# Simulation-Free Schrödinger Bridges via Score and Flow Matching

**Alexander Tong**[†]
Mila – Québec AI Institute
Université de Montréal

**Nikolay Malkin**[†]
Mila – Québec AI Institute
Université de Montréal

**Kilian Fatras**[†]
Mila – Québec AI Institute
McGill University

**Lazar Atanackovic**
University of Toronto
Vector Institute

**Yanlei Zhang**
Mila – Québec AI Institute
Université de Montréal

**Guillaume Huguet**
Mila – Québec AI Institute
Université de Montréal

**Guy Wolf**
Mila – Québec AI Institute
Université de Montréal
Canada CIFAR AI Chair

**Yoshua Bengio**
Mila – Québec AI Institute
Université de Montréal
CIFAR Senior Fellow

## Abstract

We present *simulation-free score and flow matching* ([SF]$^2$M), a simulation-free objective for inferring stochastic dynamics given unpaired samples drawn from arbitrary source and target distributions. Our method generalizes both the score-matching loss used in the training of diffusion models and the recently proposed flow matching loss used in the training of continuous normalizing flows. [SF]$^2$M interprets continuous-time stochastic generative modeling as a Schrödinger bridge problem. It relies on static entropy-regularized optimal transport, or a minibatch approximation, to efficiently learn the SB without simulating the learned stochastic process. We find that [SF]$^2$M is more efficient and gives more accurate solutions to the SB problem than simulation-based methods from prior work. Finally, we apply [SF]$^2$M to the problem of learning cell dynamics from snapshot data. Notably, [SF]$^2$M is the first method to accurately model cell dynamics in high dimensions and can recover known gene regulatory networks from simulated data. Our code is available in the `TorchCFM` package at https://github.com/atong01/conditional-flow-matching.

## 1 INTRODUCTION

Score-based generative models (SBGMs), including diffusion models, are a powerful class of generative models that can represent complex distributions over high-dimensional spaces (Sohl-Dickstein et al., 2015; Song and Ermon, 2019; Ho et al., 2020; Nichol and Dhariwal, 2021; Dhariwal and Nichol, 2021). SBGMs typically generate samples by simulating the evolution of a source density – nearly always a Gaussian – according to a stochastic differential equation (SDE) (Song et al., 2021b). Despite their empirical success, SBGMs are restricted by their assumption of a Gaussian source, which is essential for optimization with the simulation-free denoising objective. This assumption is often violated in the temporal evolution of physical or biological systems, such as in the case of single-cell gene expression data, which prevents the use of SBGMs for learning the underlying dynamics.

An approach of choice in such problems has been to use flow-based generative models, synonymous with continuous normalizing flows (CNFs) (Chen et al., 2018; Grathwohl et al., 2019; Finlay et al., 2020). Flow-based models assume a *deterministic* continuous-time generative process and fit an ordinary differential equation (ODE) that transforms the source density to the target density. Flow-based models were previously limited by inefficient simulation-based training objectives that require an expensive integration of the ODE at training time. However, recent work has introduced simulation-free training objectives that make CNFs competitive with SBGMs when a Gaussian source is assumed (Lipman et al., 2023; Liu et al., 2023b; Pooladian et al.,
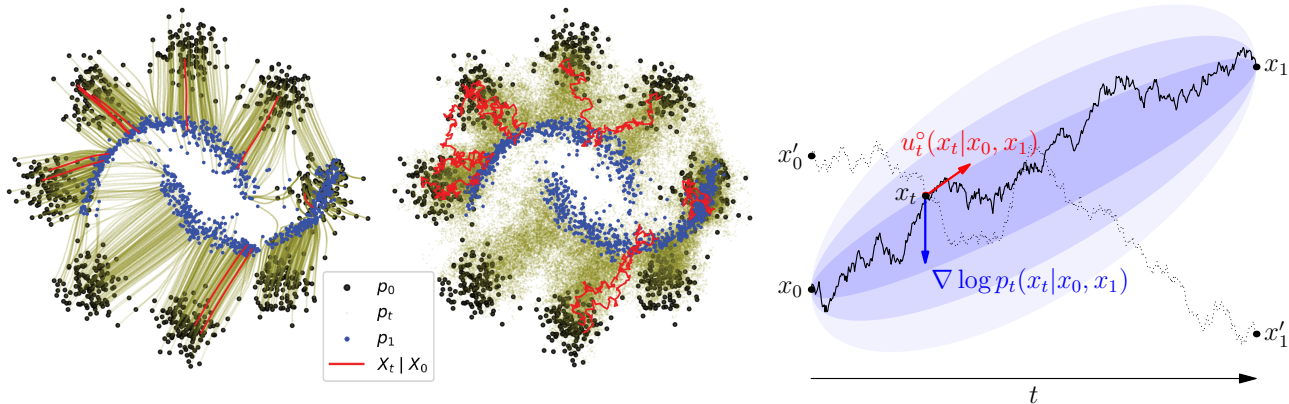
Figure 1: **Left:** ODE and SDE paths from 8gaussians to moons, sampled from a model trained using [SF]$^2$M. [SF]$^2$M makes it possible to vary the diffusion schedule at inference time and thus interpolate between ODEs and SDEs that have the same marginal densities. **Right:** Illustration of the stochastic regression objective in [SF]$^2$M. Given a source point $x_0$ and target point $x_1$ sampled from an entropic OT plan between marginals, an intermediate point $x_t$ is sampled from the Brownian bridge (marginal in light blue) in a simulation-free way. Neural networks are regressed to the ODE drift $u_t^{\circ}(x_t|x_0, x_1)$ and to the conditional score $\nabla \log p_t(x_t|x_0, x_1)$. The regression objective is stochastic, as the same point $x_t$ may appear on different conditional paths, *e.g.*, the dotted path from $x_0'$ to $x_1'$. The stochastic regression recovers dynamics that transform the marginal at time 0 to that at time 1.

2023) and extended these objectives to the case of arbitrary source distributions (Liu, 2022; Albergo and Vanden-Eijnden, 2023; Tong et al., 2024). However, these objectives do not yet apply to learning stochastic dynamics, which can be beneficial both for generative modeling and for recovering the dynamics of systems.

The Schrödinger bridge (SB) problem – the canonical probabilistic formulation of stochastically mapping between two arbitrary distributions – considers the most likely evolution between a source and target probability distributions under a given reference process (Schrödinger, 1932; Léonard, 2014b). The SB problem has been applied in a wide variety of problems, including generative modeling (De Bortoli et al., 2021; Vargas et al., 2021; Chen et al., 2022; Wang et al., 2021; Song and Ermon, 2019), modeling natural stochastic dynamical systems (Schiebinger et al., 2019; Holdijk et al., 2022; Koshizuka and Sato, 2023), and mean field games (Liu et al., 2022). Except for a small number of special cases (*e.g.*, Gaussian (Mallasto et al., 2022; Bunne et al., 2022a)), the SB problem typically does not have a closed-form solution, but can be approximated with iterative algorithms that require simulating the learned stochastic process (De Bortoli et al., 2021; Chen et al., 2022; Bunne et al., 2022a). While theoretically sound, these methods present numerical and practical issues that limit their scalability to high dimensions (Shi et al., 2023).

**This paper introduces a simulation-free objective for the Schrödinger bridge problem** called *simulation-free score and flow matching* ([SF]$^2$M). [SF]$^2$M simultaneously generalizes (1) the simulation-free objectives for CNFs (Tong et al., 2024; Liu et al.,

2023b) to the case of stochastic dynamics and (2) the denoising training objective for diffusion models to the case of arbitrary source distributions (Fig. 1). Our algorithm uses a connection between the SB problem and entropic optimal transport (OT) to express the Schrödinger bridge as a mixture of Brownian bridges (De Bortoli et al., 2021; Léonard, 2014b). In contrast to dynamic SB algorithms that require simulating an SDE at every iteration, [SF]$^2$M takes advantage of *static* entropic OT maps between source and target distributions, which are efficiently computed by the Sinkhorn algorithm (Sinkhorn, 1964).

We demonstrate the effectiveness of [SF]$^2$M on both synthetic and real-world datasets. On synthetic data, we show that [SF]$^2$M performs better than related prior work and finds a better approximation to the true Schrödinger bridge. As an application to real data, we consider modeling sequences of cross-sectional measurements (*i.e.*, unpaired time series observations) by a sequence of Schrödinger bridges. While there are many prior methods on modeling cells with Schrödinger bridges in the static setting (Schiebinger et al., 2019; Huguet et al., 2022b; Lavenant et al., 2021; Nolan et al., 2023) or low-dimensional dynamic setting (Bunne et al., 2022b,a; Koshizuka and Sato, 2023), [SF]$^2$M is the first method able to scale to thousands of gene dimensions, as its training is completely simulation-free. We also introduce a static manifold geodesic map which improves cell interpolations in the dynamic setting, demonstrating one of the first practical applications of Schrödinger bridge approximations with non-Euclidean costs. Finally, we show that unlike with static optimal transport, we are able to directly model and recover the gene-gene

interaction network driving the cell dynamics.

We summarize our **main contributions** below:

- We present [SF]$^2$M, the first simulation-free objective for the Schrödinger bridge problem, and prove its correctness.
- We study effective empirical and minibatch approximations to the entropic OT plan used in [SF]$^2$M.
- We validate our proposed method on synthetic distributions and in several single-cell dynamics problems.

## 2 PRELIMINARIES

We consider a pair of compactly supported distributions over $\mathbb{R}^d$ with (unknown) densities $q(x_0)$ and $q(x_1)$ (also denoted $q_0, q_1$). We assume access to a finite dataset of samples from $q_0$ and $q_1$. The problem of continuous-time stochastic generative modeling, or SDE inference, consists in finding a stochastic mapping $f$ that transforms $q_0$ to $q_1$. Samples from $q_1$ can then be generated by drawing a sample from $q_0$ and applying $f$ to obtain a sample from $q_1$.

### 2.1 Schrödinger bridges via entropic OT

The Schrödinger bridge problem asks to find most likely evolution between two probability measures $q_0$ and $q_1$ with respect to a reference stochastic process $\mathbb{Q}$. Formally, the Schrödinger bridge is the solution of:

$$\mathbb{P}^\star = \argmin_{\mathbb{P}:p_0=q_0,p_1=q_1} \mathrm{KL}(\mathbb{P} \,\|\, \mathbb{Q}), \qquad (1)$$

where $\mathbb{P}$ is a stochastic process (distribution over continuous paths $[0,1] \to \mathbb{R}^d$) with law $p$ (with marginals denoted $p_t$).

**SDEs and diffusion processes.** The stochastic processes we consider can be represented as Itô SDEs of the form $dx = u_t(x)\,dt + g(t)\,dw_t$, where $u_t$ is a smooth vector field and $dw_t$ a Brownian motion. A density $p(x_0)$ evolved according to a SDE induces marginal distributions $p_t(x_t)$, viewed as a function $p : [0,1] \times \mathbb{R}^d \to \mathbb{R}_+$. They are characterized by the initial conditions $p_0$ and the *Fokker-Planck equation* $\partial_t p_t = -\nabla \cdot (p_t u_t) + \frac{g^2(t)}{2}\Delta p_t$, where $\Delta p_t = \nabla \cdot (\nabla p_t)$ is the Laplacian.

In this work, we consider $\mathbb{Q} = \sigma\mathbb{W}$, where $\mathbb{W}$ is the standard Brownian motion defined by the SDE $dx = dw_t$, in which case the solution to (1) is known as the *diffusion Schrödinger bridge* (De Bortoli et al., 2021; Bunne et al., 2022a). We refer to Léonard (2014a,b) for a full discusssion on Schrödinger bridges.

**Entropically-regularized optimal transport.** The entropic OT problem is defined as follows:

$$\pi_\varepsilon^*(q_0, q_1) = \qquad\qquad\qquad\qquad (2)$$

$$\argmin_{\pi \in U(q_0,q_1)} \int d(x_0, x_1)^2 \, d\pi(x_0, x_1) + \varepsilon \, \mathrm{KL}(\pi \| q_0 \otimes q_1),$$

where $U(q_0, q_1)$ is the set of admissible transport plans (joint distributions over $x_0$ and $x_1$ whose marginals are equal to $q_0$ and $q_1$), $d(\cdot, \cdot)$ is the ground cost, $\varepsilon$ is the regularization parameter, and $q_0 \otimes q_1$ is the joint distribution over $x_0, x_1$ in which $x_0$ and $x_1$ are independent. When $\varepsilon \to 0$, we recover *exact optimal transport*. We now recall a cornerstone theorem that connects the SB problem to the entropic OT plan:

**Proposition 2.1** (Föllmer (1988)). *Let the reference process be a Brownian motion (i.e., $\mathbb{Q} = \sigma\mathbb{W}$). Then the Schrödinger bridge problem admits a unique solution $\mathbb{P}^*$ having the form of a mixture of Brownian bridges weighted by an entropic OT plan:*

$$\mathbb{P}^*((x_t)_{t \in [0,1]}) = \int \mathbb{Q}((x_t)_t \mid x_0, x_1) \, d\pi_{2\sigma^2}^\star(x_0, x_1) \quad (3)$$

*where $\mathbb{Q}((x_t)_{t \in (0,1)} \mid x_0, x_1)$ is the Brownian bridge between $x_0$ and $x_1$ with diffusion rate $\sigma$.*

Motivated by this result, the algorithm we propose stochastically regresses the parameters defining an unconditional SDE to those defining Brownian bridges.

### 2.2 Neural SDEs and probability flows

In this section, we consider an SDE $dx = u_t(x)\,dt + g(t)\,dw_t$. We review some important properties and discuss the approximation of $u_t$ by a neural network.

**Score and flow parametrization.** In the degenerate case $g(t) \equiv 0$, an SDE becomes an ODE and the Fokker-Planck equation recovers the *continuity equation* $\frac{\partial p}{\partial t} = -\nabla \cdot (p_t u_t)$. From the Fokker-Planck and continuity equations, it can easily be derived that the ODE

$$dx = \underbrace{\left[u_t(x) - \frac{g(t)^2}{2}\nabla \log p_t(x)\right]}_{u_t^\circ(x)} dt, \qquad (4)$$

together with a distribution over initial conditions $p(x_0)$, induces the same marginal distributions $p_t(\cdot)$ as the SDE; therefore, (4) is called the *probability flow ODE* of the stochastic process. Conversely, if the probability flow ODE's drift $u_t^\circ(x)$, the diffusion schedule $g(\cdot)$, and the *score function* $\nabla \log p_t(x)$ are known, then the SDE's drift term can be recovered via

$$u_t(x) = u_t^\circ(x) + \frac{g(t)^2}{2}\nabla \log p_t(x). \qquad (5)$$

Therefore, **specifying an SDE is tantamount to specifying the probability flow ODE and its**

**score function**. By reversing the sign of $u_t^\circ$ in the ODE (4) and converting it to an SDE using (5), we also get the time reversal formula from Anderson (1982):

$$dx = \left[-u_t^\circ(x) + \frac{g(t)^2}{2}\nabla \log p_t(x)\right] dt + g(t)\,dw_t$$
$$= \left[-u_t(x) + g(t)^2 \nabla \log p_t(x)\right] dt + g(t)\,dw_t, \quad (6)$$

which induces the same distribution on $x_{1-t}$ as the original SDE does on $x_t$.

**Approximating SDEs with neural networks.** If the marginal $p_t(x)$ can be tractably sampled and one knows the probability flow ODE's drift $u_t^\circ(x)$ and score $\nabla \log p_t(x)$, both can be approximated by neural networks. Specifically, time-varying vector fields $v_\theta(\cdot,\cdot) : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ and $s_\theta(\cdot,\cdot) : [0,1] \times \mathbb{R}^d \to \mathbb{R}_d$ can be trained with the *(unconditional) score and flow matching* objective

$$\mathcal{L}_{\mathrm{U[SF]^2M}}(\theta) = \quad (7)$$
$$\mathbb{E}\Big[\underbrace{\|v_\theta(t,x) - u_t^\circ(x)\|^2}_{\text{flow matching loss}} + \lambda(t)^2 \underbrace{\|s_\theta(t,x) - \nabla \log p_t(x)\|^2}_{\text{score matching loss}}\Big],$$

where the expectation is over $t \sim \mathcal{U}(0,1)$, $x \sim p_t(x)$ and $\lambda(\cdot)$ is some choice of positive weights. (In practice, it can be more stable to approximate $g(t)^2 \nabla \log p_t(x)$ rather than $\nabla \log p_t(x)$, a simple parametrization change that does not change the learning problem or the objective.) Once trained, $v_\theta$ and $s_\theta$ can be used to simulate the SDE from source samples $x_0$. This procedure is described in Alg. 2.

Remarkably, with a separate parametrization of the probability flow ODE and the score, we can simulate the SDE at inference time with an arbitrary diffusion rate $g(\cdot)$ that need not match the one used at training time. If the global optimum of (7) is attained (Fig. 1), we are ensured to obtain samples from the same marginals for any arbitrary diffusion rate $g(\cdot)$. For example, we can simulate the probability flow ODE by setting $g(t) \equiv 0$. Similarly, the backward SDE can be simulated starting at samples $x_1$ using the time reversal formula (6).

