# Simulation-Free Schrödinger Bridges via Score and Flow Matching

**Alexander Tong** [1 2] **Nikolay Malkin** [1 2] **Kilian Fatras** [1 3] **Lazar Atanackovic** [4 5] **Yanlei Zhang** [1 2]
**Guillaume Huguet** [1 2] **Guy Wolf** [1 2] **Yoshua Bengio** [1 2 6]

## Abstract

We present *simulation-free score and flow matching* ([SF]$^2$M), a simulation-free objective for inferring stochastic dynamics given unpaired source and target samples drawn from arbitrary 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 (SB) 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.

Code: 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 & Ermon, 2019; Ho et al., 2020; Nichol & Dhariwal, 2021; Dhariwal & 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, but recent work has introduced simulation-free training objectives that make CNFs competitive with SBGMs when a Gaussian source is assumed (Lipman et al., 2023; Pooladian et al., 2023) and extended these objectives to the case of arbitrary source distributions (Liu, 2022; Albergo & Vanden-Eijnden, 2023; Tong et al., 2023b). 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 natural 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, 2014). 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 & Ermon, 2019), modeling natural stochastic dynamical systems (Schiebinger et al., 2019; Holdijk et al., 2022; Koshizuka & 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 Bor-

[1]Mila – Québec AI Institute [2]Université de Montréal [3]McGill University [4]University of Toronto [5]Vector Institute [6]CIFAR Fellow. Correspondence to: Alexander Tong <alexander.tong@mila.quebec>.

*Accepted to ICML workshop on Frontiers in Learning, Control, and Dynamical Systems.* 

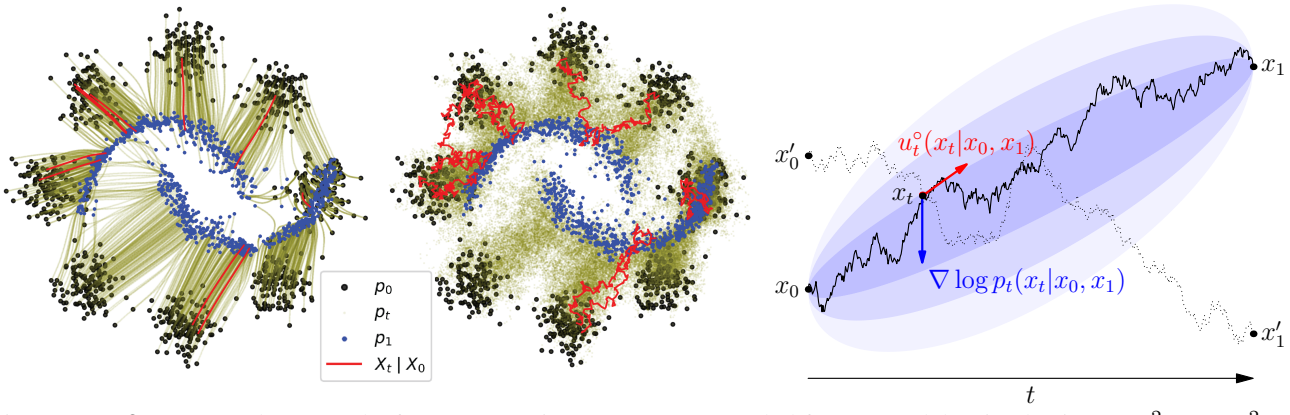


Figure 1: **Left:** ODE and SDE paths from 8-Gaussians to moons, sampled from a model trained using [SF]$^2$M. [SF]$^2$M makes it possible to vary the diffusion schedule at inference time and thus interpolate between ODEs and SDEs that have the same marginal densities. **Right:** Illustration of the 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.

toli 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., 2023b; Pooladian et al., 2023) to the case of stochastic dynamics and (2) the denoising training objective for diffusion models to the case of arbitrary source distributions (Figure 1). Our algorithm uses a connection between the SB problem and entropic optimal transport (OT) to express the Schrödinger bridge as the Markovization of a mixture of Brownian bridges (De Bortoli et al., 2021; Léonard, 2014). In contrast to dynamic SB algorithms that require simulating an SDE on every iteration, [SF]$^2$M can take advantage of *static* entropic OT maps between source and target distributions, which are efficiently approximated by the Sinkhorn algorithm (Sinkhorn, 1964; Cuturi, 2013; Altschuler et al., 2017) or stochastic algorithms (Genevay et al., 2016; Seguy et al., 2018).

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 in generative modeling metrics 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 & 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 in the static optimal transport case, we are able to directly model and recover the gene-gene interaction network driving the cell dynamics.

**Outline.** In §2, we review the mathematical background needed to formulate [SF]$^2$M. §3 presents and analyzes the proposed algorithm. §4 reviews related work, and §5 is dedicated to experiments.

## 2. Preliminaries: Neural SDEs and Schrödinger bridges

In this section we review the mathematical background and common assumptions needed to state our new algorithms. The exposition closely follows that of Tong et al. (2023b) but extends it to the stochastic case.

We will consider a pair of compactly supported distributions over $\mathbb{R}^d$ with densities $q(x_0)$ and $q(x_1)$ (also denoted $q_0, q_1$). The densities may not be explicitly known, but instead we may have finite datasets of samples available. 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 $q_0$ – which could be an easy-to-sample distribution, such as a Gaussian – and transforming it by $f$ to obtain a sample from $q_1$.

