East Asian Journal on Applied Mathematics
doi: 10.4208/eajam.2025-085.181125

Vol. **xx**, No. x, pp. 1-35
xxx 2026

Solving Schrödinger Bridge Problem via Continuous Normalizing Flow

Yang Jing[1,*], Lei Li[1,2,3] and Jingtong Zhang[1]

[1]*School of Mathematical Sciences, Shanghai Jiao Tong University, Shanghai 200240, P.R.China.*
[2]*Institute of Natural Sciences, MOE-LSC, Shanghai Jiao Tong University, Shanghai 200240, P.R.China.*
[3]*Shanghai Artificial Intelligence Laboratory.*

Received 28 April 2025; Accepted (in revised version) 18 November 2025.

Abstract. The Schrödinger bridge problem (SBP), which can be understood as an entropy-regularized optimal transport, seeks to compute stochastic dynamic mappings connecting two given distributions. SBP has shown significant theoretical importance and broad practical potential, with applications spanning a wide range of interdisciplinary fields. While theoretical aspects of the SBP are well-understood, practical computational solutions for general cases have remained challenging. This work introduces a computational framework that leverages continuous normalizing flows and score matching methods to approximate the drift in the dynamic formulation of the SBP. The learned drift term can be used for building generative models, opening new possibilities for applications in probability flow-based methods. We also provide a rigorous Γ—convergence analysis for our algorithm, demonstrating that the neuron network solutions converge to the theoretical ones as the regularization parameter tends to infinity. Lastly, we validate our algorithm through numerical experiments on fundamental cases.

AMS subject classifications: 49Q22, 68T07

Key words: Schrödinger bridge, continuous normalizing flow, Γ-convergence.

1. Introduction

Finding mappings between two probability distributions is the focus of many modern machine learning applications, from generative modeling to understanding physical systems [34]. While deterministic methods such as optimal transport (OT) have offered robust frameworks for addressing this task, many real-world phenomena are inherently governed by stochastic processes, necessitating more sophisticated approaches. The Schrödinger bridge problem, originally proposed by Schrödinger [44], addresses this challenge by seeking the most likely stochastic process connecting two probability distributions. SBP is of

*Corresponding author. *Email addresses:* sharkjingyang@sjtu.edu.cn (Y. Jing), leili2010@sjtu.edu.cn (L. Li), sffred@sjtu.edu.cn (J. Zhang)

great theoretical importance in mathematical physics, as they provide deeper insights into the evolution of physical systems and the study of quantum information. It also demonstrates broad practical potential across various applications in interdisciplinary fields such as modeling natural stochastic dynamical systems [16, 43], image processing [31] and shape correspondence [9].

Let $\Omega \subset \mathbb{R}^d$ be a convex domain in $\mathbb{R}^d$ with a smooth boundary, which could possibly be unbounded in general. The Schrödinger problem can be stated as follows [26]: For two probability distributions ρ_0 and ρ_1 on Ω, find a path measure $\mathbb{P}^* \in \mathscr{P}(C([0,T];\Omega))$ such that

$$\mathbb{P}^* = \underset{\mathbb{P}\in\mathscr{D}(\rho_0,\rho_1)}{\operatorname{argmin}} \ \mathbb{KL}(\mathbb{P}\|\mathbb{Q}), \tag{1.1}$$

where $\mathbb{Q} \in \mathscr{P}(C([0,T];\Omega))$ is a reference measure. Here, $\mathscr{P}(C([0,T];\Omega))$ indicates the set of measures on the path space $C([0,T];\Omega)$ (continuous curves taking values in Ω), and $\mathscr{D}(\rho_0,\rho_1) \subset \mathscr{P}(C([0,T];\Omega))$ is the set with marginal measures ρ_0 at time $t=0$ and ρ_1 at time $t=T$. The KL divergence between two probability measure μ and ν on Ω is defined by

$$\mathbb{KL}[\mu\|\nu] = \begin{cases} \displaystyle\int_\Omega \log\left(\frac{d\mu}{d\nu}\right) d\mu, & \text{if} \quad \mu \ll \nu, \\ \infty, & \text{otherwise,} \end{cases}$$

where $d\mu/d\nu$ denotes the Radon-Nikodym derivative of μ with respect to ν. Note that the KL divergence is non-negative by Jensen's inequality and achieves zero only if $\mu = \nu$. Moreover, it is a convex functional with respect to both arguments. In practice, $\mathbb{Q}$ is often taken to be the path measure induced by Brownian motion or some relatively simple diffusion process.

Previous research has demonstrated a close relationship between SBP and the optimal stochastic control problem [40], thereby offering an alternative approach to solving the SBP. Given that $\mathbb{Q} = \sqrt{2}\sigma\mathbb{W}$, where $\mathbb{W}$ is the path measure induced by the standard Brown motion, then solving SBP (1.1) is equivalent to finding the optimal stochastic control of the optimization problem — cf. [40],

$$\begin{aligned} \min_{u} \quad & \mathbb{E}\left[\frac{1}{2}\int_0^T |u_t(X_t,t)|^2 dt\right], \\ \text{s.t.} \quad & dX_t = u_t(X_t,t)dt + \sqrt{2}\sigma\, dW_t, \\ & X_0 \sim \rho_0, \quad X_T \sim \rho_1, \end{aligned} \tag{1.2}$$

or the problem with the Fokker-Planck equation constraint

$$\begin{aligned} \min_{\rho,u} \quad & \frac{1}{2}\int_0^T \int_\Omega \rho(x,t)|u(x,t)|^2\, dx\, dt, \\ \text{s.t.} \quad & \partial_t\rho + \nabla\cdot(\rho u) = \sigma^2\Delta\rho, \\ & \rho(x,0) = \rho_0(x), \quad \rho(x,T) = \rho_1(x). \end{aligned} \tag{1.3}$$

Here the single bars $|\cdot|$ represent Euclidean length in $\mathbb{R}^d$. We will use double bars $\|\cdot\|$ for norms involving the spatial integral $\int_\Omega$, and single bars $|\cdot|$ for Euclidean length in $\mathbb{R}^d$, and for the absolute values of scalars in the following text. Under mild assumptions on ρ_0 and ρ_1, Schrödinger bridge problem (1.3) admits a unique minimizer $(\rho, u = \nabla\Phi)$ [26], where (ρ, Φ) is the classical solutions of the Euler-Lagrange equations

$$\begin{aligned}\partial_t\rho + \nabla\cdot(\nabla\Phi\rho) &= \sigma^2\Delta\rho,\\ \partial_t\Phi + \frac{1}{2}|\nabla\Phi|^2 &= -\sigma^2\Delta\Phi.\end{aligned}$$

If $\mathbb{Q}$ is the measure generated by the Langevin diffusion $dX_t = -\nabla U(X_t)dt + \sqrt{2}\sigma dW$, where $U(x)$ is a given potential, then solving corresponding SBP is equivalent to finding the optimal control u such that the diffusion process $dX_t = [u(X_t,t) - \nabla U(X_t)]\,dt + \sqrt{2}\sigma dW$ bridges the distributions ρ_0 and ρ_1 and minimizes the L^2 cost

$$\begin{aligned}\min_{\rho,u}\quad & \frac{1}{2}\int_0^T\int_\Omega \rho(x,t)|u(x,t)|^2 dx dt,\\ \text{s.t.}\quad & \partial_t\rho + \nabla\cdot\big(\rho(u-\nabla U)\big) = \sigma^2\Delta\rho,\\ & \rho(x,0) = \rho_0(x),\quad \rho(x,1) = \rho_1(x).\end{aligned} \tag{1.4}$$

Traditional approaches to solving the SBP primarily rely on iterative proportional fitting [25] and Sinkhorn-Knopp algorithms [47, 52]. Iterative methods suffer from cumulative numerical errors [8], and crucially, reliance on mesh-based discretization makes them computationally intractable for high-dimensional problems.

Recent advances in deep learning, particularly in the realm of continuous normalizing flows (CNFs) and diffusion models, have opened new possibilities for addressing such challenges [5,50], which have demonstrated remarkable success in modeling complex probability distributions and have natural connections to differential equations. One can model the distribution using CNFs and enforce the constraints of the associated PDEs (such as Fokker-Plank equation or continuity equation) by minimizing the residuals [7, 54], or incorporate regularization during CNF training to ensure the minimizers converge to the target solutions [10, 35]. Thus, the framework of CNFs is particularly promising for the SBP due to its ability to track density evolution and its natural alignment with transport-based formulations. The recently developed flow-matching technique [29, 32] could effectively find a particular CNF velocity field to connect two distributions.

Making use of the score-matching and flow-matching techniques, the learning-based approaches for Schrödinger bridge problems become a very active research direction in recent years [17, 30, 31, 46, 51]. For example, De Bortoli *et al.* [1] framed score-based generative models as SBP and proposed the diffusion Schrödinger bridge (DSB), an algorithm that approximates the classical IPF procedure by iteratively learning the drifts of the forward and backward SDEs. By decomposing the path measures into a mixture of two-point bridges, the flow-matching technique has been built in to solve the Schrödinger bridge problems. For example, Shi *et al.* [46] has generalized the DSB to the diffusion Schödinger bridge matching method. Related works include the iterated diffusion bridge mixture procedure [38]

and the unified bridge algorithm [23]. An iterative scheme based on flow-matching for a class of generalized Schrödinger bridge problems has been proposed in [30]. The iterative schemes, however, could suffer from simulation-inference mismatch, leading to error accumulation. The iterative Markovian fitting proposed in [46] is claimed to preserve initial and terminal distributions, but real simulation and sampling also introduce errors which could still accumulate through iterations. Recently, Tong *et al.* [51] proposed a simulation-free method to learn the drift that avoids the iterative process. Using the flow-matching technique, the problem is then changed to solving the regularized optimal transport problem. However, solving the OT problem itself is equivalently difficult. Though the Sinkhorn method could be used, the problem is still challenging if the number of sample points is large, and moreover sampling from the learned transport plan is also not straightforward.

In this work, we aim to solve the SBP using the framework of original CNFs, employing the dynamic formulation of the SBP. The terminal distribution has a known density up to a normalizing constant, while we allow the initial distribution to be represented by some samples. The hard terminal constraint is replaced by a relaxed soft penalty using the KL divergence. The Schrödinger bridge problem with relaxed terminal constraint was theoretically studied concurrently by Garg *et al.* [12]. The SDE backbone of the SBP dynamic formulation introduces challenges in estimating density using the change of variables formula (see the discussion in Section 3). To address this, we propose a hypothetical velocity field to transform the Fokker-Planck equation into the transport equation form, aligning the SBP with the framework of CNFs. Building on this, our algorithm introduces a mesh-free framework that leverages CNFs to solve the dynamic formulation of the SBP. This approach offers several key advantages:

- Though the terminal constraint is relaxed into a soft penalty based on KL divergence, the approach is non-iterative and there is no error accumulation for the initial and terminal constraints.
- It provides a theoretically grounded method for finding the connecting stochastic differential equation (SDE) without requiring mesh discretization, making it scalable to higher dimensions. Since it is mesh-free, the approach learns a continuous representation of the solution, enabling efficient sampling and density estimation at arbitrary points in the path.
- The framework naturally incorporates the stochastic aspects of the SBP through a combination of CNFs and score matching, providing a more faithful representation of the underlying probabilistic dynamics.

A portion of our algorithm draws inspiration from a series of works on regularized CNFs. By recognizing that both optimal transport and CNFs can be framed within a transport framework, Finlay *et al.* [10] were the first to introduce optimal transport regularization in CNFs and Onken *et al.* [35] proposed OT-Flow, an improved version of the CNFs combined with optimal transport. The added regularization enforces the solutions to approximate the theoretical solutions of the traditional OT problem. Jing *et al.* [21] also employed a similar methodology to design a machine learning framework for learning geodesics under

the spherical Wasserstein-Fisher-Rao metric based on unbalanced optimal transport theory. Note that the SBP can be interpreted as an entropy-regularized optimal transport problem, which motivates us to approach SBP using a framework similar to that of regularized CNFs.

Another fundamental challenge in adapting machine learning approaches to classical mathematical problems lies in ensuring that the computational solutions faithfully approximate the theoretical ones. In our context, while CNFs provide a feasible computational framework, we must establish that our numerical solutions converge to the true solution of the SBP. We address this through the mathematical framework of Γ-convergence, which allows us to study the convergence of minimizers in optimization problems. By formulating our computational approach as a sequence of relaxed optimization problems with increasing penalty on the terminal distribution constraint, we can rigorously demonstrate that the solutions of our algorithm converge to those of the classical SBP. This theoretical foundation bridges the gap between computational efficiency and mathematical rigor.

The rest of the paper is organized as follows. Section 2 is devoted to a brief review of the necessary concepts and tools established in previous works. Next, we present our approach and derive the necessary equations to solve the dynamic formulation of the SBP in Section 3. In Section 4, we propose our algorithm to solve the SBP, including training hypothetical velocity field via CNFs and recovering optimal control through score matching. A convergence analysis is developed in Section 5. In Section 6, we provide some numerical experiments to validate the algorithm.

2. Preliminaries

In this section, we review several necessary concepts, tools and related preliminary results for our use later. In particular, we will introduce the CNFs, the Γ-convergence and the Benamou-Brenier functional.

2.1. Continuous normalizing flows

As mentioned above, we will develop a practical algorithm for the SBP using CNFs. The CNFs [5, 10, 13] are a class of generative models based purely on particle transportation. They aim to construct continuous and invertible mappings between an arbitrary distribution ρ_0 and the standard reference distribution ρ_1 through the flows induced by a velocity field $v(\cdot, t)$, which is represented by a neural network.

