cdot)P^{s,\star}(dx_0,dx_1)$. This means that $P^\star_{01} = P^{s,\star}$ and $(P^\star)^{x_0x_1} = R^{x_0x_1}\ P^{s,\star} - a.e.$*

Proof. To begin with, ($\mathrm{S_{dyn}}$) and ($\mathrm{S_{stat}}$) are both strictly convex problems and as such admit at most one solution.

Let's consider $\phi = (X_0, X_1) : (\omega_s)_{0\le s\le 1} \in \Omega \mapsto (\omega_0, \omega_1) \in \mathcal{X}^2$ and apply the additive property of the KL divergence. For all $P \in \mathrm{P}(\Omega)$:

$$\mathrm{KL}(P|R) = \mathrm{KL}(P_{01}|R_{01}) + \mathbb{E}_{(X_0,X_1)\sim P_{01}}\big[\mathrm{KL}\big(P^{X_0X_1}\big|R^{X_0X_1}\big)\big]$$

1. Assume that ($\mathrm{S_{dyn}}$) has solution $P^\star$. We know this solution must be of the form $P^\star(\cdot) = \int_{\mathcal{X}^2} R^{x_0x_1}(\cdot)P^\star_{01}(dx_0,dx_1)$ where $P^\star_{01}$ minimizes $\mathrm{KL}(\cdot|R_{01})$. Thus $P^\star_{01}$ is the solution of ($\mathrm{S_{stat}}$) i.e. $P^{s,\star} = P^\star_{01}$.
2. Assume that ($\mathrm{S_{stat}}$) has solution $P^{s,\star}$. The above equation implies that $\mathrm{KL}(P|R) \ge \mathrm{KL}(P_{01}|R_{01})$ with equality if and only if $P^{x_0x_1} = R^{x_0x_1}\ P_{01} - a.e.$. In addition, $\mathrm{KL}(P_{01}|R_{01})$ is by definition minimized by setting $P_{01} = P^{s,\star}$. We have thus defined the bridges $P^{x_0x_1}$ and the endpoints P_{01} of the optimal solution and we finally obtain the desired expression for $P^\star$ using the disintegration formula.

□

3 Solving the Schrödinger Bridge

In Section 2 we have introduced the *continuous* SB problem in its *dynamic* and *static* forms and demonstrated the equivalence between the two. In this Section, we first present the *discrete* version of the SB problems in both forms. We then describe the IPF algorithm which solves the *discrete dynamic* SB problem. We finally introduce the Diffusion Schrödinger Bridge [3] as a variational approach to IPF.

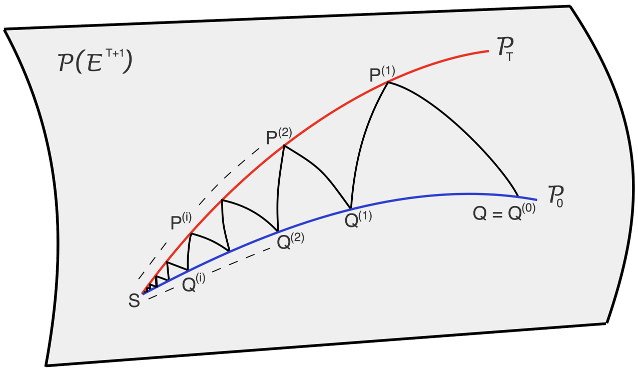

Figure 2: The IPF algorithm amounts to iteratively solving half-bridges (i.e. projecting with respect to the KL divergence) until convergence to a fixed point denoted by S here. Figure taken from [1].

3.1 Discretized Schrödinger Bridge

Let's first denote once and for all $\mathcal{X} = \mathbb{R}^d$ where d is the dimension of the process. Then, instead of considering continuous time $t \in [0, 1]$ we discretize the process into N time steps, such that Ω becomes $(\mathbb{R}^d)^N$. The reference measure is still denoted $R \in \mathrm{M}_+(\Omega)$ and we still have the extremal conditions $\nu_0, \nu_1 \in \mathrm{P}(\mathbb{R}^d)$.

The *discrete dynamic* and *discrete static* SB problems can thus be formulated as follows:

$$P^\star = \arg\min\{\mathrm{KL}(P|R) \mid P \in \mathrm{P}((\mathbb{R}^d)^N),\ P_0 = \nu_0,\ P_N = \nu_1\} \tag{DS$_{\text{dyn}}$}$$

$$P^{s,\star} = \arg\min\{\mathrm{KL}(P^s|R_{0N}) \mid P^s \in \mathrm{P}((\mathbb{R}^d)^2),\ P^s_0 = \nu_0,\ P^s_N = \nu_1\} \tag{DS$_{\text{stat}}$}$$

(DS$_{\text{stat}}$) can be seen as a *discrete entropy-regularized optimal transport* problem and therefore it can be readily solved using the Sinkhorn algorithm.

(DS$_{\text{dyn}}$) is more complicated however, and requires the *dynamic* equivalent of the Sinkhorn algorithm, which is the IPF algorithm.

3.2 Iterative Proportional Fitting

We now present the IPF algorithm which solves the *discrete dynamic* SB problem.

```
Algorithm 1 Iterative Proportional Fitting (IPF)
Input: prior distribution $p_{\text{prior}}$, data distribution $p_{\text{data}}$, reference dynamics $R$
Initialize $P^0 = p_{\text{data}} \prod_{k=0}^{N-1} R_{k+1|k}$
for $n \geq 1$ do
    Update $P^{2n+1}$ as:
        $P^{2n+1} = \arg\min \mathrm{KL}\left(P|P^{2n}\right), \quad P_N = p_{\text{prior}}$
    Update $P^{2n+2}$ as:
        $P^{2n+2} = \arg\min \mathrm{KL}\left(P|P^{2n+1}\right), \quad P_0 = p_{\text{data}}$
end for
```

This algorithm can be seen as an alternate projection scheme with respect to the KL divergence [2]. It iteratively solves a system of two coupled equations until it reaches a fixed point. Intuitively, each iteration amounts to solving a half-bridge problem which starts from $\mathbf{x}_0 \sim p_{\text{data}}$ or from $\mathbf{x}_N \sim p_{\text{prior}}$ and evolves according to dynamics close to the reference dynamic R.