### 2.1. SDEs, scores, and probability flows

We consider a stochastic differential equation[1] defined by a smooth time-dependent vector field $u : [0, 1] \times \mathbb{R}^d \to \mathbb{R}^d$ and a continuous, positive diffusion function $g : [0, 1] \to \mathbb{R}_{>0}$:

$$dx = u_t(x)\, dt + g(t)\, dw_t, \tag{1}$$

where $u_t(x)$ is used interchangeably with $u(t, x)$ and $dw_t$ is the standard Brownian motion. Given a distribution over initial conditions $p(x_0)$, this SDE defines a stochastic process, or distribution over paths, which is a joint distribution over variables $x_t$ indexed by $t \in [0, 1]$.

Although the evolution of points according to the SDE over short time ranges is simple – $p(x_{t+\Delta t}|x_t)$ is approximated by the Gaussian $\mathcal{N}(x_t + u_t(x_t)\Delta t, g(t)^2\Delta t)$ – the long-range conditional distributions $p(x_t \mid x_0)$ induced by the SDE may be quite complex. This property enables continuous-time stochastic models, such as diffusion models, to sample multimodal distributions by iterative addition of conditional Gaussians to an initial noise sample.

A density $p(x_0)$ evolved according to the SDE (1) induces marginal distributions $p(x_t)$, with their densities also denoted $p_t$. The marginal densities, viewed as a function $p : [0, 1] \times \mathbb{R}^d \to \mathbb{R}$, are characterized by the initial conditions $p_0$ and the *Fokker-Planck equation*:

$$\frac{\partial p}{\partial t} = -\nabla \cdot (p_t u_t) + \frac{g^2(t)}{2} \Delta p_t, \tag{2}$$

where $\Delta p_t = \nabla \cdot (\nabla p_t)$ is the Laplacian. In the degenerate case $g(t) \equiv 0$, the SDE becomes an ODE and we recover the *continuity equation* $\frac{\partial p}{\partial t} = -\nabla \cdot (p_t u_t)$. From the Fokker-Plank and continuity equations, it can easily be derived that the ODE

$$dx = \underbrace{\left[u_t(x) - \frac{g(t)^2}{2} \nabla \log p_t(x)\right]}_{u_t^\circ(x)} dt, \tag{3}$$

together with distribution over initial conditions $p(x_0)$, induces the same marginal distributions $p_t(\cdot)$ as the SDE (1); therefore, (3) is called the *probability flow ODE* of the 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

$$u_t(x) = u_t^\circ(x) + \frac{g(t)^2}{2} \nabla \log p_t(x). \tag{4}$$

Therefore, **specifying an SDE is tantamount to specifying its probability flow ODE and its score function**. By reversing the sign in of $u_t^\circ$ in the ODE (3) and converting to an SDE using (4), we also recover the formula for the reverse-time SDE from Anderson (1982):

$$\begin{aligned} dx &= \left[-u_t^\circ(x) + \frac{g(t)^2}{2} \nabla \log p_t(x)\right] dt + g(t)\, dw_t \\ &= \left[-u_t(x) + g(t)^2 \nabla \log p_t(x)\right] dt + g(t)\, dw_t, \end{aligned} \tag{5}$$

which induces the same distribution on $x_{1-t}$ as (1) 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

$$\begin{aligned} \mathcal{L}_{\text{U[SF]}^2\text{M}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}(0,1), x\sim p_t(x)} \Big[ & \underbrace{\|v_\theta(t, x) - u_t^\circ(x)\|^2}_{\text{flow matching loss}} \\ & + \lambda(t)^2 \underbrace{\|s_\theta(t, x) - \nabla \log p_t(x)\|^2}_{\text{score matching loss}} \Big], \end{aligned} \tag{6}$$

where $\lambda(\cdot)$ 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 Algorithm 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* and still obtain samples from the same marginals if the global optimum of (6) is attained (Figure 1). 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 (5).

**Gaussian marginals and Brownian bridges.** We will require some facts about ODEs and SDEs whose marginals are Gaussian, i.e., $p_t(x) = \mathcal{N}(x; \mu_t, \sigma_t^2)$. By Theorem 3 of Lipman et al. (2023) or Theorem 2.1 of Tong et al. (2023b), these marginals are generated by the ODE defined by

$$u_t^\circ(x) = \frac{\sigma_t'}{\sigma_t}(x - \mu_t) + \mu_t', \tag{7}$$

[1] Because the diffusion term depends only on $t$, the SDE can be either in the Itô or the Stratonovich sense.

where $\sigma_t', \mu_t'$ denote time derivatives, and have score $\nabla \log p_t(x) = -(x - \mu_t)/\sigma_t^2$. A case of particular interest is the *Brownian bridge* from $x_0$ to $x_1$ with constant diffusion rate $g(t) = \sigma$, which has $p_t(x) = \mathcal{N}(x; tx_1 + (1-t)x_0, \sigma^2 t(1-t))$ and therefore satisfies

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

### 2.2. Schrödinger bridges via entropic optimal transport

In this section, we briefly recall the entropically-regularized optimal transport problem and its connections to Schrödinger bridges.