For a given time T, one is trying to obtain a mapping $z : \mathbb{R}^d \times [0, T] \to \mathbb{R}^d$, which defines a continuous evolution $x \mapsto z(x, t)$ driven by the velocity field for every $x \in \mathbb{R}^d$ such that the distribution at T matches ρ_1. Define

$$\ell(x, t) := \log |\det \nabla z(x, t)|,$$

then $z(x, t)$ and $\ell(x, t)$ satisfy the following ODE system:

$$\partial_t \begin{bmatrix} z(x,t) \\ \ell(x,t) \end{bmatrix} = \begin{bmatrix} v(z(x,t), t; \boldsymbol{\theta}) \\ \operatorname{tr}(\nabla v(z(x,t), t; \boldsymbol{\theta})) \end{bmatrix}, \quad \begin{bmatrix} z(x,0) \\ \ell(x,0) \end{bmatrix} = \begin{bmatrix} x \\ 0 \end{bmatrix}. \tag{2.1}$$

Then the density $\rho(\cdot,t)=(z(\cdot,t))_{\#}\rho_0$ (the pushforward of ρ_0 under the flow map $z(\cdot,t)$) satisfies

$$\log\rho_0(x)=\log\rho\big(z(x,t),t\big)+\log|\det\nabla z(x,t)| \quad \text{for all} \quad x\in\mathbb{R}^d. \tag{2.2}$$

To train the dynamics, CNFs minimize the expected negative log-likelihood given by the right-hand-side in (2.1), or equivalently the KL divergence between target distribution and final distribution under the constraint (2.1) [13,36,37,41]

$$J=\mathbb{KL}\big[\rho(\boldsymbol{z}(\boldsymbol{x},T))\|\rho_1(\boldsymbol{z}(\boldsymbol{x},T))\big].$$

Notably, CNFs exhibit strong capability in modeling density changes, and we leverage this advantage of CNFs to develop our algorithm for solving the SBP.

2.2. Γ− convergence

Another main part of this work is to establish the theoretical foundation for our approach. Especially, we employ Γ-convergence theory [3] to demonstrate that the relaxed formulation converges to the original SBP. The Γ-convergence has been proved to be a powerful framework to analyze the convergence of optimization problems and their corresponding optimizers.

Definition 2.1. *Let X be a topological space. Let (f_n) be a sequence of functionals on X. Define*

$$\Gamma-\limsup_{n\to\infty} f_n(x)=\sup_{N_x}\limsup_{n\to\infty}\inf_{y\in N_x} f_n(y),$$
$$\Gamma-\liminf_{n\to\infty} f_n(x)=\sup_{N_x}\liminf_{n\to\infty}\inf_{y\in N_x} f_n(y),$$

where N_x ranges over all neighbourhoods of x. If there exists a functional f defined on X such that

$$\Gamma-\limsup_{n\to\infty} f_n=\Gamma-\liminf_{n\to\infty} f_n=f,$$

then we say the sequence (f_n) Γ-converges to f.

Proposition 2.1. *Any cluster point of the minimizers of a Γ-convergent sequence (f_n) is a minimizer of the corresponding Γ-limit functional f.*

The properties of Γ-convergence make it an optimal framework for analyzing the convergence of minimizers across a sequence of optimization problems. While direct verification of Γ-convergence through Definition 2.1 often presents significant technical challenges, the following two propositions offers an alternative approach to establishing Γ-convergence [4,53].

Proposition 2.2. *Suppose that X is a first-countable topological space. It holds that*

$$\inf_{x^n\to x}\limsup_{n\to\infty} f_n(x_n)=\Gamma\text{-}\limsup_{n\to\infty} f_n,$$
$$\inf_{x^n\to x}\liminf_{n\to\infty} f_n(x_n)=\Gamma\text{-}\liminf_{n\to\infty} f_n,$$

where $\inf_{x^n \to x}$ *indicates that the infimum is taken over all sequences* $\{x^n\}$ *that converge to* x. *Consequently, if*

$$\inf_{x^n \to x} \limsup_{n\to\infty} f_n(x_n) = \inf_{x^n \to x} \liminf_{n\to\infty} f_n(x_n) := f$$

exists, then (f_n) *is* Γ*-convergent to* f.

The following gives a criterion for the Γ-convergence of a special case which is applicable for our problem later.

Proposition 2.3. *Let* f_n *and* f *be functionals defined on a first-countable topological space* X *such that* $f_n \uparrow f$ *pointwise and* f_n *is lower-semicontinuous. Then* f_n *is* Γ*-convergent to* f.

The proof of Proposition 2.3 is attached in Appendix A.

2.3. Benamou-Brenier functional

The Benamou-Brenier functional gives the rigorous definition of the L^p norm $\int |v|^p \rho(dx)$ of the velocity field with general probability measure ρ in the optimal transport related problems, which is defined as follows:

$$\mathscr{B}_p(\rho, m) := \sup\left\{ \int_X a\, d\rho + \int_X b \cdot dm : (a,b) \in C_b(X; K_q) \right\},$$

where

$$K_q := \left\{ (a,b) \in \mathbb{R} \times \mathbb{R}^d : a + \frac{1}{q}|b|^q \le 0 \right\}, \quad \frac{1}{p} + \frac{1}{q} = 1.$$

The following proposition characterizes key properties of the Benamou-Brenier functional.

Proposition 2.4. *The Benamou-Brenier functional* $\mathscr{B}_p$ *is convex and lower semi-continuous on the space* $\mathscr{M}(X) \times \mathscr{M}^d(X)$ *with the weak convergence topology. Moreover, the following properties hold:*

1. $\mathscr{B}_p(\rho, m) \ge 0$.
2. *If both* ρ *and* m *are absolutely continuous with respect to a same positive measure* λ *on* X, *we can write*

$$\mathscr{B}_p(\rho, m) = \int_X f_p\big(\rho(x), m(x)\big) d\lambda(x),$$

where $\rho(x)$ *and* $m(x)$ *are densities with respect to* λ, *and* $f_p : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ *is defined by*

$$f_p(t,x) := \sup_{(a,b)\in K_q} (at + b \cdot x) = \begin{cases} \dfrac{1}{p}\dfrac{|x|^p}{t^{p-1}}, & \text{if} \quad t > 0, \\ 0, & \text{if} \quad t = 0, \quad x = 0, \\ +\infty, & \text{if} \quad t = 0, \quad x \neq 0 \quad \text{or} \quad t < 0. \end{cases}$$

3. $\mathscr{B}_p(\rho, m) < +\infty$ *only if* $\rho \geq 0$ *and* $m \ll \rho$.

4. *For* $\rho \geq 0$ *and* $m \ll \rho$, *we have* $m = u \cdot \rho$ *and* $\mathscr{B}_p(\rho, m) = \int (1/p)|u|^p d\rho$.

In Section 5, we will employ the Benamou-Brenier functional to define the relevant functionals and examine the properties of feasible points.

3. Governing Equations in the CNF Framework

The dynamic formulation (1.2) (or (1.4)) of the Schrödinger bridge problem naturally leads to a stochastic differential equation framework similar to the CNF setting. Below, we mainly focus on (1.2) as the example, as the generalization to others is similar.

We begin by considering the following SDE:

$$dX_t = u(X_t, t)dt + \sqrt{2}\sigma dW_t. \tag{3.1}$$

Similar to (2.2), the law at t is given by

$$\rho(z, t) = \mathbb{E}_W \Big[\rho_0\big(X_t^{-1}(z)\big) \big(\det \nabla X_t(x) \big)^{-1} \Big],$$

where $\mathbb{E}_W$ indicates the expectation over the stochastic mapping $x \mapsto z = X_t(x)$, given by the strong solution of the SDE (3.1). Exactly due to this expectation, different stochastic trajectories are then coupled together, making the loss construction using the KL divergence in [5, 35] infeasible. One may use other penalty like the maximum mean discrepancy (MMD) but the practical performance seems not excellent. Hence, our approach is to transform the SDE into an ODE formulation that gives the same time marginal distributions. This approach allows us to leverage CNFs to develop an algorithm for solving the SBP.

Consider the corresponding Fokker-Planck equation of (3.1)

$$\partial_t \rho + \nabla \cdot (\rho u) = \sigma^2 \Delta \rho. \tag{3.2}$$

By noting that $\Delta \rho = \nabla \cdot \nabla \rho$, we can rewrite (3.2) in the form of continuity equation

$$\partial_t \rho + \nabla \cdot \big[\rho \big(u - \sigma^2 \nabla \log \rho \big) \big] = 0. \tag{3.3}$$

The transformation from (3.2) to (3.3) is crucial as it reveals the underlying transport structure of the problem. By expressing the diffusion term using the score function $\nabla \log \rho$, we separate the deterministic and stochastic components of the dynamics, which will be essential for our neural network approximation. The time-varying hypothetical velocity field

$$f := u - \sigma^2 \nabla \log \rho$$

represents a crucial decomposition of the dynamics. While similar transformations have been used in related contexts [19, 45], our application to the SB problem enables a novel

computational approach through neural network approximation. It is important to note that f is not explicit, as we do not have access to the score function $\nabla \log \rho$ at this stage.

The complete dynamical system we derive combines three key elements: the particle positions $z(x,t)$, the log-determinant of the transformation

$$\ell(x,t) := \log|\det \nabla z(x,t)|, \tag{3.4}$$

and the score function

$$s(x,t) := \nabla \log \rho\big(z(x,t),t\big). \tag{3.5}$$

This coupled system allows us to simultaneously track the evolution of particles and their associated probability density, while maintaining the necessary regularity conditions for the SB problem.

By transforming Fokker-Plank equation (3.2) into the form of transport equation allows us to track the evolution of particles $z(x,t)$ with corresponding $\ell(x,t)$ along the trajectories the same as (2.1). One can also track the score function $s(x,t)$ along the trajectories with the help of following proposition:

Proposition 3.1. *The log-determinant of the transformation $\ell(x,t)$ introduced in* (3.4) *satisfies the following ODE:*

$$\partial_t \ell(x,t) = \nabla_z \cdot f\big(z(x,t),t\big).$$

Meanwhile, the score function along the trajectory $s(x,t)$ defined in (3.5) *satisfies*

$$\partial_t s(x,t) = -\nabla(\nabla \cdot f\big(z(x,t),t)\big) - \mathscr{J}_f^\top\big(z(x,t),t\big)s(x,t),$$

where $\mathscr{J}_f$ denotes the Jacobian matrix of the vector-valued function f, and is given by $\mathscr{J}_f = (\nabla_z f(z,t))^\top$ with

$$(\nabla_z f)_{ij} = \partial_i f_j.$$

A proof can be found in Appendix A. Thus, the whole ODE system in our framework can be given by

$$\begin{aligned}
\partial_t \begin{bmatrix} z(x,t) \\ \ell(x,t) \\ s(x,t) \end{bmatrix} &= \begin{bmatrix} f(z,t) \\ \nabla_z \cdot f(z(x,t),t) \\ -\nabla_z(\nabla_z \cdot f(z(x,t),t)) - \mathscr{J}_f^\top(z(x,t),t)s(x,t) \end{bmatrix}, \\
\begin{bmatrix} z(x,0) \\ \ell(x,0) \\ s(x,T) \end{bmatrix} &= \begin{bmatrix} x \\ 0 \\ \nabla \log \rho_1(z(x,T)) \end{bmatrix}.
\end{aligned} \tag{3.6}$$

Here ρ_1 is the target distribution whose density expression is known, possibly up to a normalizing constant. In applications, ρ_1 is often a standard Gaussian, but clearly other distributions are also allowed in this framework. The first two equations is solved from 0 to T, while the third equation is supposed to be solved backward from T to 0 after having the

terminal position $z(x,T)$ of particles. Recall the dynamic formulation of Schödinger bridge problem, we are devoted to solve a relaxed version

$$\begin{aligned} \min_{\rho,u} \quad & \frac{1}{2}\int_0^T \int_\Omega \rho(x,t)|u(x,t)|^2 dx dt + \alpha \mathbb{KL}\big[\rho(x,T)||\rho_1(x)\big], \\ \text{s.t.} \quad & \partial_t \rho + \nabla\cdot(\rho u) = \sigma^2 \Delta \rho, \\ & \rho(x,0) = \rho_0(x). \end{aligned} \tag{3.7}$$

Note that the above formulation can be easily modified if the reference measure $\mathbb{Q}$ is given by other simple diffusion process. For example, if $\mathbb{Q}$ is the measure generated by the Langevin diffusion stated in (1.4), it suffices to adjust the hypothetical velocity field f by adding the given prior drift. We refer the readers to Section 6.3 for further details and corresponding numerical experiments.

In above problem, KL divergence plays a soft constraint for terminal condition. As the regularization coefficient α goes to infinity, the minimizers of (3.7) should converge to the ones of classical Schödinger bridge problem. We will resolve this in Section 5 with the help of $\Gamma-$convergence.

4. CNF Based Machine Learning Framework

In this section, we propose a machine learning framework to solve the SBP via continuous normalizing flows and score matching algorithms. The training process of hypothetical velocity field is illustrated in Section 4.1. Additional implementation details and techniques are also discussed. Then, in Section 4.2, we recover the optimal drift based on the pre-trained hypothetical velocity field using classical score matching methods.

