SOLVING THE SCHRÖDINGER BRIDGE VIA BAYESIAN INFERENCE IN PARAMETER SPACE

Anonymous authors
Paper under double-blind review

ABSTRACT

Solving transport problems, i.e., finding a map that transports one given distribution to another, has numerous applications in machine learning. Currently, the Schrödinger Bridge (SB) is commonly used to describe the mapping between two arbitrary distributions. However, most existing SB solutions either suffer from slow convergence and sampling due to the difficulty of directly optimizing the likelihood or are highly sensitive to initialization and hyperparameters. To address these, we propose a novel Bayesian Schrödinger Bridge (BSB) formalism to address the SB problem by explicit likelihood factorization leveraging Bayesian inference. By explicitly introducing Bayesian inference in the continuous parameter space of a Gaussian mixture model, BSB allows tractable and explicit likelihood-based optimization of the likelihood instead of the variational lower bound, while enabling more accurate modeling of complex probability density transitions. BSB analytically derives closed-form Bayesian inference at each time step, enabling the design of SDE and higher-order ODE solvers for few-step high-quality generation. Extensive experiments validate our effectiveness, achieving faster convergence, robust training, and high-quality few-step SB solution.

1 INTRODUCTION

Learning a transport between probability distributions is fundamental to generative modeling Ho et al. (2020); Song et al. (2021); Song & Ermon (2019). Flow Matching Lipman et al. (2022) and Rectified Flow Liu et al. (2022a) formulate this task through an ODE connecting two distributions, providing promising alternatives to diffusion models Ho et al. (2020). Its stochastic counterpart, the Schrödinger Bridge (SB) Schrödinger (1932); Chen et al. (2021), extends transport to SDEs and characterizes entropy-regularized transport between distributions. While Iterative Proportional Fitting (IPF) Fortet (1940); Kullback (1968) and Iterative Markovian Fitting (IMF) Peluchetti (2023b;a); Shi et al. (2023) provide principled frameworks for solving SBs, their practical optimization remains challenging due to complex joint likelihood objectives Bernton et al. (2019); Gushchin et al. (2024b); Finlay et al. (2020); Caluya & Halder (2021); Pavon et al. (2021).

For IPF-based methods in Fig. 1(a), Diffusion Schrödinger Bridge (DSB) De Bortoli et al. (2021) approximates joint-distribution optimization through diffusion processes, while Simplified-DSB Tang et al. (2024) simplifies the objective and leverages pretrained SGMs for faster convergence. Nevertheless, repeated marginal correction makes IPF computationally expensive and slow to converge De Bortoli et al. (2021). IMF, illustrated in Fig. 1(b), instead combines Markovian projection with conditional-expectation correction, connecting SBs with flow-based learning and improving optimization efficiency. Existing IMF methods, however, remain limited by unstable adversarial training Gushchin et al. (2024b) or accumulated sampling errors in iterative methods such as α-DSBM De Bortoli et al. (2024) and IPMF Kholkin et al. (2026). Moreover, their typically simple Gaussian transition kernels restrict the modeling of complex transport and may hinder likelihood optimization and convergence under general distributions Kholkin et al. (2026). Conversely, Bayesian Flow Networks (BFNs) Graves et al. (2023) perform explicit Bayesian updates in distribution-parameter space and directly optimize data likelihoods, providing a principled foundation for flexible distribution transport.

To address the problems of existing SB solutions and reveal the advantages of solving SB problem in parameter space, we introduce the *Bayesian Schrödinger Bridge* (BSB) as a novel solution shown in Fig. 1(c) to the SB problem. We propose a novel Iterative Proportional Bayesian Fitting (IPBF)

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

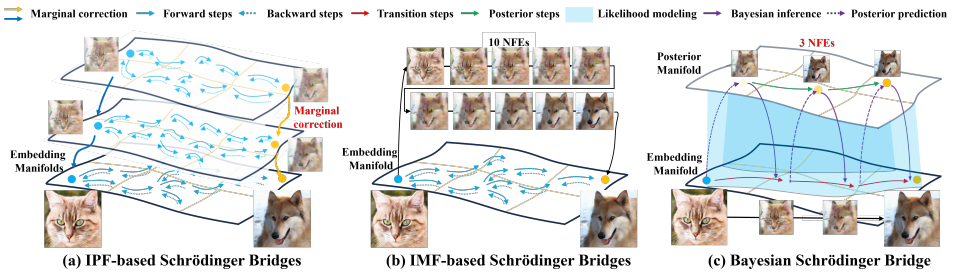

Figure 1: Comparison of IPF-based, IMF-based method, and our Bayesian Schrödinger Bridge.

procedure to characterize the transport process in BSB. The IPBF process mainly consists of two projection steps: posterior projection and Bayesian inference projection. The posterior projection employs a neural network to predict the posterior distribution from the current transition distribution, while the Bayesian inference projection fuses the posterior information with the current transition distribution to update the transition distribution via closed-form distribution conflation. IPBF alternately performs posterior and Bayesian inference projections to accomplish the transport. The posterior and Bayesian inference projections are performed in the parameter space of the distributions, thereby enabling tractable and explicit likelihood-based optimization. To perform IPBF more accurately, BSB employs a Gaussian Mixture Model (GMM) as the transport kernel, which has been identified as a universal approximator Huix et al. (2024). Based on the IPBF procedure in the parameter space using GMMs, BSB offers two key advantages: 1) By explicitly modeling the influence of the posterior in transport, it enables tractable and explicit optimization of the likelihood function without relying on variational lower-bound approximations; 2) We can analytically derive closed-form Bayesian inference, enabling the introduction of novel SDE and ODE solvers to accelerate BSB sampling while maintaining high accuracy.

2 Preliminaries

In this section, we briefly present the elementary notations, background, and underlying ideas of IPF, IMF, and BFN. We use p_{data} and p_{prior} to refer to the two endpoint distributions in the transport process.

Schrödinger Bridge. The SB problem with a Wiener prior in the discrete-time setting De Bortoli et al. (2021); Kholkin et al. (2026) aims to find a transition $q(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_1)$ path that minimizes the divergence from the reference process, given two endpoint distributions $p_{\text{prior}} = p(\mathbf{x}_0)$ and $p_{\text{data}} = p(\mathbf{x}_1)$ and a reference transition $p(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_1)$, is stated as

$$\min_{q \in \Pi_N(p_0, p_1)} \mathrm{KL}(q(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_1) \| p(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_1)), \tag{1}$$

where $\Pi_N(p_0, p_1) \subset \mathcal{P}_{2,ac}(\mathbb{R}^{D \times (N+2)})$ is the subset of discrete stochastic processes with marginals $q(\mathbf{x}_0) = p_0(\mathbf{x}_0)$, $q(\mathbf{x}_1) = p_1(\mathbf{x}_1)$. Kholkin et al. (2026) demonstrates that Eq. 1 can be transformed into the static SB problem:

$$\min_{q \in \Pi(p_0, p_1)} \mathrm{KL}\big(q(\mathbf{x}_0, \mathbf{x}_1) || p(\mathbf{x}_0, \mathbf{x}_1)\big), \tag{2}$$

where $\Pi(p_0, p_1) \subset \mathcal{P}_{2,ac}(\mathbb{R}^{D \times D})$ is the subset of joint distributions $q(\mathbf{x}_0, \mathbf{x}_1)$ s.t. $q(\mathbf{x}_0) = p_0(\mathbf{x}_0)$, $q(\mathbf{x}_1) = p_1(\mathbf{x}_1)$. Eq. 2 is equivalent to the objective of the entropic optimal transport (EOT) problem with the quadratic cost up to an additive constant Cuturi (2013); Peyré et al. (2019); Léonard (2013); Genevay (2019).