**ODEs and SDEs for Brownian bridges.** For processes whose marginals $p_t(x)$ are Gaussian, Theorem 3 of Lipman et al. (2023) or Theorem 2.1 of Tong et al. (2024) yield expressions for the flow and score. The main case of interest is the Brownian bridge from $x_0$ to $x_1$ with constant diffusion rate $g(t) = \sigma$. The marginals are given by $p_t(x) = \mathcal{N}(x; tx_1 + (1-t)x_0, \sigma^2 t(1-t))$, and the ODE and score are computed using the aforementioned result:

$$u_t^\circ(x) = \frac{1-2t}{t(1-t)}(x - (tx_1 + (1-t)x_0)) + (x_1 - x_0),$$
$$\nabla \log p_t(x) = \frac{tx_1 + (1-t)x_0 - x}{\sigma^2 t(1-t)}. \quad (8)$$

The Schrödinger bridge approximation algorithm we will propose leverages fast solutions to the entropy-regularized OT problem (2) and the closed-form $u_t^\circ$ and $\nabla \log p_t$ of Brownian bridges (8).

## 3 SIMULATION-FREE SDE TRAINING

We next describe our simulation-free method to learn SDEs through score and flow matching, summarized in Alg. 1. We present the general case in §3.1, then consider the Schrödinger bridge case in §3.2.

### 3.1 Matching the conditional flow and score

Tong et al. (2024) described a simulation-free stochastic regression objective, conditional flow matching (CFM), that fits an ODE generating marginal distributions given by mixtures of simpler probability paths. We generalize CFM to matching stochastic dynamics.

Suppose the stochastic process $\mathbb{P}((x_t)_{t \in [0,1]})$, with marginals $p_t(x)$, is a mixture over a latent variable $z$ with density $q(z)$, *i.e.*,

$$\mathbb{P}((x_t)_{t \in [0,1]}) = \int \mathbb{P}((x_t)|z)q(z)\,dz. \quad (9)$$

Suppose that $\mathbb{P}((x_t)|z)$ is defined by the SDE $dx = u_t(x|z)\,dt + g(t)\,dw_t$ with initial conditions $p_0(x|z)$, and let $u_t^\circ(x|z)$ be drift of the corresponding probability flow ODE given by (4). One then has expressions for the probability flow ODE and score that generate the marginals of the process $\mathbb{P}$ given initial conditions $p_0(x) = \int_z p_0(x|z)q(z)\,dz$:

$$u_t^\circ(x) = \mathbb{E}_{q(z)} \frac{u_t^\circ(x|z)p_t(x|z)}{p_t(x)},$$
$$\nabla \log p_t(x) = \mathbb{E}_{q(z)}\left[\frac{p_t(x|z)}{p_t(x)}\nabla \log p_t(x|z)\right]. \quad (10)$$

To be precise, we generalize Theorem 3.1 from Tong et al. (2024) to stochastic settings:

**Theorem 3.1.** *Under mild regularity conditions, the ODE $dx = u_t^\circ(x)\,dt$ generates the marginals $p_t$ of $\mathbb{P}$ from initial conditions $p_0$, and the score is given by (10). The SDE $dx = [u_t^\circ(x) + \frac{1}{2}g(t)^2 \nabla \log p_t(x)]\,dx + g(t)\,dt$ generates the Markovization of $\mathbb{P}$.*

We emphasize that, in general, the SDE in Theorem 3.1 does not recover $\mathbb{P}$, but only its Markovization (*i.e.*, the process with the same infinitesimal transition kernel). A process $\mathbb{P}$ of the form (9) is not necessarily Markovian and may not be generated by any SDE.

**A stochastic regression objective.** The marginal ODE drift and score expressions in (10) motivate objectives for fitting $u_t^\circ(x)$ and $\nabla \log p_t(x)$ with neural

---

**Algorithm 1** Score and Flow Matching Training

**Input:** Efficiently samplable $q(z)$, $p_t(x|z)$, computable $u_t(x|z)$, initial networks $v_\theta$ and $s_\theta$.
**while** Training **do**

> $z \sim q(z);\quad t \sim \mathcal{U}(0,1);\quad x \sim p_t(x|z)$
> $\mathcal{L}_{[SF]^2M} \leftarrow \|v_\theta(t,x) - u_t^\circ(x|z)\|^2$
> $\quad + \lambda(t)^2 \|s_\theta(t,x) - \nabla_x \log p_t(x|z)\|^2 \quad \triangleright \text{ see (11)}$
> $\theta \leftarrow \text{Update}(\theta, \nabla_\theta \mathcal{L}_{[SF]^2M})$

**return** $v_\theta, s_\theta$

---

networks when only the *conditional* ODEs and scores are known. Generalizing (7), we define the *(conditional) simulation-free score and flow matching* objective ([SF]$^2$M) for neural networks $v_\theta(\cdot,\cdot)$ approximating the ODE drift and $s_\theta(\cdot,\cdot)$ approximating the score:

$$\mathcal{L}_{[SF]^2M}(\theta) = \mathbb{E}_{Q'} \underbrace{\|v_\theta(t,x) - u_t^\circ(x|z)\|^2}_{\text{conditional flow matching loss}} \quad (11)$$

$$+ \mathbb{E}_{Q'} \lambda(t)^2 \underbrace{\|s_\theta(t,x) - \nabla \log p_t(x|z)\|^2}_{\text{conditional score matching loss}},$$

where, as in (7), $\lambda(\cdot)$ is some choice of positive weights and $Q' = (t \sim \mathcal{U}(0,1)) \otimes q(z) \otimes p_t(x|z)$. This objective can be used to approximate the quantities defined in (10), provided the conditional ODEs and scores are known and $p_t(x|z)$ can be tractably sampled. Correctness is guaranteed by the following Theorem:

**Theorem 3.2** (Equality of conditional gradients). *If $p_t(x) > 0$ for all $x \in \mathbb{R}^d$ and $t \in [0,1]$, then $\nabla_\theta \mathcal{L}_{U[SF]^2M}(\theta) = \nabla_\theta \mathcal{L}_{[SF]^2M}(\theta)$, where $\mathcal{L}_{U[SF]^2M}(\theta)$ is the unconditional score and flow matching loss (7).*

This result generalizes Theorem 3.2 of Tong et al. (2024). It provides a simulation-free way to train neural networks sufficient to simulate an SDE generating marginals $p_t(x)$ with arbitrary diffusion rate $g(\cdot)$ (cf. the discussion following (7)). The training and inference algorithms are summarized in Alg. 1, Alg. 2.

In our approach, the SDE recovered from the ODE and score defined via (10) is the Markovization of the mixture of stochastic processes indexed by $z$.

**Sources of conditional ODEs and scores.** Although the [SF]$^2$M framework can handle general conditioning information $z$, in this paper we consider the case where $z$ is identified with a pair $(x_0, x_1)$ of a source and target point. For a given $z = (x_0, x_1)$, we will take the conditional probability path $p_t(x|z)$ to be a Brownian bridge with constant diffusion scale $\sigma$, so that $u_t^\circ(x|z)$ and $\nabla \log p_t(x|z)$ are given by (8). To avoid numerical issues for $t$ close to 0 or 1, we add a small smoothing constant to the variance. The conditional distributions are thus peaky at $x_0$ and $x_1$ at $t = 0$ and $t = 1$. (An extension to nonconstant diffusion scale is described in Appendix E.)

For the resulting marginal $p_t(x)$ to satisfy the boundary conditions $p_0(x) = q_0(x)$ and $p_1(x) = q_1(x)$, $q(x_0, x_1)$ must be a coupling of $q_0$ and $q_1$ (*i.e.*, a transport plan). This is formalized in the following theorem:

**Theorem 3.3** ([SF]$^2$M recovers marginals from bridges). *If $q(\cdot,\cdot) \in U(q_0, q_1)$ and $v_\theta^*, s_\theta^*$ globally minimize $\mathcal{L}_{[SF]^2M}(\theta)$, the SDE with drift $[v_\theta^* + \frac{1}{2}g(t)^2 s_\theta^*]$ and diffusion $g$, and initial conditions $p_0 = q_0$, is the Markovization of the mixture of Brownian bridges from $x_0$ to $x_1$ over $q(x_0, x_1)$. In particular, if the SDE generates marginals $p_t$, then $p_1 = q_1$.*

This theorem tells us that as long as our joint distribution $q(x_0, x_1)$ has the correct marginals, [SF]$^2$M will recover a valid generative model which pushes $q_0$ to $q_1$.

### 3.2 Building Schrödinger bridges via [SF]$^2$M and entropic optimal transport

In the previous section, we showed that our method, [SF]$^2$M, can approximate the marginal probability $p_t$ of a mixture of processes of the form (9). In this section, we explain how our [SF]$^2$M approximates the SB.

**[SF]$^2$M approximates the Schrödinger bridge.** In order to achieve an efficient approximation of the SB, we leverage Proposition 2.1. The SB can be expressed as a mixture of Brownian bridges weighted by an entropic optimal transport plan (3). Therefore, to approximate the SB with [SF]$^2$M, we set the distribution $q(x_0, x_1)$ to be equal to the entropic OT plan $\pi_{2\sigma^2}^\star(q_0, q_1)$ and train the networks $v_\theta$ and $s_\theta$ using Alg. 1. We show that this procedure recovers the SB:

**Proposition 3.4** ([SF]$^2$M with entropic OT recovers the SB process). *Let $\mathbb{P}^*$ be the Schrödinger bridge between $q_0$ and $q_1$ with respect to $\mathbb{Q} = \sigma\mathbb{W}$. If $v_\theta^\star, s_\theta^\star$ globally minimize $\mathcal{L}_{[SF]^2M}$, with coupling $\pi_{2\sigma^2}^\star(q_0, q_1)$, then $\mathbb{P}^*$ is defined by the SDE with drift $[v_\theta^* + \frac{1}{2}g(t)^2 s_\theta^*]$, diffusion $g$, and initial conditions $p_0 = q_0$.*

**Empirical approximation.** Unfortunately, the real distributions $q_0$ and $q_1$ are usually unknown and we only have access to *i.i.d.* samples forming empirical distributions $\hat{q}_0$ and $\hat{q}_1$ of size $n$. Therefore, we can only approximate the true entropic OT plan by computing the entropic OT plan $\pi_{2\sigma^2}^\star(\hat{q}_0, \hat{q}_1)$ between the empirical distributions (Cuturi, 2013; Altschuler et al., 2017). This empirical OT plan can be used in [SF]$^2$M to construct an *empirical* Schrödinger bridge.

Fortunately, the true entropic OT can be efficiently approximated using empirical distributions, even in high-dimensional spaces (Genevay et al., 2019; Mena and Niles-Weed, 2019), and it was recently shown that the Schrödinger bridge inherits this property (Stromme, 2023, Theorem 5). In turn, the entropic OT plan between empirical distributions can be efficiently com-

Table 1: Comparison of SB algorithms (see §6). $[SF]^2M$ is the first algorithm that does not assume paired samples, require SDE integration during training, or use an IPF outer loop. DSBM uses simulation only in the outer loop.

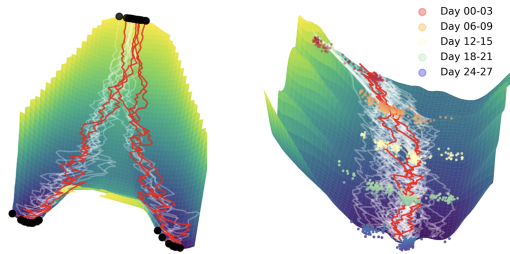| Algorithm → | I²SB/ASB | DSB/NLSB | DSBM/IDBM | $[SF]^2M$ |
|---|---|---|---|---|
| Unpaired samples | ✗ | ✓ | ✓ | ✓ |
| Bridge matching | ✓ | ✗ | ✓ | ✓ |
| Single loop | ✓ | ✗ | ✗ | ✓ |
| Sim.-free training | ✓ | ✗ | ✗/✓ | ✓ |
| No explicit $(x_0, x_1)$ pairing | ✓ | ✓ | ✓ | ✗ |



Figure 2: Visualization of learned Waddington's landscape $W$ with a bifurcating trajectory for one Gaussian to two Gaussians (**left**) and for the Embryoid Body (EB) data (Moon et al., 2019) (**right**). The dimensions are space (left-right), time (forward-back), and potential (up-down).

puted using the Sinkhorn algorithm (Cuturi, 2013), which has $\mathcal{O}(n^2)$ computational complexity (Altschuler et al., 2017), or using stochastic algorithms (Genevay et al., 2016; Seguy et al., 2018). However, if this cost is to high (*e.g.*, if $n$ is too large or if one has the true generative process, as in the Gaussian-to-data setting), the plan can be further approximated using minibatch OT (Fatras et al., 2020, 2021a); see Appendix A.

The use of an entropic OT plan and marginalization via stochastic regression distinguishes $[SF]^2M$ from existing neural SB algorithms (Table 1). Such past approaches include mean-matching (DSB and NLSB, De Bortoli et al., 2021; Koshizuka and Sato, 2023), and bridge-matching approaches (DSBM and IDBM, Shi et al., 2023; Peluchetti, 2023), both of which require an outer iterative proportional fitting loop with an inner training loop. Others have studied the problem assuming paired source and target data (I²SB and ASB, Liu et al., 2023a; Somnath et al., 2023); $SF^2M$ can be thought of as *inferring* the pairing while jointly fitting the SDE.

See §C.3 for further discussion of the implications of these choices and practical recommendations.

## 4 LEARNING CELL DYNAMICS WITH $[SF]^2M$

Modeling cell dynamics is a major open problem in single-cell data science, as it is important for understanding – and eventually intervening in – cellular programs of development and disease (Lähnemann et al., 2020). In this section, we show how $[SF]^2M$ can be used for and adapted to modeling single-cell dynamics.

The cellular dynamics between time-resolved snapshot data, representing observations of cells lying in the space of gene activations, are commonly modeled using Schrödinger bridges (Hashimoto et al., 2016; Schiebinger et al., 2019; Bunne et al., 2022a; Koshizuka and Sato, 2023). The applicability of the SB formulation to cell dynamics relies upon the principle of least action, which is thought to hold for cellular systems over short timescales (Schiebinger, 2021), and motivates our choice to apply $[SF]^2M$ to these problems.

**Learning flows on cell manifolds.** Cells are thought to lie on a low-dimensional manifold in the space of gene expressions (Moon et al., 2018), which has motivated work on density-adhering regularizations (Tong et al., 2020; Koshizuka and Sato, 2023) and manifold embeddings (Huguet et al., 2022a). Because $[SF]^2M$ can use a coupling between marginals $q(x_0, x_1)$ defined by entropic OT with an arbitrary cost function, we can take advantage of these embeddings to compute the pairing using a ground cost that is adapted to the geometry of the manifold. Specifically, we use the Geodesic Sinkhorn method (Huguet et al., 2022b), which computes the entropic OT plan with cost

$$c_{\text{geo}}(x_0, x_1) = \sqrt{-\log \mathcal{H}_t(x_0, x_1)}. \quad (12)$$

The matrix $\mathcal{H}_t$ approximates the heat kernel defined via the Laplace-Beltrami operator on the manifold, efficiently approximated using a $k$-nearest-neighbour graph. We find using this cost leads to more accurate trajectories in high dimensions (see Table 5).

**Learning developmental landscapes.** A common model of cell development, known as Waddington's epigenetic landscape (Waddington, 1942), assumes that cells evolve and differentiate in the space of gene expressions by following (noisy) gradient ascent on an energy function. While a few heuristic methods have been proposed to approximate this energy function from single-cell data before (Tang, 2017; Qin et al., 2023), we propose a novel approach to model the landscape directly. In our approach, the negative energy can be directly interpreted as an action potential, inspired by the modeling in Neklyudov et al. (2023).

To do this, we impose a Langevin dynamics parametrization on the flow and score in $[SF]^2M$: $v_\theta(t, x) = -\nabla_x E_v(t, x)$ and $s_\theta(t, x) = -\nabla_x E_s(t, x)$, where $E_v$ and $E_s$ are neural networks. We can define the Waddington's landscape by $W := E_v + \frac{1}{2}g(t)^2 E_s$. The drift of the SDE is then $u_t(x) = -\nabla_x W(t, x)$, meaning that the time-evolution of a cell follows gradient dynamics on $W$ with added Gaussian noise of

Table 2: Two-dimensional data: generative modeling performance ($\mathcal{W}_2$) and dynamic OT optimality (NPE) divided into SDE methods (top) and ODE methods (bottom). [SF]²M performs the best on 3 of 4 datasets and is similar to OT-CFM, which is equivalent to [SF]²M as $g(t) \to 0$. *Indicates results taken from Shi et al. (2023).

| Metric → | $\mathcal{W}_2$ ($\downarrow$) | | | | Normalized Path Energy ($\downarrow$) | | | |
|---|---|---|---|---|---|---|---|---|
| Algorithm ↓ \| Dataset → | $\mathcal{N}{\to}$8gaussians | moons$\to$8gaussians | $\mathcal{N}{\to}$moons | $\mathcal{N}{\to}$scurve | $\mathcal{N}{\to}$8gaussians | moons$\to$8gaussians | $\mathcal{N}{\to}$moons | $\mathcal{N}{\to}$scurve |
| [SF]²M-Exact | **0.275**±0.058 | 0.726±0.137 | **0.124**±0.023 | **0.128**±0.005 | **0.016**±0.012 | 0.045±0.031 | **0.053**±0.038 | 0.034±0.024 |
| [SF]²M-I | 0.393±0.054 | 1.482±0.151 | 0.185±0.028 | 0.201±0.062 | 0.160±0.019 | 2.577±0.323 | 0.855±0.130 | 0.845±0.106 |
| DSBM-IPF (Shi et al., 2023)* | 0.315±0.079 | 0.812±0.092 | 0.140±0.006 | 0.140±0.005 | 0.022±0.020 | 0.244±0.027 | 0.383±0.034 | 0.297±0.036 |
| DSBM-IMF (Shi et al., 2023)* | 0.338±0.091 | 0.838±0.098 | 0.144±0.005 | 0.145±0.037 | 0.029±0.017 | 0.345±0.049 | 0.230±0.028 | 0.286±0.033 |
| DSB (De Bortoli et al., 2021)* | 0.411±0.084 | 0.987±0.324 | 0.190±0.049 | 0.272±0.065 | — | — | — | — |
| OT-CFM (Tong et al., 2024) | 0.303±0.043 | **0.601**±0.027 | 0.130±0.016 | 0.144±0.028 | 0.031±0.027 | **0.015**±0.010 | 0.083±0.009 | **0.027**±0.012 |
| SB-CFM (Tong et al., 2024) | 2.314±2.112 | — | 0.434±0.594 | 0.341±0.468 | 1.000±0.000 | — | 0.995±0.000 | 0.745±0.039 |
| RF (Liu, 2022) | 0.421±0.071 | 1.525±0.330 | 0.283±0.045 | 0.345±0.079 | 0.044±0.031 | 0.203±0.090 | 0.130±0.078 | 0.099±0.066 |
| I-CFM (Tong et al., 2024) | 0.373±0.103 | 1.557±0.407 | 0.178±0.014 | 0.242±0.141 | 0.202±0.055 | 2.680±0.292 | 0.891±0.120 | 0.856±0.031 |
| FM (Lipman et al., 2023) | 0.343±0.058 | — | 0.209±0.055 | 0.198±0.037 | 0.190±0.054 | — | 0.762±0.099 | 0.743±0.116 |

scale $g(t)$. We visualize these landscapes in Fig. 2 with further details in §F.6.