4.1. Training hypothetical velocity field

In our case, we aim to connect a real data distribution to the normal distribution via the Schröndinger bridge. A neural network is employed to approximate the hypothetical velocity field. The network is trained to minimize a loss function that combines the KL divergence with the L^2 norm of the drift as a regularization term

$$\begin{aligned} &\mathscr{J} = \alpha \mathscr{J}_{\mathbb{KL}} + \mathscr{J}_B, \\ &\mathscr{J}_{\mathbb{KL}} = \mathbb{KL}\big[\rho(z(x,T))\|\rho_1(z(x,T))\big], \\ &\mathscr{J}_B = \frac{1}{2}\int_0^T \int_\Omega \big|f\big(z(x,t),t\big) + \sigma^2 \nabla \log \rho\big(z(x,t),t\big)\big|^2 \rho_0(x) dx dt. \end{aligned}$$

Applying the same push-forward trick as in CNFs [13] to simplify the KL term yields

$$\begin{aligned} &\mathbb{KL}\big[\rho(\boldsymbol{z}(\boldsymbol{x},T))\|\rho_1(\boldsymbol{z}(\boldsymbol{x},T))\big] \\ =& \int_\Omega \log\Big(\frac{\rho(\boldsymbol{z}(\boldsymbol{x},T))}{\rho_1(\boldsymbol{z}(\boldsymbol{x},T))}\Big)\rho\big(\boldsymbol{z}(\boldsymbol{x},T)\big)\det\big(\nabla \boldsymbol{z}(\boldsymbol{x},T)\big)\mathrm{d}\boldsymbol{x} \end{aligned}$$

$$= \int_\Omega \log\Big(\frac{\rho_0(\boldsymbol{x})}{\rho_1(\boldsymbol{z}(\boldsymbol{x},T))\det(\nabla \boldsymbol{z}(\boldsymbol{x},T))}\Big)\rho_0(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$
$$= \int_\Omega \log\big(\rho_0(\boldsymbol{x})\big)\rho_0(\boldsymbol{x})\mathrm{d}\boldsymbol{x} + \mathbb{E}_{\rho_0(x)}\big[-\log\big(\rho_1(\boldsymbol{z}(\boldsymbol{x},T))\big) - \log\det\big(\nabla \boldsymbol{z}(\boldsymbol{x},T)\big)\big].$$

Note that the first part is a constant independent of the neural network and can therefore be dropped during training. By substituting the normal distribution expression for ρ_1, we obtain the final loss function

$$\begin{aligned} \mathscr{J} &= \mathbb{E}_{\rho_0(x)}\big[\alpha C(x,T) + B(x,T)\big], \\ C(x,T) &= -\ell(x,T) + \frac{1}{2}|\boldsymbol{z}(\boldsymbol{x},T)|^2 + \frac{d}{2}\log(2\pi), \\ B(x,T) &= \frac{1}{2}\int_0^T \big|f(z(x,t),t) + \sigma^2 s(x,t)\big|^2 dt, \end{aligned} \tag{4.1}$$

which can be approximated with the Monte Carlo approximation. Here, $\ell(x,t)$ and $s(x,t)$ are given in (3.4) and (3.5). Algorithm 4.1 presents our training process for the hypothetical velocity field.

Algorithm 4.1 Solving SBP via Normalizing Flow

Require: Data samples $\{x_i\}_{i=1}^N$, time interval $[0,T]$, initializing network f_θ.
1: **for** number of training iterations **do**
2: Solve ODE system (3.6) with f_θ to obtain $z(x_i,T)$, $\ell(x_i,T)$ and $s(x_i,t)$.
3: Calculate cost function $\mathscr{J} = \alpha\mathscr{J}_{\mathbb{KL}} + \mathscr{J}_B$.
4: Use ADAM optimizer [24] optimizer to update network parameter of f_θ.
5: **end for**

In training, particularly in high-dimensional cases, several methods can be used to reduce the computational cost.

1. **Random mini-batch method.** Inspired by the random batch method in stochastic gradient descent [2] and interacting particle systems [20], one can select a mini-batch of indices to give an estimation of $\nabla \cdot f$.

2. **Unbiased linear-time log-density estimation.** In general, for a vector-valued function $f : \mathbb{R}^d \to \mathbb{R}^d$, computing the trace of the Jacobian $\mathrm{Tr}(\nabla f)$ (or equivalently, the divergence $\nabla \cdot f$) exactly incurs a computational cost of $\mathscr{O}(d^2)$. To avoid computing each entry of the diagonal of the Jacobian explicitly, we can leverage Hutchinson's trace estimator, as employed in CNFs [10, 13]. This approach enables efficient approximation of $\nabla \cdot f$ when solving the second ODE in (3.6), or when estimating data density during inference. The Hutchinson's trace estimator provides an unbiased estimate of the trace of a matrix A as follows:

$$\mathrm{Tr}(A) = \mathbb{E}_{\rho(\lambda)}\big[\lambda^T A\lambda\big],$$

where λ is a d-dimensional random vector with zero mean $\mathbb{E}[\lambda] = 0$ and identity

covariance $\mathbf{Cov}(\lambda) = I$. Common choices for the distribution $\rho(\lambda)$ include the standard Gaussian distribution or the Rademacher distribution. To compute $\mathrm{Tr}(\nabla f)$, we use automatic differentiation to calculate $\nabla(\lambda^T f)$ first. Then we get an unbiased estimate of trace by multiplying λ again

$$\mathrm{Tr}(\nabla f) = \nabla \cdot f \approx \nabla(\lambda^T f) \cdot \lambda.$$

3. **Matrix-Free Jacobian Computations.** A significant computational challenge in our method arises from the ODE system in Eq. (3.6), which requires evaluating terms involving the Jacobian of the neural network J_f. Specifically, the dynamics of the log-determinant $\ell(x,t)$ and the score $s(x,t)$ depend on the divergence of the vector field $\nabla \cdot f$ and products involving $J_f^\top$. A naive approach would require explicitly forming the Jacobian matrix $J_f \in \mathbb{R}^{d\times d}$, an operation with quadratic complexity that is computationally prohibitive in high dimensions. To overcome this bottleneck, we employ a matrix-free approach that leverages the automatic differentiation capabilities of PyTorch. The function `torch.autograd.grad` allows for the efficient computation of vector-Jacobian products (VJPs) of the form $v^\top J_f$ without instantiating the full matrix. We also use this capability to implement Hutchinson's trace estimator for the divergence, where $\nabla \cdot f = \mathrm{Tr}(J_f) \approx \lambda^\top J_f \lambda$ is computed with a single VJP per sample. This matrix-free methodology is critical for the scalability of our algorithm to high-dimensional problems.

CNFs are trained to minimize the loss function in (4.1), which involves the solution to an initial value problem with an ODE parameterized by θ. Chen *et al.* [5] employ the adjoint method to compute the gradient and subsequently perform backpropagation, which allows one to train the models with constant memory cost as a function of depth. Typically, for any scalar loss function which operates on the solution to an initial value problem

$$L\big(\mathbf{z}(t_1)\big) = L\left(\int_{t_0}^{t_1} f\big(\mathbf{z}(t), t; \theta\big) dt\right).$$

Then Pontryagin [39] shows that its derivative takes the form of another initial value problem

$$\frac{dL}{d\theta} = -\int_{t_1}^{t_0} \left(\frac{\partial L}{\partial \mathbf{z}(t)}\right)^T \frac{\partial f(\mathbf{z}(t), t; \theta)}{\partial \theta} dt. \tag{4.2}$$

The quantity $\partial L/\partial z(t)$ is known as the adjoint state of the ODE. Chen *et al.* [5] proposed to use a black-box ODE solver to compute $z(t_1)$, and then another call to a solver to compute (4.2) with the initial value $\partial L/\partial z(t_1)$. The adjoint method can be employed to solve high-dimensional cases of the SBP.

4.2. Recovering the optimal drift

It is important to note that we only have access to the score function along the trajectory. To recover the drift u, we can employ classical score matching algorithms to approximate the score function with a pre-trained hypothetical velocity field f_θ.

Given i.i.d. samples $\{x_1, x_2, \cdots, x_N\} \subset \mathbb{R}^d$ from an underlying distribution π, the classical score matching [18] uses a neuron network $s_\phi(x) \in \mathbb{R}^d$ to approximate the score function $\nabla \log \pi(x)$ and minimize the L^2 loss function $\mathbb{E}_\pi |s_\phi(x) - \nabla \log \pi(x)|_2^2/2$, which can be shown to be equivalent to the following up to a constant:

$$J_{SM}(\phi) \triangleq \mathbb{E}_\pi \left[\operatorname{tr}\big(\nabla_x s_\phi(x)\big) + \frac{1}{2}|s_\phi(x)|^2 \right].$$

After applying the Hutchinson's trace estimator mentioned above, the sliced score matching [49] minimize the following loss function:

$$J_{SSM}(\phi) \triangleq \mathbb{E}_\pi \mathbb{E}_{\rho(\lambda)} \left[\lambda^T \nabla_x s_\phi(x) \lambda + \frac{1}{2}\big(\lambda^T s_\phi(x)\big)^2 \right],$$

where $\nabla_x s_\phi$ denotes the Hessian of the score function. $\lambda \sim \rho_\lambda$ and $x \sim \pi$ are independent. For the sake of computational convenience, one may model the distribution ρ_λ of λ as either a multivariate Rademacher distribution or a multivariate normal distribution, as both satisfy $\mathbf{E}(\lambda) = 0$ and $\mathbf{Cov}(\lambda) = I$.

In our case, to recover the drift, we can parameterize it as a neural network $u_\phi(x,t)$ and expect the function

$$s_\phi(x,t) := \frac{u_\phi(x,t) - f_\theta(x,t)}{\sigma^2} \tag{4.3}$$

to approximate the ground truth score $\nabla \log \rho_t(x)$. Note that the score function $s_\phi(x,t)$ here is different from that in Eq. (3.5) as the x variable here indicates a general location at current time point instead of the initial location for a trajectory. Similar to noise conditional score networks (NCSNs) [48] and diffusion models [15], the joint score matching objective is given as follows:

$$J_{JSM}(\phi) \triangleq \mathbb{E}_{t\sim\mathscr{U}(0,1)} \mathbb{E}_{x_t\sim\rho_t} \mathbb{E}_{\lambda\sim\rho(\lambda)} w(t) \left[\lambda^T \nabla_x s_\phi(x_t,t) \lambda + \frac{1}{2}\big(\lambda^T s_\phi(x_t,t)\big)^2 \right],$$

where $w(t)$ is a predefined weight function. Thus, as long as we obtain a pre-trained hypothetical velocity field f_θ, we can sample from standard Gaussian and generate trajectories, which will be used in the training of score matching to obtain drift function u_ϕ. Algorithm 4.2 presents the process of recovering optimal control via score matching. We remark that if σ is very small, one may want to train s_ϕ as an independent neural network and then recover $u_\phi = f_\theta + \sigma^2 s_\phi$ for better stability.

Algorithm 4.2 Drift Recovering in SBP

Require: Pre-trained hypothetical velocity f_θ, a time discretization $0 = t_0 < t_1 < \cdots < t_L = T$, initializing network s_ϕ.

1: Sample from Gaussian distribution and use f_θ to generate trajectories $\{x_i^{t_k}\}_{i=1}^N$ by solving the first equation of the ODE system (3.6) backward in time.
2: **for** number of training iterations **do**
3: Select a batch of sample pairs $(x_i^{t_k}, t_k)$ from trajectories.

4: Calculate cost function J_{JSM}.
5: Minimize $J_{JSM}(\phi)$ with ADAM [24] optimizer.
6: **end for**
7: Calculate the drift u_ϕ by (4.3) with trained s_ϕ.
8: **return** Optimal drift u_ϕ.

5. Convergence of the Optimization Problem to SBP

We introduced a relaxation factor α for the terminal distribution in (4.1) to render the loss function computationally tractable. We now rigorously demonstrate that the solution to our relaxed optimization problem converges to that of the original Schrödinger bridge problem as α approaches infinity by leveraging the framework of Γ-convergence, as outlined earlier. See similar analysis of the convergence from OT-Flow to optimal transport in [22].

For the convenience of analysis, we will assume that $\Omega \subset \mathbb{R}^d$ is a bounded domain with smooth boundary in this section.

5.1. Mathematical formulation of problems

Following [42], we make use of the variables (ρ, m) where $m = \rho u$ so that there is better convexity. Then the optimization problem of SBP (1.3) can be reformulated as

$$\begin{aligned} \min_{\rho,m} \quad & \int_0^T \int_\Omega \frac{|m|^2}{2\rho} dx dt + \chi_E^\infty, \\ \text{s.t.} \quad & \partial_t \rho + \nabla \cdot m = \sigma^2 \Delta \rho, && x \in \Omega, \\ & (m - \sigma^2 \nabla \rho) \cdot n = 0, && x \in \partial\Omega, \\ & \rho(x,0) = \rho_0(x), && x \in \Omega, \end{aligned} \tag{5.1}$$

where E is the set of the terminal constraints $\rho(x,T) = \rho_1(x)$ and χ_E^∞ is the indicator function

$$\chi_E^\infty = \begin{cases} 0, & \text{if} \quad x \in E, \\ +\infty, & \text{otherwise.} \end{cases}$$

The corresponding optimization problem of our algorithm (3.7) is given by

$$\begin{aligned} \min_{\rho,m} \quad & \int_0^T \int_\Omega \frac{|m|^2}{2\rho} dx dt + \alpha \mathbb{KL}\big[\rho(x,T) || \rho_1(x)\big], \\ \text{s.t.} \quad & \partial_t \rho + \nabla \cdot m = \sigma^2 \Delta \rho, && x \in \Omega, \\ & (m - \sigma^2 \nabla \rho) \cdot n = 0, && x \in \partial\Omega, \\ & \rho(x,0) = \rho_0(x), && x \in \Omega. \end{aligned} \tag{5.2}$$

In general, ρ and m are measures and $\rho(\cdot, T)$ may not be well-defined either before the time regularity in time has been verified. Hence, we need to put the above problem into a rigorous framework.