Iterative Proportional Fitting. Early SB solutions, especially diffusion bridges Vargas et al. (2021); De Bortoli et al. (2021); Tang et al. (2024), are based on the IPF procedure Kullback (1968). The IPF-based algorithm is started by setting the process $q(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_1) = p(\mathbf{x}_0)p(\mathbf{x}_t, \mathbf{x}_1 | \mathbf{x}_0)$. Then, the algorithm alternates between two types of IPF projections proj_1 and proj_0:

$$q^{2k+1} = \text{proj}_1\big(q^{2k}\big) \overset{\text{def}}{=} p_1(\mathbf{x}_1) \prod_{n=0}^{N} q^{2k}(\mathbf{x}_{t_n} | \mathbf{x}_{t_{n+1}}), q^{2k+2} = \text{proj}_0\big(q^{2k+1}\big) \overset{\text{def}}{=} p_0(\mathbf{x}_0) \prod_{n=1}^{N+1} q^{2k+1}(\mathbf{x}_{t_n} | \mathbf{x}_{t_{n-1}}). \tag{3}$$

In practice, each IPF iteration samples $p(\mathbf{x}_0)$ or $p(\mathbf{x}_1)$ from the neural network trained in the previous epoch. This not only couples the likelihoods of the forward and backward processes but

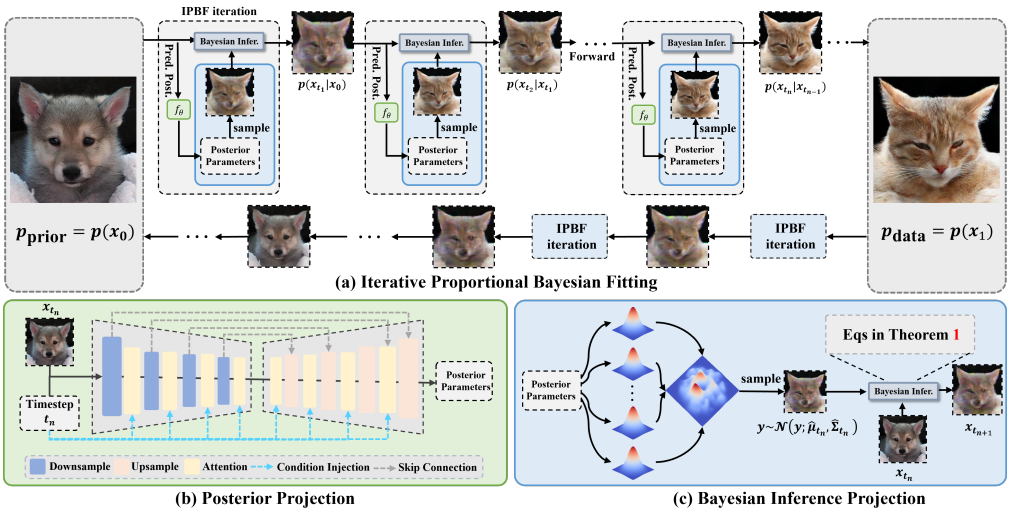

Figure 2: Overview of our Bayesian Schrödinger Bridge. (a) Iterative Proportional Bayesian Fitting: We alternately perform posterior projection and Bayesian inference projection to accomplish the transport in BSB. (b) Posterior Projection: We use neural networks to predict the posterior parameters of the current transition distribution. (c) Bayesian Inference Projection: We update the current transition distribution using posterior observations through Bayesian inference.

also leads to prior forgetting Kholkin et al. (2026), since the network in early training stage cannot accurately capture the terminal distributions in each direction, thereby hindering convergence in subsequent epochs.

Iterative Markovian Fitting. To address the prior forgetting issue in IPF and the limitation of updating only the marginals $q(\mathbf{x}_0)$ and $q(\mathbf{x}_1)$ at each iteration, the Iterative Markovian Fitting (IMF) Peluchetti (2023a); Shi et al. (2023); Gushchin et al. (2024b) directly models a Markov process that satisfies the margin distribution constraints, which does not suffer from the "prior forgetting". The procedure is initialized with any $q^0 \in \Pi_N(p_0, p_1)$:

$$q^{2k+1} = \text{proj}_{\mathcal{R}}(q^{2k}) \overset{\text{def}}{=} q^{2k}(\mathbf{x}_0, \mathbf{x}_1)p(\mathbf{x}_t|\mathbf{x}_0, \mathbf{x}_1), \tag{4}$$

$$q^{2k+2} = \text{proj}_{\mathcal{M}}(q^{2k+1}) \overset{\text{def}}{=} q^{2k+1}(\mathbf{x}_0) \prod_{n=1}^{N+1} q^{2k+1}(\mathbf{x}_{t_n}|\mathbf{x}_{t_{n-1}}) \overset{\text{def}}{=} q^{2k+1}(\mathbf{x}_1) \prod_{n=0}^{N} q^{2k+1}(\mathbf{x}_{t_n}|\mathbf{x}_{t_{n+1}}). \tag{5}$$

The reciprocal projection $\text{proj}_{\mathcal{R}}$ in Eq. 4 creates a new reference process combining the distribution $q(\mathbf{x}_0, \mathbf{x}_1)$ and $p(\mathbf{x}_t|\mathbf{x}_0, \mathbf{x}_1)$. The Markovian projection $\text{proj}_{\mathcal{M}}$ in Eq. 5 uses the set of transitional densities $\{q(\mathbf{x}_{t_n}|\mathbf{x}_{t_{n-1}})\}$ or $\{q(\mathbf{x}_{t_n}|\mathbf{x}_{t_{n+1}})\}$ to create a new Markovian process starting from $q(\mathbf{x}_0)$ or $q(\mathbf{x}_1)$ respectively. At each iteration, IMF approximates the entire path law, making it compatible with modern learning paradigms such as score-based methods and velocity fields shown in De Bortoli et al. (2024).