**Learning gene regulatory networks.** Finally, we use [SF]²M to learn gene regulatory networks from population snapshots of gene expressions, a persisting challenge in cellular biology (Pratapa et al., 2020). Following previous work in discovering sparse interaction structure from continuous-time systems (Tank et al., 2021; Aliee et al., 2021; Bellot and Branson, 2022; Aliee et al., 2022; Atanackovic et al., 2023), we define the *gene regulatory network* as the directed graph whose vertices are genes (dimensions of the space) and an edge $i \to j$ is present if and only if $\frac{\partial (v_\theta(t,x))_j}{\partial x_i} \neq 0$. This directed graph is expected to be sparse.

Previous work resorted to performing inference of trajectories in a low-dimensional (and dense) representation (Tong et al., 2023; Bunne et al., 2022b), which complicated the discovery of the sparse graph structure in gene space. [SF]²M is the first Schrödinger bridge method to scale to high dimensions. This allows us to learn a dynamic directly in the gene space and recover the sparse gene interactions. To accomplish this, we use a specialized parametrization of $v_\theta$, inspired by Bellot and Branson (2022), which enables the graph structure to be read out from the sparsity pattern of the initial layer of the trained model (see §F.7 for details).

## 5 RELATED WORK

**Stochastic continuous-time modeling.** Our framework is related to both flow-based (Chen et al., 2018; Grathwohl et al., 2019; Albergo et al., 2023; Albergo and Vanden-Eijnden, 2023; Neklyudov et al., 2022; Liu, 2022) and score-based (Sohl-Dickstein et al., 2015; Song and Ermon, 2019, 2020; Song et al., 2021b; Ho et al., 2020; Winkler et al., 2023; Dhariwal and Nichol, 2021; Watson et al., 2023) generative modeling. Both have drawn attention due to their stability and efficiency in training and high quality of generated samples. See Appendix C for further discussion.

**Schrödinger bridge approximation methods.** While there is significant theoretical work on the SB problem (Léonard, 2014b; Stromme, 2023; Albergo et al., 2023), practical solutions have assumed paired samples from the Schrödinger bridge or required simulation during training. Algorithms based on iterative proportional fitting (DSB and DSBM; De Bortoli et al., 2021; Shi et al., 2023) have the advantage of yielding the exact Schrödinger bridge if trained to optimality on each iteration, but may accumulate error with each outer-loop step due to underfitting and function approximation. On the other hand, our proposed [SF]²M requires neither training-time integration nor outer-loop iteration and therefore will converge to the exact SB – if the neural network function class and learning algorithm so allow – but, unlike DSB and DSBM, require knowledge of the entropic OT plan and the conditional paths (see Table 1).

The relative advantages of these algorithms and practical recommendations are further discussed in §3.2, §C.3. Furthermore, in Appendix D we show that the simulation-based outer loop in Peluchetti (2023); Shi et al. (2023) can be combined with [SF]²M to improve the SB marginals at the cost of generative performance.

**Applications to cell dynamics.** When the observer seeks to recover dynamics from multiple snapshots with scRNA-seq data, the machinery of optimal transport can be used (Schiebinger et al., 2019; Yang et al., 2020; Tong et al., 2020; Bunne et al., 2022b; Huguet et al., 2022a; Bunne et al., 2022a; Koshizuka and Sato, 2023). However, these methods all require simulation during training, which scales poorly to high dimensions.

## 6 EXPERIMENTS

In this section we empirically evaluate [SF]²M with respect to optimal transport, generative modeling, and single-cell interpolation criteria. We compare:

- Minibatch [SF]²M with exact OT minibatches ([SF]²M-Exact), with entropic OT (Sinkhorn) mini-

Table 3: Gaussian-to-Gaussian Schrödinger bridges with $10^4$ datapoints between a Gaussian with parameters estimated from empirical samples $(p_t)$ with error to the continuous Schrödinger bridge marginals $(p_t^*)$ either at the target distribution (left) or averaged across 21 timepoints (right).

| Metric → | $KL(p_1, p_1^*)$ | | | Mean $KL(p_t, p_t^*)$ | | |
|---|---|---|---|---|---|---|
| Alg. ↓ \| Dim. → | 5 | 20 | 50 | 5 | 20 | 50 |
| [SF]²M-Exact | $0.007_{\pm0.000}$ | $\mathbf{0.029}_{\pm0.002}$ | $\mathbf{0.124}_{\pm0.003}$ | $0.006_{\pm0.000}$ | $\mathbf{0.028}_{\pm0.001}$ | $0.258_{\pm0.001}$ |
| DSBM-IPF | $0.015_{\pm0.005}$ | $0.132_{\pm0.004}$ | $0.528_{\pm0.013}$ | $\mathbf{0.005}_{\pm0.002}$ | $0.050_{\pm0.002}$ | $\mathbf{0.221}_{\pm0.004}$ |
| DSB | $8.757_{\pm-}$ | $49.963_{\pm-}$ | $221.213_{\pm-}$ | $8.757_{\pm-}$ | $49.963_{\pm-}$ | $221.213_{\pm-}$ |
| SB-CFM | $\mathbf{0.001}_{\pm0.000}$ | $0.034_{\pm0.003}$ | $0.170_{\pm0.002}$ | $0.008_{\pm0.000}$ | $0.086_{\pm0.002}$ | $0.447_{\pm0.003}$ |

Table 4: Single-cell comparison over three datasets, averaged over leaving out different intermediate timepoints on 5 PCs. For each left-out point, we measure the 1-Wasserstein distance between the imputed and ground truth distributions at the left-out time point, following Tong et al. (2020). *Indicates values taken from aforementioned work.

| Algorithm ↓ \| Dataset → | Cite | EB | Multi |
|---|---|---|---|
| [SF]²M-Geo | $1.017_{\pm0.104}$ | $0.879_{\pm0.148}$ | $1.255_{\pm0.179}$ |
| [SF]²M-Exact | $\mathbf{0.920}_{\pm0.049}$ | $\mathbf{0.793}_{\pm0.066}$ | $\mathbf{0.933}_{\pm0.054}$ |
| [SF]²M-Sink | $1.054_{\pm0.087}$ | $1.198_{\pm0.342}$ | $1.098_{\pm0.308}$ |
| DSBM (Shi et al., 2023) | $1.705_{\pm0.160}$ | $1.775_{\pm0.429}$ | $1.873_{\pm0.631}$ |
| DSB (De Bortoli et al., 2021) | $0.953_{\pm0.140}$ | $0.862_{\pm0.023}$ | $1.079_{\pm0.117}$ |
| OT-CFM (Tong et al., 2024) | $\mathbf{0.882}_{\pm0.058}$ | $\mathbf{0.790}_{\pm0.068}$ | $\mathbf{0.937}_{\pm0.054}$ |
| I-CFM (Tong et al., 2024) | $0.965_{\pm0.111}$ | $0.872_{\pm0.087}$ | $1.085_{\pm0.099}$ |
| SB-CFM (Tong et al., 2024) | $1.067_{\pm0.107}$ | $1.221_{\pm0.380}$ | $1.129_{\pm0.363}$ |
| Reg. CNF (Finlay et al., 2020)* | — | 0.825 | — |
| TrajectoryNet (Tong et al., 2020)* | — | 0.848 | — |
| NLSB (Koshizuka and Sato, 2023) | — | 0.970 | — |

Table 5: Leave-one-timepoint-out testing of dynamics interpolation methods measuring the error between the predicted and ground truth left out timepoint using the 1-Wasserstein distance. We test on 50 and 100 principal components as well as 1000 highly variable genes.

| Dim. → | 50 | | 100 | | 1000 | |
|---|---|---|---|---|---|---|
| Alg. ↓ \| Dataset → | Cite | Multi | Cite | Multi | Cite | Multi |
| [SF]²M-Geo | $\mathbf{38.52}_{\pm0.29}$ | $44.80_{\pm1.91}$ | $44.50_{\pm0.42}$ | $52.20_{\pm1.96}$ | $40.09_{\pm1.53}$ | $51.29_{\pm0.09}$ |
| [SF]²M-Exact | $40.01_{\pm0.78}$ | $45.34_{\pm2.83}$ | $46.53_{\pm0.43}$ | $52.89_{\pm1.99}$ | $43.66_{\pm0.72}$ | $53.15_{\pm1.86}$ |
| DSBM | $53.81_{\pm7.74}$ | $66.43_{\pm14.39}$ | $58.99_{\pm7.62}$ | $70.75_{\pm14.03}$ | $50.09_{\pm4.81}$ | $61.71_{\pm13.90}$ |
| OT-CFM | $38.76_{\pm0.40}$ | $47.58_{\pm6.62}$ | $45.39_{\pm0.42}$ | $54.81_{\pm5.86}$ | $43.25_{\pm0.73}$ | $52.29_{\pm1.55}$ |
| I-CFM | $41.83_{\pm3.28}$ | $49.78_{\pm4.43}$ | $48.28_{\pm3.28}$ | $57.26_{\pm3.86}$ | $44.12_{\pm0.52}$ | $52.99_{\pm1.50}$ |

batches (-Sink), with independent couplings (-I), and with Geodesic OT (-Geo) when applicable.

- A variety of (ODE) flow-based models, including optimal transport conditional flow matching (OT-CFM, Tong et al., 2024), rectified flow (RF, Liu, 2022), and flow matching (FM, Lipman et al., 2023).
- Schrödinger bridge models: diffusion Schrödinger bridges (DSB, De Bortoli et al., 2021) and diffusion Schrödinger bridge matching (DSBM, Shi et al., 2023), which is equivalent to work on iterated diffusion mixture transport (IDBM, Peluchetti, 2023).
- Single-cell dynamics models: Neural Lagrangian Schrödinger bridges (NLSB, Koshizuka and Sato, 2023), TrajectoryNet (Tong et al., 2020).

See Appendix F for all experiment details. All results are presented as mean ± std. over five seeds.

**[SF]²M is a competitive generative model for low-dimensional data.** We first evaluate in Table 2 how well various methods approximate dynamic optimal transport on low-dimensional datasets (8gaussians, moons, and scurve). We train Schrödinger bridges between a Gaussian and each dataset, and between 8gaussians and moons, using [SF]²M. We report the 2-Wasserstein distance between the predicted distribution and the target distribution with samples of size 10,000. Following Tong et al. (2024), we also report the Normalized Path Energy relative to the 2-Wasserstein distance, defined as $\mathrm{NPE}(p,q) := |\int \|v_\theta(t,x)\|^2 dt - \mathcal{W}_2^2(p,q)|/\mathcal{W}_2^2(p,q)$. This metric is equal to zero if and

only if $v_\theta$ solves the dynamic optimal transport problem. Table 2 summarizes our results, showing that [SF]²M outperforms all methods, both stochastic (top) and deterministic (bottom). Despite minibatch OT bias (which can be seen as a form of regularization like the entropic regularization (Fatras et al., 2020)), we find [SF]²M-Exact approximates the Schrödinger bridge best, with the OT computation accounting for only 1% of the training time on batch sizes of 512.

**[SF]²M recovers the SB.** Next we evaluate how well [SF]²M can model Schrödinger bridge marginals. We use a Gaussian-to-Gaussian Schrödinger bridge because it has closed-form Gaussian marginals (Mallasto et al., 2022; Bunne et al., 2022a) following De Bortoli et al. (2021). After training all methods, we evaluate the quality of empirical marginals with respect to the ground truth by sampling trajectories using Alg. 2. We compute the KL divergence between a Gaussian approximation of the empirical marginal and the Gaussian marginal of the ground truth Schrödinger bridge at multiple timepoints. This evaluation is shown in Table 3 at just the last timepoint ($t = 1$) and an average over 21 equally spaced timepoints. We train each method for an equal number of steps, using 20 outer loops for DSB and DSBM, which require iterative optimization of forward and backward models. We find that in low dimensions SB-CFM, which corresponds to [SF]²M's probability ODE flow, performs the best, closely followed by [SF]²M-Exact. In high dimensions, [SF]²M-Exact better matches the target distribution, and performs similarly to DSBM and significantly better than DSB on the intermediate marginals.

**[SF]²M accurately models high-dimensional single cell dynamics.** We train our method [SF]²M on single cell dynamics, as described in §4, on three real-world datasets in the setup established by Tong et al. (2020) (see §F.6) and gathered our results for different dimensions in Table 4 and Table 5. Given $K$ unpaired data distributions representing a cell population at $K$ different timepoints, we solve a SB problem between every two successive time points, sharing parameters between the models. To test the interpolation ability of the trained models, we perform leave-one-out interpolation, predicting timepoint $k$ using a model trained on
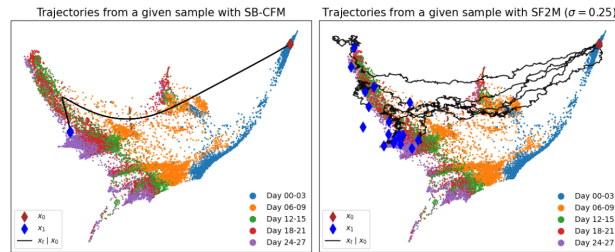
Figure 3: Simulation of trajectories from a given cell on 2D EB data. **Left:** Probability flow ODE trajectory, approximated by SB-CFM (Tong et al., 2024). **Right:** Five SDE trajectories from [SF]$^2$M; more target samples (20) in blue.

Table 6: GRN recovery from simulated time-lapsed single-cell gene expression. Shows structure predictive performance in terms of area under the receiver operator characteristic (AUC-ROC) and average precision (AP).

| GRN → | Bifurcating System | | Trifurcating System | |
|---|---|---|---|---|
| Alg. ↓ \| Metric → | AUC-ROC (↑) | AP (↑) | AUC-ROC (↑) | AP (↑) |
| NGM-[SF]$^2$M$_{\sigma=0}$ | **0.786**$_{\pm0.081}$ | **0.521**$_{\pm0.160}$ | **0.764**$_{\pm0.066}$ | **0.485**$_{\pm0.105}$ |
| NGM-[SF]$^2$M$_{\sigma=0.1}$ | 0.723$_{\pm0.014}$ | 0.444$_{\pm0.030}$ | 0.731$_{\pm0.077}$ | 0.453$_{\pm0.091}$ |
| Spearman | 0.755$_{\pm0.003}$ | 0.438$_{\pm0.002}$ | 0.718$_{\pm0.00}$ | 0.413$_{\pm0.005}$ |
| Pearson | 0.744$_{\pm0.000}$ | 0.415$_{\pm0.000}$ | 0.710$_{\pm0.00}$ | 0.405$_{\pm0.002}$ |
| DREMI | 0.594$_{\pm0.017}$ | 0.293$_{\pm0.011}$ | 0.419$_{\pm0.02}$ | 0.205$_{\pm0.007}$ |
| Granger | 0.664$_{\pm0.013}$ | 0.421$_{\pm0.04}$ | 0.613$_{\pm0.04}$ | 0.343$_{\pm0.039}$ |
| SCODE | 0.570$_{\pm0.036}$ | 0.370$_{\pm0.028}$ | 0.570$_{\pm0.06}$ | 0.332$_{\pm0.077}$ |

timepoints $[1, \ldots, k-1, k+1, \ldots, K]$. We consider four data representations of different dimensionality: using the first 5, 50, or 100 whitened principal components and using the 1000 dimensions corresponding to the most highly variable genes (Wolf et al., 2018). [SF]$^2$M performs the best among the Schrödinger bridge methods and similarly to the ODE-based OT-CFM on the low-dimensional data (Table 4), but is better in higher dimensions (Table 5). As the number of dimensions grows, the advantage of geodesic interpolation ([SF]$^2$M-Geo) becomes apparent.

We also compare the stochastic process modeled by [SF]$^2$M to its probability flow ODE SB-CFM. We show in Fig. 3 that [SF]$^2$M models a stochastic dependence of the output on the input, unlike SB-CFM, despite the two algorithms sharing marginal densities. This is important in the EB data as the initial stem cells are thought to be pluripotent and should evolve stochastically into differentiated cell types over time. Such differentiation cannot be modelled by ODE-based methods, thus motivating the use of a stochastic process for modeling single-cell dynamics.