To proceed, we first specify the topological space and associated constraints. The topology spaces we work on are

$$X := L^1\big([0,T];\mathscr{P}(\Omega)\big), \quad Y := L^1\big([0,1];\mathscr{M}(\Omega)^d\big).$$

Here, $\mathscr{P}(\Omega)$ is the set of probability measures on Ω and $\mathscr{M}(\Omega)^d$ is the set of $\mathbb{R}^d$-valued Radon measures on Ω. Then we equip the product space $X \times Y$ for (ρ, m) with the product weak topology, i.e., $(\rho_n, m_n) \to (\rho, m)$ means: for all $f \in C_b([0,T]\times\Omega;\mathbb{R})$ and $g \in C_b([0,T]\times\Omega;\mathbb{R}^d)$, one has

$$\int_0^T\int_\Omega f\,d\rho_n + \int_0^T\int_\Omega g\,dm_n \;\to\; \int_0^T\int_\Omega f\,d\rho + \int_0^T\int_\Omega g\,dm.$$

Obviously $X \times Y$ is closed and first-countable. Then the constraint $\partial_t\rho + \nabla\cdot m = \sigma^2\Delta\rho$ introduces a subspace $\mathscr{H}$ of X

$$\begin{aligned}\mathscr{H} := \Bigg\{(\rho,m)\in X\times Y : &-\int_0^T\left(\int_\Omega(\partial_t\varphi)\rho(dx)+\nabla\varphi\cdot m(dx)\right)dt\\ &-\int_\Omega\varphi(x,0)\rho_0(dx) = \sigma^2\int_0^T\int_\Omega\Delta\varphi\cdot\rho(dx)\,dt, \forall\varphi\in C_b^2\big([0,T]\times\Omega\big),\\ &\left.\frac{\partial\varphi}{\partial n}\right|_{\partial\Omega} = 0, \varphi(\cdot,T) = 0\Bigg\}.\end{aligned}$$

The subspace $\mathscr{H}$ contains the boundary conditions implicitly and naturally. This space is quite large for ρ, compared to the original requirement $\rho \in \mathscr{P}(C([0,T];\Omega))$. However, as we will see later, this space is more convenient for analysis and it will not include more solutions. Clearly $\mathscr{H}$ is closed due to the fact that the constraints are linear.

Similarly, E can now be rigorously defined as follows:

$$\begin{aligned}E := \Bigg\{(\rho,m)\in X\times Y : &-\int_0^T\left(\int_\Omega(\partial_t\varphi)\rho(dx)+\nabla\varphi\cdot m(dx)\right)dt\\ &+\int_\Omega\varphi(x,T)\rho_1(dx) - \int_\Omega\varphi(x,0)\rho_0(dx)\\ &= \sigma^2\int_0^T\int_\Omega\Delta\phi\cdot\rho(dx)\,dt, \forall\varphi\in C_b^2(\bar\Omega\times[0,T]), \left.\frac{\partial\varphi}{\partial n}\right|_{\partial\Omega} = 0\Bigg\}.\end{aligned}$$

One should note that E is a closed subspace of $\mathscr{H}$, and this is the rigorous definition of the set E introduced in (5.1).

Next, we treat the issue for $\rho(\cdot,T)$. We following the approach in [22] to resolve this. We consider the following:

$$\bar\rho(\cdot,T) := \lim_{\delta\to0}\frac{1}{\delta}\int_{T-\delta}^T\rho_s\,ds.$$

Clearly, if $t \mapsto \rho_t$ has a version that is continuous at T, $\bar{\rho}(\cdot, T)$ is well-defined. With this, we define the functional

$$G(\rho, \rho_1) := \begin{cases} \mathbb{KL}\big[\bar{\rho}(x,T) \| \rho_1(x)\big], & \text{if} \quad \bar{\rho}(\cdot,T) \quad \text{exists}, \\ \infty, & \text{otherwise.} \end{cases}$$

$G(\rho, \rho_1)$ actually corresponds to the KL divergence term if we later focus on the feasible points, for which ρ is actually continuous in time.

We now can define the functionals F_α and $F_\infty : \mathscr{H} \to \mathbb{R} \cup \infty$ corresponding to (5.2) and (5.1) respectively as

$$\begin{aligned} F_\alpha(\rho, m) &= \int_0^T \mathscr{B}_2(\rho_t, m_t)\, dt + \alpha G(\rho, \rho_1), \\ F_\infty(\rho, m) &= \int_0^T \mathscr{B}_2(\rho_t, m_t)\, dt + \chi_E^\infty. \end{aligned} \tag{5.3}$$

With Proposition 2.4, optimization for the functional F_α and F_∞ are then the rigorous definitions of the optimization problems (5.2) and (5.1), respectively. Similar as shown in [22], they are lower semi-continuous with respect to the topology considered.

5.2. Regularity of feasible points

It is well-known that the minimizer of F_∞ is the solution of the Schrödinger bridge problem, and the existence of solution to SBP is a well-established result [26]. Using the solutions, it is to construct a feasible point of F_α. Hence, the feasible points of F_α and F_∞ actually exists.

We show that the feasible points of F_α and F_∞ have good time regularity properties by the following proposition.

Proposition 5.1. *If (ρ, m) is a feasible solution of (5.3) (for F_α or F_∞), $m \ll \rho$ and the Radon-Nikodym derivative $u = dm/d\rho$ is in $L^1([0,T]; L^2(\rho_t))$, and there is a version of ρ such that $t \mapsto \rho_t$ is continuous in $\mathscr{W}_2(\Omega)$, where*

$$\mathscr{W}_2 := \left\{ \mu \in \mathscr{P}(\Omega) \;\middle|\; \int |x|^2 \mu(dx) < \infty \right\}.$$

Moreover, if the initial entropy is finite

$$H(\rho_0) := \int \rho_0 \log \rho_0 \, dx < \infty,$$

then ρ is absolutely continuous in $\mathscr{W}_2(\Omega)$.

Proof. If (ρ, m) is a feasible solution, then

$$\int_0^T \mathscr{B}_2(\rho_t, m_t)\, dt < +\infty.$$

By Proposition 2.4, $m \ll \rho$ for a.e. t and $u_t = dm_t/d\rho_t$ satisfies

$$\int_0^T \|u\|^2_{L^2(\rho_t)}\,dt = \int_0^T |u_t|^2\rho_t(dx) = 2\int_0^T \mathscr{B}_2(\rho_t, m_t)\,dt < +\infty.$$

So that $u \in L^1([0,T]; L^2(\rho_t))$.

Since Ω is bounded, the continuity in $\mathscr{W}_2$ is equivalent to the weak continuity, which follows from the fact that ρ is a weak solution to $\partial_t\rho + \nabla\cdot(\rho u) = 0$ as it is in $\mathscr{H}$. In fact, we take $\varphi(x,t) = \phi(x)h(t)$ such that $h(T) = 0$, $\phi \in C_b^2$ and $\partial\phi/\partial n = 0$, one has

$$\begin{aligned}&-\int_0^T h'(t)\int_\Omega \phi\rho_t(dx)\,dt - \int_0^T h(t)\int \nabla\phi\cdot u\rho(dx)\,dt - h(0)\int \phi(x)\rho_0(dx)\\ &= \sigma^2\int_0^T h(t)\int \Delta\phi\rho(dx)\,dt.\end{aligned}$$

This holds for any $h \in C^1[0,T]$ with $h(T) = 0$. This means that $t \mapsto g(t) := \int \phi\rho_t(dx)$ has a weak derivative and the weak derivative is given by

$$\frac{d}{dt}\int \phi\rho_t(dx) = \int \nabla\phi\cdot u\rho(dx) + \sigma^2\int \Delta\phi\rho(dx).$$

Since $u \in L^2(0,T; L^2(\rho_t))$, the right hand side is integrable in $(0,T)$. Hence, $t \mapsto g(t)$ is absolutely continuous in $(0,T)$. Moreover, by the equation above, one has $g(0+) = \int \phi(x)\rho_0(dx)$. This indicates that

$$\int \phi\rho_{t+h}(dx) \;\to\; \int \phi\rho_t(dx), \quad h \to 0$$

for any $\phi \in C_b^2(\Omega)$. By a standard density argument, the test function can be generalized to C_b class and thus $t \mapsto \rho_t$ is weakly continuous and thus continuous in $\mathscr{W}_2(\Omega)$, due to the boundedness of Ω.

For the absolute continuity of ρ, recall

$$f_t = u_t - \sigma^2\nabla\log\rho_t.$$

Then, ρ_t satisfies the continuity equation with this equivalent drift f_t. By the standard result in optimal transport theory [42, Theorem 5.14], if

$$\int_0^T \|f_t\|^2_{L^2(\rho_t)}\,dt < \infty,$$

then ρ_t is absolutely continuous in $\mathscr{W}_2$. By the integrability of u_t, it suffices to verify

$$\int_0^T \|\nabla\log\rho_t\|^2_{L^2(\rho_t)}\,dt < \infty.$$

Consider the entropy

$$H(\rho_t) = \int_\Omega \rho_t \log \rho_t \, dx.$$

Formal calculations give

$$\begin{aligned}\frac{d}{dt}H(\rho_t) &= \int_\Omega \rho_t u_t \cdot \nabla \log \rho_t \, dx - \sigma^2 \|\nabla \log \rho_t\|^2_{L^2(\rho_t)} \\ &\le \|u_t\|_{L^2(\rho_t)} \|\nabla \log \rho_t\|_{L^2(\rho_t)} - \sigma^2 \|\nabla \log \rho_t\|^2_{L^2(\rho_t)} \\ &\le C\|u_t\|^2_{L^2(\rho_t)} - \frac{1}{2}\sigma^2 \|\nabla \log \rho_t\|^2_{L^2(\rho_t)}.\end{aligned}$$

On bounded domain, $H(\rho_t)$ is always bounded from below. Integrating over $[0,T]$, combining the fact that

$$\int_0^T \|u_t\|_{L^2(\rho_t)} \, dt < +\infty$$

and the boundedness of $H(\rho_0)$, one has

$$\int_0^T \int_\Omega |\nabla \log \rho_t|^2 \rho_t(dx)\, dt < +\infty.$$

This calculation can be made rigorous by mollification and taking the mollification parameter to zero. Hence, $\int_0^T \|f_t\|^2_{L^2(\rho_t)} \, dt$ is bounded, and ρ_t is absolutely continuous. □

According to this proposition, ρ_t is continuous and thus

$$G(\rho, \rho_1) = \mathbb{KL}\big[\rho(x,T) \| \rho_1(x)\big].$$

That means the functional we considered indeed is the one we desire.

5.3. Convergence of functionals

We now establish the result concerning the Γ-convergence and the behavior of the minimizers.

Theorem 5.1. *Assume that F_α (resp. F_∞) has at least one feasible point over $\mathscr{H}$, then F_α (resp. F_∞) has a unique global minimizer over $\mathscr{H}$. Moreover, F_α Γ-converges to F_∞, and for any sequence $\alpha_I \to \infty$, the minimizers of F_{α_I} have a subsequence converging to the minimizers of F_∞.*

Proof. Since F_α is lower-semicontinuous (by Proposition 2.4), and $F_\alpha \uparrow F_\infty$ pointwise, one concludes that by Proposition 2.3 that F_α Γ-converges to F_∞.

By Proposition 2.4, we know that $\mathscr{B}_2(\rho, m) \ge 0$, hence F_α and F_∞ are bounded from below. Then there exists $F^*_\alpha \in [0, +\infty)$ such that

$$F^*_\alpha = \inf_{(\rho,m)\in\mathscr{H}} F_\alpha(\rho, m).$$

Consider a feasible minimizing sequence (ρ_n, m_n) such that $F_\alpha(\rho_n, m_n) \to F_\alpha^*$. Then by the Hölder inequality, we have

$$\begin{aligned}\|m_n\| &= \int_0^T \int_\Omega |m_n(dx,t)|\,dt \\ &\le \left(2\int_0^T \int_\Omega \rho_n(dx)dt \int_0^T \mathscr{B}_2(\rho_n, m_n)dt\right)^{1/2} \\ &= \sqrt{2T}\left(\int_0^T \mathscr{B}_2(\rho_n, m_n)dt\right)^{1/2} \\ &\le \sqrt{2TF_\alpha(\rho_n, m_n)}, \end{aligned} \tag{5.4}$$

where m_n is signed measure. We have put dx in the first arguments of m_n, which indicates integration with respect to x. Eq. (5.4) implies that $\sup_n \|m_n\| < +\infty$, and further $\sup_n(\|m_n\| + \|\rho_n\|) < +\infty$. According to the Banach-Alaoglu theorem, there exists a subsequence (ρ_{n_k}, m_{n_k}) converging weakly to (ρ^*, m^*) in $\mathscr{H}$. Together with the lower-semicontinuity of F_α, we have

$$F_\alpha^* \le F_\alpha(\rho^*, m^*) \le \liminf_{k\to\infty} F_\alpha(\rho_{n_k}, m_{n_k}) = F_\alpha^*.$$

Hence, the minimizer of F_α exists. Similarly, the minimizer of F_∞ exists. Moreover, since F_α is strictly convex (by the strict convexity of the Benamou-Brenier functional), the minimizer must be unique.

Let (ρ^α, m^α) be the minimizer of F_α, and (ρ^∞, m^∞) be the minimizer of F_∞. It is clear that

$$F_\alpha(\rho^\alpha, m^\alpha) \le F_\alpha(\rho^\infty, m^\infty) \le F_\infty(\rho^\infty, m^\infty),$$

then

$$\sup_\alpha \left(\|m^\alpha\| + \|\rho^\alpha\|\right) \le \sqrt{2TF_\infty(\rho^\infty, m^\infty)} + T < +\infty.$$

Again by the Banach-Alaoglu theorem, there exists a subsequence $(\rho^{\alpha_k}, m^{\alpha_k})$ converging weakly to (ρ, m) in $\mathscr{H}$. Since F_α Γ-converges to F_∞, (ρ, m) is a minimizer of F_∞. □

We also note that the original problem requires $\rho \in \mathscr{P}(C([0,T];\Omega))$, while we seek ρ in a much larger space. However, due to the uniqueness of the minimizer, if the original Schrödinger bridge problem has a solution, it must be the solution in our framework. Hence, the mathematical formulation here is fine.