Bayesian Flow Networks. BFN Graves et al. (2023) introduces a new paradigm by modeling the probability density transition process in the parameter space. Given the distribution of the D-dimension data $p(\mathbf{x} \mid \boldsymbol{\theta}) = \prod_{d=1}^{D} p(x^d \mid \boldsymbol{\theta}^d) = \prod_{d=1}^{D} \mathcal{N}(x \mid \mu, \rho^{-1})$ and the distribution of the posterior which is predicted by neural network ϕ_θ: $p(\mathbf{y}; \mathbf{x}, \alpha) = \prod_{d=1}^{D} p(y^{(d)}; x^{(d)}, \alpha) = \mathcal{N}(\mathbf{y}; \phi_\theta(\mathbf{x}), \alpha^{-1})$, BFN updates the data distribution through closed-form Bayesian inference:

$$\boldsymbol{\mu}_{t_i} \sim \mathcal{N}\big(\frac{\alpha \mathbf{y} + \boldsymbol{\mu}_{t_{i-1}} \rho_{t_{i-1}}}{\rho_{t_i}}, \frac{\alpha}{\rho_{t_i}^2}\mathbf{I}\big), \quad \rho_{t_i} = \rho_{t_{i-1}} + \alpha \tag{6}$$

Xue et al. (2024) points out that BFN is also a variant of DDPM, but operates in the parameter space for noising and denoising, thereby enabling continuous and discrete data modeling.

3 Bayesian Schrödinger Bridge

In this section, we first reformulate the Markov processes in IMF to define the Bayesian Schrödinger Bridge (BSB) via posterior projection and Bayesian inference in Sec. 3.1. We then show in Sec. 3.2

that the Bayesian inference procedure of BSB converges to a solution of the SB problem. Sec. 3.3 further provides the closed-form Bayesian inference, and Sec. 3.4 presents the training algorithm together with higher-order SDE/ODE solvers.

3.1 Main Theorem

The SB problem suffers from challenges in likelihood optimization and numerical instability. To address these challenges, we decompose the optimization procedure of SB into two steps: posterior projection $\text{proj}_{\mathcal{P}}$ and Bayesian inference projection $\text{proj}_{\mathcal{B}}$. The posterior projection maps the current transition distribution onto the posterior manifold predicted by the neural network. The Bayesian inference projection analytically derives the next-step transition distribution by updating the current transition distribution through Bayesian inference using the posterior.

Posterior projection. The posterior projection $\text{proj}_{\mathcal{P}}$ project the current transition distribution to the posterior distribution $q(\mathbf{y}|\mathbf{x})$:

$$q^{3k+2} = \text{proj}_{\mathcal{P}}\big(q^{3k+1}\big) \overset{\text{def}}{=} q^{3k+1}(\mathbf{y}_{t_{n-1}}|\mathbf{x}_{t_{n-1}}) \overset{\text{def}}{=} q^{3k+1}(\mathbf{y}_{t_{n+1}}|\mathbf{x}_{t_{n+1}}). \tag{7}$$

Typically, the parameter of the posterior distribution is predicted by a neural network. The posterior projection explicitly provides a feasible direction of distribution transition for the Bayesian inference projection.

Bayesian inference projection. Based on the posterior projection $\text{proj}_{\mathcal{P}}$, the Bayesian inference projection derives the next transition distribution using the posterior information $\mathbf{y}$ and the current transition distribution:

$$\begin{aligned} q^{3k+3} = \text{proj}_{\mathcal{B}}(q^{3k+1}, q^{3k+2}) &\overset{\text{def}}{=} q^{3k+1}(\mathbf{x}_0) \prod_{n=1}^{N+1} q^{3k+1}\big(\mathbf{x}_{t_n} \mid \mathbf{x}_{t_{n-1}}, \mathbf{y}_{t_{n-1}}\big) \\ &\overset{\text{def}}{=} q^{3k+1}(\mathbf{x}_1) \prod_{n=0}^{N} q^{3k+1}\big(\mathbf{x}_{t_n} \mid \mathbf{x}_{t_{n+1}}, \mathbf{y}_{t_{n+1}}\big). \end{aligned} \tag{8}$$

Based on $\text{proj}_{\mathcal{P}}$ and $\text{proj}_{\mathcal{B}}$, we alternately perform posterior projection and Bayesian inference projection to achieve the transport between p_{prior} and p_{data}:

$$q^{3k+1} = \text{proj}_{\mathcal{R}}\big(q^{3k}\big), q^{3k+3} = \text{proj}_{\mathcal{B}}\big(q^{3k+1}, \text{proj}_{\mathcal{P}}\big(q^{3k+1}\big)\big). \tag{9}$$

Analogous to IMF, $\text{proj}_{\mathcal{R}}$ is used to create an input for training. We specify that all the $\text{proj}_{\mathcal{R}}$, $\text{proj}_{\mathcal{P}}$, and $\text{proj}_{\mathcal{B}}$ are performed in the *parameter space* of the transition and posterior distributions. We refer to this procedure in Eq. 9 as **Iterative Proportional Bayesian Fitting (IPBF)**, as shown in Fig. 2(a). Compared to IMF, which performs transport by implicitly conducting Bayesian inference via neural networks in Markovian projection, IPBF explicitly decomposes the Markov process into posterior projection and Bayesian inference, where the posterior projection explicitly indicates the direction of distribution transport, thereby enabling tractable and explicit likelihood-based optimization and faster convergence of the transport process. Meanwhile, compared with the adversarial–based IMF methods, the IPBF process that performs transport in the parameter space exhibits greater stability.

3.2 Iterative Proportional Bayesian Fitting

For given the $\text{proj}_{\mathcal{B}}$ in Eq. 9, the joint distribution of Bayesian projection is defined as the conflation of prior and posterior:

$$\text{proj}_{\mathcal{B}}(q) = q(\mathbf{x}_0) \prod_{n=1}^{N+1} q\big(\mathbf{x}_{t_n} \mid \mathbf{x}_{t_{n-1}}, \mathbf{y}_{t_{n-1}}\big) = \frac{1}{Z} q(\mathbf{x}_0) \prod_{n=1}^{N+1} \underbrace{q(\mathbf{x}_{t_n} \mid \mathbf{x}_{t_{n-1}})}_{q^{3k+1}} \underbrace{q(\mathbf{y}_{t_{n-1}} \mid \mathbf{x}_{t_{n-1}})}_{\text{proj}_{\mathcal{P}}\left(q^{3k+1}\right)}, \tag{10}$$