The IPF algorithm is simple enough to allow us to write theoretical expressions for P^{2n} and P^{2n+1}. Writing these expressions gives an intuitive understanding of what the algorithm is doing.

1. We begin the algorithm with a distribution P^0 obtained by starting from $\mathbf{x}_0 \sim p_{\text{data}}$ and then following the reference dynamics given by R i.e. $P^0(\mathbf{x}_{0:N}) = p_{\text{data}}(\mathbf{x}_0) \prod_{k=0}^{N-1} R_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{x}_k)$.

2. We then construct P^1 as the distribution obtained by starting from $x_N \sim p_{\text{prior}}$ and then sampling according to the reverse transitions of P^0 i.e. $P^1(\mathbf{x}_{0:N}) = p_{\text{prior}}(\mathbf{x}_N) \prod_{k=0}^{N-1} P^0_{k|k+1}(\mathbf{x}_k|\mathbf{x}_{k+1})$.

3. Likewise, P^2 is constructed by first sampling $\mathbf{x}_0 \sim p_{\text{data}}$ and then by reversing the dynamics of P^1 i.e. $P^2(\mathbf{x}_{0:N}) = p_{\text{data}}(\mathbf{x}_0) \prod_{k=0}^{N-1} P^1_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{x}_k)$

4. This process is repeated until P^n converges.

Remark 1 [Link with SGM] In SGM, we set $p(\mathbf{x}_{0:N}) = p_{\text{data}}(\mathbf{x}_0) \prod_{k=0}^{N-1} p_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{x}_k)$ and we learn $P^1(\mathbf{x}_{0:N}) = p_{\text{prior}}(\mathbf{x}_N) \prod_{k=0}^{N-1} p_{k|k+1}(\mathbf{x}_k|\mathbf{x}_{k+1})$. This shows that SGM consists in approximating only the first iteration of IPF and therefore justifies the inherent superiority of DSB.

Remark 2 [Theoretical guarantees of IPF] When the state-space Ω is discrete or in the case where p_{data} and p_{prior} are compactly supported, then IPF converges at a geometric rate w.r.t. the Hilbert-Birkhoff metric, see e.g. [6] for a definition. Under milder hypothesis, De Bortoli et al. [3] prove that $(P^n)_n$ converges with $\text{KL}(P^n_0|p_{\text{data}}) = O(\frac{1}{n})$ and $\text{KL}(P^n_N|p_{\text{prior}}) = O(\frac{1}{n})$.

Following [3], we introduce recursive formulas for the densities of P^{2n} and P^{2n+1} for $n \in \mathbb{N}$. These formulas allow for a convenient representation of P^n which will be leveraged in the next paragraph on DSB.

Proposition 3.1. *Assume that* $\text{KL}(p_{\text{data}} \otimes p_{\text{prior}}|p_{0,N}) < +\infty$. *Then for any* $n \in \mathbb{N}$, P^{2n} *and* P^{2n+1} *admit positive densities w.r.t. the Lebesgue measure denoted as* p^n *resp.* q^n *and for any* $\mathbf{x}_{0:N} \in \Omega$, *we have* $p^0(\mathbf{x}_{0:N}) = p_{\text{data}}(\mathbf{x}_0) \prod_{k=0}^{N-1} R_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{x}_k)$ *and*

$$q^n(\mathbf{x}_{0:N}) = p_{\text{prior}}(\mathbf{x}_N) \textstyle\prod_{k=0}^{N-1} p^n_{k|k+1}(\mathbf{x}_k|\mathbf{x}_{k+1}),\ p^{n+1}(\mathbf{x}_{0:N}) = p_{\text{data}}(\mathbf{x}_0) \textstyle\prod_{k=0}^{N-1} q^n_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{x}_k).$$

In practice we have access to $p^n_{k+1|k}$ and $q^n_{k|k+1}$. Hence, to compute $p^n_{k|k+1}$ and $q^n_{k+1|k}$ we use Bayes rule:

$$p^n_{k|k+1}(\mathbf{x}_k|\mathbf{x}_{k+1}) = \frac{p^n_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{x}_k)p^n_k(\mathbf{x}_k)}{p^n_{k+1}(\mathbf{x}_{k+1})},\ q^n_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{x}_k) = \frac{q^n_{k|k+1}(\mathbf{x}_k|\mathbf{x}_{k+1})q^n_{k+1}(\mathbf{x}_{k+1})}{q^n_k(\mathbf{x}_k)}.$$

The densities p^n and q^n introduced in the above proposition cannot be computed in closed-form and thus need to be approximated. This is the objective of DSB, which is a variational approach relevant to generative modeling.

3.3 Diffusion Schrödinger Bridge

In SGM we use Gaussian transitions of the form

$$p_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{x}_k) = \mathcal{N}(\mathbf{x}_{k+1}|\mathbf{x}_k + \gamma f(\mathbf{x}_k), 2\gamma\mathbf{I}) \tag{4}$$

where $f(\mathbf{x}) = -\alpha\mathbf{x}^2$.

In Song et al.'s SDE approach [10] we have $\alpha = 0$ which corresponds to setting the reference dynamic R to be a Brownian motion. On the contrary, in the DPPM approach [5] we have $\alpha > 0$ which means the reference dynamic R is an Ornstein-Ulhenbeck process. This motivates the choice of initial reference dynamic $p = p_{\text{data}} \prod_{k=0}^{N-1} p_{k+1|k}$ with transitions $p_{k+1|k}$ given by (4). In this Gaussian setting, we can compute closed-form approximations for p^n and q^n, which is the object of the proposition below.

