BM^2: Coupled Schrödinger Bridge Matching

Anonymous authors
Paper under double-blind review

Abstract

A Schrödinger bridge establishes a dynamic transport map between two target distributions via a reference process, simultaneously solving an associated entropic optimal transport problem. We consider the setting where samples from the target distributions are available, and the reference diffusion process admits tractable dynamics. We thus introduce Coupled Bridge Matching (BM^2), a simple *non-iterative* approach for learning Schrödinger bridges with neural networks. A preliminary theoretical analysis of the convergence properties of BM^2 is carried out, supported by numerical experiments that demonstrate the effectiveness of our proposal.

1 Introduction

The Schrödinger bridge problem seeks a process, the Schrödinger bridge, with prescribed initial and terminal distributions, such that the distribution of the Schrödinger bridge minimizes the Kullback-Leibler (KL) divergence to the distribution of a reference process. Schrödinger bridges play a central role in measure transport theory (Marzouk et al., 2016). Notably, it is known that the initial-terminal distribution of a Schrödinger bridge provides a solution to a corresponding entropic optimal transport problem (Peyré & Cuturi, 2020). Schrödinger bridges thus provide an effective framework for finding an alignment between samples from two target distributions. Furthermore, diffusion-based generative models (Ho et al., 2020; Song et al., 2021) can be interpreted as solving trivial instances of the Schrödinger bridge problem (Peluchetti, 2023). Consequently, Schrödinger bridges offer a more general approach to contemporary generative applications.

We consider the setting where samples are readily available from both target distributions, and where the reference process is a diffusion process solution to a stochastic differential equation (SDE). We thus introduce Coupled Bridge Matching (BM^2), a novel methodology aimed at computing the Schrödinger bridge given the reference SDE and samples from the two marginal distributions of interest. BM^2 builds upon Bridge Matching (BM), introduced[1] by Peluchetti (2021). Our approach advances recent contributions by Peluchetti (2023); Shi et al. (2023) by removing the need to solve a sequence of optimization problems. A neural network is employed to jointly learn a forward drift function and a backward drift function corresponding to the forward and backward dynamics of a Schrödinger bridge. BM^2 achieves several key desiderata:

(i) non-iterative: training is conducted through standard stochastic gradient descent within a single optimization loop;

(ii) exact: the idealized version of BM^2 yields the target Schrödinger bridge without approximations; the only sources of error involved in its practical implementation are the neural network approximation error and the discretization error due to sampling the learned SDE;

(iii) efficient gradient: the gradient of the loss function with respect to neural network parameters depends solely on few random variables sampled at the current optimization step;

(iv) simple loss: the loss function avoids derivative terms with respect to neural network inputs and does not impose hard constraints (such as conservative vector field requirements) on the neural network approximator.

---

[1] Peluchetti (2021) used the term "Diffusion Bridge Mixture-Matching Transport" (DBMT), but we follow Shi et al. (2023) in using the sleeker nomenclature "Bridge Matching" for this transport.

These features collectively enhance the efficiency and applicability of BM^2 in solving Schrödinger bridge problems. Training is robust, as it does not depend on hyperparameters that are typically challenging to set without time-consuming pilot runs, such as the number of training steps per optimization iteration (i) or the level of approximation (ii). Moreover, the memory requirements are modest due to (iii). Finally, the implementation is straightforward (i, iv), as illustrated in Algorithms 1 and 2 and in the annotated PyTorch code of Listing 1.

Content: This paper is structured as follows. In Section 2, we formally introduce the Schrödinger bridge problem with associated reference process dynamics. Section 3 reviews Bridge Matching, while Section 4 introduces Coupled Bridge Matching, discussing its theoretical properties and implementation aspects. Numerical experiments are presented in Section 5, followed by a discussion of related works in Section 6. Section 7 concludes the paper. For clarity, a more general formulation of BM^2 is deferred to Appendix A, all proofs to Appendix B, an additional numerical experiment to Appendix C, and code listings to Appendix D.

Notation and Assumptions: To enhance accessibility, we refrain from discussing the more technical aspects related to the Schrödinger bridge problem in its path measure formulation. The excellent treaties of Léonard (2014b;a) and Bortoli et al. (2021, Appendices D, H) already serve this goal. We denote distributions with uppercase letters and their corresponding (Lebesgue) densities with lowercase letters. All stochastic processes considered are d-dimensional, continuous, and defined on the unit time interval $[0,1]$. For a stochastic process X with distribution P (denoted $X \sim P$), we use subscripts to specify marginal distributions, joint distributions, and conditional distributions of P. P_t: marginal distribution of X_t at time t, with density p_t; $P_{0,1}$: initial-terminal joint distribution of (X_0, X_1); $P_{|0}$: distribution of X given its initial value X_0. Superscripts indicate a distribution P's dependency on another distribution Z, as in P^Z, or a sequence of distributions, as in $P^{(i)}$, $i \geq 1$. For a d-dimensional distribution Q_0, we define the stochastic process mixture distribution $Q_0 P_{|0}$ as: $(Q_0 P_{|0})(X \in \cdot) := \int P_{|0}(X \in \cdot | x_0) Q_0(dx_0)$. From a generative perspective, $X \sim Q_0 P_{|0}$ is obtained by sampling $X_0 \sim Q_0$ and then $X \sim P_{|0}(\cdot | X_0)$ conditionally on X_0. The marginal-conditional decomposition of P over its initial value is thus $P = P_0 P_{|0}$. Similarly, for a $d \times d$-dimensional joint distribution $Q_{0,1}$, we define $Q_{0,1} P_{|0,1}$ such that $X \sim Q_{0,1} P_{|0,1}$ is obtained by sampling $(X_0, X_1) \sim Q_{0,1}$ and then $X \sim P_{|0,1}(\cdot | X_0, X_1)$ conditionally on X_0 and X_1. Time is always indexed on a common forward timescale, on which all stochastic processes' distributions are defined. The dynamics of a diffusion process $X \sim P$ can be formulated in both forward and backward time directions, through corresponding forward and backward SDEs. In backward SDEs, t decreases from 1 to 0 (dt is negative), which is denoted by $t \in [1,0]$. All Brownian motions are independent. Unless otherwise noted, each diffusion process is a Markov diffusion process which is a (weak) solution to an associated SDE.

2 Problem Setting

2.1 Schrödinger Bridges and Entropic Optimal Transport

For two target d-dimensional distributions Ψ_0 and Ψ_1, and a reference stochastic process distribution R, the *dynamic* Schrödinger bridge (SB) problem seeks to find

$$S^{\Psi_0, \Psi_1, R} := \underset{P \in \mathcal{P}(\Psi_0, \Psi_1)}{\arg\min} \; \mathbb{KL}(P \parallel R), \tag{1}$$

where $\mathbb{KL}(\cdot \parallel \cdot)$ is the KL divergence and $\mathcal{P}(\Psi_0, \Psi_1)$ is the class of distributions of stochastic processes having initial distribution Ψ_0 and terminal distribution Ψ_1. We narrow down (1) to the case where R is the distribution of a diffusion process. In this case, under suitable conditions (Léonard, 2014b), (1) admits a unique solution which is also a diffusion process. From this point forward, Ψ_0, Ψ_1 and R are considered fixed. For brevity, we will thus denote the Schrödinger bridge $S^{\Psi_0, \Psi_1, R}$ simply as S, and apply the same notation convention to any distribution dependent on these variables.

The forward and backward dynamics of $X \sim S$ are given by:

$$X_0 \sim \Psi_0, \quad dX_t = \mu_s(X_t, t)dt + \sigma dW_t, \quad t \in [0,1], \tag{S}$$

$$X_1 \sim \Psi_1, \quad dX_t = -\upsilon_s(X_t, t)dt + \sigma dW_t, \quad t \in [1,0], \tag{$\overleftarrow{S}$}$$

for the SB-optimal drift functions μ_s, υ_s. These functions are related to the Schrödinger potentials (Léonard, 2014b) and are not analytically available aside from very specific choices of Ψ_0, Ψ_1 and R.

We assume that $R_{0,1}$ admits density $r_{0,1}$. Once S is obtained, the solution to the *static* Schrödinger bridge problem is given by $S_{0,1}$:

$$
\begin{aligned}
S_{0,1} &= \underset{C_{0,1} \in \mathcal{C}(\Psi_0, \Psi_1)}{\arg\min} \ \mathbb{KL}(C_{0,1} \parallel R_{0,1}), \\
&= \underset{C_{0,1} \in \mathcal{C}(\Psi_0, \Psi_1)}{\arg\min} \ \underset{C_{0,1}}{\mathbb{E}} \left[-\log r_{1|0}(X_1|X_0) \right] + \mathbb{H}(C_{0,1}).
\end{aligned}
\tag{2}
$$

In (2), $\mathcal{C}(\Psi_0, \Psi_1)$ denotes the class of $d \times d$-dimensional joint distributions with marginal distributions Ψ_0 and Ψ_1, commonly referred to as the class of couplings of Ψ_0 and Ψ_1, and $\mathbb{H}(C_{0,1}) := \mathbb{E}_{C_{0,1}}[\log c_{1,0}(X_1, X_0)]$ is the entropy of $C_{0,1}$.

The entropic optimal transport (EOT) solution for the cost function $k(x_0, x_1)$ and regularization level ε is given by:

$$
E_{0,1} := \underset{C_{0,1} \in \mathcal{C}(\Psi_0, \Psi_1)}{\arg\min} \ \underset{C_{0,1}}{\mathbb{E}} \left[\kappa(X_1, X_0) \right] + \varepsilon \mathbb{H}(C_{0,1}).
\tag{3}
$$

Thus, for each choice of $R_{0,1}$ in (2), $S_{0,1}$ solves a corresponding problem (3). As in the following, when R is associated to (R), $S_{0,1}$ solves the EOT problem (3) for the Euclidean cost $\kappa(x_1, x_0) = \frac{1}{2}\|x_0 - x_1\|^2$ and regularization level $\varepsilon = \sigma^2$.

We refer to Peyré & Cuturi (2020); Léonard (2014b); Gushchin et al. (2023) for related background material from complementary perspectives.

2.2 Reference Dynamics

We focus on the case where R is the distribution of a scaled Brownian motion:

$$
X_0 \sim \Psi_0, \quad dX_t = \sigma dW_t, \quad t \in [0, 1],
\tag{R}
$$

with $\sigma > 0$. Our approach *is not limited* to the choice of SDE (R), BM² readily extends to the broader class of reference SDEs examined in Peluchetti (2023). The main requirement for the applicability of BM² is the analytical availability of (4, 5) for the chosen reference SDE. We address the case, commonly employed in generative applications, of $dX_t = \sigma\sqrt{\beta_t}dW_t$ for a schedule β_t explicitly in Appendix A, and refer the reader to Peluchetti (2021; 2023) for the general setting. As our developments are orthogonal to the specific choice of reference process, we focus on the simplest case for explanatory reasons.