where Z is a normalization factor, compared with the IMF, IPBF incorporates Bayesian inference in each density transfer, rather than relying solely on the previous state. This enables the transition process to converge toward the target distribution more rapidly. Consider the $\text{proj}_{\mathcal{R}}$ used to construct the reference process $\text{proj}_{\mathcal{R}}(q) = p(\mathbf{x}_{t_1}, \ldots, \mathbf{x}_{t_N}|\mathbf{x}_0, \mathbf{x}_1) q(\mathbf{x}_0, \mathbf{x}_1)$, given marginal distributions $p_0 = p_{\text{prior}}$ and $p_1 = p_{\text{data}}$ at times $t = 0$ and $t = 1$, respectively, IPBF starts with any reference transition $p(x_{t_1}, \ldots, x_{t_N}|x_0, x_1) q(x_0, x_1)$. Then, it constructs the following sequence of transition processes following Eq. 9: Starting from an arbitrary reference transition, the procedure

alternates between three projections, a reciprocal projection $\text{proj}_{\mathcal{R}}$, which enforces consistency with the reference process, a posterior projection $\text{proj}_{\mathcal{P}}$, which explicitly predicts transition direction (see Fig. 2(b)), and a Bayesian inference projection $\text{proj}_{\mathcal{B}}$ that incorporates posterior to analytically derive the next transition distribution(see Fig. 2(c)). This alternating projection scheme can be naturally interpreted as an optimization procedure over the trajectory space, where each projection minimizes the KL divergence onto the corresponding constraint set. The sequence q^k constructed by the ideal IPBF procedure (with exact posterior projection, see Appendix B) converges in KL to p^*. In particular, $q^k(x_0, x_1)$ convergence to the solution $p^*(x_0, x_1)$ of the static SB :

$$\lim_{k\to\infty} D_{\mathrm{KL}} q^k p^* = 0, \lim_{k\to\infty} D_{\mathrm{KL}} q^k(x_0, x_1) p^*(x_0, x_1) = 0. \tag{11}$$

We prove the convergence of IPBF in Appendix by introducing a lemma showing that q^{3k+3} of IPBF in Eq.9 is equivalent to the $\text{proj}_{\mathcal{M}}$ in IMF, thereby deriving the convergence result by closely following the IMF framework Shi et al. (2023).

3.3 Closed Form Parameter Space Inference

For the reciprocal projection $\text{proj}_{\mathcal{R}}$ as reference process, we model the intermediate distribution $p(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_1)$ as: $p(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_1) \overset{\text{def}}{=} \mathcal{N}\big(\mathbf{x}_t;\, (1-t)\mathbf{x}_0 + t\mathbf{x}_1,\, \epsilon^2\, t(1-t)\,\mathbf{I}\big)$, where $\epsilon > 0$ controls the overall noise intensity. For the posterior and Bayesian inference projection, unlike traditional IMF-based methods that model trajectories with marginal densities $q(\mathbf{x})$, our BSB formulation explicitly conditions the path distribution on parameter variables. To prevent the closed-form probability density from collapsing in high-dimensional spaces, we treat the D-dimensional data as an independent collection of low-dimensional distributions: $q(\mathbf{x}) \overset{\text{def}}{=} q(\mathbf{x} \mid \boldsymbol{\theta}) = \prod_{d=1}^{D} q(x^{(d)} \mid \boldsymbol{\theta}^{(d)})$. In IPBF, we model the data as the Gaussian mixture model (GMM), for each dimension d is a GMM: $q(\mathbf{x}_{t_n} \mid \boldsymbol{\theta}_{t_n}) = \sum_{k=1}^{K} A_{t_n}^k \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_{t_n}^k, \boldsymbol{\Sigma}_{t_n}^k)$, where $\boldsymbol{\theta}_t = \{A_{t_n}^k, \boldsymbol{\mu}_{t_n}^k, \boldsymbol{\Sigma}_{t_n}^k\}$ are GMM parameters, $A^k = \frac{\exp a_k}{\sum_k \exp a_k}$, K is the hyperparameter of the number of mixture components. For $t = 0$, $q(\mathbf{x}_0 \mid \boldsymbol{\theta}_0) = p_{\text{prior}}$, for $t = 1$, $q(\mathbf{x}_1 \mid \boldsymbol{\theta}_1) = p_{\text{data}}$. Unlike other SB solutions that recursively update $\mathbf{x}_{t_n}$ by network, we derive the transition of $\boldsymbol{\theta}_{t_n}$ based on a closed-form Bayesian inference:

Theorem 1 *Given the $\boldsymbol{\theta}_t$, a posterior sample $\mathbf{y} \sim q(\mathbf{y}_{t_n} \mid \mathbf{x}_{t_n}) \overset{\text{def}}{=} q(\mathbf{y} \mid \hat{\boldsymbol{\theta}}) = \mathcal{N}(\mathbf{y}; \hat{\boldsymbol{\mu}}_{t_n}, \hat{\boldsymbol{\Sigma}}_{t_n})$, the updated $q(\mathbf{x}_{t_{n+1}} \mid \mathbf{x}_{t_n}, \mathbf{y}_{t_n}) \overset{\text{def}}{=} q(\mathbf{x}_{t_{n+1}} \mid \boldsymbol{\theta}_{t_{n+1}})$ for proj$_{\mathcal{B}}$ is:*

$$\boldsymbol{\Sigma}_{t_{n+1}}^k = \Big((\boldsymbol{\Sigma}_{t_n}^k)^{-1} + \hat{\boldsymbol{\Sigma}}^{-1}\Big)^{-1}, \boldsymbol{\mu}_{t_{n+1}}^k = \boldsymbol{\Sigma}_{t_{n+1}}^k \Big((\boldsymbol{\Sigma}_{t_n}^k)^{-1} \boldsymbol{\mu}_{t_n}^k + \hat{\boldsymbol{\Sigma}}^{-1} \hat{\mu}\Big) \tag{12}$$

$$A_{t_{n+1}}^k = \frac{A_{t_n}^k\, \mathcal{N}(\hat{\mu}; \boldsymbol{\mu}_{t_n}^k, \boldsymbol{\Sigma}_{t_n}^k + \hat{\boldsymbol{\Sigma}})}{\sum_{j=1}^{K} A_{t_n}^j\, \mathcal{N}(\hat{\mu}; \boldsymbol{\mu}_{t_n}^j, \boldsymbol{\Sigma}_{t_n}^j + \hat{\boldsymbol{\Sigma}})}. \tag{13}$$

where $\hat{\boldsymbol{\theta}}_{t_n}$ is predicted by a neural network f_θ: $\hat{\boldsymbol{\theta}}_{t_n} = f_\theta(\boldsymbol{\theta}_{t_n}, t)$, and we define $\boldsymbol{\Sigma} = \sigma^2\mathbf{I}$, $\hat{\boldsymbol{\Sigma}} = \sigma^2\mathbf{I}$ as a diagonal matrix to simplify all covariance matrices.