**Entropically-regularized optimal transport.** For two probability measures $q_0$ and $q_1$, the entropically-regularized OT problem is defined as follows:

$$\mathrm{OT}_\varepsilon(q_0, q_1) = \min_{\pi \in U(q_0, q_1)} \int d(x_0, x_1)^2 \, d\pi(x_0, x_1) + \varepsilon \, \mathrm{KL}(\pi \| q_0 \otimes q_1), \tag{9}$$

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. We denote the entropic optimal transport plan – the argmin in (9) – as $\pi_\varepsilon^\star$. When $\varepsilon \to 0$, we recover the classical (Kantorovich) optimal transport, which we call *exact optimal transport*.

**Schrödinger bridge.** The Schrödinger bridge problem asks what the most likely evolution between two marginals with respect to some reference stochastic process is. Formally, the Schrödinger bridge is the solution to the following problem:

$$\mathbb{P}^\star = \min_{\mathbb{P}: p_0 = q_0, p_1 = q_1} \mathrm{KL}(\mathbb{P} \,\|\, \mathbb{Q}), \tag{10}$$

where $\mathbb{P}$ is a stochastic process (distribution over continuous paths $[0,1] \to \mathbb{R}^d$) with law $p_t$ and $\mathbb{Q}$ is a reference process. One often considers $\mathbb{Q} = \sigma\mathbb{W}$ where $\mathbb{W}$ is the standard Brownian motion defined by the SDE $dx = dw_t$, in which case the solution to (10) is known as the *diffusion Schrödinger bridge* (De Bortoli et al., 2021; Bunne et al., 2022a). Interestingly, the marginals of the Schrödinger bridge can be characterized as a mixture of Brownian bridges weighted by an entropic OT plan in the following way:

$$p_t(x) = \int p_t(x|x_0, x_1) \, d\pi^\star_{2\sigma^2}(x_0, x_1) \tag{11}$$

where $p_t(x|x_0, x_1) = \mathcal{N}(x; (1-t)x_0 + tx_1, \sigma^2 t(1-t))$ is the marginal of the Brownian bridge between $x_0$ and $x_1$ with diffusion rate $\sigma$. This formulation motivates our approach, which uses fast solutions to the entropy-regularized OT problem (9) and the closed-form $u_t^\circ$ and $\nabla \log p_t$ of Brownian bridges (8) to approximate the Schrödinger bridge.

## 3. [SF]$^2$M: Simulation-free training of SDEs

We next describe our method of simulation-free score and flow matching, which is summarized in Alg. 1. We present the general case in §3.1, then consider the discrete Schrödinger bridge case with entropy-regularized joint in §3.2, and specializations to cell dynamics with a geodesic cost in §3.3.

### 3.1. Matching the conditional flow and score

Tong et al. (2023b) described a general simulation-free stochastic regression objective, conditional flow matching (CFM), that fits an ODE generating marginal distributions $p_t(x)$ that are mixtures of Gaussian probability paths, given unpaired samples of $x_0$ and $x_1$. We generalize CFM to the stochastic setting.

Suppose the marginal probabilities $p_t(x)$ are representable as mixtures over a latent distribution $q(z)$ i.e.

$$p_t(x) = \int p_t(x|z) q(z) \, dz. \tag{12}$$

Suppose also that $p_t(x|z)$ is generated by $u_t^\circ(x|z)$ from initial conditions $p_0(x|z)$. One then has expressions for an ODE that generates $p_t(x)$ from initial conditions $p_0$, and for the score of $p_t(x)$, in terms of the conditional ODEs and scores:

$$u_t^\circ(x) = \mathbb{E}_{q(z)} \frac{u_t^\circ(x|z) p_t(x|z)}{p_t(x)},$$
$$\nabla \log p_t(x) = \mathbb{E}_{q(z)} \left[ \frac{p_t(x|z)}{p_t(x)} \nabla \log p_t(x|z) \right]. \tag{13}$$

To be precise, we have the following generalization of Theorem 3.1 from Tong et al. (2023b):

**Theorem 3.1.** *Under mild regularity conditions, the ODE $dx = u_t^\circ(x)\, dt$ generates the probability path (12) from initial conditions $p_0(x)$, and the score of $p_t$ is given by (13).*

**A stochastic regression objective.** The expressions for the marginal ODE drift and score in (13) motivate objectives for fitting $u_t^\circ(x)$ and $\nabla p_t(x)$ with neural networks when only the *conditional* ODEs and scores are known. Generalizing (6), 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:

$$\mathcal{L}_{[\mathrm{SF}]^2\mathrm{M}}(\theta) = \\ = \mathbb{E}_{t\sim\mathcal{U}(0,1),z\sim q(z),x\sim p_t(x|z)}\big[\underbrace{\|v_\theta(t,x) - u_t^\circ(x|z)\|^2}_{\text{conditional flow matching loss}} \\ + \lambda(t)^2 \underbrace{\|s_\theta(t,x) - \nabla \log p_t(x|z)\|^2}_{\text{conditional score matching loss}}\big], \quad (14)$$

where, as in (6), $\lambda(\cdot)$ is some choice of positive weights. This objective can be used to approximate the quantities defined in (13), provided the conditional ODEs and scores are known and $p_t(x|z)$ can be tractably sampled. Correctness is guaranteed by the following:

**Theorem 3.2.** *If $p_t(x) > 0$ for all $x \in \mathbb{R}^d$ and $t \in [0,1]$, then $\mathcal{L}_{[\mathrm{SF}]^2\mathrm{M}}(\theta)$ and $\mathcal{L}_{\mathrm{U}[\mathrm{SF}]^2\mathrm{M}}(\theta)$ are equal up to a constant, and $\nabla_\theta\mathcal{L}_{\mathrm{U}[\mathrm{SF}]^2\mathrm{M}}(\theta) = \nabla_\theta\mathcal{L}_{[\mathrm{SF}]^2\mathrm{M}}(\theta)$, where $\mathcal{L}_{\mathrm{U}[\mathrm{SF}]^2\mathrm{M}}(\theta)$ is the unconditional score and flow matching loss (6).*

This result generalizes Theorem 3.2 of Tong et al. (2023b). 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 (6)). The training and inference algorithms are summarized as Algs. 1 and 2.

We remark that the conditional ODEs and scores themselves arise from SDEs, then the SDE recovered from the ODE and score defined via (13) is the Markovian projection of the mixture of stochastic processes indexed by $z$. The Markovization of a mixture of SDEs, approximated in a *simulation-based* way, is also used to solve the Schrödinger bridge problem in De Bortoli et al. (2021); Shi et al. (2022; 2023).

**Algorithm 1** Simulation-Free 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**
  $z \sim q(z)$;  $t \sim \mathcal{U}(0,1)$;  $x \sim p_t(x|z)$
  $\mathcal{L}_{[\mathrm{SF}]^2\mathrm{M}} \leftarrow \|v_\theta(t,x) - u_t(x|z)\|^2 + \lambda(t)^2\|s_\theta(t,x) - \nabla_x \log p_t(x|z)\|^2$ ▷ *see eq. 14*
  $\theta \leftarrow \mathrm{Update}(\theta, \nabla_\theta\mathcal{L}_{[\mathrm{SF}]^2\mathrm{M}})$
**return** $v_\theta, s_\theta$

**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 $\sigma_{\min}$ 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 §C.)

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 $(q_0, q_1)$, which we denote as $U(q_0, q_1)$. This can be formalized in the following theorem:

**Theorem 3.3** ([SF]$^2$M solves the generative modeling problem)**.** *If $q(x_0, x_1) \in U(q_0, q_1)$ and $v_\theta^*, s_\theta^*$ globally minimize $\mathcal{L}_{[\mathrm{SF}]^2\mathrm{M}}(\theta)$ then if $p_t$ is the marginal probability induced by an SDE with drift $[v_\theta^* + \frac{1}{2}g(t)^2 s_\theta^*]$ and diffusion $g$, and initial conditions $p_0$, then $p_t(x) = q_t(x)$ for all $t$ and $x$.*

Taking $t = 1$, this tells us that as long as our joint distribution $q(x_0, x_1)$ has the correct marginals, [SF]$^2$M will optimize for a valid generative model which pushes $q_0$ to $q_1$ or vice-versa. For the ODE case, various options were studied in Tong et al. (2023b). Various choices for the stochastic case will be discussed in §3.2.

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

In the previous section, we showed that our method SF2M can approximate marginal probability $p_t$ in the form of eq. 12. In this section, we explain how our [SF]$^2$M framework is able to approximate a Schrödinger bridge as a special case.

**[SF]$^2$M approximates the Schrödinger bridge.** In order to achieve an efficient approximation of the Schrödinger bridge, we leverage a significant insight: its marginals can be expressed as a mixture of Brownian bridges, which are weighted by an entropic optimal transport plan as shown in eq. 11. Therefore, to approximate the Schrödinger bridge with [SF]$^2$M, we just need to set the distribution $q(x_0, x_1)$ to be equal to the entropic OT plan $\pi^\star_{2\sigma^2}(q_0, q_1)$ and then, to train our networks $v_\theta$ and $s_\theta$ following Algorithm 1. We show in the next proposition that our [SF]$^2$M method correctly approximates the Schrödinger bridge, which is a direct consequence of Theorem 3.3 and eq. 11.

**Proposition 3.4** ([SF]$^2$M with entropic OT computes the SB marginals)**.** *Denote the law of the Schrödinger bridge between $\hat{q}_0$ and $\hat{q}_1$ as $p$. Then if $v_\theta^\star$, $s_\theta^\star$ globally minimize $\mathcal{L}_{[\mathrm{SF}]^2\mathrm{M}}$ with respect to $\pi^\star_{2\sigma^2}(\hat{q}_0, \hat{q}_1)$, then they generate the marginals of the empirical SB $p_t$ (eq. 11).*

**Empirical distributions of finite support.** 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 between empirical distributions $\hat{q}_0$ and $\hat{q}_1$, which we denote $\pi^\star_{2\sigma^2}(\hat{q}_0, \hat{q}_1)$ (Cuturi, 2013; Altschuler et al., 2017). This also means that we only have access to an empirical Schrödinger bridge. Note that it is straightforward to adapt Proposition 3.4 to show that [SF]$^2$M approximates the empirical Schrodinger bridge. Fortunately, it has been established that the true entropic OT cost can be efficiently approximated using empirical distributions, even in high-dimensional spaces (Genevay et al., 2019; Mena & Niles-Weed, 2019), and it was recently shown that the Schrödinger bridge inherits this property (Stromme, 2023, Theorem 5) with $O(n^{1/2})$ error.