Proposition 3.2. *Let* $p^0 = p$ *and* $f^0_k = f$ *as defined above. Then for all* $n \in \mathbb{N}$, *we have the following approximations:*

$$q^n_{k|k+1}(\mathbf{x}_k|\mathbf{x}_{k+1}) \simeq \mathcal{N}(\mathbf{x}_k|\mathbf{x}_{k+1} + \gamma b^n_{k+1}(\mathbf{x}_{k+1}), 2\gamma\mathbf{I}) \tag{5}$$

$$p^{n+1}_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{x}_k) \simeq \mathcal{N}(\mathbf{x}_{k+1}|\mathbf{x}_k + \gamma f^{n+1}_k(\mathbf{x}_k), 2\gamma\mathbf{I}) \tag{6}$$

where

$$b^n_{k+1}(\mathbf{x}_{k+1}) = -f^n_k(\mathbf{x}_{k+1}) + 2\nabla \log p^n_{k+1}(\mathbf{x}_{k+1}) \tag{7}$$

$$f^{n+1}_k(\mathbf{x}_k) = f^n_k(\mathbf{x}_k) - 2\nabla \log p^n_{k+1}(\mathbf{x}_k) + 2\nabla \log q^n_k(\mathbf{x}_k) \tag{8}$$

[2] One can generalize with time-dependent step sizes i.e. $(\gamma_k)_k$ instead of a fixed γ. For the sake of simplicity we won't consider this case.

Proof. We will only prove the formula for $q^n_{k|k+1}$ as the demonstration is essentially the same for $p^{n+1}_{k+1|k}$.

$$\begin{aligned}
\log q^n_{k|k+1}(\mathbf{x}_k|\mathbf{x}_{k+1}) &= \log p^n_{k|k+1}(\mathbf{x}_k|\mathbf{x}_{k+1}) \\
&= \log p^n_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{x}_k) + [\log p^n_k(\mathbf{x}_k) - \log p^n_{k+1}(\mathbf{x}_{k+1})] \\
&\approx \log p^n_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{x}_k) + [\log p^n_{k+1}(\mathbf{x}_k) - \log p^n_{k+1}(\mathbf{x}_{k+1})] \\
&\approx \log p^n_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{x}_k) + (\mathbf{x}_k - \mathbf{x}_{k+1})^T \nabla \log p^n_{k+1}(\mathbf{x}_{k+1}) \\
&= -\frac{1}{4\gamma}||\mathbf{x}_{k+1} - (\mathbf{x}_k + \gamma f^n_k(\mathbf{x}_k))||^2 + (\mathbf{x}_k - \mathbf{x}_{k+1})^T \nabla \log p^n_{k+1}(\mathbf{x}_{k+1}) + \text{const} \\
&\approx -\frac{1}{4\gamma}||\mathbf{x}_{k+1} - \mathbf{x}_k - \gamma f^n_k(\mathbf{x}_{k+1})||^2 + (\mathbf{x}_k - \mathbf{x}_{k+1})^T \nabla \log p^n_{k+1}(\mathbf{x}_{k+1}) + \text{const} \\
&\approx -\frac{1}{4\gamma}\left[||\mathbf{x}_k||^2 - 2\mathbf{x}_k^T(\mathbf{x}_{k+1} - \gamma f^n_k f(\mathbf{x}_{k+1}) + 2\gamma \nabla \log p^n_{k+1}(\mathbf{x}_{k+1})) + \cdots\right] + \text{const} \\
&= \mathcal{N}(\mathbf{x}_k|\mathbf{x}_{k+1} + \gamma b^n_k(\mathbf{x}_{k+1}))
\end{aligned}$$

where we used $p^n_k \approx p^n_{k+1}$ in line 3, a Taylor expansion of $\log p_{k+1}(\mathbf{x}_k)$ in $\mathbf{x}_{k+1}$ in line 4, $f^n_k(\mathbf{x}_k) \approx f^n_k(\mathbf{x}_{k+1})$ in line 6 and we used the "complete the square" method in line 7 and 8. □

The problem with the above approach is that the scores $\nabla p^n_k(\mathbf{x})$ and $\nabla q^n_k(\mathbf{x})$ accumulate in the definitions of $f^n_k(\mathbf{x})$ and $b^n_k(\mathbf{x})$. At step n, we effectively need to have compute to $2n$ scores, each of which is estimated using a separate score network. This approach is thus too costly in compute and memory.

The workaround proposed by De Bortoli et al. [3] is to directly approximate the mean of the transitions $q^n_{k|k+1}$ and $p^n_{k+1|k}$, which can be *recursively* computed without requiring to store $2n$ score networks and compute $2n$ scores at each iteration n. They dubbd this variational approach *Iterative Mean-Matching Proportional Fitting* and it is detailed in the proposition below.

Proposition 3.3 (Iterative Mean-Matching Proportional Fitting (IMMPF))**.** *Assume that for any $n \in \mathbb{N}$ and $k \in \{0, \ldots, N-1\}$,*

$$q^n_{k|k+1}(\mathbf{x}_k|\mathbf{x}_{k+1}) = \mathcal{N}(\mathbf{x}_k; B^n_{k+1}(\mathbf{x}_{k+1}), 2\gamma\mathbf{I}) \tag{9}$$

$$p^n_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{x}_k) = \mathcal{N}(\mathbf{x}_{k+1}; F^n_k(\mathbf{x}_k), 2\gamma\mathbf{I}) \tag{10}$$

with $B^n_{k+1}(\mathbf{x}) = \mathbf{x} + \gamma b^n_{k+1}(\mathbf{x})$, $F^n_k(\mathbf{x}) = \mathbf{x} + \gamma f^n_k(\mathbf{x})$ for any $\mathbf{x} \in \mathbb{R}^d$.