Thm. 1 can be proved via the conflation and convolution of a Gaussian and a GMM, with details provided in Appendix. We obtain a GMM approximation to the reverse-time SDE solution by recursively sampling from $q(\mathbf{x}_{t_{n+1}} | \mathbf{x}_{t_n})$ with the following theorem:

Theorem 2 *Given Thm. 1, the general formula for $q(\mathbf{x}_{t_n} | \mathbf{x}_0)$ is given by n times Bayesian inference:*

$$\boldsymbol{\mu}_{t_n}^k = \boldsymbol{\Sigma}_{t_n}^k \Big((\boldsymbol{\Sigma}_0^k)^{-1} \boldsymbol{\mu}_0^k + \hat{\boldsymbol{\Sigma}}^{-1} \sum_{i=1}^{n} \hat{\mu}_i\Big), \tilde{A}_{t_n}^k = A_0^k \prod_{i=1}^{n} \mathcal{N}\big(\hat{\mu}_i;\, \mu_{i-1}^k,\, \mathrm{tr}(\boldsymbol{\Sigma}_{i-1}^k) + \mathrm{tr}(\hat{\boldsymbol{\Sigma}})\big), \tag{14}$$

$$\boldsymbol{\Sigma}_{t_n}^k = \big((\boldsymbol{\Sigma}_0^k)^{-1} + n\, \hat{\boldsymbol{\Sigma}}^{-1}\big)^{-1}, A_{t_n}^k = \frac{\tilde{A}_{t_n}^k}{\sum_{j=1}^{K} \tilde{A}_{t_n}^j}, \tag{15}$$

By replacing the neural network–predicted terms in Thm. 2 with the endpoint distributions $p(\mathbf{x}_0)$ and $p(\mathbf{x}_1)$, we can compute $q(\mathbf{x}_t \mid \mathbf{x}_0)$ or $q(\mathbf{x}_t \mid \mathbf{x}_1)$. Meanwhile, Thm. 2 further shows that the covariance matrices $\boldsymbol{\Sigma}_0^k$ and $\hat{\boldsymbol{\Sigma}}^{-1}$ determine the extent to which the Bayesian inference projection trusts the posterior observations, thereby allowing different covariance schedule of $\hat{\boldsymbol{\Sigma}}^{-1}$ to control the step size of the Bayesian update. We further derive the covariance schedule of the closed-form inference in Appendix.

Algorithm 1: Training algorithm of Bayesian Schrödinger Bridge.

Input: $p(\mathbf{x}_0)$, $p(\mathbf{x}_1)$, number of GMM components K
Output: $p_{0\to1}$ network $f^{\mathrm{F}}_{\boldsymbol{\theta}}$, $p_{1\to0}$ network $f^{\mathrm{B}}_{\boldsymbol{\theta}}$
Initialize all networks
while *not converge* **do**
  Sample a batch $\{\mathbf{x}_0\} \sim p(\mathbf{x}_0)$, $\{\mathbf{x}_1\} \sim p(\mathbf{x}_1)$
  Sample $t \sim U(0,1)$
  Compute $q(\mathbf{x_t} \mid \mathbf{x}_0)$, $q(\mathbf{x_t} \mid \mathbf{x}_1)$ `// Thm. 2`
  Predict $\hat{\boldsymbol{\theta}}_1 \leftarrow f^{\mathrm{F}}_{\theta}(\boldsymbol{\theta}_t, t)$, $\hat{\boldsymbol{\theta}}_0 \leftarrow f^{\mathrm{B}}_{\theta}(\boldsymbol{\theta}_t, t)$
  $\mathcal{L}^{\mathrm{F}} \leftarrow \|\boldsymbol{\mu}_1 - \hat{\boldsymbol{\mu}}_1\|^2$, $\mathcal{L}^{\mathrm{B}} \leftarrow \|\boldsymbol{\mu}_0 - \hat{\boldsymbol{\mu}}_0\|^2$
  Backpropagate and update $f^{\mathrm{F}}_{\boldsymbol{\theta}}$ and $f^{\mathrm{B}}_{\boldsymbol{\theta}}$

3.4 Training and Sampling for BSB

We train BSB by matching the network posterior to its population counterpart. By Thm. 1, the only learnable object at each step is the posterior $q(\mathbf{y} \mid \hat{\boldsymbol{\theta}}) = \mathcal{N}(\mathbf{y}; \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}(t))$; the mixture weights and covariances are updated analytically. The ideal posterior implied by the reference bridge is $\mathcal{N}(\mathbf{y}; \boldsymbol{\mu}^{\mathrm{ref}}, \hat{\boldsymbol{\Sigma}}(t))$, where $\boldsymbol{\mu}^{\mathrm{ref}}$ is the endpoint target along the transport direction. The exact likelihood objective is therefore the expected KL between two Gaussians, available in closed form without any variational bound:

$$\mathcal{L} = \mathbb{E}_{t,\mathbf{x}_0,\mathbf{x}_1} D_{\mathrm{KL}}\left(\mathcal{N}(\boldsymbol{\mu}^{\mathrm{ref}}, \hat{\boldsymbol{\Sigma}}(t)), \mathcal{N}(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}(t))\right) = \mathbb{E}\Big[\tfrac{1}{2\hat{\sigma}^2(t)} \big\|\boldsymbol{\mu}^{\mathrm{ref}} - \hat{\boldsymbol{\mu}}\big\|^2\Big]. \tag{16}$$

Dropping the time-dependent weight $1/2\hat{\sigma}^2(t)$, as is standard for simplified diffusion objectives Ho et al. (2020), yields:

$$\mathcal{L}_{\text{simple}} = \underbrace{\|\boldsymbol{\mu}_1 - \hat{\boldsymbol{\mu}}_1\|^2}_{\text{loss of } p_{\text{prior}}\to p_{\text{data}}} = \underbrace{\|\boldsymbol{\mu}_0 - \hat{\boldsymbol{\mu}}_0\|^2}_{\text{loss of } p_{\text{data}}\to p_{\text{prior}}} \tag{17}$$

Compared to other IMF solutions, our BSB naturally unifies the training objectives across all time steps t. Unlike ASBM, which relies on adversarial losses, BSB provides greater training stability. In contrast to methods such as DSBM and α-DSBM, it requires no additional reparameterization. Moreover, by employing GMM kernel for transition, BSB can capture more fine-grained transition processes.

SDE form and solver. From the general density transition formula in Thm. 2, we can derive the time-reverse SDEs: $d\mathbf{x}_t = \mathbf{v}_\theta(\mathbf{x}_t, t)dt + g(t)d\bar{\mathbf{w}}$, where $\mathbf{v}_\theta(\mathbf{x}_t, t) = f_\theta(\mathbf{x}_t, t) - \mathbf{x}_t$ denotes the transition velocity field, and $g(t)$ can be derived from the variance schedule of the reciprocal projection. This allows us to apply SDE solver steps such as the Euler–Maruyama step.