The entropic OT plan between empirical distributions can be efficiently computed using the Sinkhorn algorithm (Cuturi, 2013), which has a quadratic computational complexity in the number of samples $n$ (Altschuler et al., 2017). However, this cost may be too high, either because $n$ is too large or because arbitrarily many samples can be drawn from the source or target (such as in the Gaussian-to-data setting). Therefore, we consider the minibatch OT approximation that we discuss in the next paragraph.

**Approximating Schrödinger bridge via minibatch optimal transport.** When $n$ is too large, we consider batches of size $m \ll n$, following Fatras et al. (2020; 2021b). In this case the *minibatch optimal transport* is defined informally as the average over many smaller "minibatch" plans.

$$\mathrm{MBOT}_m(\hat{q}_0, \hat{q}_1) := \mathbb{E}_{\alpha\sim\hat{q}_0^{\otimes m}, \beta\sim\hat{q}_1^{\otimes m}} \mathrm{OT}_\varepsilon(\alpha, \beta) \quad (15)$$

The MBOT complexity complexity scales with $O(km^2)$ for a $k$-sample Monte Carlo approximation. In practice, we set $m$ equal to the stochastic gradient descent batch size following Tong et al. (2023b); Pooladian et al. (2023). This has extremely small computational overhead in practice. We also find that the minibatch $\varepsilon$ should be smaller than $2\sigma^2$ (or even equal to zero) in the minibatch setting to reach best empirical performance. We show that this might be due to the implicit entropy-inducing regularization of minibatch OT (Fatras et al., 2020; 2021b), which we detail further in §B.1 and illustrate in Figure S1.

### 3.3. Applications to learning cell dynamics and regulatory networks

The modeling of cell dynamics is one of the most important open problems in single-cell data science (Lähnemann et al., 2020). Successful modeling of cell dynamics is important for understanding – and eventually intervening in – cellular programs of development and disease (Tong et al., 2023a). 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 & 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). Over longer timescales, this assumption no longer holds; to this end, non-Euclidean costs have been explored in the static context (Huizing et al., 2022; Huguet et al., 2022b), but they are difficult to formulate in the dynamic (SB) framework.

**High-dimensional transport with geodesic costs.** Cells are thought to lie on a low-dimensional manifold in the space of gene expressions (Moon et al., 2018). Recent methods have tried to take advantage of this with density-adhering regularizations (Tong et al., 2020; Koshizuka & Sato, 2023) or embedding into a space where Euclidean distances reflect manifold geodesic distances (Huguet et al., 2022a). Because [SF]$^2$M is able to match a coupling between marginals $q(x_0, x_1)$ defined by entropic OT with an arbitrary cost function, we propose to fit a Schrödinger bridge with marginals that match a geodesic cost.

Specifically, we use the Geodesic Sinkhorn method (Huguet et al., 2022b), which computes the static joint $q(x_0, x_1)$ using (entropic) OT with cost

$$c_{geo}(x_0, x_1) = \sqrt{-\log \mathcal{H}_t(x_0, x_1)}. \quad (16)$$

The matrix $\mathcal{H}_t$ is an approximation of the heat kernel defined via the Laplace-Beltrami operator on the manifold, efficiently approximated using a $k$-nearest-neighbour graph (see Huguet et al. (2022b)). We find using this cost leads to more accurate trajectories in high dimensions.

**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 descent on an energy function. While there have been attempts to approximate this energy function from single-cell data before (Tang, 2017; Qin et al., 2023), we formalize a method to do so from single-cell sequencing data using [SF]$^2$M. To do this, we impose a Langevin dynamics parametrization on the flow and score: $v_\theta(t, x) = -\nabla_x E_v(t, x)$ 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 scale $g(t)$. We visualize these landscapes in Figure 2 with further details in §F.6.

**Learning gene regulatory networks.** Finally, we use [SF]$^2$M 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 & Branson, 2022; Aliee et al., 2022; Atanack-

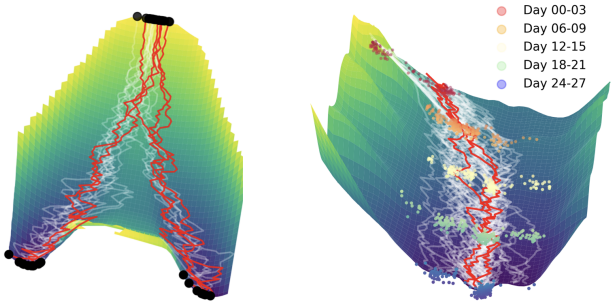


Figure 2: Visualization of learned Waddington's landscape $W$ with a bifurcating trajectory for one Gaussian to two Gaussians (**left**) and for the Embryoid Body (EB) data (**right**) (Moon et al., 2019). Three dimensions are space (left-right), time (forward-back), and potential (up-down).

ovic et al., 2023), we define the *gene regulatory network* in this setting 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., 2023a; Bunne et al., 2022b), which complicated the discovery of the sparse graph structure in gene space. [SF]$^2$M is the first Schrödinger bridge method to scale to high dimensions instead of compressing to low dimensional PCA space. This allows us to learn a dynamics model 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 & 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).