6. Numerical Experiments

In this section, we consider some numerical examples to demonstrate our deep learning framework for solving the SBP. Our code is available at `https://github.com/sharkjingyang/SchB`. In low dimensional experiments, we simply adopted the multilayer perceptron (MLP) framework, since this architecture already possesses sufficient

approximation capability to represent the hypothetical velocity field in low dimensional cases. High order adaptive ODE solver is employed and an implementation of the adjoint method is utilized for back-propagation. Meanwhile we ensure tolerance is set low enough so numerical error from ODE solver can be negligible.

Experiment setup

Hyper-parameters. In all experiments, we consider the time interval $[0,1]$, i.e. $T=1$. We take $\alpha=10$ as a default choice to make a balance for terms in the cost function.
Architectures. We use sine and cosine functions for time embedding, which are then processed through an MLP. The resulting embeddings are concatenated with the input data. The network architecture consists of three Concat-Squash linear layers with tanh activation functions.
Optimizer. We choose ADAM [24] with a learning rate of 1e-3.

6.1. 1-D Gaussian mixture

As a first example, we employ the 1-D Gaussian mixture problem to evaluate the performance of our algorithm in solving the SBP, including verifying convergence properties and observing typical behaviors. We sample from the Gaussian mixture

$$\rho_0(x)=\frac{1}{2}\cdot\frac{1}{\sqrt{2\pi}}e^{-(x+3)^2/2}+\frac{1}{2}\cdot\frac{1}{\sqrt{2\pi}}e^{-(x-3)^2/2}$$

and use our framework to learn the SB dynamics from ρ_0 to standard normal distribution ρ_1. Fig. 1 illustrates the forward process from a Gaussian mixture to a normal distribution, presented in histogram form. With a pre-trained hypothetical velocity field f_θ, one can sample trajectories and employ score matching algorithms to approximate the score function. As a result, the optimal drift can be recovered as well. Fig. 2 shows ODE trajectories and SDE paths between two marginal distributions.

We also compare our results with a traditional optimization method. We employ a similar numerical scheme described in [21] to transform the optimization problem into discrete form and use a primal-dual hybrid algorithm to solve the SBP on the grid. The details of numerical implementation of the primal-dual algorithm can be found in Appendix B. In the 1-D case, the minimizer of the discrete SBP optimization problem serves as a reference

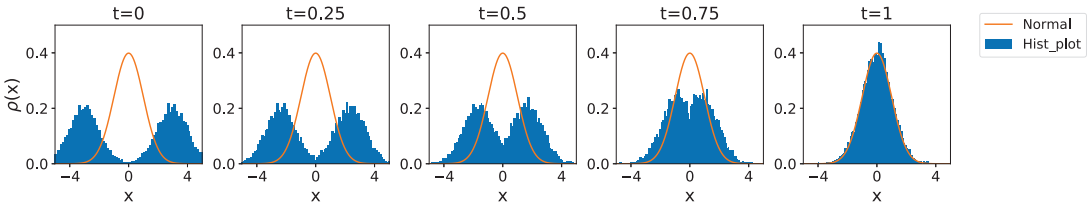
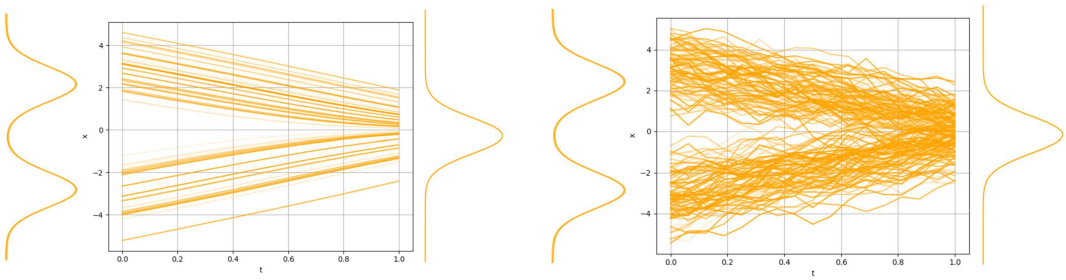


Figure 1: Evolution of particle distribution transitioning from Gaussian mixture to standard Gaussian.



a) ODE trajectories for $dX = f_\theta dt$ b) SDE paths for $dX = vdt + \sqrt{2}dW$

Figure 2: ODE interpolation trajectories and SDE interpolation paths.

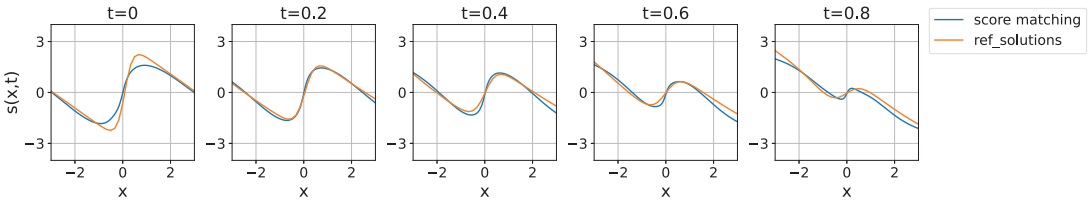


Figure 3: Comparison of score function between score matching solutions and reference ones. The pre-trained f_θ is solved with $\alpha = 10$.

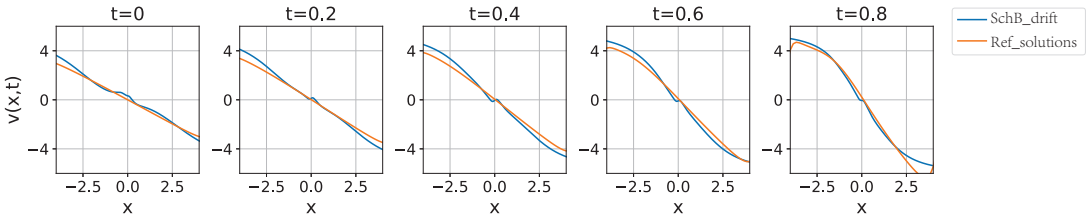


Figure 4: Comparison of drift of solutions between our algorithm ($\alpha = 10$) and tradition primal dual method.

solution to verify the accuracy of our algorithm. Fig. 3 displays the score function s_ϕ obtained through score matching, compared to the reference score, which is computed from the solutions of the discrete optimization problem (B.2) using finite differences. The consistency of the score functions suggests that our algorithm provides a good approximation of the time marginals, as the trajectories used for training score matching algorithms are generated by the well-trained hypothetical velocity field f_θ. Fig. 4 shows the SBP's optimal drift recovered after score matching as suggested in Section 4.2 and the reference drift. It can be observed that the two drifts match well in the high-density region, particularly within $[-3, 3]$. The reference drift exhibits oscillations near the boundary due to the boundary conditions imposed in the discrete Schrödinger optimization problem, whereas our SBP-solving algorithm via normalizing flows does not strictly enforce a bounded domain or mass conservation.

To assess the performance of the proposed method more quantitatively, we follow the techniques applied to generative models as in [35]. We compare the samples $\boldsymbol{X} = \{x_i\}_{i=1}^N$

obtained with forward process at time T to the samples $\boldsymbol{Q} = \{q_i\}_{i=1}^M$ drawn from ρ_1 via maximum mean discrepancy [14, 28]

$$\mathrm{MMD}(\boldsymbol{X}, \boldsymbol{Q}) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} k(x_i, x_j) + \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M} k(q_i, q_j) - \frac{2}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M} k(x_i, q_j),$$

where we choose Gaussian kernel

$$k(x_i, q_j) = \exp\left(-\frac{1}{2}\left|x_i - q_j\right|^2\right).$$

MMD tests the difference between two distributions in the form of samples. The MMD provides an external metric to evaluate our model since it is not used in the training. We set $M = N = 10000$ for validation and plot the curve of MMD results with respect to different α in Fig. 5. It can be observed that an increase in α leads to a smaller MMD between the push-forward distribution and the terminal distribution. This behavior is expected, as a larger α imposes a stronger penalty on the KL term, as confirmed by our Γ-convergence analysis. We also calculate the relative $L^2([0,T]; L^2[-3,3])$ error for the drift for different values of α compared to the drift by the primal-dual algorithm. The results are shown in Fig. 5. Each point on the curves represents the average value obtained from five independent experiments. The curve indicates quantitatively that the approach could recover the drift for Schrödinger bridge problem.

We plot the hypothetical velocity field f with different noise scale σ solved by our algorithm, and the velocity field from optimal transport as computed by OT-Flow as shown in Fig. 6. A higher noise scale corresponds to a larger velocity in the initial stage, which

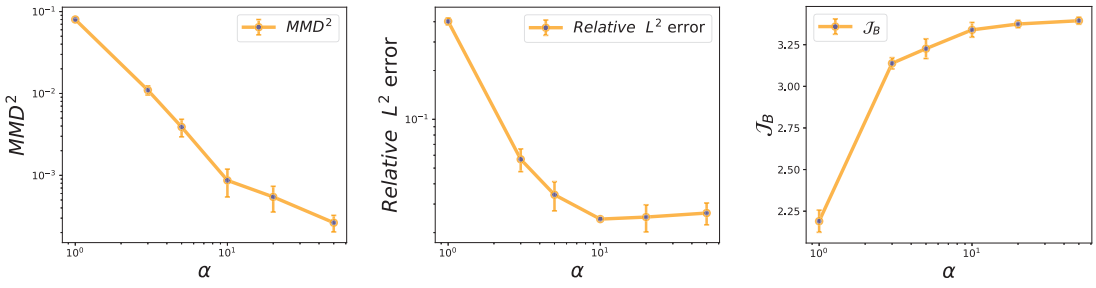


Figure 5: Left: the MMD results for different values of α. Middle: the relative $L^2([0,T]; L^2[-3,3])$ error of u for different values of α. Right: $\mathscr{J}_B$ with different values of α.

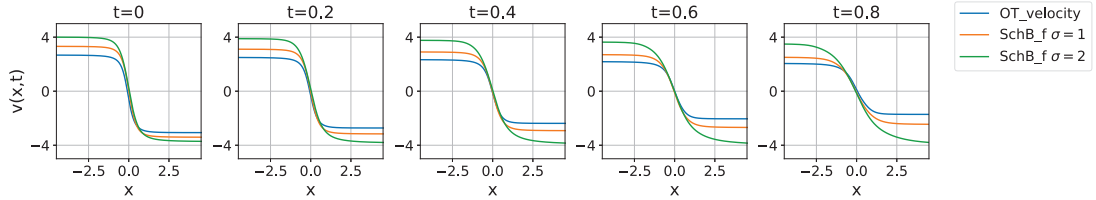


Figure 6: Comparison of SBP hypothetical velocity field with different σ and the velocity field of optimal transport problem. We fix $\alpha = 10$ in the experiments.

aligns with intuitive understanding. In our 1-D experiments, the hypothetical velocity field f in SBP is quite similar compared with the optimal velocity field in optimal transport problem. A plausible explanation is that the SBP can be interpreted as a regularized entropy optimal transport problem. As the noise scale diminishes to zero, the minimizers of SBP are anticipated to converge to those of optimal transport.

During training, we set a relatively large α as suggested by the Γ-convergence analysis. In this large α regime, as the training progresses, one can expect $\mathbb{KL}[\rho(x,T)||\rho_1(x)] \approx 0$. To verify the convergence, we plot the curve of the optimization target for different values of α in Fig. 5. Each point on the curve represents the average value obtained from five independent experiments, where the settings are the same except for the value of α. As α increases, we observe that $\mathscr{J}_B$ has a slower increasing rate, which is expected to eventually converge the minimum of the SBP. The optimization target computed by the traditional optimization method is about 3.493.

6.2. 2-D toy problems

We would like to clarify that the main goal of our article is to solve the Schrödinger bridge problem between two probability distributions. However, the trained model could clearly be used to generate samples. In this subsection, we visualize the generative performance on classical 2-D synthetic examples to evaluate the potential of our algorithm as a generative model. In all experiments, we fix $\alpha = 10$ and choose 10000 samples for visualization. The hypothetical velocity field is modeled as a neuron network of 4 Concat-Squash linear layers [5]. We utilize Dopri5 adaptive ODE solver for both the forward and backward ODEs. The model is trained for 1000 epochs using the ADAM [24] optimizer with a learning rate of $2e$-4 and a batch size of 1024.

We use trained hypothetical velocity field, or alternatively SB drift with diffusion process to simulate an inverse transformation (sampling from 2-D standard normal distribution and pushing particles back to the data distribution), from which one can compare the similarity of original data with generative distribution. The high similarity indicates that our model can generate samples to approximate ρ_0 with satisfactory accuracy even though ρ_0 has separate supports. Numerical results are illustrated in Fig. 7, which demonstrates the potential of our model to function as a generative model.

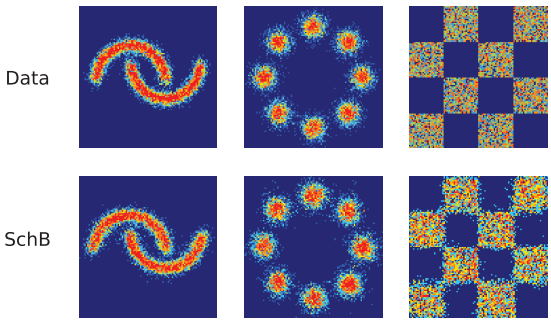


Figure 7: Comparison of data and generated samples.