ODE and high order solver. BSB naturally supports ODE solvers, as closed-form Bayesian inferences provide explicit parameter values at each time step. For the high order solver, previous methods typically adopt Adams-Bashforth-style second-order multistep methods Chen et al. (2025), where the prediction $\mathbf{x}_0$ or $\mathbf{x}_1$ from the two previous steps is extrapolated to obtain the next midpoint, followed by an Euler update. We also incorporate this into BSB. Please refer to Appendix for full sampling algorithm.

4 Related Work and Discussion

IPF-based Diffusion Schrödinger Bridges. Diffusion Schrödinger bridges have been extensively studied in generative modeling Liu et al. (2022b); Somnath et al. (2023); De Bortoli et al. (2021), including endpoint-conditioned methods Heng et al. (2021), Doob's h-transform for distribution translation Liu et al. (2022b); Somnath et al. (2023), and simulation-free formulations Delbracio & Milanfar (2023); Peluchetti (2023a). IPF-based methods De Bortoli et al. (2021); Shi et al. (2023) iteratively approximate SBs by correcting intermediate marginals. DSB De Bortoli et al. (2021) realizes IPF with dual neural networks minimizing alternating KL objectives, whose training targets lack clear physical interpretation. S-DSB Tang et al. (2024) connects DSB to score-based odeling and reparameterizes its objectives using DDPM Ho et al. (2020) and Flow Matching Lipman et al. (2022), improving convergence. Nevertheless, repeated marginal correction in IPF-based bridges may lead to compromise transport solutions Kholkin et al. (2026).

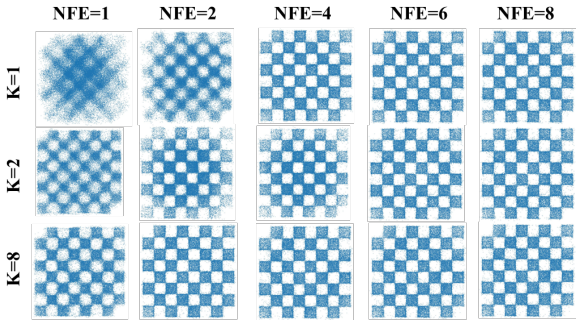

Figure 3: Ablation studies of K number, with 2nd ODE solver.

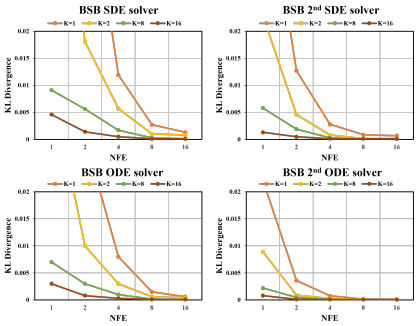

Figure 4: Ablation studies of K number on different solver.

Table 1: Module-wise ablation study on AFHQ. Each variant independently removes one module from Full BSB while keeping the remaining components unchanged.

Method	Full BSB	w/o IPBF	w/o GMM	w/o Solver
FID ↓	**13.29**	17.97 (+4.68)	16.05 (+2.76)	14.19 (+0.90)

Flow Methods and IMF. Flow Matching models deterministic bridges with ODE-based samplers Liu et al. (2022a); Lipman et al. (2022); Pooladian et al. (2023), with DDIM Song et al. (2021) admitting a related discrete-time interpretation Liu et al. (2022a). Stochastic interpolants Albergo et al. (2025) further generalize such transport by introducing stochasticity, closely connecting flow-based formulations to SBs. Recent SB methods therefore increasingly adopt IMF Kholkin et al. (2026); De Bortoli et al. (2024); Zheng et al. (2025); Gushchin et al. (2024b); Zhou et al. (2024), typically using flow-like objectives to predict local transport fields along bridging trajectories. BSB also belongs to this IMF-based family.

Discussion. Our approach differs from prior IMF-based SB methods both theoretically and practically. Theoretically, we reformulate IMF as closed-form Bayesian inference over GMM parameters, with GMMs serving as transport kernels in a continuous parameter space. This provides two key advantages: **1)** GMM priors capture complex density transitions while recovering Gaussian kernels at $K = 1$, preserving consistency with existing IMF theory and extending it to mixture transport. In contrast to LightSB Korotin et al. (2024) and LightSBM Gushchin et al. (2024a), which optimize GMM parameters via empirical risk, our formulation performs explicit Bayesian distribution updates for principled density transport. **2)** Closed-form parameter-space inference enables stable and efficient transition computation, analytical SDE/ODE solvers, and tractable likelihood-based optimization. Practically, BSB unifies training across time by predicting terminal distributions along transport trajectories, improving stability and convergence over SB methods based on time-dependent objective reparameterization.

5 Experiments

In this section, we conduct module-wise ablations on IPSB, the GMM kernel, and the solver, and analyze the effect of the number of GMM components. We then evaluate BSB on 2D point-cloud and image transport tasks, and verify its SB-solving capability on the EOT benchmark. Finally, we assess BSB on general transport tasks to demonstrate its versatility. Full experimental details are provided in the Appendix.

5.1 Ablation Studies

Please refer to Appendix for more ablation studies.

Module-wise ablation. To isolate the contribution of each proposed module, we independently remove IPBF, the GMM kernel, or the proposed solver from Full BSB while keeping all other components unchanged. As shown in Tab. 1, removing IPBF produces the largest degradation. This confirms that the explicit posterior projection and closed-form Bayesian inference in IPBF constitute the primary source of improvement by providing a unified likelihood-based transport objective.

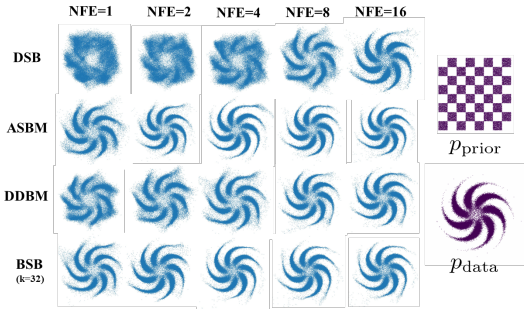

Figure 5: Comparison of 2D data transport.

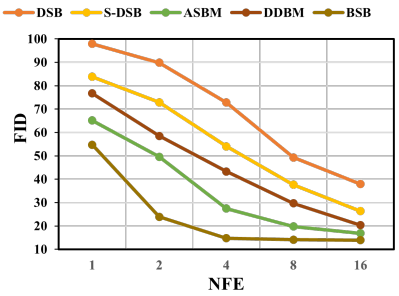

Figure 6: FID under few-step sampling.

Table 2: Comparison on Day2Night, AFHQ datasets.