We collect here various results concerning (R) that will be utilized in the following:

$$
R_{t|0}(\cdot|x_0) = \mathcal{N}(x_0, \sigma^2 t),
\tag{4}
$$

$$
R_{t|0,1}(\cdot|x_0, x_1) = \mathcal{N}(x_0(1-t) + x_1 t, \sigma^2 t(1-t)),
\tag{5}
$$

$$
\mu_{01}(x_t, t, x_1) := \sigma^2 \nabla_{x_t} \log r_{1|t}(x_1|x_t) = \frac{x_1 - x_t}{1 - t},
\tag{6}
$$

$$
\upsilon_{01}(x_t, t, x_0) := \sigma^2 \nabla_{x_t} \log r_{t|0}(x_t|x_0) = \frac{x_0 - x_t}{t}.
\tag{7}
$$

Conditioning $X \sim R$ on the endpoints $X_0 = x_0$, $X_1 = x_1$ results in the diffusion bridge distribution $R_{|0,1}$, with associated forward and backward SDEs:

$$
X_0 = x_0, \quad dX_t = \mu_{01}(X_t, t, x_1)dt + \sigma dW_t, \quad t \in [0, 1],
\tag{$R_{|0,1}$}
$$

$$
X_1 = x_1, \quad dX_t = -\upsilon_{01}(X_t, t, x_0)dt + \sigma dW_t, \quad t \in [1, 0].
\tag{$\overleftarrow{R_{|0,1}}$}
$$

3 Bridge Matching (BM)

We succinctly review Bridge Matching, and refer to Peluchetti (2021; 2023); Shi et al. (2023) for more details. BM takes as input a joint distribution $Q_{0,1}$ with marginal distributions Q_0, Q_1 and a SDE, (R). Firstly, a

stochastic process $\Pi^{Q_{0,1}}$ is constructed as a mixture of diffusion bridges ($R_{|0,1}$), such that the endpoints (X_0, X_1) of $X \sim \Pi^{Q_{0,1}}$ are distributed according to $Q_{0,1}$. This process, which is a mixture of diffusion processes, is not itself a diffusion process in general (Jamison, 1974). However, we can obtain a marginal-matching diffusion process with distribution $M^{Q_{0,1}}$ for which $M_t^{Q_{0,1}} = \Pi_t^{Q_{0,1}}, 0 \leq t \leq 1$. Consequently, $X \sim M^{Q_{0,1}}$ is a diffusion process for which $X_0 \sim Q_0$ and $X_1 \sim Q_1$, i.e. it defines a dynamic transport from Q_0 to Q_1.

Concretely, let $\Pi^{Q_{0,1}} := Q_{0,1} R_{|0,1}$. The BM transport based on $Q_{0,1}$ with distribution $M^{Q_{0,1}}$ is realized by

$$X_0 \sim Q_0, \quad dX_t = \underbrace{\mu_m^{Q_{0,1}}(X_t, t)}_{\mathbb{E}_{\Pi^{Q_{0,1}}}[\mu_{01}(X_t, t, X_1)|X_t]} dt + \sigma dW_t, \quad t \in [0, 1], \qquad (M)$$

$$X_1 \sim Q_1, \quad dX_t = -\underbrace{\upsilon_m^{Q_{0,1}}(X_t, t)}_{\mathbb{E}_{\Pi^{Q_{0,1}}}[\upsilon_{01}(X_t, t, X_0)|X_t]} dt + \sigma dW_t, \quad t \in [1, 0], \qquad (\overleftarrow{M})$$

and satisfies $M_t^{Q_{0,1}} = \Pi_t^{Q_{0,1}}, 0 \leq t \leq 1$.

As conditional expectations are mean squared error minimizers, suitable training objectives for the drift functions $\mu_m^{Q_{0,1}}$ and $\upsilon_m^{Q_{0,1}}$ are derived from

$$\mu_m^{Q_{0,1}} = \arg\min_{\mu} \mathbb{E}_{\Pi^{Q_{0,1}}} \left[\frac{1}{2} \int_0^1 \|\mu_{01}(X_t, t, X_1) - \mu(X_t, t)\|^2 dt \right], \qquad (8)$$

$$\upsilon_m^{Q_{0,1}} = \arg\min_{\upsilon} \mathbb{E}_{\Pi^{Q_{0,1}}} \left[\frac{1}{2} \int_0^1 \|\upsilon_{01}(X_t, t, X_0) - \upsilon(X_t, t)\|^2 dt \right], \qquad (9)$$

by replacing each integral with an expectation over uniform time $t \sim \mathcal{U}(0, 1)$, and then approximating both expectations with Monte Carlo estimators. While we will rely exclusively on (8, 9) in the experiments of Section 5, $\mu_m^{Q_{0,1}}$ and $\upsilon_m^{Q_{0,1}}$ can be inferred from paths $X \sim \Pi^{Q_{0,1}}$ also by performing maximum likelihood estimation or by employing a drift matching estimator (Liu et al., 2022; Peluchetti, 2023).

We conclude this section by reviewing prior BM results relevant for BM^2. Define:

$\mathcal{P} := \{d\text{-dimensional, continuous, stochastic processes on } [0, 1]\},$

$\mathcal{R} := \{P \in \mathcal{P} \mid P = P_{0,1} R_{|0,1} = \Pi^{P_{0,1}} \text{ for some } P_{0,1}\},$

$\mathcal{M} := \{P \in \mathcal{P} \mid P \text{ is a (Markov) diffusion process}\},$

$\mathcal{S} := \{P \in \mathcal{P} \mid P \text{ is a Schrödinger bridge for some target marginal distributions}\} = \mathcal{R} \cap \mathcal{M},$

where the equivalence is established by Jamison (1975) under appropriate assumptions. We additionally define the following restrictions: $\mathcal{P}(\Psi_0, \cdot) := \{P \in \mathcal{P} \mid P_0 = \Psi_0\}$, $\mathcal{P}(\cdot, \Psi_1) := \{P \in \mathcal{P} \mid P_1 = \Psi_1\}$, $\mathcal{P}(\Psi_0, \Psi_1) := \{P \in \mathcal{P} \mid P_0 = \Psi_0 \text{ and } P_1 = \Psi_1\}$. Restrictions to $\mathcal{R}, \mathcal{M}, \mathcal{S}$ and $\mathcal{C}$ employ the same notation.

For $Q \in \mathcal{P}$, it is instructive to view BM as a map between distributions defined by the composition of two projections: $Q \overset{\mathcal{R}p}{\to} \Pi^{Q_{0,1}} \overset{\mathcal{M}p}{\to} M^{Q_{0,1}}$. Here, the *reciprocal projection* $\mathcal{R}p : \mathcal{P} \to \mathcal{R}$ projects Q onto the reciprocal class $\mathcal{R}$, while the *Markovian projection* $\mathcal{M}p : \mathcal{R} \to \mathcal{M}$ projects $\Pi^{Q_{0,1}}$ onto the class of diffusion processes, see Shi et al. (2023). It follows that if $P \in \mathcal{R}$, then $P = \mathcal{R}p(P)$, and if $P \in \mathcal{M}$, then $P = \mathcal{M}p(P)$. Consequently, if $P \in \mathcal{S}$, $P = (\mathcal{M}p \circ \mathcal{R}p)(P)$ for the BM map $(\mathcal{M}p \circ \mathcal{R}p)$, and conversely if $P = (\mathcal{M}p \circ \mathcal{R}p)(P)$ then $P \in \mathcal{S}$.

3.1 Iterated Bridge Matching (I-BM) and Diffusion Iterative Proportional Fitting (DIPF)

In the dynamic setting, Peluchetti (2023); Shi et al. (2023) demonstrate that, under suitable conditions, iterative application of the BM procedure to an initial coupling $C_{0,1} \in \mathcal{C}(\Psi_0, \Psi_1)$ results in convergence toward S. Specifically, defining $I^{(0)} := M^{C_{0,1}}$ and $I^{(i)} := M^{I_{0,1}^{(i-1)}}$ for $i \geq 1$, it holds that $\mathbb{KL}(I^{(i)} \parallel S) \to 0$ as $i \to \infty$. In practical applications, the independent initial coupling given by the product distribution $C_{0,1} = \Psi_0 \otimes \Psi_1$ is frequently employed.

In the static setting, the classical procedure employed in solving problems (2, 3) is known by several names: the Sinkhorn algorithm (Peyré & Cuturi, 2020), the Iterated Proportional Fitting (IPF) procedure (Ruschendorf, 1995), or Fortet iterations (Fortet, 1940). The iterates are given by $D_{0,1}^{(0)} := \Psi_0 R_{1|0}$, $D_{0,1}^{(1)} := \Psi_1 K_{0|1}^{(0)}$, $D_{0,1}^{(2)} := \Psi_0 D_{1|0}^{(1)}$, and so on. At each iteration, one of the target marginal distributions is replaced while the remaining conditional distribution is kept fixed. Alternatively, one can start from $\Psi_1 R_{0|1}$ following the same logic. Under suitable conditions (Ruschendorf, 1995), KL convergence $\mathbb{KL}(S_{0,1} \parallel D_{0,1}^{(i)}) \to 0$ is established. The key insight of Bortoli et al. (2021); Vargas et al. (2021) is that it is possible to extend the IPF iterations to the dynamic setting. In this case, the IPF iterations are implemented by learning the time reversal of a diffusion process at each iteration. We refer to the resulting training algorithm, as proposed by Bortoli et al. (2021), as Diffusion Iterative Proportional Fitting (DIPF). Bortoli et al. (2021) establishes the convergence properties of the DIPF iterates, see their Propositions 4, 5 and Section 3.5.

4 Coupled BM (BM^2)

As a starting point for the derivation of BM^2, consider the system of equations

$$
\begin{cases}
H^{K'_{0,1}} = \Psi_0 M_{|0}^{K'_{0,1}} \\
K^{H'_{0,1}} = \Psi_1 M_{|1}^{H'_{0,1}}
\end{cases}, \tag{10}
$$

whose variables are diffusion distributions $H^{K'_{0,1}}, K^{H'_{0,1}}$ and H', K'. That is, $H^{K'_{0,1}}$ is obtained as the BM transport based on $K'_{0,1}$ conditioned to have initial distribution Ψ_0, while $K^{H'_{0,1}}$ is obtained as the BM transport based on $H'_{0,1}$ conditioned to have terminal distribution Ψ_1. Equivalently, (10) is expressed as

$$X_0 \sim \Psi_0, \quad dX_t = \mu_m^{K'_{0,1}}(X_t, t)dt + \sigma dW_t, \quad t \in [0,1], \qquad (H^{K'_{0,1}})$$

$$X_1 \sim \Psi_1, \quad dX_t = -v_m^{H'_{0,1}}(X_t, t)dt + \sigma dW_t, \quad t \in [1,0]. \qquad (\overleftarrow{K}^{H'_{0,1}})$$

All of μ_m, v_m, M are defined in Section 3. System (10) defines an update step $(H', K') \overset{(10)}{\to} (H^{K'_{0,1}}, K^{H'_{0,1}})$. We are interested in the fixed points of such updates, i.e. (H', K') such that $(H', K') \overset{(10)}{\to} (H', K')$. It holds that $H' = K' = S$ is a fixed point to (10). As $S \in \mathcal{S}(\Psi_0, \Psi_1)$, $S = \Pi^{S_{0,1}} = M^{S_{0,1}}$, see the review at the end of Section 3. Consequently, $\Psi_0 M_{|0}^{S_{0,1}} = \Psi_0 S_{|0} = S$ and $\Psi_1 M_{|1}^{S_{0,1}} = \Psi_1 S_{|1} = S$. In this case, the SB-optimal drifts μ_s and v_s of $(S, \overleftarrow{S})$ respectively replace $\mu_m^{K'_{0,1}}$ and $v_m^{H'_{0,1}}$ in $(H^{K'_{0,1}}, \overleftarrow{K}^{H'_{0,1}})$. Under the additional assumption that $H' = K'$, or equivalently that $(H^{K'_{0,1}}, \overleftarrow{K}^{H'_{0,1}})$ are the time reversal of each other, this fixed point is unique. Let $G = H' = K'$, we have $G = \Psi_0 M_{|0}^{G_{0,1}} = G_0 M_{|0}^{G_{0,1}} = M_0^{G_{0,1}} M_{|0}^{G_{0,1}} = M^{G_{0,1}}$ and $G_0 = \Psi_0, G_1 = \Psi_1$, thus $G = \mathcal{S}(\Psi_0, \Psi_1) = S$. We have shown the following:

Lemma 1 (Fixed points of (10)). *Under suitable conditions (Léonard, 2014a), the updates $(H', K') \overset{(10)}{\to} (H^{K'_{0,1}}, K^{H'_{0,1}})$, parametrized by diffusion process distributions, admit $H' = K' = S$ as fixed point. If $H' = K'$, this fixed point is unique.*

When $\mu_m^{K'_{0,1}} = \mu_s$ and $v_m^{H'_{0,1}} = v_s$, (10) has reached an equilibrium. The updates $(H', K') \overset{(10)}{\to} (H^{K'_{0,1}}, K^{H'_{0,1}})$ are realized through the computation of the drifts $\mu_m^{K'_{0,1}}$ and $v_m^{H'_{0,1}}$, i.e. by minimizing the losses (8, 9), where $Q_{0,1}$ is respectively equal to $K'_{0,1}$ and $H'_{0,1}$. Our proposal, BM^2, follows from replacing the complete minimization of (8, 9) with partial and stochastic minimization of (8, 9) through stochastic gradient descent. More precisely, consider the forward and backward SDEs with distributions $F(\theta)$ and $B(\theta)$:

$$X_0 \sim \Psi_0, \quad dX_t = \mu_f(X_t, t, \theta)dt + \sigma dW_t, \quad t \in [0,1], \qquad (F(\theta))$$

$$X_1 \sim \Psi_1, \quad dX_t = -v_b(X_t, t, \theta)dt + \sigma dW_t, \quad t \in [1,0]. \qquad (\overleftarrow{B}(\theta))$$

$\mu_f(X_t, t, \theta)$ and $v_b(X_t, t, \theta)$ are drift functions to be learned, which are implemented through a neural network with parameters θ. Let θ' represent the values of θ at a given step during training, and define the losses

$$
\begin{aligned}
\mathbb{L}_f(\theta; \theta') &\coloneqq \underset{\Pi^{B_{0,1}(\theta')}}{\mathbb{E}} \left[\frac{1}{2} \int_0^1 \|\mu_{01}(X_t, t, X_1) - \mu_f(X_t, t, \theta)\|^2 dt \right], \\
\mathbb{L}_b(\theta; \theta') &\coloneqq \underset{\Pi^{F_{0,1}(\theta')}}{\mathbb{E}} \left[\frac{1}{2} \int_0^1 \|v_{01}(X_t, t, X_1) - v_b(X_t, t, \theta)\|^2 dt \right], \\
\mathbb{L}(\theta; \theta') &\coloneqq \mathbb{L}_f(\theta; \theta') + \mathbb{L}_b(\theta; \theta').
\end{aligned}
\tag{11}
$$

At each optimization step, BM^2 attempts to minimize $\mathbb{L}(\theta; \theta')$ in θ via a step of stochastic gradient descent, starting from $\theta = \theta'$ and keeping θ' fixed, resulting in θ''. The subsequent optimization step employs $\theta' \leftarrow \theta''$. The complete training objective is presented in Algorithm 1, where sg() refers to the stop-gradient operator — $\mathbb{L}(\theta; \theta')$ is minimized in the first arguments only — and discretize() represents a generic SDE discretization scheme. For completeness, we outline the standard SGD training loop in Algorithm 2, where sgdstep() refers to an update step via a generic gradient descent optimizer.

It should be noted that merely performing coupled drift matching of F and B, wherein F learns the drift consistent with paths from B and vice versa, does not yield the Schrödinger bridge as a fixed point (Bortoli et al., 2021). The introduction of the mixing process Π is crucial in ensuring this property. Moreover, $\mathbb{L}(\theta; \theta')$ must be minimized only with respect to its first argument: the application of the stop-gradient operator sg() is not an efficiency consideration but a necessary component.

Algorithm 1 BM^2 — training loss computation

outputs: $\mathtt{l}(\theta)$: sampled loss value
inputs: θ: current parameters

1: **function** loss(θ)
2:     $\mathtt{f}_0 \sim \Psi_0$ ▷ Marginal sampling
3:     $\mathtt{f}_{\Delta t}, \dots, \mathtt{f}_1 | \mathtt{f}_0 \sim \mathrm{sg}(\mathrm{discretize}(\mathtt{f}_0, \Delta t, \mu_f(\cdot, \cdot, \theta)))$ ▷ Discretization of $(F(\theta))$
4:     $\mathtt{b}_1 \sim \Psi_1$ ▷ Marginal sampling
5:     $\mathtt{b}_{1-\Delta t}, \dots, \mathtt{b}_0 | \mathtt{b}_1 \sim \mathrm{sg}(\mathrm{discretize}(\mathtt{b}_1, \Delta t, v_b(\cdot, \cdot, \theta)))$ ▷ Discretization of $(\overleftarrow{B}(\theta))$
6:     $\mathtt{t} \sim \mathcal{U}(0, 1)$ ▷ Time sampling
7:     $\pi\mathtt{f}_\mathtt{t} \sim R_{\mathtt{t}|0,1}(\cdot | \mathtt{f}_0, \mathtt{f}_1)$ ▷ Bridge sampling (5)
8:     $\pi\mathtt{b}_\mathtt{t} \sim R_{\mathtt{t}|0,1}(\cdot | \mathtt{b}_0, \mathtt{b}_1)$ ▷ Bridge sampling (5)
9:     $\mathtt{l}_\mathtt{f}(\theta) \leftarrow {}^1\!/_2 \|\mu_{01}(\pi\mathtt{b}_\mathtt{t}, \mathtt{t}, \mathtt{b}_1) - \mu_f(\pi\mathtt{b}_\mathtt{t}, \mathtt{t}, \theta)\|^2$ ▷ BM based on $B_{0,1}$ (6, 8)
10:    $\mathtt{l}_\mathtt{b}(\theta) \leftarrow {}^1\!/_2 \|v_{01}(\pi\mathtt{f}_\mathtt{t}, \mathtt{t}, \mathtt{f}_0) - v_b(\pi\mathtt{f}_\mathtt{t}, \mathtt{t}, \theta)\|^2$ ▷ BM based on $F_{0,1}$ (7, 9)
11:    $\mathtt{l}(\theta) \leftarrow \mathtt{l}_\mathtt{f}(\theta) + \mathtt{l}_\mathtt{b}(\theta)$
12:    **return** $\mathtt{l}(\theta)$

Algorithm 2 BM^2 — training loop

outputs: θ^*: trained parameters
inputs: θ°: initial parameters

1: **function** train(θ°)
2:     $\theta \leftarrow \theta^\circ$
3:     **while** not converged **do**
4:        $\mathtt{l}(\theta) \leftarrow \mathrm{loss}(\theta)$ ▷ Sample loss with Algorithm 1
5:        $\theta \leftarrow \mathrm{sgdstep}(\theta, \nabla_\theta \mathtt{l}(\theta))$ ▷ Perform SGD step
6:     **return** θ

4.1 Implementation Aspects

The following aspects are not presented in Listing 1, but are taken care of in the codebase utilized to carry out the experiments of Section 5.

Path Caching: as in Bortoli et al. (2021); Shi et al. (2023), to enhance efficiency, we cache the initial and terminal endpoints of the paths sampled in lines 3 and 5 of Algorithm 1, and periodically refresh the cache during training. Notably, it is unnecessary to cache entire paths; only the endpoints are required for bridge sampling, which is advantageous from a memory perspective. Bridge sampling offers the additional benefit of increased sample diversity: for cached (fixed) endpoints, the samples corresponding to lines 7 and 8 differ at each step.

Model: we utilize a single neural network to parametrize both $\mu_f(x, t, \theta)$ and $v_b(x, t, \theta)$. As the training process is not iterative, it is unnecessary to introduce multiple neural networks (or parameters), one for each iteration.

Sampling EMA: as in Ho et al. (2020); Song et al. (2021), to improve the stability of training we apply the Exponential Moving Averaging (EMA) to the parameters employed in path sampling in lines 3 and 5 of Algorithm 1.

Loss Singularities: the losses of lines 9 and 10 of Algorithm 1 diverge for $\mathtt{t} \to 1$ and $\mathtt{t} \to 0$ respectively. Singularities of these kind are common to scalable losses for generative diffusion models. In our numerical experiments we simply restrict sampling of t to $\mathcal{U}(\epsilon, 1 - \epsilon)$ for a small $\epsilon > 0$. More sophisticated alternatives involve either employing the dynamics of Appendix A for an appropriate scheduling β_t, or learning terminal-value predictors in place of drift terms, recovering the latter through (6, 7).

4.2 Convergence Properties

At each training step, BM² performs a partial and stochastic minimization of the loss $\mathbb{L}(\theta; \theta')$ from (11) with respect to θ, where $\mathbb{L}(\theta; \theta')$ is defined by an expectation over a distribution dependent on θ', yielding θ''. Subsequently, θ' is updated to match θ'', and the process advances to the next training step. The alternation between expectation and maximization steps bears resemblance to the classical Expectation-Maximization (EM) algorithm (Dempster et al., 1977).

4.2.1 Complete Minimization

We start by establishing in Theorem 1 that the version of BM² where $\mathbb{L}(\theta; \theta')$ is fully minimized at each training step recovers the I-BM and DIPF iterations for two specific initialization choices of $(F(\theta), \overleftarrow{B}(\theta))$. The prior convergence results of Bortoli et al. (2021); Shi et al. (2023); Peluchetti (2023) (see the review of Section 3.1) toward S thus apply.

To facilitate the presentation of the convergence results in this section, we introduce, with a slight abuse of notation, the following functional versions of the losses (11):

$$
\begin{aligned}
\mathbb{L}_f(\mu_f; v_b') &:= \underset{\Pi^{B'_{0,1}}}{\mathbb{E}} \left[\frac{1}{2} \int_0^1 \|\mu_{01}(X_t, t, X_1) - \mu_f(X_t, t)\|^2 dt \right], \\
\mathbb{L}_b(v_b; \mu_f') &:= \underset{\Pi^{F'_{0,1}}}{\mathbb{E}} \left[\frac{1}{2} \int_0^1 \|v_{01}(X_t, t, X_1) - v_b(X_t, t)\|^2 dt \right], \\
\mathbb{L}(\mu_f, v_b; \mu_f', v_b') &:= \mathbb{L}_f(\mu_f; v_b') + \mathbb{L}_b(v_b; \mu_f').
\end{aligned}
\tag{12}
$$

In (12) we identify μ_f, v_b with F, B, and μ_f', v_b' with F', B' (the remaining quantities defining F, B, F', B' are fixed). We will use $\mathbb{L}_f(\mu_f; v_b')$, $\mathbb{L}_b(v_b; \mu_f')$ and $\mathbb{L}_f(\theta; \theta')$, $\mathbb{L}_b(\theta; \theta')$ interchangeably. We are now ready to state our first convergence result.