The top two rows of Fig. 8 compare the forward and generation processes of the SchB, demonstrating the invertibility of the mappings and the stability of the numerical solvers. The third row presents the generation process of classical CNFs. In comparing the generation processes of SchB and classical CNFs, we observe that classical CNFs do not follow a 'straight' path, whereas the SchB corresponds to deblurring in the later stages of generation.

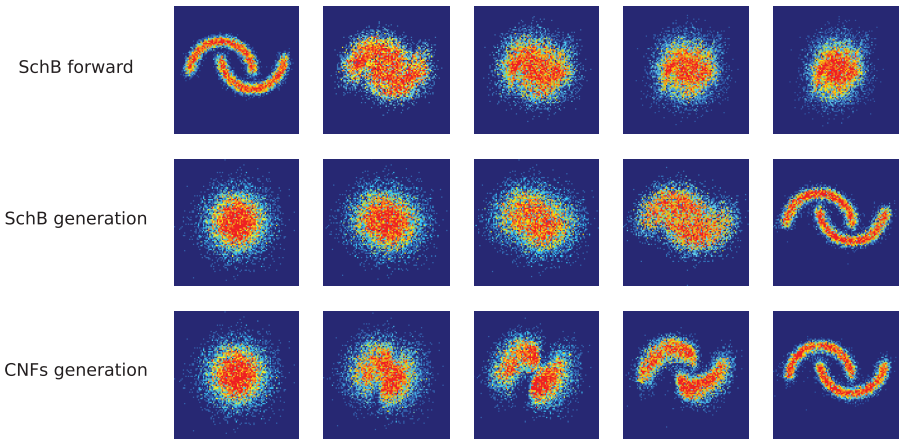


Figure 8: Top: Forward process from data of moons to the normal distribution. Middle: Generation process of SchB from Gaussian samples to new data samples. Bottom: Generation process of classical CNFs from Gaussian samples to new data samples.

6.3. 2-D double well experiment

In this section, we demonstrate our algorithm by solving the optimal control problem of transitioning from one minimizer to another, a typical application in the field of chemical reactions. Considering the double well potential, we demonstrate how to incorporate a functional prior of any form and learn the distribution of paths connecting ρ_0 and ρ_1. Similar experiments have been researched in [11,52]. To incorporate prior knowledge, we use a potential field with two local minima, and the boundary distributions ρ_0 and ρ_1 are modeled as Gaussian distributions centered at the wells.

The problem is equivalent to find the optimal control u such that the diffusion process

$$dX_t = \big[u(X_t,t) - \nabla U(X_t)\big]dt + \sqrt{2}dW$$

bridges the distributions ρ_0 and ρ_1 and minimizes the L^2 cost

$$\begin{aligned} \min_{\rho,u} \quad & \frac{1}{2}\int_0^T \int_\Omega \rho(x,t)|u(x,t)|^2 dx dt, \\ \text{s.t.} \quad & \partial_t \rho + \nabla\cdot\big(\rho(u-\nabla U)\big) = \sigma^2 \Delta\rho, \\ & \rho(x,0) = \rho_0(x), \quad \rho(x,1) = \rho_1(x). \end{aligned}$$

The aim of this experiment is to show that, with this prior, the SBP results in low-energy trajectories (as determined by the well's potential function) for particle configurations sampled from the wells. Intuitively, the learned paths are expected to avoid the high-energy peak at $x = (0, 0)$ and instead follow the 'passes' on either side. Notably, if we were to estimate the optimal transport geodesics between ρ_0 and ρ_1 or use a Brownian motion prior, the learned trajectories would pass through the center, which corresponds to the highest energy path between the wells.

The algorithm remains nearly identical. We parameterized $f_\theta := u - \sigma^2 \nabla \log \rho$ as a neural network and denotes $F(x,t) := f_\theta(x,t) - \nabla U(x)$. Consequently, the corresponding ODE system becomes

$$\partial_t \begin{bmatrix} z \\ \ell \\ s \end{bmatrix} = \begin{bmatrix} F \\ \nabla \cdot F \\ -\nabla(\nabla \cdot F) - \mathscr{J}_F^T s \end{bmatrix}, \quad \begin{bmatrix} z(x,0) \\ \ell(x,0) \\ s(x,T) \end{bmatrix} = \begin{bmatrix} x \\ 0 \\ \nabla \log \rho_1(z(x,T)) \end{bmatrix}. \tag{6.1}$$

The loss function remains unchanged. The parameterized function f_θ can be trained by minimizing this loss function and the ODE trajectories can be recovered by solving

$$dx_t = \big(f_\theta(x_t, t) - \nabla U(x_t)\big)dt.$$

Additionally the optimal control can be recovered using score matching as described in Section 4.2, enabling the generation of SDE paths for particles.

Algorithm 6.1 Solving 2-D Double Well Problems

Require: Data samples $\{x_i\}_{i=1}^N$, time interval $[0, T]$, initializing network f_θ.

1: **for** number of training iterations **do**
2: Solving ODE system (6.1) with f_θ to obtain $z(x_i, T)$, $\ell(x_i, T)$ and $s(x_i, t)$.
3: Calculate cost function $\mathscr{J} = \alpha \mathscr{J}_{\mathbb{KL}} + \mathscr{J}_B$.
4: Use ADAM optimizer [24] to update network parameter of f_θ.
5: **end for**
6: Use Algorithm 4.2 to recover the optimal control u_ϕ from f_θ.

Especially the potential and the marginal distributions in our experiment are set to be

$$U(x,y) = 0.3\left(\frac{1}{2}(x^2-1)^2 + y^2 + 8\exp\left(\frac{-((x-a)^2+y^2)}{32}\right)\right),$$
$$\rho_0 = \mathscr{N}\left(\begin{bmatrix}-1\\0\end{bmatrix}, \begin{bmatrix}0.0125 & 0\\0 & 0.15\end{bmatrix}\right), \quad \rho_1 = \mathscr{N}\left(\begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}0.0125 & 0\\0 & 0.15\end{bmatrix}\right),$$

where a is a constant controlling the potential barrier in the middle. The learned control vector field for the SDE at different time is shown in Fig. 9. The white lines indicate the contour lines (level curves) of the potential function, the black arrows denote the directions of the control vectors, and the background color represents the magnitude of the control vectors. We can observe that at $t = 0$ the control vector in the left region is opposite to

the gradient of the given potential, driving particles away from the left local minima. At $t = 1$ the control vector in the right region aligns with the gradient of the potential, guiding particles toward the target local minima.

Fig. 10 shows the ODE and SDE paths for different a. For $a = 0.5$ we can observe that the path of particles bypassing the barrier. The case where $a = 0$ presents an intriguing phenomenon: despite the presence of a symmetric potential field, the trained network consistently produces asymmetric results. Note that, however, the probability that the trained velocity field chooses the upper half region or lower half region is symmetric. This behavior suggests the presence of implicit regularization mechanisms within the neural network architecture. It is unclear at this point whether the original system has the physical symmetry breaking. Fig. 11 shows the terminal distribution of preset data and sampling from trained network. The terminal distributions matched the target distribution in both $a = 0.5$ and $a = 0$ cases, demonstrating that our algorithm can always find connecting paths that avoid high-energy peak and enable transitions between the minimizers.

6.4. High-dimensional LQG bridge

We demonstrate the effectiveness of our approach for a high-dimensional case in this subsection. The experimental setup is a specific instance of a linear quadratic Gaussian (LQG) bridge. This terminology is used because the reference dynamics are linear, the negative log-probabilities of the boundary distributions are quadratic, and the distributions themselves are Gaussian. We consider a 64-dimensional Ornstein-Uhlenbeck process as the reference dynamics, described by the following SDE:

$$dX_t = -\theta X_t dt + \sigma dW_t, \quad X_t \in \mathbb{R}^{64}.$$

The Schrödinger bridge problem we consider is between two Gaussian boundary distributions $\rho_0 = \mathcal{N}(\mu_0, \Sigma)$ and $\rho_1 = \mathcal{N}(\mu_1, \Sigma)$ over the time interval $[0, T]$. For this experiment, we set the hyper-parameters as follows: $\theta = 1.0$, $T = 1.0$, and the noise coefficient $\sigma = \sqrt{2}$. The boundary conditions are centered at $\mu_0 = -\mathbf{1} \in \mathbb{R}^{64}$ and $\mu_1 = \mathbf{1} \in \mathbb{R}^{64}$, with a shared covariance matrix $\Sigma = 0.5 \cdot \mathbf{I}$, where $\mathbf{I}$ is the 64-dimensional identity matrix.

A key advantage of this setup is that the mean of the ground-truth distribution μ_t has a closed-form solution [6]

$$\mu_t = \frac{\sinh(\theta(T-t))}{\sinh(\theta T)} e^{-\theta t} \mu_0 + \frac{\sinh(\theta t)}{\sinh(\theta T)} e^{\theta(t-T)} \mu_1,$$

which allows us to precisely measure the error of our learned solution.

For this high-dimensional problem, we employ a simple MLP architecture for the velocity field f_θ. The network consists of three hidden layers with (128, 256, 128) neurons and SiLU activations. To ensure stable training, we apply spectral normalization to each linear layer. Time is incorporated into the network via a sinusoidal positional embedding. Due to the potential for stiffness in high dimensions, we utilize a fixed-step fourth-order Runge-Kutta (RK4) solver with 50 integration steps for both the forward and backward passes.

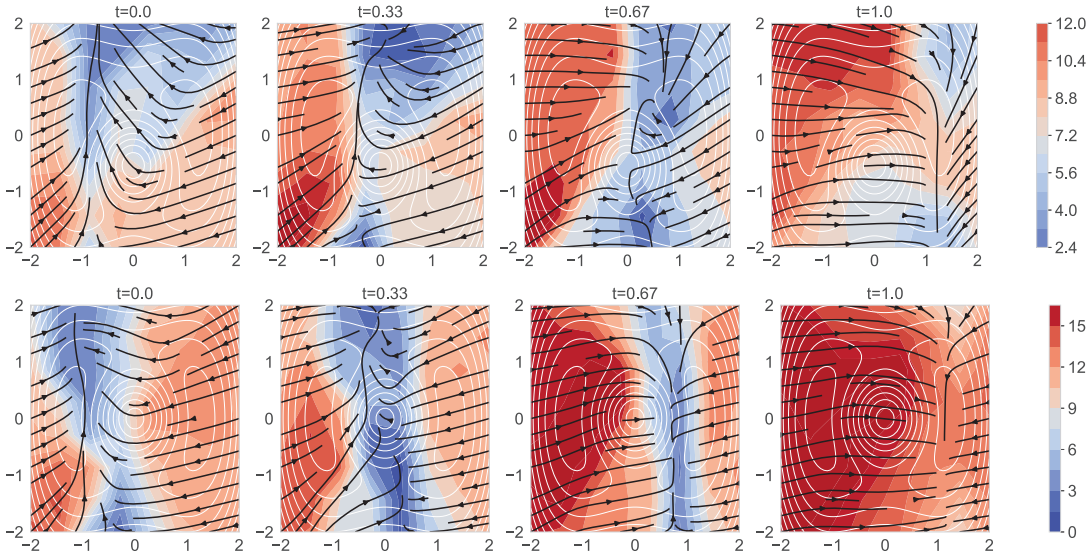
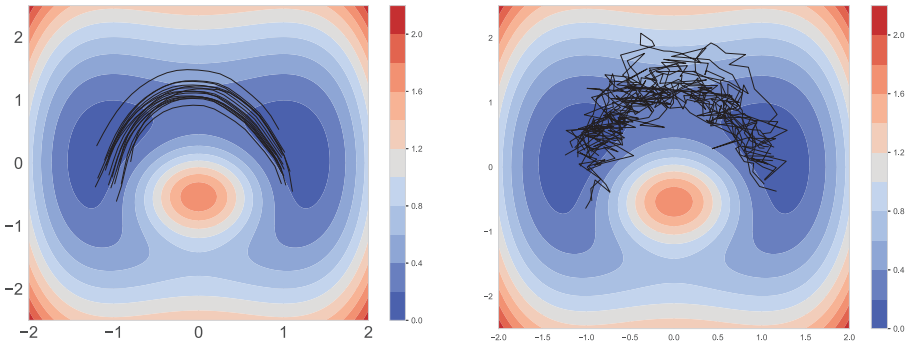
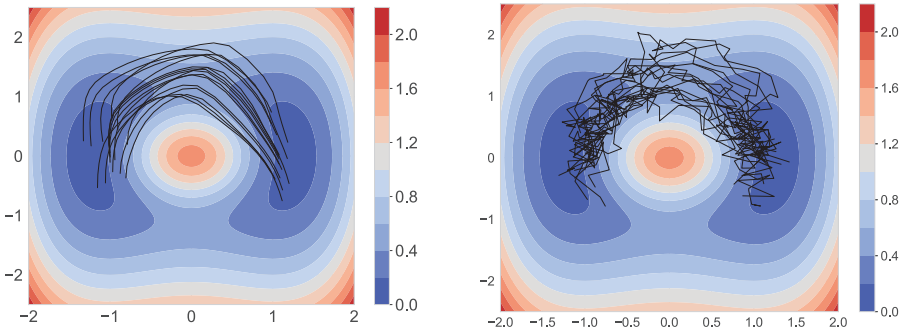


Figure 9: Optimal control u for different values of a. The white lines represent the contour lines of the potential function, the black arrows represent the directions of the control vectors, and the background color represents the magnitude of the control vectors. Upper: $a = 0.5$. Bottom: $a = 0$.