Then we have for any $n \in \mathbb{N}$ and $k \in \{0, \ldots, N-1\}$

$$B^n_{k+1} = \arg \min_{B \in L^2(\mathbb{R}^d, \mathbb{R}^d)} \mathbb{E}_{p^n_{k,k+1}}\left[\|B(X_{k+1}) - (X_{k+1} + F^n_k(X_k) - F^n_k(X_{k+1}))\|^2\right] \tag{11}$$

$$F^{n+1}_k = \arg \min_{F \in L^2(\mathbb{R}^d, \mathbb{R}^d)} \mathbb{E}_{q^n_{k,k+1}}\left[\|F(X_k) - (X_k + B^n_{k+1}(X_{k+1}) - B^n_{k+1}(X_k))\|^2\right] \tag{12}$$

Proof. We will only prove the formula for B^n_{k+1} as the demonstration is essentially the same for F^{n+1}_k. Let $n \in \mathbb{N}$ and $k \in \{0, \ldots, N-1\}$. For all $\mathbf{x}_{k+1} \in \mathbb{R}^d$ we have

$$p^n_{k+1}(\mathbf{x}_{k+1}) = \int p^n_k(\mathbf{x}_k) p^n_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{x}_k) d\mathbf{x}_k$$

such that

$$\begin{aligned}
\nabla \log p^n_{k+1}(\mathbf{x}_{k+1}) &= \frac{1}{p^n_{k+1}(\mathbf{x}_{k+1})} \nabla p^n_{k+1}(\mathbf{x}_{k+1}) \\
&= \frac{1}{p^n_{k+1}(\mathbf{x}_{k+1})} \int p^n_k(\mathbf{x}_k) \nabla p^n_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{x}_k) d\mathbf{x}_k \\
&= \frac{1}{p^n_{k+1}(\mathbf{x}_{k+1})} \int p^n_k(\mathbf{x}_k) \left[\frac{F^n_k(\mathbf{x}_k) - \mathbf{x}_{k+1}}{2\gamma} p^n_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{x}_k)\right] d\mathbf{x}_k \\
&= \frac{1}{2\gamma} \int p^n_{k|k+1}(\mathbf{x}_k|\mathbf{x}_{k+1})(F^n_k(\mathbf{x}_k) - \mathbf{x}_{k+1}) d\mathbf{x}_k \\
&= \frac{1}{2\gamma}\left(\mathbb{E}_{X_k \sim p^n_{k|k+1}(\cdot|\mathbf{x}_{k+1})}[F^n_k(X_k)] - \mathbf{x}_{k+1}\right)
\end{aligned}$$

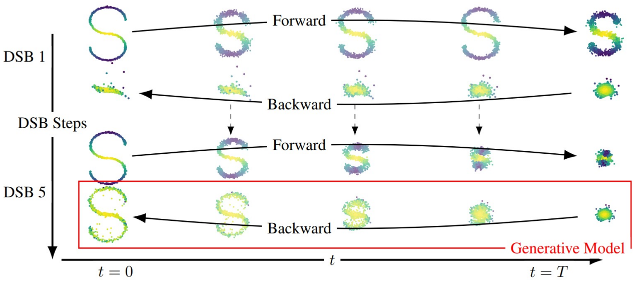

Figure 3: Illustration of the Diffusion Schrödinger Bridge algorithm.

It follows that

$$\begin{aligned} b_{k+1}^n(\mathbf{x}_{k+1}) &= -f_k^n(\mathbf{x}_{k+1}) + 2\nabla \log p_{k+1}^n(\mathbf{x}_{k+1}|\mathbf{x}_k) \\ &= \frac{1}{\gamma}(\mathbf{x}_{k+1} - F_k^n(\mathbf{x}_{k+1})) + \frac{1}{\gamma}\left(\mathbb{E}_{X_k \sim p_{k|k+1}^n(\cdot|\mathbf{x}_{k+1})}[F_k^n(X_k)] - \mathbf{x}_{k+1}\right) \\ &= \frac{1}{\gamma}\mathbb{E}_{X_k \sim p_{k|k+1}^n(\cdot|\mathbf{x}_{k+1})}[F_k^n(X_k) - F_k^n(\mathbf{x}_{k+1})] \end{aligned}$$

Finally,

$$\begin{aligned} B_{k+1}^n(\mathbf{x}_{k+1}) &= \mathbf{x}_{k+1} + \gamma b_{k+1}^n(\mathbf{x}_{k+1}) \\ &= \mathbb{E}_{X_k \sim p_{k|k+1}^n(\cdot|\mathbf{x}_{k+1})}[\mathbf{x}_{k+1} + F_k^n(X_k) - F_k^n(\mathbf{x}_{k+1})] \\ &= \mathbb{E}_{(X_k, X_{k+1}) \sim p_{k,k+1}^n}[X_{k+1} + F_k^n(X_k) - F_k^n(X_{k+1}) \big| X_{k+1} = \mathbf{x}_{k+1}] \end{aligned}$$

By definition of the conditional expectation, we obtain the desired expression for B_{k+1}^n. □

We now have recursive formulas to approximate B_{k+1}^n and F_k^{n+1}, which are defined according to the variational problem described in the above proposition.

In practice, we use neural networks to approximate $B_{\beta^n}(k,\mathbf{x}) \approx B_k^n(\mathbf{x})$ and $F_{\alpha^n}(k,\mathbf{x}) \approx F_k^n(\mathbf{x})$. The parameters β^n and α^n are learned with gradient descent to minimize the sum over k of the empirical loss functions given by (11) and (12). These empirical losses are computed using M samples and denoted as $\hat{\ell}_n^b(\beta)$ and $\hat{\ell}_n^f(\alpha)$. The resulting algorithm, which runs for $L \in \mathbb{N}$ IMMPF iterations, constitutes the DSB algorithm.

The DSB algorithm is illustrated in Figure 3 and summarized in Algorithm 2 where $Z_k^j, \tilde{Z}_k^j \sim \mathcal{N}(0, \mathbf{I})$ i.i.d.