**[SF]$^2$M can be used to recover gene regulatory networks.** We demonstrate the use-case of [SF]$^2$M for recovering gene regulatory networks (GRNs) from single-cell gene expression using the algorithm described at the end of §4. We show how we can use [SF]$^2$M to simultaneously learn dynamics and GRN structure from single-cell gene expression data. We use BoolODE (Pratapa et al., 2020) to simulate two single-cell systems given ground truth GRNs: (1) a system with bifurcating trajectories (7 genes), and (2) a system with trifurcating trajectories (9 genes). We summarize our results in Table 6. We also measure how accurately the ground truth GRN is recovered using the standard AUC-ROC and average precision (AP) metrics. We find that [SF]$^2$M-Exact with no noise (corresponding to OT-CFM (Tong et al., 2024)) performs best at inferring the underlying GRN as compared to correlation baselines (Pearson and Spearman correlation), a mutual information baseline DREMI (Krishnaswamy et al.,

2014), and pairwise Granger causality (Granger, 1969). Higher values of $\sigma$ do not perform as well, but still outperform the baselines on most metrics.

# 7 CONCLUSION

We have introduced a novel class of simulation-free objectives for learning continuous-time stochastic generative models between general source and target distributions. For sources and targets with finite support, we can directly approximate the continuous-time Schrödinger bridge without simulation by computing the entropic OT plan via efficient algorithms. We have shown how our method can be applied to learn cell dynamics and extract the gene regulatory structure. Future work can consider how to train [SF]$^2$M-like models with interventional data to improve GRN inference.

**Limitations.** The main limitation of [SF]$^2$M is that it requires knowledge of conditional path distributions (Brownian bridges). These distributions are not available in closed form if one considers more general reference processes (Fernandes et al., 2022), which may be useful to encode biological priors (Koshizuka and Sato, 2023), or on general Riemannian manifolds.

# Acknowledgments

## References

Albergo, M. S., Boffi, N. M., and Vanden-Eijnden, E. (2023). Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint 2303.08797*.

Albergo, M. S. and Vanden-Eijnden, E. (2023). Building normalizing flows with stochastic interpolants. *International Conference on Learning Representations (ICLR)*.

Aliee, H., Richter, T., Solonin, M., Ibarra, I., Theis, F., and Kilbertus, N. (2022). Sparsity in continuous-depth neural networks. *Neural Information Processing Systems (NeurIPS)*.

Aliee, H., Theis, F. J., and Kilbertus, N. (2021). Beyond predictions in neural ODEs: Identification and interventions. *arXiv preprint 2106.12430*.

Altschuler, J., Niles-Weed, J., and Rigollet, P. (2017). Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration. *Neural Information Processing Systems (NIPS)*.

Anderson, B. D. (1982). Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326.

Atanackovic, L., Tong, A., Hartford, J., Lee, L. J., Wang, B., and Bengio, Y. (2023). DynGFN: Bayesian dynamic causal discovery using generative flow networks. *Neural Information Processing Systems (NeurIPS)*.

Bellot, A. and Branson, K. (2022). Neural Graphical Modelling in Continuous Time: Consistency Guarantees and Algorithms. *International Conference on Learning Representations (ICLR)*.

Bunne, C., Hsieh, Y.-P., Cuturi, M., and Krause, A. (2022a). The Schrödinger bridge between gaussian measures has a closed form. *arXiv preprint 2202.05722*.

Bunne, C., Meng-Papaxanthos, L., Krause, A., and Cuturi, M. (2022b). Proximal optimal transport modeling of population dynamics. *Artificial Intelligence and Statistics (AISTATS)*.

Burkhardt, D., Bloom, J., Cannoodt, R., Luecken, M. D., Krishnaswamy, S., Lance, C., Pisco, A. O., and Theis, F. J. (2022). Multimodal single-cell integration across time, individuals, and batches. In *NeurIPS Competitions*.

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. (2018). Neural ordinary differential equations. *Neural Information Processing Systems (NeurIPS)*.

Chen, T., Liu, G.-H., and Theodorou, E. A. (2022). Likelihood training of Schrödinger bridge using forward-backward SDEs theory. *International Conference on Learning Representations (ICLR)*.

Cuturi, M. (2013). Sinkhorn distances: Lightspeed computation of optimal transport. *Neural Information Processing Systems (NIPS)*.

Damodaran, B. B., Kellenberger, B., Flamary, R., Tuia, D., and Courty, N. (2018). DeepJDOT: Deep joint distribution optimal transport for unsupervised domain adaptation. *European Conference on Computer Vision (ECCV)*.

De Bortoli, V., Thornton, J., Heng, J., and Doucet, A. (2021). Diffusion Schrödinger bridge with applications to score-based generative modeling. *Neural Information Processing Systems (NeurIPS)*.

Dhariwal, P. and Nichol, A. (2021). Diffusion models beat GANs on image synthesis. *Neural Information Processing Systems (NeurIPS)*.

Fatras, K., Sejourne, T., Flamary, R., and Courty, N. (2021a). Unbalanced minibatch optimal transport; applications to domain adaptation. *International Conference on Machine Learning (ICML)*.

Fatras, K., Zine, Y., Flamary, R., Gribonval, R., and Courty, N. (2020). Learning with minibatch Wasserstein: Asymptotic and gradient properties. *Artificial Intelligence and Statistics (AISTATS)*.

Fatras, K., Zine, Y., Majewski, S., Flamary, R., Gribonval, R., and Courty, N. (2021b). Minibatch optimal transport distances; analysis and applications. *arXiv preprint 2101.01792*.

Fernandes, D., Vargas, F., Ek, C. H., and Campbell, N. D. F. (2022). Shooting Schrödinger's cat. *Advances in Approximate Bayesian Inference*.

Finlay, C., Jacobsen, J.-H., Nurbekyan, L., and Oberman, A. M. (2020). How to train your neural ode: The world of jacobian and kinetic regularization. *International Conference on Machine Learning (ICML)*.

Flamary, R., Courty, N., Gramfort, A., Alaya, M. Z., Boisbunon, A., Chambon, S., Chapel, L., Corenflos, A., Fatras, K., Fournier, N., Gautheron, L., Gayraud, N. T. H., Janati, H., Rakotomamonjy, A., Redko, I., Rolet, A., Schutz, A., Seguy, V., Sutherland, D. J., Tavenard, R., Tong, A., and Vayer, T. (2021). POT: Python Optimal Transport. *Journal of Machine Learning Research (JMLR)*, 22.

Föllmer, H. (1988). Random fields and diffusion processes. In Hennequin, P.-L., editor, *École d'Été de Probabilités de Saint-Flour XV–XVII, 1985–87*, pages 101–203, Berlin, Heidelberg. Springer Berlin Heidelberg.

Genevay, A., Chizat, L., Bach, F., Cuturi, M., and Peyré, G. (2019). Sample complexity of Sinkhorn di-

vergences. *Artificial Intelligence and Statistics (AISTATS)*.

Genevay, A., Cuturi, M., Peyré, G., and Bach, F. (2016). Stochastic optimization for large-scale optimal transport. *Neural Information Processing Systems (NIPS)*.

Genevay, A., Peyre, G., and Cuturi, M. (2018). Learning generative models with sinkhorn divergences. *Artificial Intelligence and Statistics (AISTATS)*.

Granger, C. W. J. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438.

Grathwohl, W., Chen, R. T. Q., Bettencourt, J., Sutskever, I., and Duvenaud, D. (2019). Fjord: Freeform continuous dynamics for scalable reversible generative models. *International Conference on Learning Representations (ICLR)*.

Hashimoto, T. B., Gifford, D. K., and Jaakkola, T. S. (2016). Learning population-level diffusions with generative recurrent networks. *International Conference on Machine Learning (ICML)*.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Neural Information Processing Systems (NeurIPS)*.

Holdijk, L., Du, Y., Hooft, F., Jaini, P., Ensing, B., and Welling, M. (2022). Path integral stochastic optimal control for sampling transition paths. *arXiv preprint 2207.02149*.

Huguet, G., Magruder, D. S., Tong, A., Fasina, O., Kuchroo, M., Wolf, G., and Krishnaswamy, S. (2022a). Manifold interpolating optimal-transport flows for trajectory inference. *Neural Information Processing Systems (NeurIPS)*.

Huguet, G., Tong, A., Zapatero, M. R., Wolf, G., and Krishnaswamy, S. (2022b). Geodesic Sinkhorn: Optimal transport for high-dimensional datasets. *arXiv preprint 2211.00805*.

Kester, L. and van Oudenaarden, A. (2018). Single-cell transcriptomics meets lineage tracing. *Cell Stem Cell*, 23(2):166–179.

Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S. (2017). Self-normalizing neural networks. *Neural Information Processing Systems (NIPS)*.

Klein, D., Palla, G., Lange, M., Klein, M., Piran, Z., Gander, M., Meng-Papaxanthos, L., Sterr, M., Bastidas-Ponce, A., Tarquis-Medina, M., Lickert, H., Bakhti, M., Nitzan, M., Cuturi, M., and Theis, F. J. (2023). Mapping cells through time and space with moscot. *bioRxiv preprint 2023.05.11.540374*.

Koshizuka, T. and Sato, I. (2023). Neural Lagrangian Schrödinger bridge. *Interntioncal Conference on Learning Representations (ICLR)*.

Krishnaswamy, S., Spitzer, M. H., Mingueneau, M., Bendall, S. C., Litvin, O., Stone, E., Pe'er, D., and Nolan, G. P. (2014). Conditional density-based analysis of T cell signaling in single-cell data. *Science*, 346(6213):1250689.

Lähnemann, D., Köster, J., Szczurek, E., McCarthy, D. J., Hicks, S. C., Robinson, M. D., Vallejos, C. A., Campbell, K. R., Beerenwinkel, N., Mahfouz, A., Pinello, L., Skums, P., Stamatakis, A., Attolini, C. S.-O., Aparicio, S., Baaijens, J., Balvert, M., Barbanson, B. D., Cappuccio, A., Corleone, G., Dutilh, B. E., Florescu, M., Guryev, V., Holmer, R., Jahn, K., Lobo, T. J., Keizer, E. M., Khatri, I., Kielbasa, S. M., Korbel, J. O., Kozlov, A. M., Kuo, T.-H., Lelieveldt, B. P., Mandoiu, I. I., Marioni, J. C., Marschall, T., Mölder, F., Niknejad, A., Raczkowski, L., Reinders, M., Ridder, J. D., Saliba, A.-E., Somarakis, A., Stegle, O., Theis, F. J., Yang, H., Zelikovsky, A., McHardy, A. C., Raphael, B. J., Shah, S. P., and Schönhuth, A. (2020). Eleven grand challenges in single-cell data science. *Genome Biology*, 21(1):31.

Lavenant, H., Zhang, S., Kim, Y.-H., and Schiebinger, G. (2021). Towards a mathematical theory of trajectory inference.

Léonard, C. (2014a). Some properties of path measures. In Donati-Martin, C., Lejay, A., and Rouault, A., editors, *Séminaire de Probabilités XLVI*, pages 207–230. Springer.

Léonard, C. (2014b). A survey of the Schrödinger problem and some of its connections with optimal transport. *Discrete and Continuous Dynamical Systems*, 34(4):1533–1574.

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. (2023). Flow matching for generative modeling. *International Conference on Learning Representations (ICLR)*.

Liu, G.-H., Chen, T., So, O., and Theodorou, E. A. (2022). Deep generalized Schrödinger bridge. *Neural Information Processing Systems (NeurIPS)*.

Liu, G.-H., Vahdat, A., Huang, D.-A., Theodorou, E. A., Nie, W., and Anandkumar, A. (2023a). $I^2$sb: Image-to-image schrödinger bridge. *International Conference on Machine Learning (ICML)*.

Liu, Q. (2022). Rectified flow: A marginal preserving approach to optimal transport. *arXiv preprint 2209.14577*.

Liu, X., Gong, C., and Liu, Q. (2023b). Flow straight and fast: Learning to generate and transfer data with rectified flow. *International Conference on Learning Representations (ICLR)*.

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. *International Conference on Learning Representations (ICLR)*.

Mallasto, A., Gerolin, A., and Ha Quang, M. (2022). Entropy-regularized 2-Wasserstein distance between Gaussian measures. *Information Geometry*, 5.

Mena, G. and Niles-Weed, J. (2019). Statistical bounds for entropic optimal transport: sample complexity and the central limit theorem. *Neural Information Processing Systems (NeurIPS)*.

Moon, K. R., Stanley, J. S., Burkhardt, D., van Dijk, D., Wolf, G., and Krishnaswamy, S. (2018). Manifold learning-based methods for analyzing single-cell rna-sequencing data. *Current Opinion in Systems Biology*, 7:36–46.

Moon, K. R., van Dijk, D., Wang, Z., Gigante, S., Burkhardt, D. B., Chen, W. S., Yim, K., van den Elzen, A., Hirn, M. J., Coifman, R. R., Ivanova, N. B., Wolf, G., and Krishnaswamy, S. (2019). Visualizing structure and transitions in high-dimensional biological data. *Nature Biotechnology*, 37(12):1482–1492.

Neklyudov, K., Brekelmans, R., Severo, D., and Makhzani, A. (2023). Action matching: Learning stochastic dynamics from samples. *International Conference on Machine Learning (ICML)*.

Neklyudov, K., Severo, D., and Makhzani, A. (2022). Action matching: A variational method for learning stochastic dynamics from samples. *arXiv preprint 2210.06662*.

Nichol, A. and Dhariwal, P. (2021). Improved denoising diffusion probabilistic models. *International Conference on Machine Learning (ICML)*.

Nolan, T. M., Vukašinović, N., Hsu, C.-W., Zhang, J., Vanhoutte, I., Shahan, R., Taylor, I. W., Greenstreet, L., Heitz, M., Afanassiev, A., Wang, P., Szekely, P., Brosnan, A., Yin, Y., Schiebinger, G., Ohler, U., Russinova, E., and Benfey, P. N. (2023). Brassinosteroid gene regulatory networks at cellular resolution in the arabidopsis root. *Science*, 379(6639):eadf4721.

Peluchetti, S. (2023). Diffusion bridge mixture transports, Schrödinger bridge problems and generative modeling. *Journal of Machine Learning Research (JMLR)*, 24:1–51.

Pooladian, A.-A., Ben-Hamu, H., Domingo-Enrich, C., Amos, B., Lipman, Y., and Chen, R. T. (2023). Multisample flow matching: Straightening flows with minibatch couplings. *International Conference on Learning Representations (ICLR)*.

Pratapa, A., Jalihal, A. P., Law, J. N., Bharadwaj, A., and Murali, T. (2020). Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. *Nature methods*, 17.

Qin, X., Rodriguez, F. C., Sufi, J., Vlckova, P., Claus, J., and Tape, C. J. (2023). A single-cell perturbation landscape of colonic stem cell polarisation. Preprint, Cancer Biology.

Salimans, T., Zhang, H., Radford, A., and Metaxas, D. (2018). Improving GANs using optimal transport. *International Conference on Learning Representations (ICLR)*.

Schiebinger, G. (2021). Reconstructing developmental landscapes and trajectories from single-cell data. *Current Opinion in Systems Biology*, 27:100351.

Schiebinger, G., Shu, J., Tabaka, M., Cleary, B., Subramanian, V., Solomon, A., Gould, J., Liu, S., Lin, S., Berube, P., Lee, L., Chen, J., Brumbaugh, J., Rigollet, P., Hochedlinger, K., Jaenisch, R., Regev, A., and Lander, E. S. (2019). Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, 176(4):928–943.e22.

Schrödinger, E. (1932). Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique. *Annales de l'Institut Henri Poincaré*, 2(4):269–310.

Seguy, V., Damodaran, B. B., Flamary, R., Courty, N., Rolet, A., and Blondel, M. (2018). Large scale optimal transport and mapping estimation. *International Conference on Learning Representations (ICLR)*.

Shi, Y., De Bortoli, V., Campbell, A., and Doucet, A. (2023). Diffusion Schrödinger bridge matching. *arXiv preprint 2303.16852*.

Shi, Y., De Bortoli, V., Deligiannidis, G., and Doucet, A. (2022). Conditional simulation using diffusion Schrödinger bridges. *Uncertainty in Artificial Intelligence (UAI)*.

Sinkhorn, R. (1964). A relationship between arbitrary positive matrices and doubly stochastic matrices.

Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. *International Conference on Machine Learning (ICML)*.

Somnath, V. R., Pariset, M., Hsieh, Y.-P., Martinez, M. R., Krause, A., and Bunne, C. (2023). Aligned diffusion Schrödinger bridges. *Uncertainty in Artificial Intelligence (UAI)*.

Song, J., Meng, C., and Ermon, S. (2021a). Denoising diffusion implicit models. *International Conference on Learning Representations (ICLR)*.

Song, Y. and Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. *Neural Information Processing Systems (NeurIPS)*.

Song, Y. and Ermon, S. (2020). Improved techniques for training score-based generative models. *Neural Information Processing Systems (NeurIPS)*.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021b). Score-based generative modeling through stochastic differential equations. *International Conference on Learning Representations (ICLR)*.

Stromme, A. (2023). Sampling from a Schrödinger bridge. *Artificial Intelligence and Statistics (AISTATS)*.

Tang, Y. (2017). Potential landscape of high dimensional nonlinear stochastic dynamics with large noise. *Scientific Reports*.

Tank, A., Covert, I., Foti, N., Shojaie, A., and Fox, E. (2021). Neural Granger Causality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Tong, A., Fatras, K., Malkin, N., Huguet, G., Zhang, Y., Rector-Brooks, J., Wolf, G., and Bengio, Y. (2024). Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*.

Tong, A., Huang, J., Wolf, G., van Dijk, D., and Krishnaswamy, S. (2020). TrajectoryNet: A dynamic optimal transport network for modeling cellular dynamics. *International Conference on Machine Learning (ICML)*.