(a) ODE trajectories for $a = 0.5$

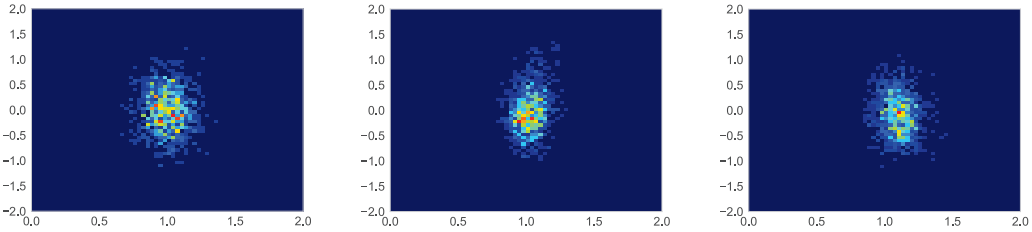
(b) SDE paths for $a = 0.5$



(c) ODE trajectories for $a = 0$

(d) SDE paths for $a = 0$

Figure 10: ODE trajectories and SDE interpolation paths for standard and symmetric breaking cases.



(a) Target distribution ρ_1 (b) Sampling with $a = 0.5$ (c) Sampling with $a = 0$

Figure 11: Comparison of terminal distributions.

The model is trained for 1000 epochs using the AdamW [33] optimizer with a learning rate of $1e$-4 and a batch size of 512.

A key computational challenge in our method is the evaluation of the divergence term $\nabla \cdot f$ required for the log-determinant dynamics, and the computation of Jacobian matrix J_f. In high-dimensional spaces, a direct computation involving d backward passes becomes prohibitively expensive. To overcome this, we employ Hutchinson's trace estimator, which provides an efficient and unbiased stochastic estimate of the divergence, and utilize the `autograd` functionality of PyTorch. A detailed description can be found in Section 4.

To quantify the accuracy of our method, we measure the time-averaged relative L^2 error between the mean of the learned particle trajectories $\hat{\mu}_t$ and the analytical mean μ_t

$$\mathscr{E} = \left(\int_0^T |\hat{\mu}_t - \mu_t|_2^2 dt \right) \left(\int_0^T |\mu_t|_2^2 dt \right)^{-1}.$$

The results are presented in Fig. 12. We observe that the accuracy of the learned trajectory is highly dependent on a proper balance between the two loss components. When α is

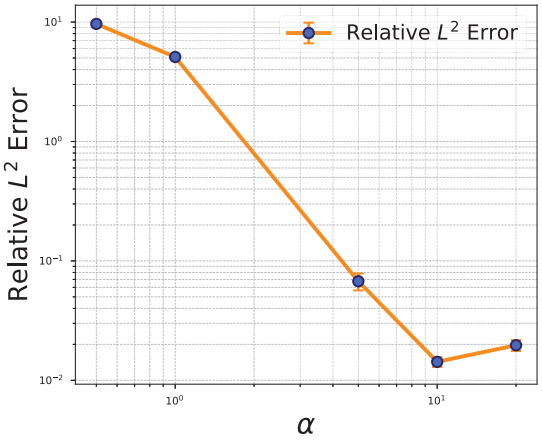


Figure 12: Relative L^2 error of the mean trajectory with respect to different values of α. The error reaches its minimum at an intermediate value of α, demonstrating the crucial trade-off between matching the terminal distribution and satisfying the path-regularizing bridge constraint.

too small, the model is not sufficiently incentivized to match the terminal distribution ρ_1, leading to a high error. Conversely, when α is excessively large, the model may find an overly aggressive or stiff velocity field to satisfy the terminal constraint, which can lead to numerical instability and a violation of the bridge dynamics, again increasing the error. The optimal performance is achieved at an intermediate value $\alpha \approx 10.0$, which correctly balances the boundary conditions and path constraints.

7. Conclusion and Discussion

In summary, we have developed a framework based on CNFs and score matching to solve the dynamic formulation of the SBP in the first part. Specifically, we introduce a hypothetical velocity field to transform the Fokker-Planck equation into a transport equation, enabling the use of the CNF framework to solve the dynamic SBP and track density evolution. In the second part, we employ Γ-convergence to establish the convergence of minimizers from our neural network solutions to the theoretical ones as the regularization coefficient $\alpha \to \infty$. Our approach provides a neural network-based solution to the SBP and contributes to a deeper understanding of the convergence properties of CNF models with regularization. The convergence analysis also offers theoretical guarantees for the stability during the training of our algorithm. Future research may explore the application of analogous methodologies to design algorithms and establish convergence analyses for addressing SDE-type mathematical tasks, such as solving optimal control problem and tracking evolution of physical systems.

Acknowledgments

This work is partially supported by the National Key R&D Program of China (Grant Nos. 2020YFA0712000, 2021YFA1002800). The work of L. Li was partially supported by the NSFC (Grant Nos. 12371400, 12031013), by the Shanghai Municipal Science and Technology Major Project (Grant No. 2021SHZDZX0102), and by the Shanghai Science and Technology Commission (Grant Nos. 21JC1403700, 21JC1402900).

Appendix A Missing Proofs

Proof of Proposition 2.3. According to Proposition 2.2, it suffices to show that

$$\inf_{x_n \to x} \limsup_{n\to\infty} f_n(x_n) \le f(x) \le \inf_{x_n \to x} \liminf_{n\to\infty} f_n(x_n).$$

Consider

$$x_n = x, \quad f(x) = \lim_{n\to\infty} f_n(x) = \limsup_{n\to\infty} f_n(x_n) \ge \inf_{x_n \to x} \limsup_{n\to\infty} f_n(x_n).$$

For the other inequality, consider any $x_n \to x$. Since $f_n \uparrow f$ pointwise,

$$f(x) = \lim_{n\to\infty} f_n(x) \le \lim_{n\to\infty} \liminf_{m\to\infty} f_n(x_m) \le \liminf_{m\to\infty} f_m(x_m).$$

The last inequality holds since for any fixed n and any m larger than n, one has $f_n(x_m) \leq f_m(x_m)$. Hence, f_n is Γ-convergent to f. □

Proof of Proposition 3.1. For the second equation,

$$\begin{aligned}\frac{\partial \ell(x,t)}{\partial t} &= \frac{1}{\det(\nabla z(x,t))}\frac{\partial \det(\nabla z(x,t))}{\partial t}\\ &= \frac{1}{\det(\nabla z(x,t))}\cdot \det\big(\nabla z(x,t)\big)\cdot \mathrm{tr}\Big[\big(\nabla z(x,t)\big)^{-1}\frac{\partial \nabla z(x,t)}{\partial t}\Big]\\ &= \frac{1}{\det(\nabla z(x,t))}\cdot \det\big(\nabla z(x,t)\big)\cdot \mathrm{tr}\Big[\big(\nabla z(x,t)\big)^{-1}\nabla z(x,t)\nabla_z f\big(z(x,t),t\big)\Big]\\ &= \mathrm{tr}\big[(\nabla_z f(z(x,t),t)\big]\\ &= \nabla_z \cdot f\big(z(x,t),t\big),\end{aligned}$$

where we have used following identities:

$$\frac{\partial \det(A)}{\partial t} = \det(A)\cdot \mathrm{tr}\Big[A^{-1}\frac{\partial A}{\partial t}\Big], \quad \mathrm{tr}(AB) = \mathrm{tr}(BA).$$

For the third equation, we first have

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}t}\nabla_z \log\rho\big(z(x,t),t\big)\\ &= \frac{\partial}{\partial t}\nabla_z \log\rho(z,t)|_{z=z(x,t)} + \frac{\mathrm{d}z}{\mathrm{d}t}\frac{\partial}{\partial z}\nabla_z \log\rho(z,t)|_{z=z(x,t)}\\ &= \nabla_z \frac{\partial}{\partial t}\log\rho(z,t)|_{z=z(x,t)} + f(z,t)\nabla_z^2 \log\rho(z,t)|_{z=z(x,t)}.\end{aligned} \tag{A.1}$$

Using the Fokker-Plank equation (3.3) and $f = u - \sigma^2\nabla\log\rho$, we derive

$$\partial_t \log\rho(z,t) = -\nabla_z \cdot f(z,t) - f(z,t)\cdot\nabla_z \log\rho(z,t),$$

which together with

$$\nabla_z\big(\nabla_z \log\rho(z,t)\cdot f(z,t)\big) = \big[\nabla_z^2 \log\rho(z,t)\big]\cdot f(z,t) + \nabla_z f(z,t)\cdot\nabla_z \log\rho(z,t),$$

taking it into (A.1) one get

$$\frac{\mathrm{d}}{\mathrm{d}t}\nabla_z \log\rho\big(z(x,t),t\big) = -\nabla_z\big(\nabla_z \cdot f(z,t)\big)|_{z=z(x,t)} - \nabla_z f(z,t)\cdot\nabla\log\rho(z,t)|_{z=z(x,t)}.$$

The proof is complete. □

Appendix B Implementation Details for Solving SBP with Optimization Method

To validate our algorithm, we use a primal-dual hybrid algorithm to solve the SBP on the grid. This algorithm has been previously employed to compute the earth mover's distance [27] and the spherical Wasserstein-Fisher-Rao metric [21], as well as to verify the convergence from optimal transport to unbalanced optimal transport [53]. We begin by taking $\sigma = 1$ for simplicity and reformulating the optimization problem into a convex form

$$\min_{\rho,m}\left\{\int_0^T\int_\Omega \frac{|m|^2}{2\rho}dxdt, \partial_t\rho+\nabla\cdot m=\Delta\rho,\ \rho(x,0)=\rho_0(x),\ \rho(x,T)=\rho_1(x)\right\}.$$

We then focus on solving the corresponding min-max problem

$$\max_{\phi}\min_{\substack{\rho,m\\ \rho(x,0)=\rho_0(x)\\ \rho(x,T)=\rho_1(x)}}\int_0^T\int_\Omega\frac{|m|^2}{2\rho}dxdt+\int_0^T\int_\Omega\phi(x,t)(\partial_t\rho+\nabla\cdot m-\Delta\rho)dxdt. \tag{B.1}$$

We employ the same discrete scheme for divergence operator and boundary conditions as [53]. For simplicity, we consider the space domain $\Omega=[0,1]^d$ and the time domain $[0,1]$. Let Ω_h be the discrete space mesh-grid of Ω with step size h, i.e. $\Omega_h=\{0,h,2h,\cdots,1\}^d$. The time domain $[0,1]$ is discreted with step size Δt. Let $N_x=1/h$ be the space grid size and $N_t=1/\Delta t$ be the time grid size. All optimization variables (ρ, m and ϕ) will be defined on the grid. To ensure clarity, we first provide some definitions on the discrete space Ω_h

$$\begin{aligned}
&\int_{\Omega_h} f(x) := \sum_{x\in\Omega_h, x_i\neq 0} f(x)h^d, \quad \langle f,g\rangle_h := \int_{\Omega_h} f(x)g(x),\\
&\nabla_h\cdot m(x) := \sum_{i=1}^d D_{h,i}m(x), x\in\Omega_h,\\
&\Delta_h\rho := \frac{1}{h^2}\sum_{i=1}^d \rho(x_1,\cdots,x_i+h,\cdots,x_d)-2\rho(x_1,\cdots,x_i,\cdots,x_d)\\
&\qquad\qquad +\rho(x_1,\cdots,x_i-h,\cdots,x_d),
\end{aligned}$$

where $D_{h,i}$ denotes discrete differential operator for i-th component in i-th dimension with step size h

$$D_{h,i}m(x)=\big(m_i(x_1,\cdots,x_i+h,\cdots,x_d)-m_i(x_1,\cdots,x_i,\cdots,x_d)\big)/h,\quad 0\le x_i\le 1-h.$$

From the discretized form, we can describe the sizes of the optimization variables individually. The sizes of discretized variables are : $(N_t+1)\times(N_x)^d$ for ρ, $N_t\times(N_x+1)^d\times d$ for m and $N_t\times(N_x)^d$ for ϕ. The boundary conditions are given as $\rho_1=\mu$, $\rho_{N_t+1}=\nu$ and $(m-\nabla\rho)\cdot\vec{n}=0$ for all $x\in\partial\Omega_h$. Then the discretization of (B.1) can be written as

$$\max_{\phi}\min_{\rho,m,\xi} L(m,\rho,\phi)=\sum_{t=1}^{N_t}\left\{\int_{\Omega_h}\frac{|m_t|^2}{2\rho_t}dx+\left\langle\phi_t,\frac{\rho_{t+1}-\rho_t}{\Delta t}+\nabla_h\cdot m_t-\Delta_h\rho_t\right\rangle_h\right\}. \tag{B.2}$$

One can add ghost points at the boundary and use discretized boundary condition to deal with central difference discretion for ρ, which ensures that the problem is well-defined. For instance, consider the left boundary: one can add a ghost point ρ_t^{-1} and the boundary condition yields

$$m_t^0 - \frac{\rho_t^0 - \rho_t^{-1}}{h} = 0,$$

which can then be incorporated into the Laplacian operator

$$\Delta_h \rho_t^0 = \frac{\rho_t^{-1} - 2\rho_t^0 + \rho_t^1}{h^2} = \frac{-\rho_t^0 + \rho_t^1}{h^2} - \frac{m_t^0}{h}.$$

As a result, (B.2) is independent of ρ_t^{-1}.