Theorem 1 (Complete BM² Iterations). *Consider the SDEs $(F(\theta), \overleftarrow{B}(\theta))$, with initial drifts $\mu_f^{(0)}, v_b^{(0)}$ and corresponding distributions $F^{(0)}, B^{(0)}$. For each $i \geq 1$, let $(\mu_f^{(i)}, v_b^{(i)}) = \arg\min_{(\mu,v)} \mathbb{L}(\mu, v; \mu_f^{(i-1)}, v_b^{(i-1)})$, resulting in the distribution iterates $F^{(i)}, B^{(i)}$. We distinguish two cases:*

(i) $\mu_f^{(0)} = v_b^{(0)} = 0$: both the iterates $F^{(0)}, B^{(1)}, F^{(2)}, \dots$ and the iterates $B^{(0)}, F^{(1)}, B^{(2)}, \dots$ are equivalent to the DIPF iterates, started respectively from the forward and from the backward time direction;

(ii) $\mu_f^{(0)} = \mu_m^{C_{0,1}}, \upsilon_b^{(0)} = \upsilon_m^{C_{0,1}}$ *for some* $C \in \mathcal{C}(\Psi_0, \Psi_1)$: $F^{(i)} = B^{(i)} = I^{(i)}$ *for each* $i \geq 0$ *where* $I^{(i)}$ *are the I-BM iterates.*

4.2.2 Partial Minimization

In the EM algorithm it suffices to perform partial maximization steps. A partial result for the setting where $\mathbb{L}(\theta; \theta')$ is partially minimized with respect to θ at each step is stated in Theorem 2, which is based on Lemma 1 and Lemma 2.

Lemma 2 (Loss Interpretation). *It holds that*

$$
\begin{aligned}
\mathbb{KL}(B_0 \parallel \Psi_0) + \mathbb{L}_f(\mu_f; \upsilon_b) &= \mathbb{KL}(\Pi^{B_{0,1}} \parallel F) + C_1(B) = \mathbb{KL}(M^{B_{0,1}} \parallel F) + C_2(B), \\
\mathbb{KL}(F_1 \parallel \Psi_1) + \mathbb{L}_b(\upsilon_b; \mu_f) &= \mathbb{KL}(\Pi^{F_{0,1}} \parallel B) + D_1(F) = \mathbb{KL}(M^{F_{0,1}} \parallel B) + D_2(F),
\end{aligned}
\tag{13}
$$

for $C_1(B), C_2(B)$ independent of F, $D_1(F), D_2(F)$ independent of B, with $0 \leq C_1(B) \leq C_2(B)$ and $0 \leq D_1(F) \leq D_2(F)$.

The losses $\mathbb{L}_f(\mu_f; \upsilon_b)$ and $\mathbb{L}_b(\upsilon_b; \mu_f)$ are easily amenable to optimization in their first arguments, as seen in Algorithm 1. Lemma 2 relates these losses to more interpretable KL divergences between distributions. By (13), a decrease of $\mathbb{L}_f(\mu_f; \upsilon_b)$ due to a change in μ_f corresponds to equivalent decreases of $\mathbb{KL}(M^{B_{0,1}} \parallel F)$ for a fixed υ_b, or B. Thus, partial minimization of $\mathbb{L}_f(\mu_f; \upsilon_b)$ brings F closer to $M^{B_{0,1}}$, the BM transport based on $B_{0,1}$, and the result of a complete minimization step, by means of reverse KL minimization. Symmetric considerations apply to $\mathbb{L}_b(\upsilon_b; \mu_f)$ as function of its first argument. Putting this result and Lemma 1 together yields Theorem 2.

Theorem 2 (Partial BM² Iterations). *At each optimization step, decreases of $\mathbb{L}_f(\theta; \theta')$ and $\mathbb{L}_b(\theta; \theta')$ in θ correspond to equivalent decreases of $\mathbb{KL}(M^{B_{0,1}(\theta')} \parallel F(\theta))$ and $\mathbb{KL}(M^{F_{0,1}(\theta')} \parallel B(\theta))$. If the losses $\mathbb{L}_f(\theta; \theta')$ and $\mathbb{L}_b(\theta; \theta')$ cannot be decreased in θ, i.e., at optimality, and if $F(\theta) = B(\theta)$, then $F(\theta) = B(\theta) = S$.*

4.2.3 Infinitesimal Minimization

We conclude our theoretical investigation by relating our proposal to the work of Karimi et al. (2023), which introduces a continuous variant of the IPF procedure. In IPF, the two target marginal distributions are replaced sequentially, one at a time. Each step corresponds to solving a static Schrödinger *half-bridge* problem (Léonard, 2014a), where in (2), $\mathcal{C}(\Psi_0, \Psi_1)$ is replaced by either $\mathcal{C}(\Psi_0, \cdot)$ or $\mathcal{C}(\cdot, \Psi_1)$. The approach proposed by Karimi et al. (2023) retains either the even or odd steps of the IPF scheme while substituting the alternate steps with partial minimizations of the corresponding half-bridge problems. In the limit of infinitesimally small improvements, this yields a dynamical system for the evolution of the iterates over continuous algorithmic time.

We demonstrate that a similar result can be obtained for a modified version of BM², where forward KL divergences are minimized instead of reverse KL divergences. The resulting dynamical system is a symmetrized version of the one obtained by Karimi et al. (2023). Let F', B' represent the current state in the optimization process. We consider a partial minimization of $\mathbb{KL}(F \parallel M^{B'_{0,1}})$, instead of $\mathbb{KL}(M^{B'_{0,1}} \parallel F)$, in F and a partial minimization of $\mathbb{KL}(B \parallel M^{F'_{0,1}})$, instead of $\mathbb{KL}(M^{F'_{0,1}} \parallel B)$, in B. As in Karimi et al. (2023), partial minimization is formulated as

$$
\begin{aligned}
F^{(\lambda)} &:= \underset{F \in \mathcal{M}(\Psi_0, \cdot)}{\arg\min} \; \lambda \mathbb{KL}(F \parallel M^{B'_{0,1}}) + (1 - \lambda) \mathbb{KL}(F \parallel F'), \\
B^{(\lambda)} &:= \underset{B \in \mathcal{M}(\cdot, \Psi_1)}{\arg\min} \; \lambda \mathbb{KL}(B \parallel M^{F'_{0,1}}) + (1 - \lambda) \mathbb{KL}(B \parallel B'),
\end{aligned}
\tag{14}
$$

where $\lambda \in [0, 1]$ controls the extent of the minimization. We begin by establishing two stability results: the updates $(F', B') \overset{(14)}{\to} (F^{(\lambda)}, B^{(\lambda)})$ preserve both $\mathcal{R}$ and $\mathcal{S}$.

Lemma 3 ($\mathcal{R}$-stability of $F^{(\lambda)}, B^{(\lambda)}$). *If $F', B' \in \mathcal{R}$, then $F^{(\lambda)}, B^{(\lambda)} \in \mathcal{R}$ for each $\lambda \in [0, 1]$.*

Lemma 4 ($\mathcal{S}$-stability of $F^{(\lambda)}, B^{(\lambda)}$). *If $F', B' \in \mathcal{S}$, then $F^{(\lambda)}, B^{(\lambda)} \in \mathcal{S}$ for each $\lambda \in [0, 1]$.*

Provided that the initial values $F', B' \in \mathcal{S}$, Lemma 4 establishes that the iterates defined by the updates $(F', B') \overset{(14)}{\to} (F^{(\lambda)}, B^{(\lambda)})$ always remain in $\mathcal{S}$. It is straightforward to ensure that $F', B' \in \mathcal{S}$ at initialization by setting the corresponding drifts to zero: $\mu'_f, v'_b = 0$, which we will assume henceforth. As $M^{B'_{0,1}} = B'$ and $M^{F'_{0,1}} = F'$, (14) can be reformulated in simpler terms:

$$
\begin{aligned}
F^{(\lambda)} &:= \underset{F \in \mathcal{M}(\Psi_0, \cdot)}{\arg\min} \ \lambda \mathbb{KL}(F \parallel B') + (1 - \lambda)\mathbb{KL}(F \parallel F'), \\
B^{(\lambda)} &:= \underset{B \in \mathcal{M}(\cdot, \Psi_1)}{\arg\min} \ \lambda \mathbb{KL}(B \parallel F') + (1 - \lambda)\mathbb{KL}(B \parallel B').
\end{aligned}
\tag{15}
$$

By Lemma 3, it suffices to solve (15) in the static setting,

$$
\begin{aligned}
F^{(\lambda)}_{0,1} &:= \underset{F_{0,1} \in \mathcal{C}(\Psi_0, \cdot)}{\arg\min} \ \lambda \mathbb{KL}(F_{0,1} \parallel B'_{0,1}) + (1 - \lambda)\mathbb{KL}(F_{0,1} \parallel F'_{0,1}), \\
B^{(\lambda)}_{0,1} &:= \underset{B_{0,1} \in \mathcal{C}(\cdot, \Psi_1)}{\arg\min} \ \lambda \mathbb{KL}(B_{0,1} \parallel F'_{0,1}) + (1 - \lambda)\mathbb{KL}(B_{0,1} \parallel B'_{0,1}).
\end{aligned}
\tag{16}
$$

The dynamic solutions are then recovered by $F^{(\lambda)}_{|0,1} = B^{(\lambda)}_{|0,1} = R_{|0,1}$.

We assume that $F'_{0,1}, B'_{0,1}, \Psi_0, \Psi_1$ admit densities. By calculus of variations, the solution to (16) is given by $f^{(\lambda)}_{0,1}(x_0, x_1) = \psi_0(x_0) f^{(\lambda)}_{1|0}(x_1|x_0)$, and $b^{(\lambda)}_{0,1}(x_0, x_1) = b^{(\lambda)}_{0|1}(x_0|x_1) \psi_1(x_1)$, where $f^{(\lambda)}_{1|0}(x_1|x_0) \propto b'_{1|0}(x_1|x_0)^\lambda f'_{1|0}(x_1|x_0)^{1-\lambda}$ and $b^{(\lambda)}_{0|1}(x_0|x_1) \propto f'_{0|1}(x_0|x_1)^\lambda b'_{0|1}(x_0|x_1)^{1-\lambda}$. The IPF iterations are recovered when $\lambda = 1$. Instead, taking the limit $\lambda \to 0$ and applying Bayes theorem, we obtain the evolution of $\log f^{(l)}_{1|0}(x_1|x_0)$ and $\log b^{(l)}_{0|1}(x_0|x_1)$ as a function of algorithmic time $l \in [0, \infty)$ through the dynamical system

$$
\begin{aligned}
\frac{d \log f^{(l)}_{1|0}(x_1|x_0)}{dl} &= -\log \frac{f^{(l)}_{1|0}(x_1|x_0)}{b^{(l)}_{0|1}(x_0|x_1)\psi_1(x_1)} + \overline{\mathbb{KL}}(f^{(l)}_{1|0}(x_1|x_0) \parallel b^{(l)}_{0|1}(x_0|x_1)\psi_1(x_1)), \quad l \in [0, \infty), \\
\frac{d \log b^{(l)}_{0|1}(x_0|x_1)}{dl} &= -\log \frac{b^{(l)}_{0|1}(x_0|x_1)}{f^{(l)}_{1|0}(x_1|x_0)\psi_0(x_0)} + \overline{\mathbb{KL}}(b^{(l)}_{0|1}(x_0|x_1) \parallel f^{(l)}_{1|0}(x_1|x_0)\psi_0(x_0)), \quad l \in [0, \infty).
\end{aligned}
\tag{17}
$$

In (17), $\overline{\mathbb{KL}}(\cdot \parallel \cdot)$ denotes the generalized KL divergence between unnormalized densities, as is the case here for the second arguments, and the initial conditions $f^{(0)}_{1|0}(x_1|x_0)$ and $b^{(0)}_{0|1}(x_0|x_1)$ are determined by $(\overrightarrow{F}(\theta), \overleftarrow{B}(\theta))$ with null drift terms. (17) can be contrasted with Karimi et al. (2023, Equation (13)). In Appendix C we report a simple numerical application of (17) to the Gaussian setting, which recovers S.