Tong, A., Kuchroo, M., Gupta, S., Venkat, A., Perez San Juan, B., Rangel, L., Zhu, B., Lock, J. G., Chaffer, C., and Krishnaswamy, S. (2023). Learning transcriptional and regulatory dynamics driving cancer cell plasticity using neural ode-based optimal transport. *bioRxiv preprint 2023.03.28.534644*.

Vargas, F., Thodoroff, P., Lawrence, N. D., and Lamacraft, A. (2021). Solving Schrödinger bridges via maximum likelihood. *Entropy*, 23(9).

Waddington, C. H. (1942). The epigenotype. *Endeavour*, 1:18–20.

Wagner, D. E. and Klein, A. M. (2020). Lineage tracing meets single-cell omics: Opportunities and challenges. *Nature Reviews Genetics*, 21(7):410–427.

Wang, G., Jiao, Y., Xu, Q., Wang, Y., and Yang, C. (2021). Deep generative learning via Schrödinger bridge. *International Conference on Machine Learning (ICML)*.

Watson, J. L., Juergens, D., Bennett, N. R., Trippe, B. L., Yim, J., Eisenach, H. E., Ahern, W., Borst, A. J., Ragotte, R. J., Milles, L. F., Wicky, B. I. M., Hanikel, N., Pellock, S. J., Courbet, A., Sheffler, W., Wang, J., Venkatesh, P., Sappington, I., Torres, S. V., Lauko, A., De Bortoli, V., Mathieu, E., Barzilay, R., Jaakkola, T. S., DiMaio, F., Baek, M., and Baker, D. (2023). Broadly applicable and accurate protein design by integrating structure prediction networks and diffusion generative models. *Nature*.

Winkler, L., Ojeda, C., and Opper, M. (2023). A score-based approach for training Schrödinger bridges for data modelling. *Entropy*, 25(2):316.

Wolf, F. A., Angerer, P., and Theis, F. J. (2018). SCANPY: Large-scale single-cell gene expression data analysis. *Genome Biology*, 19(1):15.

Yang, K. D., Damodaran, K., Venkatachalapathy, S., Soylemezoglu, A. C., Shivashankar, G. V., and Uhler, C. (2020). Predicting cell lineages using autoencoders and optimal transport. *PLOS Computational Biology*, 16:1–20.

## Checklist

1. For all models and algorithms presented, check if you include:

   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]

   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]

   (c) Source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:

   (a) Statements of the full set of assumptions of all theoretical results. [Yes]

   (b) Complete proofs of all theoretical results. [Yes]

   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:

   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]

   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]

   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]

   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

   (a) Citations of the creator if your work uses existing assets. [Yes]

   (b) The license information of the assets, if applicable. [Not Applicable]

   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]

   (d) Information about consent from data providers/curators. [Not Applicable]

   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

   (a) The full text of instructions given to participants and screenshots. [Not Applicable]

   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

# Simulation-Free Schrödinger Bridges via Score and Flow Matching: Supplementary Materials

Our code can be found at `https://github.com/atong01/conditional-flow-matching`. The supplementary material is structured as follows:

- Appendix A gives more background on optimal transport.

- Appendix B presents the proofs of our different results.

- Appendix C gives more background on the related work.

- Appendix D describes additional results and experiments.

- Appendix E discusses Schrödinger bridges with varying diffusion rate.

- Appendix F presents the experimental details of our experiments in the main paper.

## A    BACKGROUND ON OPTIMAL TRANSPORT

In this section, we review optimal transport and its application in machine learning.

---

**Algorithm 2** Simulation-Free Score and Flow Matching Inference (with Euler-Maruyama integration)

---

**Input:** Source distribution $q_0$, flow and score networks $v_\theta$ and $s_\theta$, diffusion schedule $g(\cdot)$, integration step size $\Delta t$.

$x_0 \sim q_0(x)$

**for** $t$ in $[0, 1/\Delta t)$ **do**

$\quad u_t \leftarrow v_\theta(t, x_t) + \frac{g(t)^2}{2} s_\theta(t, x_t)$

$\quad x_{t+\Delta t} \sim \mathcal{N}(x + u_t \Delta t, g(t)^2 \Delta t)$
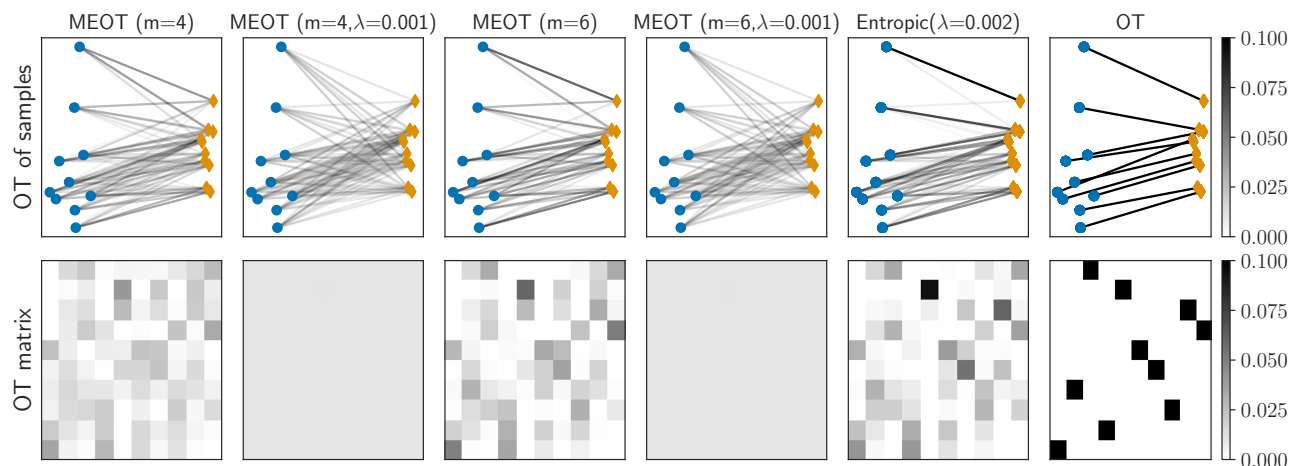
**return** Samples $x_1$

---



Figure 4: Optimal transport couplings for different OT costs and batch sizes on a 2D example. The top row represent the OT matching between samples while the bottom row represent the minibatch OT plan. We can see that coupling entropic OT with minibatches lead to a uniform plan contrary to using only entropic regularization or minibatch approximation.

### A.1 Minibatch OT

In the context of generative modeling, the source distribution is a Gaussian distribution and the target distribution is the real data distribution. This scenario corresponds to a semi-discrete optimal transport problem. Therefore, the Sinkhorn algorithm cannot be used to compute the entropic OT plan between distributions. It is nonetheless possible to compute it with stochastic algorithms (Genevay et al., 2016). Unfortunately, these stochastic algorithms are slow to converge and it is prohibitive for large scale datasets. Therefore, we chose to rely on a minibatch optimal transport approximation (Fatras et al., 2020, 2021b). Minibatch OT computes the OT between minibatches of samples and thus corresponds to the discrete-discrete optimal transport setting. It is known to have quadratic computational and memory costs in the number of samples (see §A.2 for a longer discussion).

The minibatch optimal transport approximation has been successfully before used in generative modeling (Genevay et al., 2018; Salimans et al., 2018). As it is different from the original optimal transport problem, we want to make sure that the basic properties from Schrödinger Bridges are conserved. We use the minibatch optimal transport plans definitions from Fatras et al. (2020), namely

**Definition A.1** (minibatch transport plan (Fatras et al., 2020)). Consider $\alpha_n$ and $\beta_n$ two empirical probability distributions. For each $A = \{a_1, \ldots, a_m\} \in \mathcal{P}_m(\alpha_n)$ and $B = \{b_1, \ldots, b_m\} \in \mathcal{P}_m(\beta_n)$ we denote by $\Pi_{A,B}$ the optimal plan between the random variables, considered as a $n \times n$ matrix where all entries are zero except those indexed in $A \times B$. We define the *averaged minibatch transport matrix*:

$$\Pi^m(\alpha_n, \beta_n) = \binom{n}{m}^{-2} \sum_{A \in \mathcal{P}_m(\alpha_n)} \sum_{B \in \mathcal{P}_m(\beta_n)} \Pi_{A,B}. \tag{13}$$

Following the subsampling idea, we define the subsampled minibatch transportation matrix:

$$\Pi^k(\alpha_n, \beta_n) := k^{-1} \sum_{(A,B) \in D_k} \Pi_{A,B} \tag{14}$$

where $D_k$ is a set of cardinality $k$ whose elements are drawn at random from the uniform distribution on $\Gamma := \mathcal{P}_m(\{x_0^1, \cdots, x_0^n\}) \times \mathcal{P}_m(\{x_1^1, \cdots, x_1^n\})$.

Note that $\Pi_k$ converges exponentially fast to $\Pi_m$ as $k$ grows (Fatras et al., 2020, Theorem 2). In practice, it is enough to set $k$ equal to 1 to get good performance in deep learning applications (Genevay et al., 2018; Damodaran et al., 2018; Fatras et al., 2021a). Therefore, the subsampling estimator does not have the correct marginals in general, contrary to $\Pi_m$ which has always the right marginals (Fatras et al., 2020, Proposition 1). Minibatch optimal transport has been shown to implicitly regularize the transport plan (Fatras et al., 2020, 2021b). Indeed, as we draw uniformly at random sample to build the minibatches, we create non optimal connections. This is similar to the entropic optimal transport which densifies the transport plan. Therefore, coupling the minibatch approximation with the entropic OT cost might lead to an extremely dense plan that is close to the uniform transport plan. Unfortunately such a transport plan looses all geometric insights from data and is also far from the original entropic OT cost. We illustrate this phenomenon in Fig. 4 on a toy example between two 2D data distributions. Notably, the minibatch OT plan is closer to the entropic OT plan than the minibatch entropic OT plan. That is why in our experiments, we observed that the minibatch approximation with exact optimal transport outperforms the minibatch approximation with the entropic OT cost. We leave the question of closeness between minibatch OT and entropic OT as future work. While it would have been possible to decrease these non-optimal connections with OT variants (Fatras et al., 2021a), it would have added extra hyperparameters that would have led to more compute.

### A.2 Computational complexity

The batch size we use is 512 for all experiments except for the Gaussian-to-Gaussian experiments, which use batch size 500 to match the DSBM baseline. The complexity of static discrete OT is cubic in the batch size $m$ and linear in the dimension $d$: $O(m^3 + m^2 d)$. This can be reduced to quadratic complexity in the entropy-regularized discrete OT case. However, for batches <10k, the exact minibatch OT is often actually faster than the Sinkhorn solver for the entropic OT problem thanks to the optimized simplex solver code in POT (Flamary et al., 2021). Note that the memory complexity is quadratic in the number of samples due to the storage of the ground cost matrix.

The asymptotic complexity does not tell the full story, as these discrete algorithms run much faster than the (theoretically linear time) neural network training. In practice, for high dimensional settings we find a $< 1\%$ overhead in including the OT solution per batch (much of which is in the transfer from GPU to CPU and back). We note that this is significantly faster than contemporary works such as DSBM and IBDM, which must train a bridge matching model for every iterative proportional fitting (IPF) step.

## B  PROOFS OF THEOREMS

**Theorem 3.1.** *Under mild regularity conditions, the ODE $dx = u_t^\circ(x)\, dt$ generates the marginals $p_t$ of $\mathbb{P}$ from initial conditions $p_0$, and the score is given by (10). The SDE $dx = [u_t^\circ(x) + \frac{1}{2}g(t)^2 \nabla \log p_t(x)]\, dx + g(t)\, dt$ generates the Markovization of $\mathbb{P}$.*

*Proof of Theorem 3.1.* We assume $u_t(x|z)$, $\nabla \log p_t(x|z)$, and their first derivatives are continuous in $x$, uniformly in $z$. The first statement is Theorem 3.1 from Tong et al. (2024), but we reproduce the key derivation here for completeness:

$$
\begin{aligned}
\frac{d}{dt}p_t(x) &= \frac{d}{dt}\int p_t(x|z)q(z)dz \\
&= \int \frac{d}{dt}\left(p_t(x|z)q(z)\right)dz \\
&= -\int \nabla \cdot \left(u_t^\circ(x|z)p_t(x|z)q(z)\right)dz \\
&= -\nabla \cdot \left(\int u_t^\circ(x|z)p_t(x|z)q(z)dz\right) \\
&= -\nabla \cdot \left(u_t^\circ(x)p_t(x)\right),
\end{aligned}
$$

showing that $p_t$ and $u_t^\circ$ satisfy the continuity equation $\frac{d}{dt}p_t(x) + \nabla \cdot (u_t^\circ(x)p_t(x)) = 0$, which implies that $u_t^\circ$ generates $p_t$ from initial conditions $p_0$.

To show the score is given by the expression in (10), using that $p_t(x) = E_{q(z)}p_t(x|z)$, we have

$$
\begin{aligned}
\nabla \log p_t(x) &= \frac{\nabla \mathbb{E}_{q(z)}[p_t(x|z)]}{p_t(x)} \\
&= \frac{\mathbb{E}_{q(z)}[\nabla p_t(x|z)]}{p_t(x)} \\
&= \mathbb{E}_{q(z)}\left[\frac{p_t(x|z)\nabla \log p_t(x|z)}{p_t(x)}\right],
\end{aligned}
$$

as desired.

The Markovization of a mixture of SDEs $\mathbb{P}$ with a common diffusion rate $g(t)$ is an SDE $dx = u_t(x)\, dt + g(t)\, dw$, where

$$
u_t(x_t) = \lim_{\Delta t \to 0} \mathbb{E}_{x_{t+\Delta t} \sim \mathbb{P}(x_{t+\Delta t}|x_t)}\left[\frac{x_{t+\Delta t} - x_t}{\Delta t}\right].
$$

Using the definition of $\mathbb{P}$ as a mixture in (9), we decompose this expectation over the posterior $p(z \mid x_t) = \frac{p_t(x_t|z)q(z)}{p_t(x_t)}$:

$$
\begin{aligned}
\lim_{\Delta t \to 0} \mathbb{E}_{x_{t+\Delta t} \sim \mathbb{P}(x_{t+\Delta t}|x_t)}\left[\frac{x_{t+\Delta t} - x_t}{\Delta t}\right] &= \lim_{\Delta t \to 0} \mathbb{E}_{z \sim p(z|x_t)}\mathbb{E}_{x_{t+\Delta t} \sim \mathbb{P}(x_{t+\Delta t}|x_t,z)}\left[\frac{x_{t+\Delta t} - x_t}{\Delta t}\right] \\
&= \mathbb{E}_{z \sim p(z|x_t)} \lim_{\Delta t \to 0} \mathbb{E}_{x_{t+\Delta t} \sim \mathbb{P}(x_{t+\Delta t}|x_t,z)}\left[\frac{x_{t+\Delta t} - x_t}{\Delta t}\right] \\
&= \mathbb{E}_{z \sim p(z|x_t)}\left[u_t(x_t \mid z)\right].
\end{aligned}
$$

The score and probability flow defined in (10) can equivalently be expressed as

$$
u_t^\circ(x_t) = \mathbb{E}_{z \sim p(z|x_t)}u_t^\circ(x_t \mid z), \quad \nabla \log p_t(x_t) = \mathbb{E}_{z \sim p(z|x_t)}\nabla \log p_t(x_t \mid z),
$$

which, together with $u_t(x \mid z) = u_t^\circ(x \mid z) + \frac{1}{2}g(t)^2 \nabla \log p_t(x \mid z)$, implies that

$$u_t(x) = \mathbb{E}_{z \sim p(z|x)}[u_t(x \mid z)] = u_t^\circ(x) + \frac{1}{2}g(t)^2 \nabla \log p_t(x),$$

as desired. $\qquad\square$

**Theorem 3.2** (Equality of conditional gradients). *If $p_t(x) > 0$ for all $x \in \mathbb{R}^d$ and $t \in [0, 1]$, then $\nabla_\theta \mathcal{L}_{\mathrm{U[SF]^2M}}(\theta) = \nabla_\theta \mathcal{L}_{\mathrm{[SF]^2M}}(\theta)$, where $\mathcal{L}_{\mathrm{U[SF]^2M}}(\theta)$ is the unconditional score and flow matching loss (7).*

*Proof of Theorem 3.2.* We show this individually for the flow matching and score matching parts of the losses. (Note that the equality of gradients of the flow matching parts of U[SF]$^2$M and [SF]$^2$M is equivalent to Theorem 3.2 of Tong et al. (2024).)

We claim that for any conditional vector field $w_t^\circ(x|z)$ and $w_t^\circ(x) := \mathbb{E}_{q(z)} \frac{p_t(x|z)}{p_t(x)} w_t^\circ(x|z)$, and approximating vector field $w_\theta(t, x)$, under some regularity conditions on $w_t(x|z)$, we have

$$\nabla_\theta \mathbb{E}_{z \sim q(z), x \sim p_t(x|z)} \left[ \|w_\theta(t, x) - w_t^\circ(x|z)\|^2 \right] = \nabla_\theta \mathbb{E}_{x \sim p_t(x)} \left[ \|w_\theta(t, x) - w_t^\circ(x)\|^2 \right]. \tag{15}$$

Assuming this claim, the theorem would follow from applying the claim to $w_t^\circ(x|z) = u_t^\circ(x|z)$ and to $w_t^\circ(x|z) = \nabla p_t(x|z)$ for every value of $t$, noting that in these cases $w_t^\circ(x) = u_t^\circ(x)$ and $w_t^\circ(x) = \nabla p_t(x)$, respectively, by Theorem 3.1, and integrating over $t$.