## 4. 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 & Vanden-Eijnden, 2023; Neklyudov et al., 2022; Liu, 2022) and score-based (Sohl-Dickstein et al., 2015; Song & Ermon, 2019; 2020; Song et al., 2021b; Ho et al., 2020; Winkler et al., 2023; Dhariwal & Nichol, 2021; Watson et al., 2022) generative modeling. Both have drawn attention due to their stability and efficiency in training and high quality of generated samples.

Previous studies have focused on simulation-free training of CNFs, including ones with Gaussian source distribution (Rozen et al., 2021; Ben-Hamu et al., 2022; Lipman et al., 2023; Fernandes et al., 2022) and those that use independent samples from $q_0$ and $q_1$ (Albergo & Vanden-Eijnden, 2023; Albergo et al., 2023). Recent research has also investigated the regularization of flows through dynamic OT (Liu et al., 2023b; Pooladian et al., 2023; Shi et al., 2023) and Schrödinger bridges and dynamic OT from unpaired samples (Shi et al., 2022; Wang et al., 2021; Tong et al., 2023b). De Bortoli et al. (2021); Chen et al. (2022); Vargas et al. (2021); Peluchetti (2023) also fit SDEs between unpaired distributions, but require an alternating minimization that fits both forward and backward processes, while our method can directly target the drift and score without iterative optimization in continuous space. Shi et al. (2023); Peluchetti (2023) propose an iterative procedure for training Schrödinger bridge models using similar ideas however they do not make use of fast static optimal transport solvers, and still require simulation during outer loops. We note this outer iterative looping method can also be applied to our method which improves the Schrödinger bridge marginals at the cost of generative performance (Table S1).

**Applications to cell dynamics.** When the observer seeks to recover dynamics from multiple snapshots with scRNA-seq data, the machinery of optimal transport (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 & Sato, 2023) can be used. However, these methods all require simulation during training, which causes issues when scaling to large dimensions. We introduce the first fully simulation-free Schrödinger bridge model for cell dynamics and show it can be applied to high-dimensional data, contrary to previous work.

## 5. Experiments

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

- Minibatch [SF]$^2$M with exact OT minibatches ([SF]$^2$M-Exact), with entropic OT (Sinkhorn) minibatches ([SF]$^2$M-Sink), and with Geodesic OT ([SF]$^2$M-Geo) when applicable.
- A variety of (ODE) flow-based models, including optimal transport conditional flow matching (OT-CFM, Tong et al. (2023b)), 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 the concurrent work on iterated diffusion mixture transport (IDBM, Peluchetti (2023)).

All results are presented as mean $\pm$ standard deviation over five seeds. Additional experimental details can be found in §F.

Table 1: Two-dimensional data: generative modeling performance ($\mathcal{W}_2$) and dynamic OT optimality (NPE) divided into SDE methods (top) and ODE methods (bottom). [SF]$^2$M performs the best on 3 of 4 datasets and is similar to OT-CFM, which is equivalent to [SF]$^2$M as $g(t) \to 0$. *Denotes results from Shi et al. (2023).

| Metric | $\mathcal{W}_2$ (↓) | | | | Normalized Path Energy (↓) | | | |
|---|---|---|---|---|---|---|---|---|
| Dataset | 8gaussians | moon-8gaussians | moons | scurve | 8gaussians | moon-8gaussians | moons | scurve |
| [SF]$^2$M-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 |
| DSBM-IPF (Shi et al., 2023)* | 0.315 ± 0.079 | 0.812 ± 0.092 | 0.140 ± 0.006 | 0.140 ± 0.024 | 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 ± 0.024 | 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., 2023b) | 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., 2023b) | 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., 2023b) | 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 (Tong et al., 2023b) | 0.343 ± 0.058 | — | 0.209 ± 0.055 | 0.198 ± 0.037 | 0.190 ± 0.054 | — | 0.762 ± 0.099 | 0.743 ± 0.116 |

**Low-dimensional data: Conditional generative modeling and optimal transport.** We first evaluate how well various methods approximate dynamic optimal transport in low dimensions. Table 1 summarizes our results, showing that [SF]$^2$M outperforms all methods, both stochastic (top) and deterministic (bottom). We report the 2-Wasserstein distance between the predicted distribution and the target distribution with samples of size 10,000. Following Tong et al. (2023b), 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.

**Schrödinger bridge construction.** Next we evaluate how well various methods can model a Schrödinger bridge between Gaussian distributions. The Gaussian-to-Gaussian Schrödinger bridge has closed-form Gaussian marginals (Mallasto et al., 2022; Bunne et al., 2022a). Following De Bortoli et al. (2021), it is possible to evaluate the quality of empirical marginals with respect to the ground truth by sampling trajectories using Algorithm 2 and calculating the KL divergence between the empirical Gaussian fit to the marginal at various timepoints and and the Gaussian marginal of the ground truth Schrödinger bridge. This evaluation is shown in Table 2 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 (De Bortoli et al., 2021) and DSBM (Shi et al., 2023), which require iterative optimization of forward and backward models. We find that in low dimensions SB-CFM, which is closely related to [SF]$^2$M but fits only the ODE and not the score, performs the best, closely followed by [SF]$^2$M-Exact. In high dimensions, [SF]$^2$M-Exact better matches the target distribution, performing roughly equally well as the algorithm from concurrent work (Shi et al., 2023) and significantly better than the simulation-based DSB method (De Bortoli et al., 2021).