Algorithm 2 Diffusion Schrödinger Bridge

```
for $n \in \{0, \dots, L\}$ do
    while not converged do
        Sample $\{X_k^j\}_{k,j=0}^{N,M}$, where $X_0^j \sim p_{\text{data}}$, and
        $X_{k+1}^j = F_{\alpha^n}(k, X_k^j) + \sqrt{2\gamma} Z_{k+1}^j$
        Compute $\hat{\ell}_n^b(\beta^n)$ approximating (11)
        $\beta^n \leftarrow$ Gradient Step($\hat{\ell}_n^b(\beta^n)$)
    end while
    while not converged do
        Sample $\{X_k^j\}_{k,j=0}^{N,M}$, where $X_N^j \sim p_{\text{prior}}$, and
        $X_{k-1}^j = B_{\beta^n}(k, X_k^j) + \sqrt{2\gamma} \tilde{Z}_k^j$
        Compute $\hat{\ell}_{n+1}^f(\alpha^{n+1})$ approximating (12)
        $\alpha^{n+1} \leftarrow$ Gradient Step($\hat{\ell}_{n+1}^f(\alpha^{n+1})$)
    end while
end for
Output: $(\alpha^{L+1}, \beta^L)$
```

3.3.1 Implementation of DSB

Now that DSB has been presented from a theoretical standpoint, we describe the practical tricks devised by De Bortoli et al. [3] to improve training efficiency.

Let's first define the generalized losses $\hat{\ell}^b_{n,I}$ and $\hat{\ell}^f_{n,I}$ for $I \subset I_0 = \{0, \ldots, N-1\} \times \{1, \ldots, M\}$.

$$\hat{\ell}^b_{n,I}(\beta) = \frac{1}{M} \sum_{(k,j)\in I} \|B_\beta(k+1, X^j_{k+1}) - (X^j_{k+1} + F^n_k(X^j_{k+1}) - F^n_k(X^j_k))\|^2 \tag{13}$$

$$\hat{\ell}^f_{n+1,I}(\alpha) = \frac{1}{M} \sum_{(k,j)\in I} \|F_\alpha(k, X^j_k) - (X^j_k + B^n_{k+1}(X^j_{k+1}) - B^n_{k+1}(X^j_k))\|^2 \tag{14}$$

Technique A: Cached DSB Because for each sample $j \in \{1, \ldots, M\}$, the corresponding sampled path $\{X^j_k\}_{0\leq k\leq N}$ is made up of correlated points, only a single uniformly sampled point $X^j_{K_j}$ where $K_j \sim \mathrm{U}(\{0, \ldots, N-1\})$ per path X^j is used to compute the loss per gradient step. However this approach would be hugely wasteful since for each gradient step of α or β we only use one point $X^j_{K_j}$ per path X^j i.e. we only use M of the $(N+1)M$ sampled points. Instead, we store the $(N+1)M$ points $\{X^j_k\}^{1\leq j\leq M}_{0\leq k\leq N}$ in a *cache* and compute the n_{epoch} gradient steps of α or β by repeatedly sampling a uniform subset I_{batch} of size *batch_size* from the cache $\{X^j_k\}^{1\leq j\leq M}_{0\leq k\leq N}$ and performing gradient descent on $\hat{\ell}^f_{n,I^c}$ and $\hat{\ell}^b_{n,I_{\text{batch}}}$ at each epoch. The cache is refreshed every *cache_period* epochs. This *cached* version of DSB can be improved further by tweaking the cache-size $(N+1)M$ and the refresh frequency *cache_period* based on available GPU memory.

Algorithm 3 presents for Cached DSB algorithm as proposed by De Bortoli et al. [3].

Algorithm 3 Cached Diffusion Schrödinger Bridge

```
for $n \in \{0, \ldots, L\}$ do
  while not converged do
    Sample and store $\{X^j_k\}^{N,M}_{k,j=0}$ where $X^j_0 \sim p_{\text{data}}$ and
    $X^j_{k+1} = X^j_k + \gamma f_{\alpha^n}(k, X^j_k) + \sqrt{2\gamma} Z^j_{k+1}$
    while not refreshed do
      Sample $I$ (uniform in $\{0, N-1\} \times \{1, M\}$)
      Compute $\hat{\ell}^b_{n,I}(\beta^n)$
      $\beta^n$ = Gradient Step($\hat{\ell}^b_{n,I}(\beta^n)$)
    end while
  end while
  while not converged do
    Sample $\{X^j_k\}^{N,M}_{k,j=0}$, where $X^j_N \sim p_{\text{prior}}$, and
    $X^j_k = X^j_{k+1} + \gamma b_{\beta^n}(k, X^j_k) + \sqrt{2\gamma} Z^j_k$
    while not refreshed do
      Sample $I$ (uniform in $\{0, N-1\} \times \{1, M\}$)
      Compute $\hat{\ell}^f_{n+1,I}(\alpha^{n+1})$
      $\alpha^{n+1}$ = Gradient Step($\hat{\ell}^f_{n+1,I}(\alpha^{n+1})$)
    end while
  end while
end for
Output: $(\alpha^{L+1}, \beta^L)$
```

Technique B: Tuning p_{prior} The converge of IPF is affected by the mean and covariance of the target Gaussian p_{prior}. Empirically, De Bortoli et al. find that choosing $\sigma^2_{\text{prior}} \gtrsim \sigma^2_{\text{data}}$ and $\mu_{\text{prior}} = \mu_{\text{data}}$ works well.

Technique C: Warm-start approach Instead of training each network B_{β^n} and F_{α^n} from scratch, we remark that

$$b^n_{k+1}(\mathbf{x}) = b^{n-1}_{k+1}(\mathbf{x}) + 2\nabla \log p^n_{k+1}(\mathbf{x}) - 2\nabla \log q^{n-1}_k(\mathbf{x}) \tag{15}$$

$$f^n_k(\mathbf{x}) = f^{n-1}_k(\mathbf{x}) + 2\nabla \log q^{n-1}_k(\mathbf{x}) - 2\nabla \log p^{n-1}_{k+1}(\mathbf{x}) \tag{16}$$

which motivates to the initializations $\beta^n_{\text{initial}} = \beta^{n-1}$ and $\alpha^n_{\text{initial}} = \alpha^{n-1}$. Thus instead of training each network from scratch, we are fine-tuning it from the previous iteration, which saves a lot of compute.