5 Numerical Experiments

To evaluate the performance of BM^2 on EOT problems, we utilize the benchmark developed by Gushchin et al. (2023). For the reference process (R), this benchmark provides pairs of target distributions Ψ_0, Ψ_1 with analytical EOT solution $S_{0,1}$ and analytical SB-optimal drift function μ_s. We focus on the mixtures benchmark, which consist of a centered Gaussian distribution as $S_0 = \Psi_0$ and a mixture of 5 Gaussian distributions for $S_{1|0}$. $S_1 = \Psi_1$ is not a mixture of Gaussian distributions, but has 5 distinct modes. The benchmark is constructed for dimensions $d \in \{2, 16, 64, 128\}$ and entropic regularization parameters $\varepsilon \in \{0.1, 1, 10\}$.

For each fully trained method, characterized by a stochastic process distribution P and forward drift function μ_p, we assess performance using two evaluation metrics:

- $\mathbb{KL}(S \parallel P)$ where, by Girsanov theorem (Øksendal, 2013),

$$
\mathbb{KL}(S \parallel P) = \underset{S}{\mathbb{E}} \left[\frac{1}{2\sigma^2} \int_0^1 \|\mu_s(X_t, t) - \mu_p(X_t, t)\|^2 dt \right];
\tag{18}
$$

Method	$\varepsilon=0.1$				$\varepsilon=1$				$\varepsilon=10$			
	$d=2$	$d=16$	$d=64$	$d=128$	$d=2$	$d=16$	$d=64$	$d=128$	$d=2$	$d=16$	$d=64$	$d=128$
BM^2	**0.01**	**0.20**	**1.03**	**3.06**	**0.01**	**0.11**	**1.43**	8.29	**0.11**	**2.25**	**13.13**	**40.46**
	0.01	0.02	0.07	0.16	0.00	0.00	0.03	0.36	0.01	0.04	0.13	0.49
BM^2_σ	0.43	3.76	39.55	127.2	0.04	0.43	5.36	18.66	0.15	2.64	13.78	43.42
	0.09	0.46	1.96	1.4	0.01	0.03	0.35	0.73	0.00	0.05	0.24	1.43
I-BM	0.03	0.20	1.24	5.70	0.01	0.16	1.94	**7.79**	0.16	4.09	17.17	49.17
	0.01	0.02	0.04	0.42	0.00	0.01	0.04	**0.07**	0.00	0.03	0.21	0.55
DIPF	0.59	2.39	7.93	34.77	0.23	1.21	13.13	36.51	0.81	28.25	113.8	345.8
	0.14	0.05	1.23	0.82	0.06	0.18	0.79	1.05	0.06	2.12	7.2	8.1

Table 1: Monte Carlo estimate of $\mathbb{KL}(S \,\|\, P)$ as function of ε and d, standard deviation in gray.

- $\text{cB}\mathbb{W}_2^2\text{-UVP}(S_{0,1}, P_{0,1})$, where

$$\text{cB}\mathbb{W}_2^2\text{-UVP}(S_{0,1}, P_{0,1}) \coloneqq \frac{100}{\frac{1}{2}\,\mathbb{V}_S[X_1]} \int \text{B}\mathbb{W}_2^2(S_{1|0}(X_1|X_0), P_{1|0}(X_1|X_0))S_0(dX_0), \qquad (19)$$

  $\text{B}\mathbb{W}_2^2(\cdot, \cdot)$ is the squared Bures-Wasserstein distance, i.e. the squared Wasserstein-2 distance between (assumed) multivariate Gaussian distributions (Dowson & Landau, 1982), and $\mathbb{V}_S[X_1]$ is the variance of $X_1 \sim S_1$.

We focus on the divergence $\mathbb{KL}(S \,\|\, P)$, rather than $\mathbb{KL}(P \,\|\, S)$, as a low $\mathbb{KL}(S \,\|\, P)$ more accurately indicates that P approximates S effectively across the entire support of S. The data-processing inequality implies that $\mathbb{KL}(S_{0,1} \,\|\, P_{0,1}) \leq \mathbb{KL}(S \,\|\, P)$. The $\text{cB}\mathbb{W}_2^2\text{-UVP}(\cdot, \cdot)$ metric, introduced by Gushchin et al. (2023), is a normalized and conditional extension of the standard $\text{B}\mathbb{W}_2^2(\cdot, \cdot)$ distance. Results for evaluation metrics (18) and (19) are summarized in Table 1 and Table 2, respectively.

In our benchmarking, we compare BM^2 against the I-BM and DIPF methods (Section 3.1). Each experiment is repeated five times, including both model training and metric evaluation, to obtain uncertainty quantification. We use $1{,}000$ Monte Carlo samples to estimate (18, 19). For simplicity, we employ the Euler–Maruyama scheme (Kloeden & Platen, 1992) with 200 discretization steps ($\Delta t = 0.005$) in all path sampling procedures. Each method undergoes $50{,}000$ SGD training steps with a batch size of $1{,}000$, settings similar to those used by Gushchin et al. (2023), enabling qualitative comparison of our results with theirs. We use the AdamW optimizer with a learning rate of 10^{-4} and hyperparameters: $\beta = (0.9, 0.999), \epsilon = 10^{-8}, wd = 0.01$, where wd denotes weight decay. Time is sampled as $t \sim \mathcal{U}(\epsilon, 1 - \epsilon)$ for $\epsilon = 0.0025$.

For BM^2, we employ a single feedforward neural network with 3 layers of width 768 and ReLU activation, resulting in approximately 1 million parameters. As mentioned in Section 4.1, we implement path caching and an exponential moving average for parameters used in path sampling. The cache contains $5{,}000$ initial-terminal values from both ($F(\theta)$) and ($\overleftarrow{B}(\theta)$), refreshed every 200 training steps.

For I-BM and DIPF, each outer loop iteration comprises $5{,}000$ SGD steps, totaling 10 outer loop (algorithmic) iterations. Following best practices (Bortoli et al., 2021; Shi et al., 2023), we alternate time directions over iterations for both algorithms. Each method employs two separate neural networks for forward and backward time directions, maintaining a total parameter count close to 1 million, matching BM^2's model size. As with BM^2, we implement path caching (for DIPF, entire discretized paths are cached) and EMA for sampling.

We also consider BM^2_σ, a variant of BM^2 that learns Schrödinger bridges for Ψ_0, Ψ_1 across multiple σ values. This amortized version leverages BM^2's non-iterative nature. At each optimization step, σ is sampled from $\mathcal{U}(0.1, 4)$ and utilized in discretizing SDEs ($F(\theta), \overleftarrow{B}(\theta)$) (lines 3 and 5 of Algorithm 1) and in bridge sampling (lines 7 and 8 of Algorithm 1). The neural network implementing drift functions $\mu_f(x, t, \theta)$ and $v_b(x, t, \theta)$ is modified to accept σ as an additional input, resulting in conditional drift functions $\mu_f(x, t, \theta, \sigma)$ and $v_b(x, t, \theta, \sigma)$. Path caching is adjusted to store σ values corresponding to cached paths.

In Table 2, we additionally include three baselines. EOT: sampling from the EOT solution, accounting for the bias due to Monte Carlo estimation. SB(discr): sampling from the SB solution via the SB-optimal drift

Method	$\varepsilon = 0.1$				$\varepsilon = 1$				$\varepsilon = 10$			
	$d=2$	$d=16$	$d=64$	$d=128$	$d=2$	$d=16$	$d=64$	$d=128$	$d=2$	$d=16$	$d=64$	$d=128$
EOT	0.02 0.00	0.05 0.00	0.34 0.00	0.91 0.00	0.09 0.00	0.17 0.00	0.43 0.00	1.14 0.00	0.12 0.00	0.18 0.00	0.23 0.00	0.38 0.00
SB(discr.)	0.04 0.00	0.07 0.00	0.35 0.00	0.92 0.00	0.10 0.00	0.17 0.00	0.45 0.00	1.18 0.00	0.12 0.00	1.15 0.00	5.38 0.01	10.48 0.01
$\Psi_0 \otimes \Psi_1$	195.8 6.9	186.3 2.4	162.6 0.8	145.1 2.1	136.1 4.7	127.6 1.4	113.0 1.7	93.61 1.57	8.07 0.33	4.88 0.14	4.22 0.09	4.45 0.07
BM^2	**0.73** 0.40	4.64 0.58	**6.84** 0.59	**8.28** 0.62	**0.14** 0.03	**0.41** 0.04	**1.72** 0.09	8.30 1.17	**0.14** 0.01	**2.30** 0.04	**41.14** 1.99	264.4 7.0
BM^2_σ	8.38 3.40	16.06 2.81	44.15 0.84	83.84 0.92	0.20 0.07	2.61 0.41	25.89 1.65	64.76 2.66	0.14 0.01	2.57 0.03	58.76 0.76	323.0 8.8
I-BM	1.07 0.50	**4.25** 0.66	7.19 0.28	16.63 2.07	0.20 0.09	0.53 0.04	2.20 0.35	**7.79** 0.79	0.14 0.02	5.21 0.11	135.8 1.3	578.7 9.8
DIPF	7.82 2.51	15.30 1.00	20.12 1.53	29.36 1.02	1.66 0.24	5.98 0.65	13.11 2.49	28.86 3.52	0.69 0.08	6.85 0.21	72.63 0.70	**226.1** 1.1

Table 2: Monte Carlo estimate of $\text{cB}\mathbb{W}^2_2\text{-UVP}(S_{0,1}, P_{0,1})$ as function of ε and d, standard deviation in gray.

μ_s, additionally accounting for Euler–Maruyama scheme discretization error. $\Psi_0 \otimes \Psi_1$: sampling from the independent coupling.

We now discuss the results presented in Tables 1 and 2. BM^2 demonstrates superior overall performance across dimensions and entropic regularization settings in both metrics. I-BM also shows good performance, particularly in comparison to the DIPF procedure, which aligns with the findings of Shi et al. (2023).

As expected, the performance of all methods deteriorates as the number of dimensions increases. This is because the metric(18) scales linearly with the number of dimensions, assuming a constant error rate in estimating each component of the true drift μ_s. Similar considerations apply to the metric (19).

While BM^2_σ exhibits a performance gap compared to BM^2, it yields reasonable results in low-dimensional settings ($d = 2, 16$). This gap may be due to increased pressure on model capacity or the need to normalize loss levels across σ values. All methods perform poorly in the high regularization setting ($\varepsilon = 10$), especially in high dimensions ($d = 64, 128$), which we include for completeness. It should be noted that, in such cases, sampling from the independent coupling (a trivial solution) is preferable to sampling from the SB-optimal SDE for the chosen discretization interval.

6 Related Works

Relevant works that, like BM^2, address the *dynamic* Schrödinger bridge problem (1) include:

I-BM and DIPF: The works most closely related to BM^2 are the iterative, sample-based DIPF (Bortoli et al., 2021; Vargas et al., 2021) and I-BM (Shi et al., 2023; Peluchetti, 2023) procedures, which do not satisfy desiderata (i). Built on similar bridge matching principles, BM^2 can be viewed as a modification of I-BM that employs a single optimization loop, resulting in a simpler algorithm that we have empirically shown to be competitive.

Forward-Backward SB SDE: Chen et al. (2022) proposes two training algorithms addressing the dynamic SB problem. Both approaches employ loss functions that require divergence computations (violating desiderata (iv)) and the use of two distinct neural networks. The first method is iterative, resembling DIPF (violating desiderata (i)), while the second method involves differentiating through entire discretized paths, resulting in high memory consumption (violating desiderata (iii)).