**Single cell dynamics.** To test the ability of all methods to model real cellular dynamics, we evaluate them on three datasets in the setup established by Tong et al. (2020). Given $K$ timepoints, we perform leave-one-out interpolation, interpolating timepoint $k$ using a model trained on timepoints $[0, \ldots, k-1, k+1, \ldots, K]$ (solving a SB problem between every two successive time points, but sharing parameters between the models used on different intervals). We first measure the error on three datasets using the first five whitened principal components (mean zero standard deviation one) in Table 3. We find that [SF]$^2$M performs the best among Schrödinger bridge methods and approximately the same as the recent OT-CFM method (Tong et al., 2023b). As this benchmark may be somewhat saturated, we also consider higher-dimensional PC (50 and 100 dimensions, Table 4), as well as the 1000 dimensions corresponding to the most highly variable genes (Wolf et al., 2018). In higher dimensions, the advantage of geodesic interpolation becomes apparent with [SF]$^2$M-Geo outperforming all other methods.

**Gene regulatory network recovery.** Here 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 §3.3. 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 5. 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 with no noise (collapsing to OT-CFM (Tong et al., 2023b)) 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 work quite as well, but still outperform the baselines on most metrics.

Table 2: Gaussian-to-Gaussian Schrödinger bridges with 10,000 datapoints between a Gaussian with parameters estimated from empirical samples ($p_t$) with error to the continuous Schrödinger bridge marginals ($q_t$) either at the target distribution (left) or averaged across 21 timepoints (right).

| Metric | $\mathrm{KL}(p_1, q_1)$ | | | | Mean $\mathrm{KL}(p_t, q_t)$ | | | |
|---|---|---|---|---|---|---|---|---|
| dim | 2 | 5 | 20 | 50 | 2 | 5 | 20 | 50 |
| $[\mathrm{SF}]^2$M-Exact | $0.003 \pm 0.000$ | $0.007 \pm 0.000$ | $\mathbf{0.029 \pm 0.002}$ | $\mathbf{0.124 \pm 0.003}$ | $0.004 \pm 0.000$ | $0.006 \pm 0.000$ | $\mathbf{0.028 \pm 0.001}$ | $0.258 \pm 0.001$ |
| DSBM-IPF (Shi et al., 2023) | $0.006 \pm 0.003$ | $0.015 \pm 0.005$ | $0.132 \pm 0.004$ | $0.528 \pm 0.013$ | $\mathbf{0.002 \pm 0.001}$ | $\mathbf{0.005 \pm 0.002}$ | $0.050 \pm 0.002$ | $\mathbf{0.221 \pm 0.004}$ |
| DSB (De Bortoli et al., 2021) | $2.783 \pm$ — | $8.757 \pm$ — | $49.963 \pm$ — | $221.213 \pm$ — | $2.783 \pm$ — | $8.757 \pm$ — | $49.963 \pm$ — | $221.213 \pm$ — |
| SB-CFM (Tong et al., 2023b) | $\mathbf{0.001 \pm 0.000}$ | $\mathbf{0.001 \pm 0.000}$ | $0.034 \pm 0.003$ | $0.170 \pm 0.002$ | $0.003 \pm 0.000$ | $0.008 \pm 0.000$ | $0.086 \pm 0.002$ | $0.447 \pm 0.003$ |

Table 3: 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.

| | Cite | EB | Multi |
|---|---|---|---|
| $[\mathrm{SF}]^2$M-Geo | $1.017 \pm 0.104$ | $0.879 \pm 0.148$ | $1.255 \pm 0.179$ |
| $[\mathrm{SF}]^2$M-Exact | *0.920 ± 0.049* | *0.793 ± 0.066* | $\mathbf{0.933 \pm 0.054}$ |
| $[\mathrm{SF}]^2$M-Sink | $1.054 \pm 0.087$ | $1.198 \pm 0.342$ | $1.098 \pm 0.308$ |
| DSBM (Shi et al., 2023) | $1.705 \pm 0.160$ | $1.775 \pm 0.429$ | $1.873 \pm 0.631$ |
| DSB (De Bortoli et al., 2021) | $0.953 \pm 0.140$ | $0.862 \pm 0.023$ | $1.079 \pm 0.117$ |
| OT-CFM (Tong et al., 2023b) | $\mathbf{0.882 \pm 0.058}$ | $\mathbf{0.790 \pm 0.068}$ | *0.937 ± 0.054* |
| I-CFM (Tong et al., 2023b) | $0.965 \pm 0.111$ | $0.872 \pm 0.087$ | $1.085 \pm 0.099$ |
| SB-CFM (Tong et al., 2023b) | $1.067 \pm 0.107$ | $1.221 \pm 0.380$ | $1.129 \pm 0.363$ |
| Reg. CNF (Finlay et al., 2020)* | — | $0.825 \pm$ — | — |
| T. Net (Tong et al., 2020)* | — | $0.848 \pm$ — | — |