We proceed to prove the claim. We drop the distributions in the expectations for conciseness, noting that because $p_t(x) = \mathbb{E}_{z \sim q(z)} p(x|z)$, no ambiguity is caused: the marginal distribution over $x$ that stands under the expectation is the same in $\mathbb{E}_{x \sim p_t(x)}$ and in $\mathbb{E}_{z \sim q(z), x \sim p_t(x|z)}$.

$$\nabla_\theta \left( \mathbb{E}_{z,x} \left[ \|w_\theta(t, x) - w_t^\circ(x|z)\|^2 \right] - \mathbb{E}_x \left[ \|w_\theta(t, x) - w_t^\circ(x)\|^2 \right] \right)$$
$$= \nabla_\theta \left( -2\mathbb{E}_{z,x}\langle w_\theta(t, x), w_t^\circ(x|z)\rangle + 2\mathbb{E}_x\langle w_\theta(t, x), w_t^\circ(x)\rangle \right)$$

where we rewrote the squared norms as inner products and used that $w_t^\circ(x|z), w_t^\circ(x)$ are independent of $\theta$ and that $w_\theta(t, x)$ is independent of $z$. To conclude, we compute

$$\mathbb{E}_x \langle w_\theta(t, x), w_t^\circ(x) \rangle = \iint \langle w_\theta(t, x), w_t^\circ(x) \rangle p_t(x) \, dx$$
$$= \int \left\langle w_\theta(t, x), \int \frac{p_t(x|z)}{p_t(x)} w_t^\circ(x|z) q(z) \, dz \right\rangle p_t(x) \, dx$$
$$= \iint \langle w_\theta(t, x), w_t^\circ(x|z) \rangle p_t(x|z) q(z) \, dx \, dz$$
$$= \mathbb{E}_{z,x} \langle w_\theta(t, x), w_t^\circ(x|z) \rangle,$$

showing that the difference of gradients vanishes. Note that the above derivation required exchanging the order of integration, which requires some assumptions of regularity at infinity. (Absolute convergence of the integrals is sufficient, and in particular, guaranteed by polynomial growth of in $x$ of $w_\theta$ $w_t^\circ(x|z)$ and exponential decay of $p_t(x|z)$ uniformly in $z$.) $\qquad\square$

**Theorem 3.3** ([SF]$^2$M recovers marginals from bridges). *If $q(\cdot, \cdot) \in U(q_0, q_1)$ and $v_\theta^*, s_\theta^*$ globally minimize $\mathcal{L}_{\mathrm{[SF]^2M}}(\theta)$, the SDE with drift $[v_\theta^* + \frac{1}{2}g(t)^2 s_\theta^*]$ and diffusion $g$, and initial conditions $p_0 = q_0$, is the Markovization of the mixture of Brownian bridges from $x_0$ to $x_1$ over $q(x_0, x_1)$. In particular, if the SDE generates marginals $p_t$, then $p_1 = q_1$.*

*Proof of Theorem 3.3.* The probability flow drift $u_t^\circ$ and score $\nabla \log p_t$ of the mixture in question are given by Theorem 3.1 shows. By Theorem 3.2, assuming sufficient regularity, optimization of $\mathcal{L}_{\mathrm{[SF]^2M}}(\theta)$ is equivalent to optimization of $\mathcal{L}_{\mathrm{U[SF]^2M}}(\theta)$, which is globally minimized when $v_\theta(t, x) = u_t^\circ(x)$ and $s_\theta(t, x) = \nabla \log p_t(x)$ for (Lebesgue,$p_t$)-almost all $t \in [0, 1]$ and $x \in \mathbb{R}^d$. Moreover, 'almost all' implies 'all' if $u_t^\circ, \nabla \log p_t, v_\theta, s_\theta$ are continuous in $t$ and $x$ and $p_t$ has full support.

By hypothesis, the vector field $u_t^\circ(x)$ satisfies the continuity equation jointly with $p_t(x) = \mathbb{E}_{q(z)} p_t(x|z)$, which was assumed to satisfy $p_0(x) = q_0(x)$ and $p_1(x) = q_1(x)$. By the algebraic derivation in §2, $u_t^\circ(x) + \frac{1}{2}g(t)^2 \nabla \log p_t(x)$

and $p_t(x)$ jointly satisfy the Fokker-Planck equation. Assuming all derivatives appearing in the Fokker-Planck equation exist and are continuous everywhere, the given SDE generates marginal probabilities $p_t$ from initial conditions $p_0$. $\qquad\square$

**Proposition 3.4** ([SF]$^2$M with entropic OT recovers the SB process). *Let $\mathbb{P}^*$ be the Schrödinger bridge between $q_0$ and $q_1$ with respect to $\mathbb{Q} = \sigma\mathbb{W}$. If $v_\theta^\star$, $s_\theta^\star$ globally minimize $\mathcal{L}_{[SF]^2M}$, with coupling $\pi_{2\sigma^2}^\star(q_0, q_1)$, then $\mathbb{P}^*$ is defined by the SDE with drift $[v_\theta^* + \frac{1}{2}g(t)^2 s_\theta^*]$, diffusion $g$, and initial conditions $p_0 = q_0$.*

*Proof of Proposition 3.4.* This is an immediate consequence of Theorem 3.3, which shows that the SDE learned by [SF]$^2$M is the Markovization of a mixture of Brownian bridges, and of Proposition 2.1, which characterizes the SB as a mixture of Brownian bridges with mixture weights given by the entropic OT plan. $\qquad\square$

# C   FURTHER DISCUSSION ON RELATED WORK

Next we discuss similarities and differences in related algorithms, including deterministic flow models and stochastic score-based generative models.

## C.1   Flow models

Recently, there has been significant advances in simulation-free training of flow models, which were originally trained using maximum likelihood training in what is termed as continuous normalizing flows (Chen et al., 2018; Grathwohl et al., 2019). In this section, we discuss the more recent flow matching techniques, which allow for simulation-free training of flow models. In this paper we show that any deterministic flow model can be converted into a stochastic model with the addition of a conditional score matching loss. This generalizes flow matching ideas to SDEs and provides a simple link between score-based generative modelling and flow-based generative modeling.

Conditional flow matching, terminology introduced by Lipman et al. (2023), and extended to dynamic optimal transport in Tong et al. (2024); Pooladian et al. (2023), trains with the *conditional flow matching* loss,

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}(0,1),z\sim q(z),x\sim p_t(x|z)}\|v_\theta(t,x) - u_t^\circ(x|z)\|^2. \tag{16}$$

for some predefined conditional paths $z, q(z), u_t(x|z)$, and $p_t(x|z)$. Depending on the choice of conditioning and probability paths we can recover most known flow matching techniques, such as the originally described flow matching (Lipman et al., 2023), action matching (Neklyudov et al., 2022), stochastic interpolants (Albergo and Vanden-Eijnden, 2023; Albergo et al., 2023), the 1-rectified flow (Liu, 2022), optimal transport conditional flow matching (OT-CFM) (Tong et al., 2024) also known as multisample flow matching (Pooladian et al., 2023), and its deterministic Schrödinger bridge counterpart, Schrödinger bridge conditional flow matching (SB-CFM) (Tong et al., 2024).

**SDEs vs. ODEs.** Flow models are a powerful way to learn deterministic dynamics which are often faster to sample from Song et al. (2021a), particularly with ideas from dynamic optimal transport (Tong et al., 2024; Pooladian et al., 2023). However, recent work (Liu et al., 2023a; Shi et al., 2023) has noted advantages of stochastic dynamics which we also observe (see Table 2 and Table 3), particularly in high dimensional settings and in terms of generative performance. In this work, we also seek to model an inherently stochastic system, where there is randomness introduced based on a variety of external factors. In particular, the fact that a single-cell develops into a whole organism, and similarly, the fact that a single population of stem cells develops into a multitude of different cell types, requires the modelling with stochastic dynamics that can model complex conditional distributions. While it is possible to approximate the conditional distribution with lineage-tracing techniques (Kester and van Oudenaarden, 2018; Wagner and Klein, 2020; Klein et al., 2023), these are biologically complex, and we leave their analysis to future work.

## C.2   Learning Schrödinger bridges

Schrödinger bridge models have been used as the source-conditioned variation of score-based generative models, which can be conditioned on a variety of source distributions including dirac (Wang et al., 2021), and from data or noised data (Somnath et al., 2023; Liu et al., 2023a). Most algorithms are based on mean-matching, which

is simulation-based and requires simulating, storing, and matching whole trajectories (De Bortoli et al., 2021), with similar algorithms introduced by Vargas et al. (2021); Chen et al. (2022). In particular, these algorithms parameterize a forward drift $v_\theta^f$ and backwards drift $v_\theta^b$, simulating trajectories in one direction and training the reverse direction to match them. Recent work extends this to Markovian bridges (termed bridge matching) (Shi et al., 2023; Peluchetti, 2023), which greatly improves performance.

### C.3 Practical implications of Schrödinger bridge modeling choices

**Approximation error.** All known algorithms for learning Schrödinger bridges for general distributions can only approximate the Schrödinger bridge. Approximation error accumulates from a number of sources, which can roughly be categorized into the following areas:

1. Error in the static joint optimization $\hat{\pi}(x_0, x_1)$ approximating the true joint $\pi^\star(x_0, x_1)$. Since there is no closed form for this optimization, all approximations are iterative and discrete. Error in this distribution accumulates in two places, and can be divided into error on the marginals $|\hat{\pi}(x_0, \cdot) - q(x_0)|$, $|\hat{\pi}(\cdot, x_1) - q(x_1)|$, and error on the joint $|\hat{\pi}(x_0, x_1) - \pi^\star(x_0, x_1)|$:

    (a) Marginal and Joint error from using a finite number of iterations (IPF underfitting). IPF both in continuous space (as used by mean-matching and bridge matching) and IPF in discrete space (as used by [SF]$^2$M) are run for a finite number of iterations. Continuous space IPF is much more expensive (involving fitting and simulating neural networks) and is thus is used with tens of iterations, often $L = 20$ (De Bortoli et al., 2021; Shi et al., 2022). In contrast, discrete space IPF uses many more iterations. We use the default python optimal transport (POT) parameters with $L_{max} = 1000$ with early stopping if the marginal error is $< 10^{-9}$ (Flamary et al., 2021). The discrete OT computation adds a negligible ($< 1\%$) computational overhead in [SF]$^2$M.

    (b) Joint error from minibatch approximation (discrete IPF only). By using discrete OT solvers, we accumulate discretization error between by using finite batch sizes. This affects the recovery of the true Schrödinger bridge, but does not affect the marginal error, which is the important error for generative modeling performance at the endpoints. As batch size $\to \infty$, this error goes to zero; for a finite dataset of small enough size, [SF]$^2$M can use full-batch discrete entropic OT, where the batch size used for OT can be larger than that used for neural network training.

    In practice, [SF]$^2$M, by using discrete OT solvers, trades off increased error in the joint via minibatch approximation error for reduced error from IPF underfitting. It is difficult to bound these errors theoretically, however, our experiments show that this tradeoff is useful in practice. We leave further theoretical investigation to future work.

2. Finite sample error. Datasets are often finite (*i.e.*, discrete). Approximating the continuous densities from discrete samples is challenging and accumulates error.

3. Neural network fitting approximation (applicable to all neural SDE approximations to Schrödinger bridges).

4. Discretization error. Simulation-free objectives like [SF]$^2$M treat time as a continuous variable, but, at inference time, error accumulates from integration in discrete time (*e.g.*, using an Euler-Maruyama scheme). We note that simulation-based objectives, such as that of DSB (De Bortoli et al., 2021), also *train* in a fixed time discretization, inducing a further approximation error.

**Iterative fitting.** Both mean-matching and bridge matching Schrödinger bridge methods require an outer IPF step, which samples data to train for each inner neural network optimization step (Table 1). At each step, new data is resampled to determine the boundary distributions of the next iteration (Alg. 3). This multiplies the training time by $L$ (the number of outer loops), and cannot guarantee that that the neural networks match the right endpoint marginals until the outer IPF step has converged.

In contrast, [SF]$^2$M is always optimizing for the correct marginals, but at the cost of a biased Schrödinger bridge due to bias in the minibatch OT. We hypothesize that this is one of the reasons that [SF]$^2$M has better generative modeling performance than mean-matching or bridge-matching methods.

In summary, we recommend using DSBM / IDBM when approximating the true Schrödinger bridge is important and the computational budget is large. We recommend using [SF]$^2$M when the computational budget is small or when the generative performance is more important than matching the true Schrödinger bridge process.

Table 7: Gaussian-to-Gaussian Schrödinger bridges with 10,000 datapoints. Here we text Sinkhorn-Exact which uses the exact OT (default), against the Sinkhorn algorithm for the static OT within batches, and using outer loops where we simulate 10,000 trajectories for further training 20 times following Shi et al. (2023).

|  | Dim | $[SF]^2M$-Sinkhorn | $[SF]^2M$-Exact | $[SF]^2M$-Exact-Looped |
|---|---|---|---|---|
| $KL(p_1, q_1)$ | 2 | **0.002 ± 0.000** | 0.003 ± 0.000 | 0.032 ± 0.009 |
|  | 5 | **0.004 ± 0.001** | 0.007 ± 0.000 | 0.088 ± 0.011 |
|  | 20 | **0.029 ± 0.001** | 0.029 ± 0.002 | 0.293 ± 0.021 |
|  | 50 | **0.122 ± 0.003** | 0.124 ± 0.003 | 0.610 ± 0.033 |
|  | 100 | 0.493 ± 0.010 | **0.486 ± 0.005** | 1.578 ± 0.026 |
| Mean $KL(p_t, q_t)$ | 2 | **0.001 ± 0.000** | 0.004 ± 0.000 | 0.006 ± 0.002 |
|  | 5 | **0.004 ± 0.000** | 0.006 ± 0.000 | 0.019 ± 0.003 |
|  | 20 | 0.042 ± 0.001 | **0.028 ± 0.001** | 0.080 ± 0.006 |
|  | 50 | 0.276 ± 0.001 | 0.258 ± 0.001 | **0.243 ± 0.012** |
|  | 100 | 1.000 ± 0.007 | 0.977 ± 0.003 | **0.792 ± 0.008** |

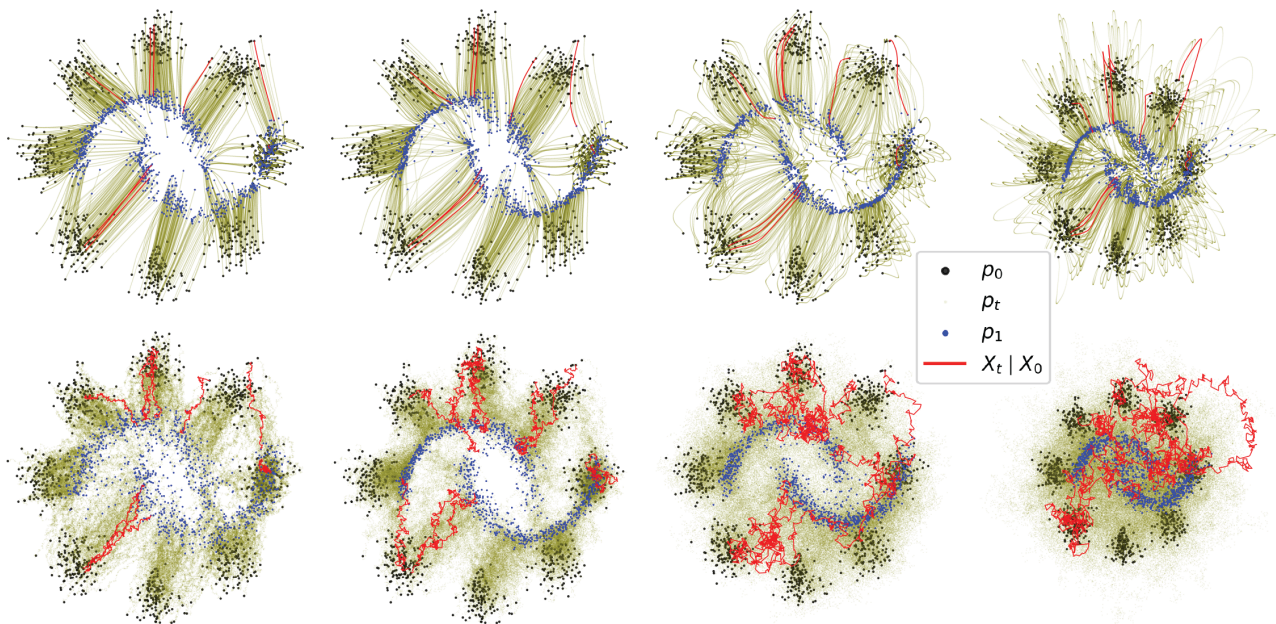# D  ADDITIONAL RESULTS AND ABLATIONS



Figure 5: Learned ODE (top) and SDE (bottom) simulations for $\sigma \in [0.1, 1, 2, 3]$ from left to right for trained $[SF]^2M$ model. The marginals match regardless of the chosen $\sigma$. Trajectory initializations are matched between runs.

## D.1  Looped $[SF]^2M$

As previously discussed, the majority of Schrödinger bridge algorithms to date have used outer iterations to perform iterative proportional fitting on the continuous distributions. This can create marginals closer to the true Schrödinger bridge marginals at the cost of additional computation, and potentially worse generative modelling performance. $[SF]^2M$ is the first Schrödinger bridge method to approximate Schrödinger bridges without performing the iterative proportional fitting in continuous time and space, instead using much more efficient iterations in the static, discrete OT setting.