The subsequent works concentrate on solving the *static* Schrödinger bridge (2), or EOT (3), problem. Once this is achieved, solutions to the dynamic problem are trivially obtained through the standard decomposition $S = S_{0,1} R_{|0,1}$. Although these works differ in nature and objectives, we include them here due to their shared characteristic with BM^2: the non-iterative nature of the algorithm.

Light SB: In two notable works, Korotin et al. (2023) and Gushchin et al. (2024) propose non-iterative, sample-based EOT solvers for the Euclidean cost function, i.e., for the specific choice of reference dynamics (R). Korotin et al. (2023) introduces an approximation to (an adjusted version of) the Schrödinger potential for Ψ_1 via a mixture of Gaussian distributions, resulting in a mixture of Gaussian distributions approximation to $S_{1|0}$. Gushchin et al. (2024) builds upon this approximation and introduces an additional sample-based training objective that takes as input any coupling $C_{0,1} \in \mathcal{C}(\Psi_0, \Psi_1)$, whereas Korotin et al. (2023) requires the independent coupling $\Psi_0 \otimes \Psi_0$. While also non-iterative, the proposals of Korotin et al. (2023); Gushchin et al. (2024) differ from BM^2 in two key aspects: (a) they learn a solution in the static setting instead of the dynamic one, and (b) they employ mixture of Gaussian distributions approximations, rather than neural network approximators for the drift functions. Consequently, these methods may face challenges in scaling to modern generative ML applications. Light SB, in both variants, demonstrates strong performance in the benchmark presented in Section 5 (Gushchin et al., 2024, Table 1). However, it is worth noting that this benchmark is particularly well-suited for Light SB, as acknowledged by its authors, since the target $S_{0,1}$ is constructed such that $S_{1|0}$ is itself a mixture of 5 Gaussian distributions.

7 Conclusions

In this work we introduced Coupled Bridge Matching (BM^2), a novel approach for learning Schrödinger bridges from samples. BM^2 builds on the principles of Bridge Matching while addressing key limitations of existing iterative methods. Our approach offers several advantages, including a simple single-loop optimization procedure, exactness in the idealized setting, modest memory requirements, and a straightforward loss function. The numerical experiments demonstrate that BM^2 is competitive with and often outperforms existing iterative diffusion-based methods like I-BM and DIPF across various dimensions and entropic regularization settings.

On the theoretical front, there is substantial room for improvement. Firstly, while bearing some resemblance to the standard convergence result for the EM algorithm, Theorem 2 lacks a quantity analogous to the likelihood being maximized in the EM algorithm. It remains unclear whether decreases in $\mathbb{KL}(M^{B_{0,1}(\theta')} \,\|\, F(\theta))$ and $\mathbb{KL}(M^{F_{0,1}(\theta')} \,\|\, B(\theta))$ can be linked to decreases in $\mathbb{KL}(F(\theta) \,\|\, S)$ and $\mathbb{KL}(B(\theta) \,\|\, S)$. Secondly, the requirement that $F(\theta) = B(\theta)$, equivalently that $(F(\theta))$ and $(\overleftarrow{B}(\theta))$ are time-reversals of each other, appears unnecessary. Notably, all numerical simulations conducted do not explicitly enforce this condition, which emerges naturally during the training process. Thirdly, it would be valuable to study problem (14) where reverse KL divergences are partially minimized, aligning more closely with the BM^2 algorithm. In this scenario, Lemma 4 no longer holds, and it may be necessary to impose a corresponding additional constraint to maintain tractable analytical computations. The attractors of (17), and of a corresponding dynamical system arising from reverse KL minimization, can be investigated to assess further convergence properties of BM^2.

On the empirical front, the applications of BM^2 in contemporary generative machine learning tasks remain unexplored. Given the promising results from previous studies employing Bridge Matching, such as those by Liu et al. (2023) and Somnath et al. (2023), it is anticipated that BM^2 could be effectively applied to various domains, including image generation, audio synthesis, and molecular design. Future work could investigate the scalability and performance of BM^2 in these domains.

References

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion Schrödinger Bridge with Applications to Score-Based Generative Modeling. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

Tianrong Chen, Guan-Horng Liu, and Evangelos Theodorou. Likelihood Training of Schrödinger Bridge using Forward-Backward SDEs Theory. In *International Conference on Learning Representations*, 2022.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data Via the EM Algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977. ISSN 0035-9246.

D. C Dowson and B. V Landau. The Fréchet distance between multivariate normal distributions. *Journal of Multivariate Analysis*, 12(3):450–455, 1982. ISSN 0047-259X.

Robert Fortet. Résolution d'un systeme d'équations de M. Schrödinger. *J. Math. Pure Appl. IX*, 1:83–105, 1940.

Nikita Gushchin, Alexander Kolesov, Petr Mokrov, Polina Karpikova, Andrei Spiridonov, Evgeny Burnaev, and Alexander Korotin. Building the Bridge of Schrödinger: A Continuous Entropic Optimal Transport Benchmark. In *Thirty-Seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

Nikita Gushchin, Sergei Kholkin, Evgeny Burnaev, and Alexander Korotin. Light and Optimal Schrödinger Bridge Matching. In *Forty-First International Conference on Machine Learning*, 2024.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. arXiv, 2020.

Benton Jamison. Reciprocal processes. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 30 (1):65–86, 1974. ISSN 1432-2064.

Benton Jamison. The Markov processes of Schrödinger. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 32(4):323–331, 1975. ISSN 1432-2064.

Mohammad Reza Karimi, Ya-Ping Hsieh, and Andreas Krause. Sinkhorn Flow: A Continuous-Time Framework for Understanding and Generalizing the Sinkhorn Algorithm, 2023.

Peter E. Kloeden and Eckhard Platen. *Numerical Solution of Stochastic Differential Equations*. Springer Berlin Heidelberg, 1992.

Alexander Korotin, Nikita Gushchin, and Evgeny Burnaev. Light Schrödinger Bridge. In *The Twelfth International Conference on Learning Representations*, 2023.

Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos Theodorou, Weili Nie, and Anima Anandkumar. I^{2}SB: Image-to-Image Schrödinger Bridge. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 22042–22062. PMLR, 2023.

Xingchao Liu, Lemeng Wu, Mao Ye, and Qiang Liu. Let us Build Bridges: Understanding and Extending Diffusion Generative Models. In *NeurIPS 2022 Workshop on Score-Based Methods*, 2022.

Christian Léonard. Some Properties of Path Measures. In Catherine Donati-Martin, Antoine Lejay, and Alain Rouault (eds.), *Séminaire de Probabilités XLVI*, Lecture Notes in Mathematics, pp. 207–230. Springer International Publishing, 2014a.

Christian Léonard. A survey of the Schrödinger problem and some of its connections with optimal transport. *Discrete & Continuous Dynamical Systems*, 34(4):1533, 2014b.

Anton Mallasto, Augusto Gerolin, and Hà Quang Minh. Entropy-regularized 2-Wasserstein distance between Gaussian measures. *Information Geometry*, 5(1):289–323, 2022. ISSN 2511-249X.

Youssef Marzouk, Tarek Moselhy, Matthew Parno, and Alessio Spantini. Sampling via Measure Transport: An Introduction. In Roger Ghanem, David Higdon, and Houman Owhadi (eds.), *Handbook of Uncertainty Quantification*, pp. 1–41. Springer International Publishing, 2016.

Stefano Peluchetti. Non-Denoising Forward-Time Diffusions. 2021.

Stefano Peluchetti. Diffusion Bridge Mixture Transports, Schrödinger Bridge Problems and Generative Modeling. *Journal of Machine Learning Research*, 24(374):1–51, 2023. ISSN 1533-7928.

Gabriel Peyré and Marco Cuturi. Computational Optimal Transport. 2020.

Ludger Ruschendorf. Convergence of the Iterative Proportional Fitting Procedure. *The Annals of Statistics*, 23(4):1160–1174, 1995. ISSN 0090-5364.

Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion Schrödinger Bridge Matching. In *Thirty-Seventh Conference on Neural Information Processing Systems*, 2023.

Vignesh Ram Somnath, Matteo Pariset, Ya-Ping Hsieh, Maria Rodriguez Martinez, Andreas Krause, and Charlotte Bunne. Aligned diffusion Schrödinger bridges. In Robin J. Evans and Ilya Shpitser (eds.), *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pp. 1985–1995. PMLR, 31 Jul–04 Aug 2023.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*, 2021.

Francisco Vargas, Pierre Thodoroff, Austen Lamacraft, and Neil Lawrence. Solving Schrödinger Bridges via Maximum Likelihood. *Entropy*, 23(9):1134, 2021.

B. K. Øksendal. *Stochastic Differential Equations: An Introduction with Applications*. Universitext. Springer, 6th ed., 6th corrected printing edition, 2013.

A Additional Dynamics

In this section we consider a simple extension to the dynamics of Section 2.2, and refer the reader to Peluchetti (2021; 2023) for the more general case. Here, we consider the case where the reference distribution R is given by the solution to:

$$X_0 \sim \Psi_0, \quad dX_t = \sigma\sqrt{\beta_t}dW_t, \quad t \in [0,1], \tag{20}$$

with $\sigma \geq 0$, $\beta_t : [0,1] \to \mathbb{R}_{>0}$ strictly positive and continuous. With $b_{s:t} := \int_s^t \beta_u du$, $0 \leq s \leq t \leq 1$, β_t is chosen such that $b_{0:1} = 1$, to disentangle the contribution of β_t from the contribution of σ. Indeed, under these conditions, β_t defines a time-warping: if X_t is the solution to (R), then $X_{b_{0:t}}$ has the same distribution as the solution to (20). Consequently, the solutions to (2) and (3) are independent of β_t.

When employing (20), the definitions in Section 2.2 are replaced as follows:

$$R_{t|0}(\cdot|x_0) = \mathcal{N}(x_0, \sigma^2 b_{0:t}), \tag{21}$$

$$R_{t|0,1}(\cdot|x_0, x_1) = \mathcal{N}(x_0 b_{t:1} + x_1 b_{0:t}, \sigma^2 b_{0:t} b_{t:1}), \tag{22}$$

$$\mu_{01}(x_t, t, x_1) := \sigma^2 \beta_t \nabla_{x_t} \log r_{1|t}(x_1|x_t) = \frac{\beta_t}{b_{t:1}}(x_1 - x_t), \tag{23}$$

$$v_{01}(x_t, t, x_0) := \sigma^2 \beta_t \nabla_{x_t} \log r_{t|0}(x_t|x_0) = \frac{\beta_t}{b_{0:t}}(x_0 - x_t). \tag{24}$$

B Proofs

Theorem 1 (Complete BM² Iterations). *Consider the SDEs $(F(\theta), \overleftarrow{B}(\theta))$, with initial drifts $\mu_f^{(0)}, v_b^{(0)}$ and corresponding distributions $F^{(0)}, B^{(0)}$. For each $i \geq 1$, let $(\mu_f^{(i)}, v_b^{(i)}) = \arg\min_{(\mu,v)} \mathbb{L}(\mu, v; \mu_f^{(i-1)}, v_b^{(i-1)})$, resulting in the distribution iterates $F^{(i)}, B^{(i)}$. We distinguish two cases:*

(i) *$\mu_f^{(0)} = v_b^{(0)} = 0$: both the iterates $F^{(0)}, B^{(1)}, F^{(2)}, \dots$ and the iterates $B^{(0)}, F^{(1)}, B^{(2)}, \dots$ are equivalent to the DIPF iterates, started respectively from the forward and from the backward time direction;*

(ii) *$\mu_f^{(0)} = \mu_m^{C_{0,1}}, v_b^{(0)} = v_m^{C_{0,1}}$ for some $C \in \mathcal{C}(\Psi_0, \Psi_1)$: $F^{(i)} = B^{(i)} = I^{(i)}$ for each $i \geq 0$ where $I^{(i)}$ are the I-BM iterates.*

Proof. Define Q associated with

$$X_1 \sim \Psi_1, \quad dX_t = \sigma dW_t, \quad t \in [1,0], \qquad (\overleftarrow{Q})$$

which is not the time reversal of (R), but $R_{|0,1} = Q_{|0,1}$.