Method	Day2Night		AFHQ-256		AFHQ-512	
	FID↓	LPIPS↓	FID↓	Percision↑	FID↓	Percision↑
DSB De Bortoli et al. (2021)	41.33	0.893	9.06	0.923	8.41	0.921
S-DSB Tang et al. (2024)	28.94	0.667	7.27	0.928	7.33	0.929
DDBM Zhou et al. (2024)	27.63	0.549	6.82	0.931	6.72	0.932
ASBM Gushchin et al. (2024b)	15.77	0.376	6.29	0.939	5.93	0.941
BSB	**13.28**	**0.361**	**5.81**	**0.942**	**5.53**	**0.946**



Figure 7: Left: cat to dog (top) and day to night translation (bottom); Right: dog to cat (top) and night to day translation (bottom)

Replacing the GMM kernel with a single Gaussian demonstrates that the mixture parameterization provides the second-largest contribution through more expressive modeling of multimodal density transitions. Removing the proposed solver results in an slight FID increase. Its smaller effect is expected because the solver is primarily designed to reduce discretization error and accelerate few-step sampling rather than determine the underlying transport objective.

Number of GMM Components. We investigate the effect of the number of GMM components K under different sampling budgets and solvers. As shown in Fig. 3, increasing K consistently improves few-step sampling. Fig. 4 further shows that larger K reduces KL divergence across solvers, with the largest gains under low-step settings. This indicates that richer GMM kernels better approximate complex target distributions and compensate for the limited expressiveness of few-step posterior updates.

5.2 Experimental Protocal

Please refer to Appendix for more detailed experiments.

2D Gaussian Comparison. We compare BSB with DSB, S-DSB, ASBM, and DDBM on 2D synthetic datasets in Fig. 5. BSB converges substantially faster, benefiting from GMM priors and closed-form Bayesian inference in parameter space, which enable tractable likelihood-based optimization with minimal approximation. Under low NFEs, IPF-based DSB and S-DSB exhibit the largest quality degradation, while BSB consistently outperforms other IMF-based methods. The GMM kernel better captures multimodal and local structures, preserving sample quality with few NFEs, while parameter-space modeling further reduces sensitivity to sampling noise.

Image-to-image Translation. We further compare BSB with existing SB methods on AFHQ and Day2Night datasets. As reported in Tab. 2, BSB consistently achieves the best performance, indi-

Table 3: Comparison with varying D values.

Method	D=2	D=16	D=64	D=128
DSB Shi et al. (2023)	3.36	8.79	20.37	41.67
DSBM Shi et al. (2023)	0.23	5.45	68.9	362
SF^{2}M Tong et al. (2024)	0.31	4.9	319	819
ASBM Gushchin et al. (2024b)	0.13	0.4	1.9	4.7
Ours	**0.12**	**0.33**	**1.57**	**4.02**

Table 4: FID comparison of different tasks.

Method	e→b	coloring	inpainting
I^{2}SB Liu et al. (2023)	7.43	37.85	7.61
DDIB Su et al. (2022)	186.84	48.93	52.88
DDBM Zhou et al. (2024)	1.83	22.83	6.84
ASBM Gushchin et al. (2024b)	1.76	21.37	9.27
Ours	**1.37**	**18.94**	**5.07**

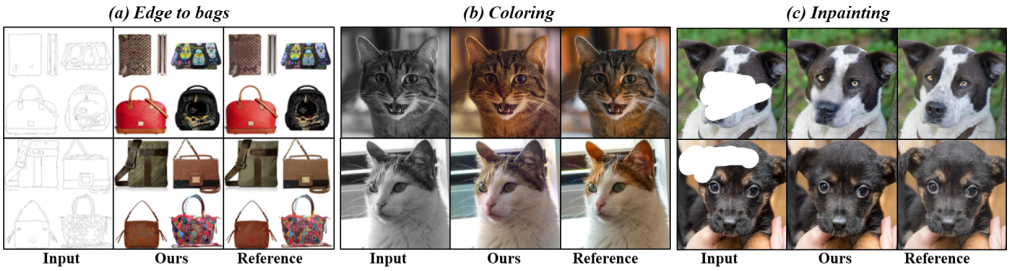

Figure 8: Applications of our BSB to general image-to-image translation tasks.

cating superior transport accuracy and sample quality across both tasks. This advantage is mainly due to the explicit GMM prior and closed-form Bayesian inference, which improve representational flexibility and sampling stability. Under reduced NFEs, Fig. 6 shows that BSB remains highly competitive, consistent with our 2D results and highlighting the robustness of Bayesian updates under limited-step sampling. As illustrated in Fig. 7, BSB also supports direct high-resolution image synthesis and yields consistent transport on both AFHQ and Day2Night.

Evaluation on EOT Benchmark. We evaluate BSB on the EOT benchmark Gushchin et al. (2023), which provides known static SB solutions in different dimensions, using the recommended cB$\mathbb{W}_2^2$-UVP metric (lower is better). As shown in Tab. 3, BSB outperforms others in most settings, validating its effectiveness in approximating SB transport. SF^{2}M degrades substantially in high dimensions, where sample sparsity limits minibatch-based approximation of the optimal coupling. DSBM also deteriorates with dimensionality due to timestep-dependent training objectives, whereas BSB maintains a unified objective aligned with the terminal distribution. Compared with ASBM, BSB further benefits from parameter-space transport with expressive GMM kernels, yielding more accurate density modeling and transport. Appendix further shows that BSB consistently outperforms other methods on the EOT benchmark across different noise levels.

General Translation Tasks. We further evaluate BSB on diverse image-to-image translation tasks, including edge-to-RGB translation, colorization, and inpainting. As shown in Tab. 4, BSB outperforms I^{2}SB and other baselines across all tasks, benefiting from explicit Bayesian posterior updates in IPBF. For edge-to-RGB translation, Fig. 8(a) shows that BSB preserves geometric boundaries while synthesizing plausible textures, enabled by GMM-based transport in parameter space. Fig. 8(b) further demonstrates realistic color recovery from grayscale inputs, indicating effective modeling of the conditional distribution between grayscale and color images. In inpainting, Fig. 8(c) shows coherent reconstruction under varying occlusion, where Bayesian parameter updates condition generation on observed regions. Additional visual comparisons are provided in the Appendix.

6 Conclusion

In this paper, we introduce Bayesian Schrödinger Bridge (BSB), a novel solution to Schrödinger Bridge problem. models the distribution transport in parameter space using closed-form Bayesian inferences of Gaussian mixture models. BSB naturally unifies the training objective across all time steps, ensures precise end-point distribution alignment, and captures fine-grained multi-modal transitions. Compared to both IPF- and IMF-based methods, BSB achieves superior sampling quality, higher stability, and better performance under limited NFEs. We believe that BSB is a significant contribution towards a general framework for distribution translation.

Limitations. Using full covariance matrices in GMM kernels may further improve performance.

REFERENCES

Michael Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

Espen Bernton, Jeremy Heng, Arnaud Doucet, and Pierre E Jacob. Schr\" odinger bridge samplers. *arXiv preprint arXiv:1912.13170*, 2019.

Kenneth F Caluya and Abhishek Halder. Wasserstein proximal algorithms for the schrödinger bridge problem: Density control with nonlinear drift. *IEEE Transactions on Automatic Control*, 67(3): 1163–1178, 2021.