However, $[SF]^2M$ is compatible with outer looping. In Table 7 we see that outer looping (with 20 outer loops following Shi et al. (2023)) produces better Schrödinger bridge marginals, but worse marginals at time 1 indicating worse generative performance. We note that outer looping takes much longer to train as $[SF]^2M$ effectively takes a
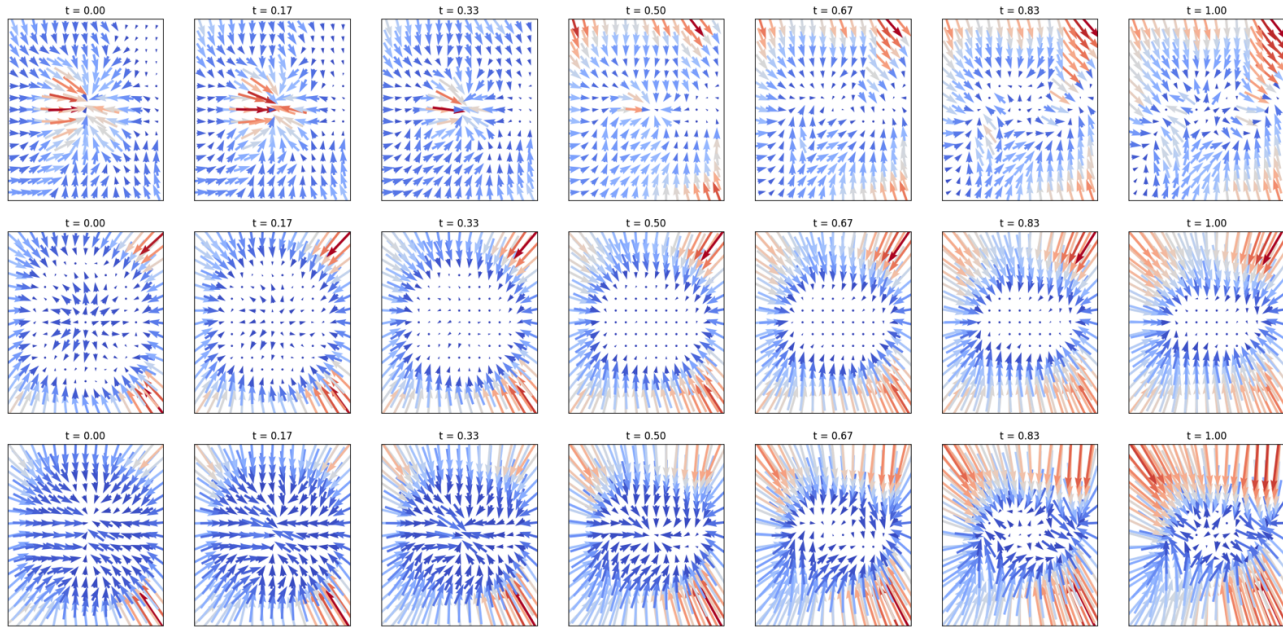
Figure 6: From top to bottom $v_\theta(t, x)$, $s_\theta(t, x)$ and $u_t(x)$ inferred through (5) for the 8-Gaussians to moons dataset.

| [SF]$^2$M | I-CFM | OT-CFM | DSBM | FM | RF |
|-----------|-------|--------|------|-----|-----|
| 4.145 | 4.381 | 4.456 | 4.511 | 4.611 | 6.01 |

Table 8: CIFAR-10 **FID** using 100-step Euler integration. First 4 models trained with batch size 128 for 16 A100-hours. Last 2 models from the DSBM paper, trained for $\sim$200 A100-hours. [SF]$^2$M performs better than DSBM (with 1/12 the compute) and the deterministic methods (I-CFM, OT-CFM, FM). Better performance can be obtained by considering higher-order or adaptive step solvers.

single outer loop, and may have advantages even over a single outer loop as shown in Tong et al. (2024); Pooladian et al. (2023) where the static solution sped up training in the deterministic setting.

Here we accomplish outer looping by simulating 5,000 (stochastic) trajectories from the backwards SDE and 5,000 trajectories from the forwards SDE. We then use the start and end points of these trajectories as samples from the approximate static OT matrix in the next iteration. This algorithm is detailed in Alg. 3. We always set $n$ (the number of samples per outer loop) to the size of the empirical dataset, fix the number of outer loops to 20, and the number inner loops to the [SF]$^2$M without outer loops divided by 20. This gives the same number of gradient descent steps for all methods, but we note that the outer loop methods require simulation for each outer loop. This adds additional computation cost.

### D.2 On the choice of static OT method: Sinkhorn vs. Exact

As the minibatch size gets large, [SF]$^2$M with entropic OT with $\epsilon = 2\sigma^2$ is the correct choice. However, with minibatching, we want a smaller $\epsilon$. In practice we often use $\epsilon = 0$ corresponding to exact (unregularized) optimal transport. We test this on the Gaussian-to-Gaussian experiment in Table 7. Here we see that [SF]$^2$M with sinkhorn OT works better in low dimensions, but struggles in high dimensions. We believe this is because minibatch-OT effectively adds more entropy in higher dimensions. More experimentation and theory is needed to determine the optimal setting of $\epsilon$ for a given dataset and minibatch size.

### D.3 Extra experiments on Cifar10

We have conducted experiments on Cifar 10 using the code from Tong et al. (2024) and the experimental evaluation from Shi et al. (2023). Results are gathered in Table 8.

---

**Algorithm 3** Looped Minibatch Simulation-Free Score and Flow Matching Training

---

**Input:** Samplable source and target $q_0, q_1$, number of outer loop iterations $L$, inner loop iterations $I$, cache size $n$, noise term $\sigma$, weighting schedule $\lambda(t)$, initial networks $v_\theta$ and $s_\theta$.

**for** Outer loop $l \in [1, \ldots, L]$ **do**
    **for** Inner loop $i \in [1, \ldots, I]$ **do**
        **if** $l = 0$ **then**
            $x_0, x_1 \sim q_0^{\otimes m}, q_1^{\otimes m}$
            $\pi \leftarrow \text{Sinkhorn}(x_0, x_1, 2\sigma^2)$          ▷ *Or* $\text{OT}(x_0, x_1)$ *see §A.1*
            $x_0, x_1 \sim \pi^{\otimes m}$          ▷ *Resample OT pairs from* $\pi$
        **else**
            $x_0, x_1 \sim \mathbb{T}^{\otimes m}$
        $t \sim \mathcal{U}(0, 1)$
        $p_t(x|x_0, x_1) \leftarrow \mathcal{N}(x; tx_1 + (1-t)x_0, \sigma^2 t(1-t))$
        $x \sim p_t(x|x_0, x_1)$
        $\mathcal{L}_{[\text{SF}]^2\text{M}} \leftarrow \|v_\theta(t, x) - u_t^\circ(x|x_0, x_1)\|^2 + \lambda(t)^2 \|s_\theta(t, x) - \nabla_x \log p_t(x|x_0, x_1)\|^2$
        $\theta \leftarrow \text{Update}(\theta, \nabla_\theta \mathcal{L}_{[\text{SF}]^2\text{M}})$
    $\mathbb{T}_f \leftarrow (x_0, \hat{x}_1)^{\otimes n//2}$          ▷ *where* $\hat{x}_1$ *is sampled according to Alg. 2*
    $\mathbb{T}_b \leftarrow (\hat{x}_0, x_1)^{\otimes n//2}$          ▷ *where* $\hat{x}_0$ *is sampled according to the backwards analog of Alg. 2*
    $\mathbb{T} \leftarrow [\mathbb{T}_f, \mathbb{T}_b]$
**return** $v_\theta, s_\theta$

---

Table 9: GRN recovery and leave-last-timepoint-out testing using single-cell gene expression over two simulated datasets. We measure performance of predicting the distribution of the final left-out timepoints (2-Wasserstein and radial basis function maximum mean discrepancy) as well accuracy of structure recovery (AUC-ROC and AP).

| | Bifurcating System | | | | Trifurcating System | | | |
|---|---|---|---|---|---|---|---|---|
| | $\mathcal{W}_2$ (↓) | RBF-MMD (↓) | AUC-ROC (↑) | AP (↑) | $\mathcal{W}_2$ (↓) | RBF-MMD (↓) | AUC-ROC (↑) | AP (↑) |
| OT-CFM | **0.782 ± 0.105** | **0.054 ± 0.004** | — | — | 0.921 ± 0.142 | 0.068 ± 0.006 | — | — |
| [SF]²M | 0.791 ± 0.097 | 0.056 ± 0.005 | — | — | 0.932 ± 0.159 | **0.062 ± 0.006** | — | — |
| NGM-[SF]²M$_{\sigma=0}$ | 0.783 ± 0.112 | 0.055 ± 0.006 | **0.786 ± 0.081** | **0.521 ± 0.160** | **0.912 ± 0.161** | 0.064 ± 0.005 | **0.764 ± 0.066** | **0.485 ± 0.105** |
| NGM-[SF]²M$_{\sigma=0.1}$ | 0.835 ± 0.089 | 0.064 ± 0.011 | 0.723 ± 0.014 | 0.444 ± 0.030 | 1.049 ± 0.195 | 0.080 ± 0.014 | 0.731 ± 0.077 | 0.453 ± 0.091 |
| NGM-[SF]²M$_{\sigma=0.01}$ | 0.813 ± 0.101 | 0.064 ± 0.008 | 0.715 ± 0.047 | 0.442 ± 0.033 | 0.956 ± 0.121 | 0.069 ± 0.005 | 0.726 ± 0.081 | 0.451 ± 0.094 |
| NGM-[SF]²M$_{\sigma=0.001}$ | 0.844 ± 0.063 | 0.082 ± 0.018 | 0.699 ± 0.043 | 0.418 ± 0.060 | 1.005 ± 0.150 | 0.094 ± 0.014 | 0.725 ± 0.080 | 0.445 ± 0.082 |
| Spearman | — | — | 0.755 ± 0.003 | 0.438 ± 0.002 | — | — | 0.718 ± 0.005 | 0.413 ± 0.005 |
| Pearson | — | — | 0.744 ± 0.000 | 0.415 ± 0.000 | — | — | 0.710 ± 0.002 | 0.405 ± 0.002 |
| DREMI (Krishnaswamy et al., 2014) | — | — | 0.594 ± 0.017 | 0.293 ± 0.011 | — | — | 0.419 ± 0.021 | 0.205 ± 0.007 |
| Granger (Granger, 1969) | — | — | 0.664 ± 0.013 | 0.421 ± 0.043 | — | — | 0.613 ± 0.048 | 0.343 ± 0.039 |

# E   SCHRÖDINGER BRIDGES WITH VARYING DIFFUSION RATE

While we consider constant diffusion for the majority of this paper, the theory also applies for time varying diffusion with the variation as specified in the following Lemma.

**Lemma E.1** (Brownian bridge with time-varying diffusion)**.** *Suppose* $\mathbf{x}_t$ *is a stochastic process with values in functions* $[0, 1] \to \mathbb{R}^d$, *defined by initial conditions* $\mathbf{x}_0 = \mathbf{a}$ *and SDE* $d\mathbf{x}_t = \sigma(t) \, d\mathbf{w}_t$, *where* $\sigma(t)$ *is continuous and positive on* $(0, 1)$. *Define* $F(t) = \int_0^t \sigma^2(s) \, ds$. *Then* $\mathbf{x}_t$ *satisfies*

$$\mathbf{x}_t \sim \mathcal{N}(\mathbf{a}, F(t))$$

$$\mathbf{x}_t \mid \mathbf{x}_1 = \mathbf{b} \sim \mathcal{N}\left(\mathbf{a} + (\mathbf{b} - \mathbf{a})\frac{F(t)}{F(1)}, F(t) - \frac{F(t)^2}{F(1)}\right).$$

*Proof.* The constraints on $\sigma$ guarantee that $F$ has a unique inverse $F^{-1}$ on $[0, F(1)]$. Consider the process

$$\mathbf{y}_t = \frac{\mathbf{x}_{F^{-1}(F(1)t)} - \mathbf{a}}{\sqrt{F(1)}},$$

which is equivalently characterized by

$$\mathbf{x}_t = \mathbf{a} + \sqrt{F(1)}\mathbf{y}_{F(t)/F(1)}.$$

A straightforward computation using the chain rule shows that $\mathbf{y}_t$ satisfies $\mathbf{y}_0 = \mathbf{0}$ and $d\mathbf{y}_t = d\mathbf{w}_t$, *i.e.*, $\mathbf{y}_t$ is Brownian motion.

By standard facts about Brownian bridges, we have

$$\mathbf{y}_t \sim \mathcal{N}(0, t)$$

$$\mathbf{y}_t \mid \mathbf{y}_1 = \frac{\mathbf{b} - \mathbf{a}}{\sqrt{F(1)}} \sim \mathcal{N}\left(\frac{\mathbf{b} - \mathbf{a}}{\sqrt{F(1)}} t, t(1 - t)\right).$$

The result follows by applying the reverse transformation to obtain the marginals of $\mathbf{x}_t$. □

Because Markovization commutes with time reparametrization, and the SB is the Markovization of a mixture of Brownian bridges, this immediately implies:

**Corollary E.2.** *The solution $p$ to the SB problem with reference process $d\mathbf{x}_t = \sigma(t)\,d\mathbf{w}_t$ has marginal $p(x_0, x_1) = \pi_{2F(1)}(x_0, x_1)$.*

## F EXPERIMENTAL DETAILS

---
**Algorithm 4** Minibatch Optimal Transport Simulation-Free Score and Flow Matching Training

---
**Input:** Samplable source and target $q_0, q_1$, noise term $\sigma$, weighting schedule $\lambda(t)$, initial networks $v_\theta$ and $s_\theta$.
**while** Training **do**
 $\quad t \sim \mathcal{U}(0, 1)$
 $\quad x_0, x_1 \sim q_0^{\otimes m}, q_1^{\otimes m}$
 $\quad \pi \leftarrow \text{Sinkhorn}(x_0, x_1, 2\sigma^2)$ $\qquad\qquad\qquad\qquad\qquad\qquad$ ▷ *Or* $\text{OT}(x_0, x_1)$ *see §A.1*
 $\quad x_0, x_1 \sim \pi^{\otimes m}$ $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad$ ▷ *Resample OT pairs from $\pi$*
 $\quad p_t(x|x_0, x_1) \leftarrow \mathcal{N}(x; tx_1 + (1-t)x_0, \sigma^2 t(1-t))$
 $\quad x \sim p_t(x|x_0, x_1)$
 $\quad \mathcal{L}_{[\text{SF}]^2\text{M}} \leftarrow \|v_\theta(t, x) - u_t(x|x_0, x_1)\|^2 + \lambda(t)^2\|s_\theta(t, x) - \nabla_x \log p_t(x|x_0, x_1)\|^2$ $\qquad$ ▷ *see (8, 11)*
 $\quad \theta \leftarrow \text{Update}(\theta, \nabla_\theta \mathcal{L}_{[\text{SF}]^2\text{M}})$
**return** $v_\theta, s_\theta$

---

### F.1 Implementation details and settings

Throughout we use networks of three layers of width 64 with SeLU activations (Klambauer et al., 2017) except for the 1000 dimensional experiment where we use width 256, and for the neural graphical model (NGM) model used in the gene regulatory network recovery task. For optimization we use ADAM-W (Loshchilov and Hutter, 2019) with learning rate $10^{-3}$ and weight decay $10^{-5}$. We train for 1,000 epochs unless otherwise noted. The flow network and score networks always have exactly the same structure. We provide a more detailed picture of the algorithm used in Alg. 4. For sampling we always take 100 integration steps with either the Euler integrator for ODEs or Euler-Maruyama integrator for SDEs, except for the Gaussian experiments where we take 20 steps to match the setup of De Bortoli et al. (2021); Shi et al. (2023). When there are multiple timepoints (*e.g.*, single-cell trajectories) we take 100 steps between each timepoint for a total of $100k$ steps for all methods. We use $\sigma = 1$ unless otherwise noted.

#### F.1.1 Weighting schedule $\lambda(t)$

Similar to other score and diffusion-based losses, the $[\text{SF}]^2\text{M}$ loss is defined with a weighting schedule $\lambda(t)$. Since $\nabla \log p_t(x|z)$ goes to infinity as $t$ tends to zero or one, we must standardize the loss to be roughly even over time. We set $\lambda(t)$ such that the target has zero mean and unit variance, *i.e.*, we predict the noise added in sampling $x$ from $\mu_t$ before multiplying by $\sigma\sqrt{t(1-t)}$. This leads to the setting:

$$\lambda(t) = \sigma_t = \sigma\sqrt{t(1-t)} \tag{17}$$

this weighting schedule ensures that the regression target for $s_\theta$ is distributed $\mathcal{N}(0, 1)$.

We also experiment with directly regressing $s_\theta$ against the scaled target $\frac{1}{2}\sigma^2\nabla_x \log p_t(x|z)$, with a different weighting function

$$\lambda(t) = \frac{2}{\sigma^2}\sigma_t = \frac{2\sqrt{t(1-t)}}{\sigma} \tag{18}$$

which also ensures the regression target for $\sigma$ is distributed $\mathcal{N}(0,1)$. With this $\lambda$, and since $\nabla_x \log p_t(x|z)$ with a Gaussian $p_t(x|z)$ can be simplified to $-\epsilon/\sigma_t$ where $\epsilon \sim \mathcal{N}(0,1)$ we have the simplified objective:

$$\lambda(t)^2 \left\| s_\theta(t,x) - \nabla_x \log p_t(x|x_0,x_1) \right\|^2 = \left\| \lambda(t)s_\theta(t,x) - \lambda(t)\nabla_x \log p_t(x|x_0,x_1) \right\|^2 \tag{19}$$

$$= \left\| \lambda(t)s_\theta(t,x) + \epsilon_t \right\|^2 \tag{20}$$

This is also numerically stable as it avoids dividing by anything that approaches zero. We leave improved weighting schedules to future work. In practice we use the schedule in (18) and the simplified objective in (20).