Firstly, consider the case of initial null drifts: $\mu_f^{(0)} = v_b^{(0)} = 0$, corresponding to $F^{(0)} = \Psi_0 R_{|0} = \Psi_0 R_{1|0} R_{|0,1} = F_{0,1}^{(0)} R_{|0,1} \in \mathcal{S}$ and $B^{(0)} = \Psi_1 Q_{|1} = \Psi_1 Q_{0|1} R_{|0,1} = B_{0,1}^{(0)} R_{|0,1} \in \mathcal{S}$. As $B^{(0)} = \Pi^{B_{0,1}^{(0)}} = M^{B_{0,1}^{(0)}}$, we have $F^{(1)} = \Psi_0 M_{|0}^{B_{0,1}^{(0)}} = \Psi_0 B_{|0}^{(0)} = \Psi_0 B_{1|0}^{(0)} R_{|0,1} \in \mathcal{S}$. As $F^{(0)} = \Pi^{F_{0,1}^{(0)}} = M^{F_{0,1}^{(0)}}$, $B^{(1)} = \Psi_1 M_{|1}^{F_{0,1}^{(0)}} = \Psi_1 F_{|1}^{(0)} = \Psi_1 F_{0|1}^{(0)} R_{|0,1} \in \mathcal{S}$. By induction, $F^{(i)} = \Psi_0 B_{1|0}^{(i-1)} R_{|0,1} \in \mathcal{S}$ and $B^{(i)} = \Psi_1 F_{0|1}^{(i-1)} R_{|0,1} \in \mathcal{S}$, $i \geq 1$. We now construct two forward-backward sequences. For the sequence $F^{(0)}, B^{(1)}, F^{(2)}, \dots$, we have $F_{0,1}^{(0)} = \Psi_0 R_{1|0}$, $B_{0,1}^{(1)} = \Psi_1 F_{0|1}^{(0)}$, $F_{0,1}^{(2)} = \Psi_0 B_{1|0}^{(1)}$, $\dots$ which are the static IPF iterates: one marginal gets replaced at a time keeping the conditional distribution fixed. In the same way, for $B^{(0)}, F^{(1)}, B^{(2)}, \dots$, we have $B_{0,1}^{(0)} = \Psi_1 Q_{0|1}$, $F_{0,1}^{(1)} = \Psi_0 B_{1|0}^{(0)}$, $B_{0,1}^{(2)} = \Psi_1 F_{0|1}^{(1)}$, $\dots$ which are again the static IPF iterates (for the backward formulation of the dynamic SB problem, i.e. via $\overleftarrow{Q}_{|0}$ as reference measure instead of $R_{|0}$, and switched marginal distributions). As each pair $F^{(i)}, B^{(i)}$ is of the form $F^{(i)} = F_{0,1}^{(i)} R_{|0,1}$, $B^{(i)} = B_{0,1}^{(i)} R_{|0,1}$, we also recover the dynamic DIPF iterates.

Secondly, consider $\mu_f^{(0)}$ and $v_b^{(0)}$ both corresponding to the BM transport based on the given coupling: $I^{(0)} = M^{C_{0,1}}$, $F^{(0)} = B^{(0)} = I^{(0)}$. Then, looking separately at either of the sequences $F^{(i)}$, $i \geq 1$, and $B^{(i)}$, $i \geq 1$, we obtain that $F^{(i)} = B^{(i)} = I^{(i)}$, $i \geq 1$. $\qquad\square$

Lemma 2 (Loss Interpretation). *It holds that*

$$
\begin{aligned}
\mathbb{KL}(B_0 \,\|\, \Psi_0) + \mathbb{L}_f(\mu_f; v_b) &= \mathbb{KL}(\Pi^{B_{0,1}} \,\|\, F) + C_1(B) = \mathbb{KL}(M^{B_{0,1}} \,\|\, F) + C_2(B), \\
\mathbb{KL}(F_1 \,\|\, \Psi_1) + \mathbb{L}_b(v_b; \mu_f) &= \mathbb{KL}(\Pi^{F_{0,1}} \,\|\, B) + D_1(F) = \mathbb{KL}(M^{F_{0,1}} \,\|\, B) + D_2(F),
\end{aligned}
\tag{13}
$$

for $C_1(B), C_2(B)$ independent of F, $D_1(F), D_2(F)$ independent of B, with $0 \leq C_1(B) \leq C_2(B)$ and $0 \leq D_1(F) \leq D_2(F)$.

Proof. We consider only $\mathbb{L}_f(\mu_f; v_b)$, the arguments for $\mathbb{L}_b(v_b; \mu_f)$ are symmetric. By Girsanov Theorem (Øksendal, 2013) and by the marginal-conditional decomposition of Kullback-Leibler divergences we have

$$
\mathbb{KL}(M^{B_{0,1}} \,\|\, F) = \mathbb{KL}(B_0 \,\|\, \Psi_0) + \mathop{\mathbb{E}}_{\Pi^{B_{0,1}}} \left[\frac{1}{2} \int_0^1 \|\mu_f(X_t, t) - \mu_m^{B_{0,1}}(X_t, t)\|^2 dt \right],
$$

$$
\mathbb{KL}(\Pi^{B_{0,1}} \,\|\, F) = \mathbb{KL}(B_0 \,\|\, \Psi_0) + \mathop{\mathbb{E}}_{\Pi^{B_{0,1}}} \left[\frac{1}{2} \int_0^1 \|\mu_f(X_t, t) - \mu_\pi^{B_{0,1}}(X_t, t, X_0)\|^2 dt \right],
$$

$$
\mathbb{KL}(\Pi^{B_{0,1}} \,\|\, \mathcal{F}) = \mathbb{KL}(B_0 \,\|\, \Psi_0) + \mathop{\mathbb{E}}_{\Pi^{B_{0,1}}} \left[\frac{1}{2} \int_0^1 \|\mu_f(X_t, t) - \mu_{01}(X_t, t, X_1)\|^2 dt \right]
$$

$$
= \mathbb{KL}(B_0 \,\|\, \Psi_0) + \mathbb{L}_f(\mu_f; v_b),
$$

where $\mu_\pi^{B_{0,1}}(X_t, t, X_0) := \mathbb{E}_{\Pi^{B_{0,1}}}[\mu_{01}(X_t, t, X_1) | X_t, X_0]$, $\mu_m^{B_{0,1}}(X_t, t) := \mathbb{E}_{\Pi^{B_{0,1}}}[\mu_{01}(X_t, t, X_1) | X_t]$, and $\mathcal{F}$ is distribution of the non-Markov diffusion solution to the auxiliary SDE

$$
X_0 \sim \Psi_0, \quad dX_t = [\mu_f(X_t, t) - \mu_{01}(X_t, t, X_1) + \mu_\pi^{B_{0,1}}(X_t, t, X_0)]dt + \sigma dW_t, \quad t \in [0, 1]. \tag{$\mathcal{F}$}
$$

By the tower property of conditional expectations and by the conditional Jensen inequality it follows that

$$
\mathbb{KL}(\Pi^{B_{0,1}} \,\|\, \mathcal{F}) - \mathbb{KL}(\Pi^{B_{0,1}} \,\|\, F)
$$

$$
= \mathop{\mathbb{E}}_{\Pi^{B_{0,1}}} \left[\frac{1}{2} \int_0^1 \|\mu_f(X_t, t) - \mu_{01}(X_t, t, X_1)\|^2 - \|\mu_f(X_t, t) - \mu_\pi^{B_{0,1}}(X_t, t, X_0)\|^2 dt \right]
$$

$$
= \mathop{\mathbb{E}}_{\Pi^{B_{0,1}}} \left[\frac{1}{2} \int_0^1 \|\mu_{01}(X_t, t, X_1)\|^2 - \|\mu_\pi^{B_{0,1}}(X_t, t, X_0)\|^2 dt \right] = C_1(B) \geq 0.
$$

By the Pythagorean property of the BM transport (Liu et al., 2022; Peluchetti, 2023)

$$
\mathbb{KL}(\Pi^{B_{0,1}} \,\|\, F) - \mathbb{KL}(M^{B_{0,1}} \,\|\, F) = \mathbb{KL}(\Pi^{B_{0,1}} \,\|\, M^{B_{0,1}}) = K(B) \geq 0.
$$

Taking $C_2(B) = C_1(B) + K(B)$ completes the proof. $\qquad\square$

Lemma 3 ($\mathcal{R}$-stability of $F^{(\lambda)}, B^{(\lambda)}$). *If $F', B' \in \mathcal{R}$, then $F^{(\lambda)}, B^{(\lambda)} \in \mathcal{R}$ for each $\lambda \in [0, 1]$.*

Proof. By the marginal-conditional decomposition of Kullback-Leibler divergences

$$
\mathbb{KL}(F \,\|\, B') = \mathbb{KL}(F_{0,1} \,\|\, B'_{0,1}) + \mathop{\mathbb{E}}_{F_{0,1}} [\mathbb{KL}(F_{|0,1} \,\|\, B'_{|0,1})],
$$

$$
\mathbb{KL}(F \,\|\, F') = \mathbb{KL}(F_{0,1} \,\|\, F'_{0,1}) + \mathop{\mathbb{E}}_{F_{0,1}} [\mathbb{KL}(F_{|0,1} \,\|\, F'_{|0,1})],
$$

and $B'_{|0,1} = F'_{|0,1} = R_{|0,1}$, hence

$$
F^{(\lambda)} := \mathop{\arg\min}_{F \in \mathcal{P}(\Psi_0, \cdot)} \lambda \mathbb{KL}(F_{0,1} \,\|\, B'_{0,1}) + (1 - \lambda)\mathbb{KL}(F_{0,1} \,\|\, F'_{0,1}) + \mathop{\mathbb{E}}_{F_{0,1}} [\mathbb{KL}(F_{|0,1} \,\|\, R_{|0,1})],
$$

$$
B^{(\lambda)} := \mathop{\arg\min}_{B \in \mathcal{P}(\cdot, \Psi_1)} \lambda \mathbb{KL}(B_{0,1} \,\|\, F'_{0,1}) + (1 - \lambda)\mathbb{KL}(B_{0,1} \,\|\, B'_{0,1}) + \mathop{\mathbb{E}}_{B_{0,1}} [\mathbb{KL}(B_{|0,1} \,\|\, R_{|0,1})],
$$

and thus $F^{(\lambda)}_{|0,1} = B^{(\lambda)}_{|0,1} = R_{|0,1}$, which completes the proof. $\qquad\square$
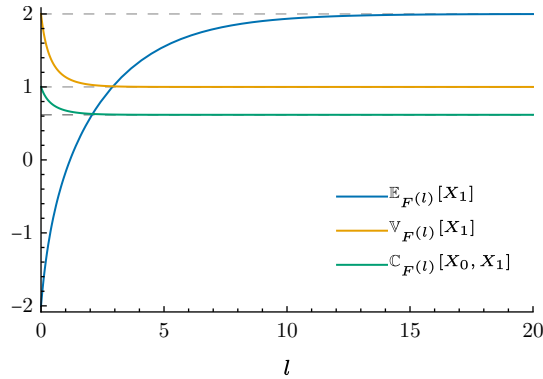
Figure 1: Algorithmic-time l evolution of $\mathbb{E}_{F^{(l)}}[X_1]$, $\mathbb{V}_{F^{(l)}}[X_1]$, $\mathbb{C}_{F^{(l)}}[X_0, X_1]$, compared with $\mathbb{E}_S[X_1]$, $\mathbb{V}_S[X_1]$, $\mathbb{C}_S[X_0, X_1]$ as dashed gray lines.