Hansheng Chen, Kai Zhang, Hao Tan, Zexiang Xu, Fujun Luan, Leonidas Guibas, Gordon Wetzstein, and Sai Bi. Gaussian mixture flow matching models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 9783–9802, 2025.

Yongxin Chen, Tryphon T Georgiou, and Michele Pavon. Optimal transport in systems and control. *Annual Review of Control, Robotics, and Autonomous Systems*, 4, 2021.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 34:17695–17709, 2021.

Valentin De Bortoli, Iryna Korshunova, Andriy Mnih, and Arnaud Doucet. Schrodinger bridge flow for unpaired data translation. *Advances in Neural Information Processing Systems*, 37:103384–103441, 2024.

Mauricio Delbracio and Peyman Milanfar. Inversion by direct iteration: An alternative to denoising diffusion for image restoration. *arXiv preprint arXiv:2303.11435*, 2023.

Chris Finlay, Augusto Gerolin, Adam M Oberman, and Aram-Alexandre Pooladian. Learning normalizing flows from entropy-kantorovich potentials. *arXiv preprint arXiv:2006.06033*, 2020.

Robert Fortet. Résolution d'un système d'équations de m. schrödinger. *Journal de Mathématiques Pures et Appliquées*, 19(1-4):83–105, 1940.

Aude Genevay. *Entropy-regularized optimal transport for machine learning*. PhD thesis, Paris Sciences et Lettres (ComUE), 2019.

Alex Graves, Rupesh Kumar Srivastava, Timothy Atkinson, and Faustino Gomez. Bayesian flow networks. *arXiv preprint arXiv:2308.07037*, 2023.

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. *Advances in Neural Information Processing Systems*, 36:18932–18963, 2023.

Nikita Gushchin, Sergei Kholkin, Evgeny Burnaev, and Alexander Korotin. Light and optimal schrödinger bridge matching. In *Forty-first International Conference on Machine Learning*, 2024a.

Nikita Gushchin, Daniil Selikhanovych, Sergei Kholkin, Evgeny Burnaev, and Alexander Korotin. Adversarial schrödinger bridge matching. *Advances in Neural Information Processing Systems*, 37:89612–89651, 2024b.

Jeremy Heng, Valentin De Bortoli, Arnaud Doucet, and James Thornton. Simulating diffusion bridges with score matching. *arXiv preprint arXiv:2111.07243*, 2021.

Jonathan Ho, Ajay Jain, Pieter Abbeel, Pieter Abbeel, Pieter Abbeel, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Tom Huix, Anna Korba, Alain Durmus, and Eric Moulines. Theoretical guarantees for variational inference with fixed-variance mixture of gaussians. *ICML*, 2024.

Sergei Kholkin, Grigoriy Ksenofontov, David Li, Nikita Kornilov, Nikita Gushchin, Alexandra Suvorikova, Alexey Kroshnin, Evgeny Burnaev, and Alexander Korotin. Diffusion & adversarial schr\" odinger bridges via iterative proportional markovian fitting. *The Fourteenth International Conference on Learning Representations*, 2026.

Aleksandr Korotin, Nikita Gushchin, and Evgeny Burnaev. Light schrödinger bridge. In *International Conference on Learning Representations*, volume 2024, pp. 15354–15383, 2024.

Solomon Kullback. Probability densities with given marginals. *The Annals of Mathematical Statistics*, 39(4):1236–1243, 1968.

Christian Léonard. A survey of the schr\" odinger problem and some of its connections with optimal transport. *arXiv preprint arXiv:1308.0215*, 2013.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A Theodorou, Weili Nie, and Anima Anandkumar. I ^{2} sb: Image-to-image schrödinger bridge. *International Conference on Machine Learning*, 2023.

Xingchao Liu, Chengyue Gong, Qiang Liu, Qiang Liu, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022a.

Xingchao Liu, Lemeng Wu, Mao Ye, and Qiang Liu. Let us build bridges: Understanding and extending diffusion generative models. *arXiv preprint arXiv:2208.14699*, 2022b.

Michele Pavon, Giulio Trigila, and Esteban G Tabak. The data-driven schrödinger bridge. *Communications on Pure and Applied Mathematics*, 74(7):1545–1573, 2021.

Stefano Peluchetti. Diffusion bridge mixture transports, schrödinger bridge problems and generative modeling. *Journal of Machine Learning Research*, 24(374):1–51, 2023a.

Stefano Peluchetti. Non-denoising forward-time diffusions. *arXiv preprint arXiv:2312.14589*, 2023b.

Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky T. Q. Chen. Multisample flow matching: Straightening flows with minibatch couplings. 2023.

Erwin Schrödinger. Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique. In *Annales de l'institut Henri Poincaré*, volume 2, pp. 269–310, 1932.

Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion schrödinger bridge matching. *Advances in neural information processing systems*, 36:62183–62223, 2023.

Vignesh Ram Somnath, Matteo Pariset, Ya-Ping Hsieh, Maria Rodriguez Martinez, Andreas Krause, and Charlotte Bunne. Aligned diffusion schr\" odinger bridges. *arXiv preprint arXiv:2302.11419*, 2023.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *International Conference on Learning Representations*, 2021.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Xuan Su, Jiaming Song, Chenlin Meng, and Stefano Ermon. Dual diffusion implicit bridges for image-to-image translation. In *The Eleventh International Conference on Learning Representations*, 2022.

Zhicong Tang, Tiankai Hang, Shuyang Gu, Dong Chen, and Baining Guo. Simplified diffusion schr\" odinger bridge. *arXiv preprint arXiv:2403.14623*, 2024.

Alexander Y Tong, Nikolay Malkin, Kilian Fatras, Lazar Atanackovic, Yanlei Zhang, Guillaume Huguet, Guy Wolf, and Yoshua Bengio. Simulation-free schrödinger bridges via score and flow matching. In *International Conference on Artificial Intelligence and Statistics*, pp. 1279–1287. PMLR, 2024.

Francisco Vargas, Pierre Thodoroff, Austen Lamacraft, and Neil Lawrence. Solving schrödinger bridges via maximum likelihood. *Entropy*, 23(9):1134, 2021.

Kaiwen Xue, Yuhao Zhou, Shen Nie, Xu Min, Xiaolu Zhang, Jun Zhou, and Chongxuan Li. Unifying bayesian flow networks and diffusion models through stochastic differential equations. In *International Conference on Machine Learning*, pp. 55656–55681, 2024.

Kaiwen Zheng, Guande He, Jianfei Chen, Fan Bao, and Jun Zhu. Diffusion bridge implicit models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Linqi Zhou, Aaron Lou, Samar Khanna, and Stefano Ermon. Denoising diffusion bridge models. In *The Twelfth International Conference on Learning Representations*, 2024.