4 Experiments on Gaussians

Now that we have fully described DSB and how to implement it in practice, we will test it in a Gaussian setting i.e. with Gaussian p_{data} and p_{prior}. In this case, the solution to the SB problem is known in closed form, and thus we can compare DSB results to the ground truth to assess DSB performance in various settings.

In this Section we first formulate the closed-form solution to the Gaussian SB problem, and then we use it to gauge DSB performance with different parameters from p_{data} and p_{prior}.

4.1 Closed-form solution to the Gaussian SB problem

Let $p_{\text{data}} \sim \mathcal{N}(\mu, \Sigma)$ and $p_{\text{prior}} \sim \mathcal{N}(\mu', \Sigma')$. Recall that the reference dynamic p is given by $p_0 = p_{\text{data}}$ and $p_{k+1|k}(\mathbf{x}_{k+1}|\mathbf{x}_k) = \mathcal{N}(\mathbf{x}_{k+1}|\mathbf{x}_k, 2\gamma\mathbf{I})$ where we set $\alpha = 0$ and thus $f = 0$ for simplicity. In particular, we have $p_{N|0}(\mathbf{x}_N|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_N|\mathbf{x}_0, \sigma^2\mathbf{I})$ where $\sigma^2 = 2N\gamma$.

We will solve the discrete *static* SB problem which we know from Subsection 2.3.1 can be seen as an entropy-regularized optimal transport problem. Recall the *discrete static* SB problem:

$$\pi^{s,\star} = \arg\min\{\mathrm{KL}(\pi^s|p) \mid \pi^s \in \mathrm{P}((\mathbb{R}^d)^2), \pi^s_0 = p_{\text{data}}, \pi^s_N = p_{\text{prior}}\} \tag{$\mathrm{S_{stat}}$}$$

We can develop the KL term:

$$\begin{aligned}\mathrm{KL}(\pi|p_{0,N}) &= \mathbb{E}_{\pi^s}[-\log p_{0,N}(X_0, X_N)] - H(\pi^s)\\ &= \mathbb{E}_{\pi^s}[-\log p_0(X_0)] + \mathbb{E}_{\pi^s}[-\log p_{N|0}(X_N|X_0)] - H(\pi^s)\\ &= \mathbb{E}_{p_{\text{data}}}[-\log p_0(X_0)] + \mathbb{E}_{\pi^s}[\frac{||X_N - X_0||^2}{2\sigma^2}] - H(\pi^s)\end{aligned}$$

Such that ($\mathrm{S_{stat}}$) can be rewritten as:

$$\pi^{s,\star} = \arg \min_{\pi^s \in \Gamma(p_{\text{data}}, p_{\text{prior}})} \mathbb{E}_{\pi^s}[||X_N - X_0||^2] - 2\sigma^2 H(\pi^s) \tag{17}$$

Using e.g. [4] to solve (17), we obtain a closed-form solution for $\pi^{s,\star}$:

$$\pi^{s,\star} \sim \mathcal{N}\left(\begin{bmatrix}\mu\\ \mu'\end{bmatrix}, \begin{bmatrix}\Sigma & C_\sigma\\ C_\sigma^\top & \Sigma'\end{bmatrix}\right). \tag{18}$$

where

$$\begin{aligned}D_\sigma &= \left(4\Sigma^{\frac{1}{2}}\Sigma'\Sigma^{\frac{1}{2}} + \sigma^4 I\right)^{\frac{1}{2}}\\ C_\sigma &= \frac{1}{2}\left(\Sigma^{\frac{1}{2}} D_\sigma \Sigma^{-\frac{1}{2}} - \sigma^2 I\right)\end{aligned}$$

4.2 Testing DSB in various settings

We will now propose various Gaussian p_{data} and p_{prior} and in each case we fit the DSB which amounts to computing an approximate Schrödinger bridge $(X_0, X_N) \sim \mathrm{DSB}(p_{\text{data}}, p_{\text{prior}})$. We then compare this approximate bridge to the true bridge $\mathrm{SB}(p_{\text{data}}, p_{\text{prior}})$, which can be computed in closed-form using the above equations.

4.2.1 Test families considered

Let $U = U([1, 10])$[3] and consider the following three family of tests:

1. **spherical** with $p_{\text{data}} \sim \mathcal{N}(0, \sigma^2 I)$, $p_{\text{prior}} \sim \mathcal{N}(0, \sigma'^2 I)$ where $\sigma, \sigma' \sim U$
2. **diagonal** with $p_{\text{data}} \sim \mathcal{N}(0, \mathrm{Diag}[(\sigma_i^2)_i])$, $p_{\text{prior}} \sim \mathcal{N}(0, \mathrm{Diag}[(\sigma_i'^2)_i])$ where $\sigma_i, \sigma_i' \sim U$
3. **general** with $p_{\text{data}} \sim \mathcal{N}(0, \Sigma)$, $p_{\text{prior}} \sim \mathcal{N}(0, \Sigma')$ where $\Sigma = ODO^T$ where O is an orthogonal matrix sampled uniformly with respect to the Haar measure and $D = \mathrm{Diag}[(\sigma_i)_i]$ where $\sigma_i \sim U$; Σ' is constructed in the same fashion

Note that for the sake of simplicity, we set $\mu = \mu' = 0$ in all cases as modeling the mean of a distribution isn't challenging ; all the difficulty lies in the covariance.

Each family of test will conducted in dimension $d \in \{1, 5, 10\}$ as we expect performance to worsen in higher dimension[4]. Contrary to [3] we keep the same DSB model size (30k parameters) regardless of d, for an apples to apples comparison. Like [3], we take $\gamma = \frac{1}{40}$ and $N = 20$ such that $T = 2N\gamma = 1$, and we train the DSB for $L = 20$ iterations.

[3] We initially used $U = U([0, 1])$ but the results proved unstable when the variance was close to 0, hence we used $U([1, 10])$ instead, which greatly improved the results.