Lemma 4 ($\mathcal{S}$-stability of $F^{(\lambda)}, B^{(\lambda)}$). *If $F', B' \in \mathcal{S}$, then $F^{(\lambda)}, B^{(\lambda)} \in \mathcal{S}$ for each $\lambda \in [0, 1]$.*

Proof. In view of Lemma 3, we have to verify that $F_{0,1}^{(\lambda)}, B_{0,1}^{(\lambda)}$ solve the EOT problems (3) for some marginal distributions if $F_{0,1}', B_{0,1}'$ do. For simplicity, we assume that all of $F_{0,1}^{(\lambda)}, B_{0,1}^{(\lambda)}, F_{0,1}', B_{0,1}'$ admits positive densities on $\mathbb{R}^{d \times d}$, and that Ψ_0 and Ψ_1 admits positive densities on $\mathbb{R}^d$. The steps of this proof carry over to the more general measure-theoretic setting.

We know that $f_{0,1}^{(\lambda)}(x_0, x_1) = \psi_0(x_0) f_{1|0}^{(\lambda)}(x_1|x_0)$ and $b_{0,1}^{(\lambda)}(x_0, x_1) = b_{0|1}^{(\lambda)}(x_0|x_1)\psi_1(x_1)$, where $f_{1|0}^{(\lambda)}(x_1|x_0) \propto b_{1|0}'(x_1|x_0)^\lambda f_{1|0}'(x_1|x_0)^{1-\lambda}$ and $b_{0|1}^{(\lambda)}(x_0|x_1) \propto f_{0|1}'(x_0|x_1)^\lambda b_{0|1}'(x_0|x_1)^{1-\lambda}$ (see Section 4.2). On the other hand

$$f_{0,1}'(x_0, x_1) = \exp\left\{\phi_0^{f'}(x_0) + \phi_1^{f'}(x_1) - \frac{\kappa(x_0, x_1)}{\varepsilon}\right\},$$

$$b_{0,1}'(x_0, x_1) = \exp\left\{\phi_0^{b'}(x_0) + \phi_1^{b'}(x_1) - \frac{\kappa(x_0, x_1)}{\varepsilon}\right\},$$

for the Schrödinger potentials[2] $\phi_0^{f'}(x_0), \phi_1^{f'}(x_1)$ and $\phi_0^{b'}(x_0) + \phi_1^{b'}(x_1)$ (Léonard, 2014a). It follows by direct computation that $f_{0,1}^{(\lambda)}(x_0, x_1)$ and $b_{0,1}^{(\lambda)}(x_0, x_1)$ satisfy:

$$f_{0,1}^{(\lambda)}(x_0, x_1) = \exp\left\{\phi_0^{f,\lambda}(x_0) + \phi_1^{f,\lambda}(x_1) - \frac{\kappa(x_0, x_1)}{\varepsilon}\right\},$$

$$b_{0,1}^{(\lambda)}(x_0, x_1) = \exp\left\{\phi_0^{b,\lambda}(x_0) + \phi_1^{f,\lambda}(x_1) - \frac{\kappa(x_0, x_1)}{\varepsilon}\right\},$$

for some other Schrödinger potentials $\phi_0^{f,\lambda}(x_0), \phi_1^{f,\lambda}(x_1)$ and $\phi_0^{b,\lambda}(x_0), \phi_1^{b,\lambda}(x_1)$. $\qquad\square$

C Infinitesimal Minimization, Gaussian Case

Consider the one-dimensional case $d = 1$, with target Gaussian marginal distributions $\Psi_0 = \mathcal{N}(\mu_0, \sigma_0^2)$ and $\Psi_1 = \mathcal{N}(\mu_1, \sigma_1^2)$, and a reference diffusion distribution R associated with (R). In this setting, the solution to the static Schrödinger bridge problem (2) is known analytically and is given by a bivariate Gaussian distribution (Mallasto et al., 2022).

We hypothesize that conditional Gaussian densities for $f_{1|0}^{(l)}(x_1|x_0)$ and $b_{0|1}^{(l)}(x_0|x_1)$ solve (17). Specifically, we propose $F_{1|0}^{(l)} = \mathcal{N}(A_l^f x_0 + a_l^f, v_l^f)$ and $B_{0|1}^{(l)} = \mathcal{N}(A_l^b x_1 + a_l^b, v_l^b)$, where $A_l^f, a_l^f, A_l^b, a_l^b \in \mathbb{R}$ and $v_l^f, v_l^b \in \mathbb{R}_{>0}$ are algorithmic-time dependent scalar parameters. By construction, $F_0^{(l)} = \mathcal{N}(\mu_0, \sigma_0^2)$ and $B_1^{(l)} = \mathcal{N}(\mu_1, \sigma_1^2)$

---

[2]We formulate the potential with respect to the Lebesgue measure on $\mathbb{R}^d$.

for each $l \geq 0$. Substituting these expressions for $f_{1|0}^{(l)}(x_1|x_0)$ and $b_{0|1}^{(l)}(x_0|x_1)$ into (17) yields a six-dimensional ODE system in the parameters. The initial conditions are $A_0^b = A_0^f = 1$, $a_0^f = a_0^b = 0$, $v_0^f = v_0^b = \sigma^2$, corresponding to initial null drift terms for $(F(\theta), \overleftarrow{B}(\theta))$, as discussed in Section 4.2.

To numerically solve the ODE and determine the values of $A_l^f, a_l^f, v_l^f, A_l^b, a_l^b, v_l^b$ over $l \in [0, L]$ for some $L > 0$, we evaluate (17) for three different pairs of (x_0, x_1). This provides sufficient constraints to identify the parameters. Subsequently, we verify that the proposed functional forms for $f_{1|0}^{(l)}(x_1|x_0)$ and $b_{0|1}^{(l)}(x_0|x_1)$ indeed solve (17).

We examine the scenario where $\mu_0 = -2, \mu_1 = 2, \sigma_0 = \sigma_1 = \sigma = 1$. Figure 1 illustrates the evolution of $\mathbb{E}_{F^{(l)}}[X_1]$, $\mathbb{V}_{F^{(l)}}[X_1]$, and $\mathbb{C}_{F^{(l)}}[X_0, X_1]$ over algorithmic time l. These quantities represent the mean and variance of X_1 and the covariance between X_0 and X_1 according to $F^{(l)}$, respectively. The corresponding values $\mathbb{E}_S[X_1], \mathbb{V}_S[X_1]$, and $\mathbb{C}_S[X_0, X_1]$ for the static Schrödinger bridge solution $S_{0,1}$ from Mallasto et al. (2022) are depicted as dashed gray lines, demonstrating convergence.

D Python Code

```python
1  # dimensions: B: batch; D: data; T: time_steps + 1
2  # required: sample_0(batch_dim, device), sample_1(batch_dim, device), fwd_drift_fn(x, t), bwd_drift_fn(x, t)
3  import torch as th
4
5  # sampling from R_{t|0,1} (5): (B, D), (B, D), (B,), () -> (B, D)
6  def sample_bridge(x_0, x_1, t, sigma):
7      B, D = x_0.shape
8      mean_t = (1 - t[..., None]) * x_0 + t[..., None] * x_1  # (B, D)
9      var_t = sigma**2 * t[..., None] * (1 - t[..., None])    # (B, D)
10     z_t = th.randn_like(x_0)                                # (B, D)
11     x_t = mean_t + th.sqrt(var_t) * z_t                     # (B, D)
12     return x_t
13
14 # fwd BM target (6): (B, D), (B, D), (B,) -> (B, D)
15 def fwd_target(x_t, x_1, t):
16     return (x_1 - x_t) / (1 - t[..., None])  # (B, D)
17
18 # fwd BM target (7): (B, D), (B, D), (B,) -> (B, D)
19 def bwd_target(x_t, x_0, t):
20     return (x_0 - x_t) / t[..., None]  # (B, D)
21
22 # Euler-Maruyama dicretization scheme: fn(x, t), (B, D), (T), () -> (B, D)
23 def discretization(drift_fn, initial_value, times, sigma):
24     B, D = initial_value.shape
25     times = times[..., None].expand(-1, B)                              # (T, B)
26     x_prev_t = initial_value                                            # (B, D)
27     for prev_t, t in zip(times[:-1], times[1:]):                        # (B), (B)
28         dt = t - prev_t                                                 # (B)
29         drift_t = drift_fn(x_prev_t, prev_t)                            # (B, D)
30         drift_part_t = drift_t * dt[..., None]                          # (B, D)
31         eps_t = th.randn_like(x_prev_t)                                 # (B, D)
32         diffusion_part_t = (sigma * th.sqrt(th.abs(dt)))[..., None] * eps_t  # (B, D)
33         x_t = x_prev_t + drift_part_t + diffusion_part_t               # (B, D)
34         x_prev_t = x_t                                                  # (B, D)
35     return x_t
36
37 # BM^2 loss computation: fn(b, d), fn(b, d), fn(x, t), fn(x, t), (), (), (), () -> ()
38 def sample_loss(sample_0, sample_1, fwd_drift_fn, bwd_drift_fn, batch_dim, time_steps, sigma, device):
39     # sample from the target marginals:
40     f_0 = sample_0(batch_dim, device)                                  # (B, D)
41     b_1 = sample_1(batch_dim, device)                                  # (B, D)
42     # sample according to current (F(θ)) and (B⃖(θ)):
43     fwd_times = th.linspace(0.0, 1.0, time_steps + 1, device=device)   # [0, 1/time_steps, ..., 1]
44     bwd_times = th.linspace(1.0, 0.0, time_steps + 1, device=device)   # [1, ..., 1/time_steps, 0]
45     f_1 = discretization(fwd_drift_fn, f_0, fwd_times, sigma).detach()  # (B, D)
46     b_0 = discretization(bwd_drift_fn, b_1, bwd_times, sigma).detach()  # (B, D)
47     # sample time and mixture processes based on F_{0,1}(θ) and B_{0,1}(θ):
48     t = th.rand((batch_dim,), device=device)                           # (B)
49     pi_f_t = sample_bridge(f_0, f_1, t, sigma)                         # (B, D)
50     pi_b_t = sample_bridge(b_0, b_1, t, sigma)                         # (B, D)
51     # define regression targets and model predictions:
52     target_f_t = fwd_target(pi_b_t, b_1, t)                            # (B, D)
53     target_b_t = bwd_target(pi_f_t, f_0, t)                            # (B, D)
54     prediction_f_t = fwd_drift_fn(pi_b_t, t)                           # (B, D)
55     prediction_b_t = bwd_drift_fn(pi_f_t, t)                           # (B, D)
56     # compute loss:
57     loss_f_t = th.sum((target_f_t - prediction_f_t)**2, dim=1) / 2     # (B)
58     loss_b_t = th.sum((target_b_t - prediction_b_t)**2, dim=1) / 2     # (B)
59     loss_t = th.mean(loss_f_t + loss_b_t)                              # ()
60     return loss_t
```

Listing 1: Basic implementation of BM^2 loss computation (Algorithm 1) in PyTorch.