Then, the primal-dual hybrid algorithm proceeds with the following three steps for the variables with subscript t:

1. $\left(m_t^{k+1}, \rho_t^{k+1}\right) = \arg\min_{m^\star, \rho^\star} L\left(m^\star, \rho^\star, \phi^k\right) + \frac{1}{2\mu}\left(|m^\star - m_t^k|^2 + |\rho^\star - \rho_t^k|^2\right)$,
2. $\tilde{m}_t^{k+1} = 2m_t^{k+1} - m_t^k, \quad \tilde{\rho}_t^{k+1} = 2\rho_t^{k+1} - \rho_t^k$,
3. $\phi_t^{k+1} = \arg\max_{\phi^\star} L\left(\tilde{m}_t^{k+1}, \tilde{\rho}_h^{k+1}, \phi^\star\right) - \frac{1}{2\tau}\left(|\phi^\star - \phi_t^k|^2\right)$.

The first step of the algorithm is to update ρ and m, which is equivalent to solve the following equations:

$$\begin{aligned} m^\star &= \frac{\rho^\star(m_t^k - \mu \mathrm{div}_h^* \phi_t^k)}{\rho^\star + \mu}, \\ -\frac{m^{\star 2}}{2\rho^{\star 2}} + \frac{\phi_{t-1}^k - \phi_t^k}{\Delta t} - \Delta_h^\star \phi_t^k &+ \frac{1}{\mu}\left(\rho^\star - \rho_t^k\right) = 0, \end{aligned} \tag{B.3}$$

where div_h^* represents the conjugate operator of divergence operator and $\Delta_h^\star$ represents the conjugate operator of Laplacian operator. By the definition of conjugate operator, we have

$$\begin{aligned} \left\langle \nabla_h \cdot f, g \right\rangle_h &= \left\langle f, \mathrm{div}_h^* g \right\rangle_h = \left\langle f, -\nabla_h g \right\rangle_h, \\ \left\langle \Delta_h \rho, \phi \right\rangle_h &= \left\langle \rho, \Delta_h^\star \phi \right\rangle_h = \left\langle \rho, \Delta_h \phi \right\rangle_h. \end{aligned}$$

Thus, $\mathrm{div}_h^* = -\nabla_h$ and $\Delta_h^* = \Delta_h$, where $\nabla_h u = (\partial_{h,1} u, \partial_{h,2} u, \cdots, \partial_{h,d} u)$ and each $\partial_{h,i}$ denotes discrete differential operator in i-th dimension with step size h

$$\partial_{h,i} u(x) = \big(u(x_1, \cdots, x_i + h, \cdots, x_d) - u(x_1, \cdots, x_i, \cdots, x_d)\big)/h, \quad 0 \le x_i \le 1 - h.$$

Meanwhile, the term ϕ_{t-1}^k in (B.3) is undefined when $t = 1$. Observe that the term $(\phi_{t-1}^k - \phi_t^k)/\Delta t$, arising from taking the variation of (B.2) with respect to ρ_t, which should be $-\phi_1^k/\Delta t$ in the case of $t = 1$. This naturally suggests the introduction of a boundary

condition $\phi_0 = 0$. Solving above system (B.3) requires us to solve the roots for a third order polynomial, where $\rho^\star$ should be the largest real root. The third step of the algorithm is to update dual variables

$$\phi_t^{k+1} = \phi_t^k + \tau\left(\frac{\tilde{\rho}_{t+1}^{k+1} - \tilde{\rho}_t^{k+1}}{\Delta t} + \nabla_h \cdot \tilde{m}_t^{k+1} - \Delta_h \tilde{\rho}_t^{k+1}\right).$$

References

[1] V.D. Bortoli, J. Thornton, J. Heng and A. Doucet, *Diffusion Schrödinger bridge with applications to score-based generative modeling*, Adv. Neural Inf. Process. Syst. **34**, 17695–17709 (2021).

[2] L.E. Bottou, *Online Learning and Stochastic Approximations*, Online Learning in Neural Networks, Cambridge University Press (1998).

[3] A. Braides, *Gamma-Convergence for Beginners*, Clarendon Press (2002).

[4] G. Buttazzo and G.D. Maso, *Γ-convergence and optimal control problems*, J. Optim. Theory Appl. **38**, 385–407 (1982).

[5] R.T.Q. Chen, Y. Rubanova, J. Bettencourt and D.K. Duvenaud, *Neural ordinary differential equations*, Adv. Neural Inf. Process. Syst. **31**, 6571–6583 (2018).

[6] Y. Chen, T.T. Georgiou and M. Pavon, *On the relation between optimal transport and Schrödinger bridges: A stochastic control viewpoint*, J. Optim. Theory Appl. **169**, 671–691 (2016).

[7] X. Feng, L. Zeng and T. Zhou, *Solving time dependent Fokker-Planck equations via temporal normalizing flow*, Commun. Comput. Phys. **32**, 401–423 (2022).

[8] D. Fernandes, F. Vargas, C.H. Ek and N. Campbell, *Shooting Schrödinger's cat*, in: *Proceedings of the Fourth Symposium on Advances in Approximate Bayesian Inference*, (2022).

[9] J. Feydy, B. Charlier, F.X. Vialard and G. Peyré, *Optimal transport for diffeomorphic registration*, in: *Medical Image Computing and Computer Assisted Intervention*, 291–299 (2017).

[10] C. Finlay, J.H. Jacobsen, L. Nurbekyan and A. Oberman, *How to train your neural ODE: The world of Jacobian and kinetic regularization*, in: *Proceedings of the International Conference on Machine Learning*, 3154–3164 (2020).

[11] Y. Gao, J.G. Liu and O. Tse, *Optimal control formulation of transition path problems for Markov jump processes*, arXiv:2311.07795 (2023).

[12] J. Garg, X.Y. Zhang and Q. Zhou, *Soft-constrained Schrödinger bridge: A stochastic control approach*, in: *Proceedings of the International Conference on Artificial Intelligence and Statistics*, **238**, 4429–4437 (2024).

[13] W. Grathwohl, R.T.Q. Chen, J. Bettencourt, I. Sutskever and D. Duvenaud, *FFJORD: Free-form continuous dynamics for scalable reversible generative models*, in: *Proceedings of the International Conference on Learning Representations*, (2019).

[14] A. Gretton, K.M. Borgwardt, M.J. Rasch, B. Schölkopf and A. Smola, *A kernel two-sample test*, J. Mach. Learn. Res. **13**, 723–773 (2012).

[15] J. Ho, A. Jain and P. Abbeel, *Denoising diffusion probabilistic models*, Adv. Neural Inf. Process. Syst. **33**, 6840–6851 (2020).

[16] L. Holdijk, Y.Q. Du, F. Hooft, P. Jaini, B. Ensing and M. Welling, *Stochastic optimal control for collective variable free sampling of molecular transition paths*, Adv. Neural Inf. Process. Syst. **36**, 79647–79669 (2024).

[17] H.W. Huang, *One-step data-driven generative model via Schrödinger bridge*, arXiv:2405.12453 (2024).

[18] A. Hyvärinen and P. Dayan, *Estimation of non-normalized statistical models by score matching*, J. Mach. Learn. Res. **6**, 695–709 (2005).
[19] V. Ilin, J.W. Hu and Z.F. Wang, *Transport based particle methods for the Fokker-Planck-Landau equation*, arXiv:2405.10392 (2024).
[20] S. Jin, L. Li and J.G. Liu, *Random batch methods (RBM) for interacting particle systems*, Comput. Phys. **400**, Paper No. 108877 (2020).
[21] Y. Jing, J.H. Chen, L. Li and J.F. Lu, *A machine learning framework for geodesics under spherical Wasserstein-Fisher-Rao metric and its application for weighted sample generation*, J. Sci. Comput. **98**, Paper No. 5 (2024).
[22] Y. Jing and L. Li, *Convergence analysis of OT-Flow for sample generation*, Numer. Math. Theor. Meth. Appl. **18**, 325–352 (2025).
[23] M. Kim, *A unified framework for diffusion bridge problems: Flow matching and Schrödinger matching into one*, arXiv:2503.21756 (2025).
[24] D.P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, in: *Proceedings of the International Conference on Learning Representations*, (2015).
[25] S. Kullback, *Probability densities with given marginals*, Ann. Math. Statist. **39**, 1236–1243 (1968).
[26] C. Léonard, *A survey of the Schrödinger problem and some of its connections with optimal transport*, Discrete Contin. Dyn. Syst. Ser. A **34**, 1533–1574 (2014).
[27] W.C. Li, E.K. Ryu, S. Osher, W.T. Yin and W. Gangbo, *A parallel method for earth mover's distance*, J. Sci. Comput. **75**, 182–197 (2018).
[28] Y.J. Li, K. Swersky and R. Zemel, *Generative moment matching networks*, in: *Proceedings of the International Conference on Machine Learning*, 1718–1727 (2015).
[29] Y. Lipman, R.T.Q. Chen, H.B. Hamu, M. Nickel and M. Le, *Flow matching for generative modeling*, in: *Proceedings of the International Conference on Learning Representations*, (2023).
[30] G.H. Liu, Y. Lipman, M. Nickel, B. Karrer, E.A. Theodorou and R.T.Q Chen, *Generalized Schrödinger bridge matching*, arXiv:2310.02233 (2023).
[31] G.H. Liu, A. Vahdat, D.A. Huang, E.A. Theodorou, W.L. Nie and A. Anandkumar, *I^2sb: Image-to-image Schrödinger bridge*, in: *Proceedings of the International Conference on Machine Learning*, 22042–22062 (2023).
[32] X.C. Liu, C.Y. Gong and Q. Liu, *Flow straight and fast: Learning to generate and transfer data with rectified flow*, in: *Proceedings of the International Conference on Learning Representations*, (2023).
[33] I. Loshchilov and F. Hutter, *Decoupled weight decay regularization*, in: *Proceedings of the International Conference on Learning Representations*, (2019).
[34] K. Neklyudov, R. Brekelmans, D. Severo and A. Makhzani, *Action matching: Learning stochastic dynamics from samples*, in: *Proceedings of the International Conference on Machine Learning*, 25858–25889 (2023).
[35] D. Onken, S.W. Fung, X.J. Li and L. Ruthotto, *OT-Flow: Fast and accurate continuous normalizing flows via optimal transport*, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, 9223–9232 (2021).
[36] G. Papamakarios, E. Nalisnick, D.J. Rezende, S. Mohamed and B. Lakshminarayanan, *Normalizing flows for probabilistic modeling and inference*, J. Mach. Learn. Res. **22**, 2617–2680 (2021).
[37] G. Papamakarios, T. Pavlakou and I. Murray, *Masked autoregressive flow for density estimation*, Adv. Neural Inf. Process. Syst. **30**, 2338–2347 (2017).
[38] S. Peluchetti, *Diffusion bridge mixture transports, Schrödinger bridge problems and generative modeling*, J. Mach. Learn. Res. **24**, 1–51 (2023).

[39] L.S. Pontryagin, *Mathematical Theory of Optimal Processes*, Routledge (2018).
[40] P.D. Pra, *A stochastic control approach to reciprocal diffusion processes*, Appl. Math. Optim. **23**, 313–329 (1991).
[41] D. Rezende and S. Mohamed, *Variational inference with normalizing flows*, in: *Proceedings of the International Conference on Machine Learning*, 1530–1538 (2015).
[42] F. Santambrogio, *Benamou-Brenier and other continuous numerical methods*, in: *Optimal Transport for Applied Mathematicians. Progress in Nonlinear Differential Equations and Their Applications*, Vol. **87**, 219–248 (2015).
[43] G. Schiebinger et al., *Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming*, Cell **176**, 928–943 (2019).
[44] E. Schrödinger, *Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique*, Annales de l'institut Henri Poincaré (1932).
[45] Z.B. Shen and Z.F. Wang, *Entropy-dissipation informed neural network for McKean-Vlasov type PDEs*, Adv. Neural Inf. Process. Syst. **36**, 59388–59411 (2024).
[46] Y.Y. Shi, V.D. Bortoli, A. Campbell and A. Doucet, *Diffusion Schrödinger bridge matching*, Adv. Neural Inf. Process. Syst. **36**, 62183–62223 (2023).
[47] R. Sinkhorn and P. Knopp, *Concerning nonnegative matrices and doubly stochastic matrices*, Pac. J. Math. **21**, 343–348 (1967).
[48] Y. Song and S. Ermon, *Generative modeling by estimating gradients of the data distribution*, Adv. Neural Inf. Process. Syst. **32**, 11895–11907 (2019).
[49] Y. Song, S. Garg, J.X. Shi and S. Ermon, *Sliced score matching: A scalable approach to density and score estimation*, in: *Proceedings of the 35th Uncertainty in Artificial Intelligence Conference*, PMLR **115**, 574–584 (2019).
[50] Y. Song, J. Sohl-Dickstein, D.P. Kingma, A. Kumar, S. Ermon and B. Poole, *Score-based generative modeling through stochastic differential equations*, in: *Proceedings of the International Conference on Learning Representations*, (2021).
[51] A. Tong, N. Malkin, K. Fatras, L. Atanackovic, Y.L. Zhang, G. Huguet, G. Wolf and Y. Bengio, *Simulation-free Schrödinger bridges via score and flow matching*, in: *Proceedings of the International Conference on Artificial Intelligence and Statistics*, **238**, 1279–1287 (2024).
[52] F. Vargas, P. Thodoroff, A. Lamacraft and N. Lawrence, *Solving Schrödinger bridges via maximum likelihood*, Entropy **23**, Paper No. 1134 (2021).
[53] Z. Xiong, L. Li, Y.N. Zhu and X.Q. Zhang, *On the convergence of continuous and discrete unbalanced optimal transport models for 1-Wasserstein distance*, SIAM J. Numer. Anal. **62**, 749–774 (2024).
[54] L. Zeng, X.L. Wan and T. Zhou, *Adaptive deep density approximation for fractional Fokker-Planck equations*, J. Sci. Comput. **97**, Paper No. 68 (2023).