[4] We initially tried $d = 50$ too, but results were more difficult to analyze and compare with smaller dimensions because the error was much higher and there was much more noise. We thus decided to drop this case.

4.2.2 Evaluation protocol

The evaluation protocol of DSB is the following:

1. set p_{data} and p_{prior}
2. train the DSB for L iterations.
3. generate $X_N^j \sim p_{\text{prior}}$ for $1 \leq j \leq M$ which are the starting ends of M distinct bridges
4. use the DSB to complete each bridge into $(X_0^j, \ldots, X_N^j)$ for $1 \leq j \leq M$
5. compute the empirical covariance $\hat{\Sigma} = \frac{1}{M}\sum_{j=1}^{M} X_0^j (X_0^j)^T$ of the modeled data distribution i.e. the empirical covariance of the $(X_0^j)_j$
6. compute the empirical cross-covariance matrix $\hat{C} = \frac{1}{M}\sum_{j=1}^{M} X_0^j (X_N^j)^T$ between the modeled data distribution and the prior distribution i.e. the empirical cross-covariance between the $(X_0^j)_j$ and the $(X_N^j)_j$
7. compare $\hat{\Sigma}, \hat{C}$ to the ground truths Σ, C_σ (as given in Equation (18)) using the Frobenius norm

Note that steps 4 to 7 are performed for $n = 1$ to $n = L$ to see the improvement of DSB over each iteration.

For each family of test, we run the above protocol for $n_{\text{exp}} = 25$ randomly generated $p_{\text{data}},\ p_{\text{prior}}$ to average the results obtained in step 7.

Remark [Choice of M] In order for our empirical covariance matrices to be close to the actual covariance matrix of the DSB, we need to have enough samples. Ideally we want $M \to +\infty$ but generating paths is computationally expensive. We make a trade-off between precision and computational cost by taking $M = 250,000$ such that in the worst-case scenario when $d = 10$, we are using M data points to estimate a random variable of dimension d^2, and we know that the corresponding Monte Carlo estimator will have a relative error of order $\sqrt{\frac{d^2}{M}} = \sqrt{\frac{100}{250{,}000}} = 0.02$, which is sufficient in our case.

Remark [Implementation details] Following [3], we used an Adam optimizer with learning rate 10^{-3} and we trained for 10,000 epochs per IPF iteration, with a batch size of 128. We adapted the cache to our available GPU memory and set *cache_size* $= 10,000$ and *cache_period* $= 1000$. We trained on NVIDIA RTX 4000 GPUs with 20 GB of RAM each.

4.3 Results

Figure 4 to 10 present the results obtained.

The solid lines represent the mean error (computed over the n_{exp} experiments) as a function of the DSB iteration n, and the shaded areas depict the uncertainty over the estimated errors (±1std computed over the n_{exp} experiments). In all plots, the error increases with d and decreases with n, as expected. Likewise, the variance of the error tends to diminish with n, which makes sense because each IPF step brings us closer to convergence.

Note that we have added the graphs for $\hat{\Sigma'}$ as they gives us an estimate of the error term due to the fact that we are using a finite number of sample bridges $M = 250,000$ to compute our empirical covariance matrices. Thus, we cannot hope for the error on $\hat{\Sigma}$ and $\hat{C}$ to go below the noise threshold given by the graphs for $\hat{\Sigma'}$.

Let's now compare the results for each covariance type (*spherical*, *diagonal*, *general*) for $\hat{\Sigma}$ and $\hat{C}$.

1. For $\hat{\Sigma}$, we observe that the error follows the same decreasing pattern and the overall error levels follow the expected order of difficulty *spherical* < *diagonal* < *general*. We also note the increased variance in the spherical case, which could be explained by the fact that it's quite likely to get a high distance $|\sigma - \sigma'|$, in which case DSB may struggle more to learn the bridge. Finally, despite the cogent shape of the error curve, the final error values (i.e. for $n = L$) are still one order of magnitude higher than the noise threshold, which implies that the IPF algorithm used by DSB is still far from convergence.
2. For $\hat{C}$, the error again follows the same decreasing pattern and the overall error levels follow *spherical* < *diagonal* < *general*, as expected. The decrease in variance over DSB iterations is more marked than for $\hat{\Sigma}$, which makes sense because the variance of $\hat{C}$ depends on that of X_0 and X_N, but X_N is sampled directly from p_{prior} so there is not additional noise from this term. Hence we expect the variance of $\hat{C}$ to be somewhat "twice less" than that of $\hat{\Sigma}$.

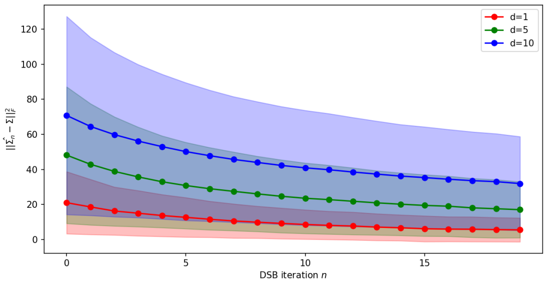

Figure 4: Error on $\hat{\Sigma}$ over DSB iterations in the spherical case.

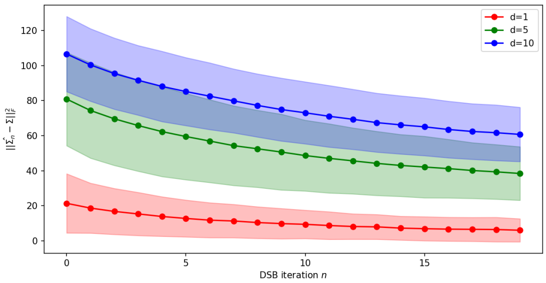

Figure 5: Error on $\hat{\Sigma}$ over DSB iterations in the diagonal case.