### F.1.2 Static optimal transport

As discussed in §A.1 it is sometimes preferable to use exact optimal transport instead of entropic optimal transport. In addition to the "entropy" added by minibatching, there are also numerical and practical considerations. In practice, we find that for batches of size $< 10000$, implementations of exact OT through the Python Optimal Transport package (POT) (Flamary et al., 2021) are often faster than implementations of the Sinkhorn algorithm (Cuturi, 2013), as the Sinkhorn algorithm is known to have numerical difficulties and needs many iterations for good approximation of the true entropic transport for small values of $\sigma$.

### F.2 Computational resources

All experiments were performed on a shared heterogeneous high-performance-computing cluster. This cluster is primarily composed of GPU nodes with RTX8000, A100, and V100 Nvidia GPUs, and CPU nodes with 32 and 64 CPUs.

### F.3 Two-dimensional experimental details

For the two-dimensional experiments we follow the setup from Shi et al. (2023). This is adapted from the setup of Tong et al. (2024) except using larger test sets for lower variance in the empirical estimation of the Wasserstein distance. We use a training set size of 10,000, a validation set size of 10,000, and a test set size of 10,000 for all methods and models.

We evaluate the empirical 2-Wasserstein distance for 10,000 forward samples from our model pushing the source distribution to the target. The number reported for $\mathcal{W}_2$ is then

$$\mathcal{W}_2 = \left( \min_{\pi \in U(\hat{p}_1,q_1)} \int \|x-y\|_2^2 d\pi(x,y) \right)^{1/2} \tag{21}$$

where $\hat{p}_1$ is sampled via Alg. 2, and $q_1$ is the test set.

As mentioned in the main text, we also measure the Normalized Path Energy. We note that this is only defined for ODE integration, hence for stochastic methods, (*i.e.*, [SF]$^2$M, DSBM) we measure the normalized path energy of the probability flow ODE (see (4)). The normalized path energy measures the relative deviation of the path energy of the model ($\int \|v_\theta(t,x)\|^2$) to the path energy of the optimal paths ($\mathcal{W}_2^2(q_0,q_1)$), which is equivalent to the squared 2-Wasserstein distance between the test set source and the test set target. More formally, the normalized path energy can be calculated as

$$NPE(q_0,q_1,v_\theta) = \frac{|\mathbb{E}_{x(0)\sim q_0} \int \|v_\theta(t,x_t)\|^2 dt - \mathcal{W}_2^2(q_0,q_1)|}{\mathcal{W}_2^2} \tag{22}$$

where $x_t$ is the solution of the probability flow ODE with $dx = v_\theta(t,x)dt$ with initial condition $x_0$. This measures how close the paths defined by $v_\theta(t,x)$ are to the optimal transport paths in terms of average energy. We note that measuring the path energy rather than the length ($\mathcal{W}_2^2$ instead of $\mathcal{W}_1$) has the additional benefit that the energy differentiates between models that follow the same paths, but at different rates, encouraging constant rate models.

### F.4 Gaussian-to-Gaussian experimental details

Similar to experiments in De Bortoli et al. (2021); Shi et al. (2023), we train on a sample of 10,000 points from a $\mathcal{N}(-0.1, I)$ source to 10,000 points sampled from a $\mathcal{N}(0.1, I)$ target, for dimensions 5, 20, and 50. We could not find all the details of those previous experiments but we match what we can here. Since we know the closed-form solution to the Schrödinger bridge from Mallasto et al. (2022); Bunne et al. (2022a) $q_t$, we can compare our estimate of the marginal at time $t$ ($\hat{p}_t$) with the true distribution at time $t$. In particular, we compare a Gaussian approximation of $\hat{p}_t$, $\tilde{p}_t$ with mean and covariance estimated from 10,000 samples of the model with $q_t$ which has the form

$$q_t(x) = \mathcal{N}\left(x; 0.2t - 0.1, (t(1-t)\sqrt{4 + \sigma^4} + (1-t)^2 + t^2)I\right) \tag{23}$$

with the KL divergence.

We compare either the average KL over 21 timepoints (including the start and end timepoints) *i.e.*,

$$\overline{\text{KL}} = \frac{1}{20}\sum_{k=0}^{20} \text{KL}(\tilde{p}_{k/20}||q_{k/20}) \tag{24}$$

to measure how closely the learned flow matches the true Schrödinger bridge marginals, and we also compare the KL divergence at time $t = 1$, to measure the performance as a generative model.

$$\text{KL}(\tilde{p}_1||q_1) \tag{25}$$

### F.5 Waddington's landscape experimental details

We use two Waddington landscapes, one Gaussian to two Gaussians and cross-sectional measurements of Embryoid Body (EB) data, to demonstrate the versatility of [SF]$^2$M for trajectory inference. The three dimensions of the landscape are space, time and potential.

More specifically, the space dimension is evolved according to the drift of the SDE by $u_t(x) = -\nabla_x W(t, x)$. The potential dimension is the Waddington's landscape by $W := E_v + \frac{1}{2}g(t)^2 E_s$, where $v_\theta(t, x) = -\nabla_x E_v(t, x)$ and $s_\theta(t, x) = -\nabla_x E_s(t, x)$ are the flow and score for Langevin dynamics. $E_s$ and $E_v$ are both parameterized by three layer neural networks of width 64, the only difference with our standard implementation of $v_\theta$ and $s_\theta$ is that they have output dimension width of one instead of $d$.

For the experiment of one Gaussian to two Gaussians, we train on a sample of 256 points from the one-dimensional source $\mathcal{N}(0, 0.1)$ to 256 points sampled from the one-dimensional target $\mathcal{N}(-1, 0.1) \cup \mathcal{N}(1, 0.1)$ for 10,000 steps. We then plot 20 trajectories from the source to the target with the potentials following from the gradient descend of $W$.

For the cross-sectional measurements from the embryoid body (EB) data, we first embed the data in one dimension with the non-linear dimensionality reduction technique PHATE (Moon et al., 2019), which we then whiten to ensure the data is at a reasonable scale for the neural network initialization. We train the [SF]$^2$M model following Alg. 5 for 50,000 steps and plot 100 stochastic trajectories along with the height of $W(t, x)$ normalized so to gradually descend over time.

### F.6 Single-cell interpolation experimental details

Here we perform two comparisons, the first matching the setup of Tong et al. (2020) in low dimensions and the second exploring higher dimensional single-cell interpolation. Following Huguet et al. (2022b), we repurpose the CITE-seq and Multiome datasets from a recent NeurIPS competition for this task (Burkhardt et al., 2022) as well as the Embryoid-body data from Moon et al. (2019); Tong et al. (2020), which has 5 population measurements over 30 days.

For the Embryoid body (EB) data, we use the same processed artifact which contains the first 100 principal components of the data. For our tests in Table 4, we truncate to the first five dimensions, then whiten each dimension following Tong et al. (2020) before interpolation. For the Embryoid body (EB) dataset which consists of 5 timepoints collected over 30 days we train separate models leaving out times $1, 2, 3$ in turn. During testing we

---

**Algorithm 5** Trajectory Simulation-Free Score and Flow Matching Training

---

**Input:** Samplable source and target $Q = \{q_0, \cdots, q_{K-1}\}$, noise term $\sigma$, weighting schedule $\lambda(t)$, initial networks $v_\theta$ and $s_\theta$.

**while** Training **do**
    **for** $k \in [0, \cdots, K-2]$ **do**
        $x_k, x_{k+1} \sim q_k^{\otimes m}, q_{k+1}^{\otimes m}$
        $\pi \leftarrow \text{Sinkhorn}(x_k, x_{k+1}, 2\sigma^2)$            ▷ *Or* $\text{OT}(x_k, x_{k+1})$ *see §A.1*
        $\mathbb{X}_k \sim \pi^{\otimes m}$            ▷ *Resample OT pairs from* $\pi$
    $k \sim \mathcal{U}(1, K)^{\otimes m}$
    $t \sim \mathcal{U}(0, 1)^{\otimes m}$
    $x_0^i, x_1^i \leftarrow \mathbb{X}_k^i$
    $p_t(x|x_0, x_1) \leftarrow \mathcal{N}(x; tx_1 + (1-t)x_0, \sigma^2 t(1-t))$
    $x \sim p_t(x|x_0, x_1)$
    $\mathcal{L}_{[\text{SF}]^2\text{M}} \leftarrow \|v_\theta(t+k, x) - u_t(x|x_0, x_1)\|^2 + \lambda(t)\|s_\theta(t+k, x) - \nabla_x \log p_t(x|x_0, x_1)\|^2$
    $\theta \leftarrow \text{Update}(\theta, \nabla_\theta \mathcal{L}_{[\text{SF}]^2\text{M}})$
**return** $v_\theta, s_\theta$

---

push forward all observed points $X_{t-1}$ to time $t$ then measure the 1-Wasserstein distance between the predicted and true distribution.

For the Cite and Multi datasets these are sourced from the Multimodal Single-cell Integration challenge at NeurIPS 2022, a NeurIPS challenge hosted on Kaggle where the task was multi-modal prediction (Burkhardt et al., 2022). Here, we repurpose this data for the task of time series interpolation. Both of these datasets consist of four timepoints from CD34+ hematopoietic stem and progenitor cells (HSPCs) collected on days 2, 3, 4, and 7. For more information and the raw data see the competition site.[1] We preprocess this data slightly to remove patient specific effects by focusing on a single donor (donor 13176).

Since these data have the full (pre-processed) gene level single-cell data, we try interpolating on higher-dimensional unwhitened principle components, and on the first 1000 highly variable genes, which is a standard preprocessing step in single-cell data analysis. To our knowledge, [SF]$^2$M is the first method to scale to the gene space of single-cell data. In Table 5, we again measure the 1-Wasserstein distance between the push forward predicted distribution and the ground truth distribution.

### F.6.1 Geodesic ground costs

We also introduce the Geodesic Sinkhorn method from Huguet et al. (2022b) for dynamic Schrödinger bridge interpolation. Here the cost is a geodesic cost based on a $k$-nearest-neighbour graph between cells.

### F.7 Gene regulatory network recovery experimental details

Using the neural graphical model (NGM), we can parameterize the gene-gene interaction graph directly within the ODE drift model $v_\theta(t, x)$. To do so, following from Bellot and Branson (2022) we can define:

$$v_{\theta_j}(t, x) = \phi(\cdots \phi(\phi(x\theta_j^{(1)})\theta_j^{(2)}) \cdots )\theta_j^{(K)}, \qquad j = 1, \ldots, d, \tag{26}$$

where $\theta_j^{(1)} \in \mathbb{R}^{d \times h}$ represents a continuous adjacency matrix of the gene-gene interactions, $\theta_j^{(k)} \in \mathbb{R}^{h \times h}, k = 2, \ldots, K-1$ are parameters of each proceeding hidden layer, $\theta_j^{(K)} \in \mathbb{R}^{h \times 1}$, $x \in \mathbb{R}^d$, and $\phi(\cdot)$ is an activation function. Then we can consider $v_\theta(t, x) = (v_{\theta_1}(t, x), \ldots, v_{\theta_d}(t, x))$ as $h$ ensembles over structure $\theta^{(1)}$. We can then use Algorithm Alg. 1 to train the NGM model with the addition of an $L^1$ penalty over structure to enforce sparsity on gene-gene interactions, i.e $\lambda_1 \|\theta^{(1)}\|_1$. We include bias terms in our implementation of (26).

Using BoolODE (Pratapa et al., 2020), we generate simulated single-cell gene expression trajectories for a bifurcating system and a trifurcating system. For the bifurcating system we consider 7 synthetic genes and generate trajectories over 1000 cells using a simulation time of 5 and an initial condition on gene 1 at a value of 1.

---

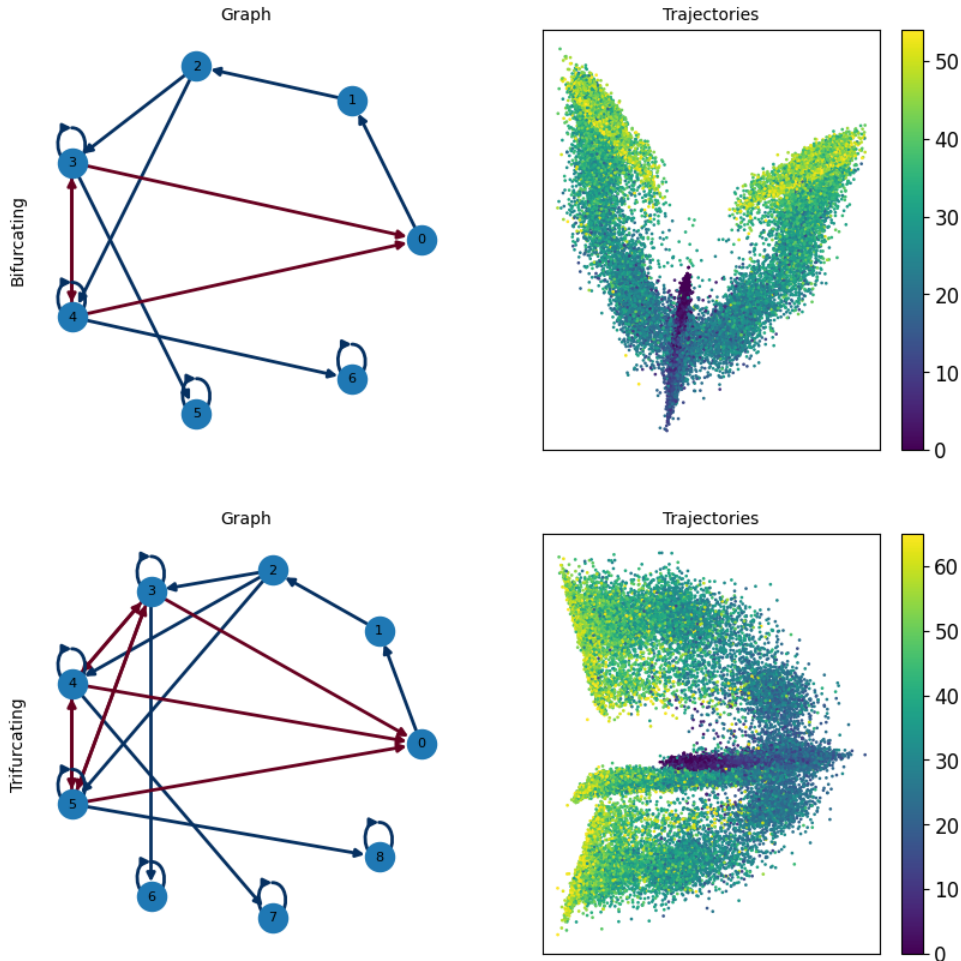[1]https://www.kaggle.com/competitions/open-problems-multimodal/data

Figure 7: Simulated single-cell trajectories given synthetic GRNs emulating bifurcating (top) and trifurcating (bottom) systems. GRNs contain directed edges of Boolean relationships between genes. For example, a red edge between gene 3 and gene 0 (top left) indicates that the rule for gene 0 is (*not* gene 3). Likewise, a blue edge between gene 6 and gene 4 indicates that rule for gene 6 is (gene 4 and gene 6). This follows from the procedure for defining synthetic GRNs using the BoolODE framework (Pratapa et al., 2020). The color bar indicates the scale of temporal progression.

We post-process the data and sub-sample to 55 timepoints and scramble the cell pairing to emulate real-world data. For the trifurcating system we consider 9 synthetic genes and generate trajectories over 800 cells using a simulation time of 6 and an initial condition on gene 1 at a value of 1. We post-process the data and sub-sample to 66 timepoints and scramble the cell pairing to emulate real-world data. We use a train-test data split of $\{0.8, 0.2\}$ respectively, and leave out the end timepoints for trajectory prediction evaluation. We the show underlying synthetic GRNs and simulated single-cell trajectories in Fig. 7.

For OT-CFM (*i.e.*, [SF]$^2$M with $\sigma = 0$), we parameterize the NGM model with two hidden layers where $\theta_j^{(1)} \in \mathbb{R}^{d \times h}$ with $h = 100$ and $d$ represents the number of input genes. Then the second layer (*i.e.*, $k = 2$) is $\theta_j^{(2)} \in \mathbb{R}^{h \times 1}$. We use this parameterization for both the bifurcating system and trifurcating systems. For [SF]$^2$M with $\sigma > 0$, we use two heads stemming from $\theta_j^{(1)}$ for the flow matching model and score matching model, respectively. Specifically, we use an additional layer $\tilde{\theta}_j^{(2)} \in \mathbb{R}^{h \times 1}$ such that $s_{\theta_j}(x, t) = \phi(\phi(x\theta_j^{(1)})\theta_j^{(2)}), j = 1, \ldots, d$. We use the SeLU activation functions for both models. To train [SF]$^2$M and NGM-[SF]$^2$M models on the bifurcating system, we use the Adam optimizer with a learning rate of 0.01 and batch size of 128 and use $\lambda_1 = 10^{-5}$. On the trifurcating system, we use the Adam optimizer with a learning rate of 0.01 and batch size of 64 and use $\lambda_1 = 10^{-6}$. We generate results for 5 model seeds. For baseline methods (*i.e.*, Spearman, Pearson, DREMI, and Granger) we generate results over 5 cell-pair scramble seeds. To evaluate GRN recovery performance, we compute the area

under the receiver operator characteristic (AUC-ROC) and average precision (AP) scores of the predicted GRNs compared to the ground truth GRNs used for generating the simulated data. We mask out the diagonal elements (self regulation loops) of the predicted and ground truth GRNs for computing the AUC-ROC and AP. We provide the full results of the GRN recovery experiments in Table 9.