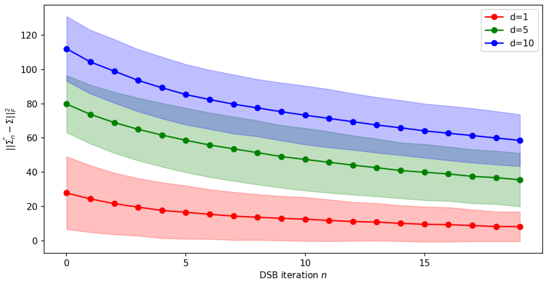

Figure 6: Error on $\hat{\Sigma}$ over DSB iterations in the general case.

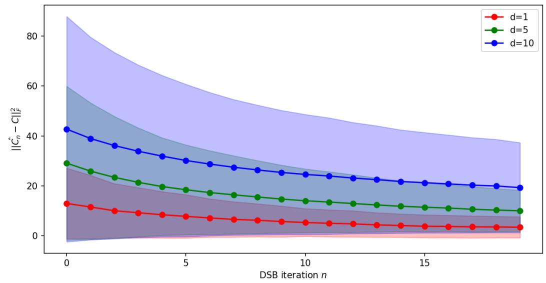

Figure 7: Error on $\hat{C}$ over DSB iterations in the spherical case.

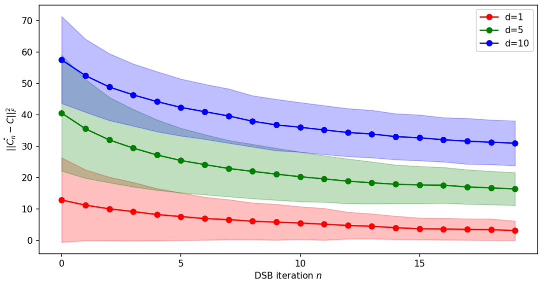

Figure 8: Error on $\hat{C}$ over DSB iterations in the diagonal case.

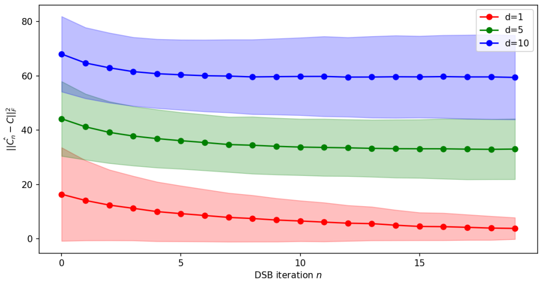

Figure 9: Error on $\hat{C}$ over DSB iterations in the general case.

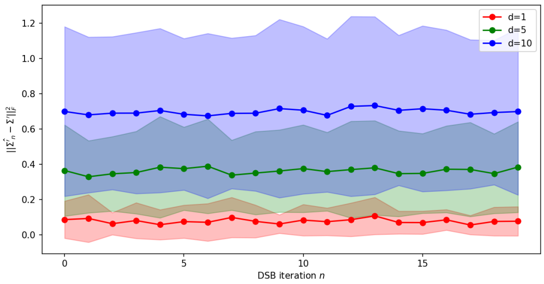

Figure 10: Error on $\hat{\Sigma}'$ over DSB iterations in the spherical case. The plots for the diagonal and general case are very similar and thus not displayed.

5 Conclusion

In this paper, we have presented the Schrödinger Bridge problem and its link to score-based generative modeling. After a mathematical introduction to the topic, we have implemented the DSB model proposed by [3]. Finally, our contribution was to leverage the availability of closed-form solutions to the Schrödinger Bridge problem in the Gaussian case to assess the performance of the DSB model in various settings by varying the dimension ($d \in \{1, 5, 10\}$) and the complexity (spherical, diagonal and general covariances for p_{data} and p_{prior}) of the problem. Our results confirm that the DSB model learns since the error on p_{data} decreases with the number of DSB iterations n. Besides, we observe empirically that DSB struggles more in higher dimensions, which follows intuition. Finally, and perhaps most interestingly, we see that after $L = 20$ DSB iterations (value used in [3]), the DSB model is still far from convergence as the error is an order of magnitude above the noise threshold. Crucially, this could mean that $L = 20$ amounts to under-training the DSB.

References

[1] Espen Bernton et al. *Schrödinger Bridge Samplers.* 2019. URL: `https://arxiv.org/abs/1912.13170`.

[2] Valentin De Bortoli. *Generative Modeling.* Accessed: 2024-06-14. 2023. URL: `https://vdeborto.github.io/project/generative_modeling/`.

[3] Valentin De Bortoli et al. *Diffusion Schrödinger Bridge with Applications to Score-Based Generative Modeling.* 2023. URL: `https://arxiv.org/abs/2106.01357`.

[4] Charlotte Bunne et al. *The Schrödinger Bridge between Gaussian Measures has a Closed Form.* 2023. URL: `https://arxiv.org/abs/2202.05722`.

[5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising Diffusion Probabilistic Models". In: (2020). URL: `https://arxiv.org/abs/2006.11239`.

[6] Bas Lemmens and Roger Nussbaum. *Birkhoff's version of Hilbert's metric and its applications in analysis.* 2013. arXiv: `1304.7921 [math.MG]`. URL: `https://arxiv.org/abs/1304.7921`.

[7] Christian Léonard. *A survey of the Schrödinger problem and some of its connections with optimal transport.* 2013. arXiv: `1308.0215 [math.PR]`. URL: `https://arxiv.org/abs/1308.0215`.

[8] Christian Léonard. *Some properties of path measures.* 2013. URL: `https://arxiv.org/abs/1308.0217`.

[9] Erwin Schrödinger. "Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique". In: *Ann. Inst. H. Poincaré* (1932).

[10] Yang Song et al. "Score-Based Generative Modeling through Stochastic Differential Equations". In: (2020). URL: `https://arxiv.org/abs/2011.13456`.

[11] Pascal Vincent. "A Connection Between Score Matching and Denoising Autoencoders". In: *Neural Computation* 23.7 (2011), pp. 1661–1674. DOI: `10.1162/NECO_a_00142`.