Table 4: 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 | | dim = 100 | | dim = 1000 | |
|---|---|---|---|---|---|---|
| | Cite | Multi | Cite | Multi | Cite | Multi |
| $[\mathrm{SF}]^2$M-Geo | $\mathbf{38.524 \pm 0.293}$ | $\mathbf{44.795 \pm 1.911}$ | $\mathbf{44.498 \pm 0.416}$ | $\mathbf{52.203 \pm 1.957}$ | $\mathbf{40.092 \pm 1.529}$ | $\mathbf{51.292 \pm 0.090}$ |
| $[\mathrm{SF}]^2$M-Exact | $40.009 \pm 0.783$ | $45.337 \pm 2.833$ | $46.530 \pm 0.426$ | $52.888 \pm 1.986$ | $43.656 \pm 0.715$ | $53.149 \pm 1.859$ |
| DSBM (Shi et al., 2023) | $53.811 \pm 7.736$ | $66.427 \pm 14.39$ | $58.994 \pm 7.623$ | $70.751 \pm 14.03$ | $50.085 \pm 4.809$ | $61.708 \pm 13.90$ |
| OT-CFM (Tong et al., 2023b) | $38.756 \pm 0.398$ | $47.576 \pm 6.622$ | $45.393 \pm 0.416$ | $54.814 \pm 5.858$ | $43.246 \pm 0.728$ | $52.289 \pm 1.553$ |
| I-CFM (Tong et al., 2023b) | $41.834 \pm 3.284$ | $49.779 \pm 4.430$ | $48.276 \pm 3.281$ | $57.262 \pm 3.855$ | $44.117 \pm 0.523$ | $52.991 \pm 1.502$ |

Table 5: 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).

| | Bifurcating System | | Trifurcating System | |
|---|---|---|---|---|
| | AUC-ROC (↑) | AP (↑) | AUC-ROC (↑) | AP (↑) |
| NGM-$[\mathrm{SF}]^2\mathrm{M}_{\sigma=0}$ | $\mathbf{0.786 \pm 0.081}$ | $\mathbf{0.521 \pm 0.160}$ | $\mathbf{0.764 \pm 0.066}$ | $\mathbf{0.485 \pm 0.105}$ |
| NGM-$[\mathrm{SF}]^2\mathrm{M}_{\sigma=0.1}$ | $0.723 \pm 0.014$ | $0.444 \pm 0.030$ | $0.731 \pm 0.077$ | $0.453 \pm 0.091$ |
| Spearman | $0.755 \pm 0.003$ | $0.438 \pm 0.002$ | $0.718 \pm 0.005$ | $0.413 \pm 0.005$ |
| Pearson | $0.744 \pm 0.000$ | $0.415 \pm 0.000$ | $0.710 \pm 0.002$ | $0.405 \pm 0.002$ |
| DREMI (Krishnaswamy et al., 2014) | $0.594 \pm 0.017$ | $0.293 \pm 0.011$ | $0.419 \pm 0.021$ | $0.205 \pm 0.007$ |
| Granger (Granger, 1969) | $0.664 \pm 0.013$ | $0.421 \pm 0.043$ | $0.613 \pm 0.048$ | $0.343 \pm 0.039$ |

## 6. 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 this can be applied to learn cell dynamics to extract the gene regulatory structure from snapshot data.

One limitation of $[\mathrm{SF}]^2$M is that it requires closed-form conditional flows. Closed-form conditional flows are not available if one considers Schrödinger bridges with more general reference processes (Fernandes et al., 2022), which may be useful to encode further biological priors (Koshizuka & Sato, 2023). Future work can consider how to train $[\mathrm{SF}]^2$M-like models with interventional data, which can substantially improve gene regulatory network inference.

## Acknowledgments

We would like to thank Hananeh Aliee, Paul Bertin, Valentin de Bortoli, Stefano Massaroli, and Austin Stromme for productive conversations. The authors acknowledge funding from CIFAR, Genentech, Samsung, IBM, Microsoft, and Google. In addition, K.F. acknowledges funding from NSERC (RGPIN-2019-06512) and G.W. acknowledges funding from NSERC Discovery grant 03267 and NIH grant R01GM135929.

## References

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

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

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

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

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

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

Atanackovic, L., Tong, A., Hartford, J., Lee, L. J., Wang, B., and Bengio, Y. DynGFN: Bayesian dynamic causal discovery using generative flow networks. *arXiv preprint 2302.04178*, 2023.

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

Ben-Hamu, H., Cohen, S., Bose, J., Amos, B., Grover, A., Nickel, M., Chen, R. T. Q., and Lipman, Y. Matching normalizing flows and probability paths on manifolds. *International Conference on Machine Learning (ICML)*, 2022.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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. POT: Python Optimal Transport. *Journal of Machine Learning Research (JMLR)*, 22, 2021.

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

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

Genevay, A., Chizat, L., Bach, F., Cuturi, M., and Peyré, G. Sample complexity of Sinkhorn divergences. *Artificial Intelligence and Statistics (AISTATS)*, 2019.

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

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

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

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

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

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

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

Huizing, G.-J., Cantini, L., and Peyré, G. Unsupervised ground metric learning using Wasserstein eigenvectors. *International Conference on Machine Learning (ICML)*, 2022.

Kester, L. and van Oudenaarden, A. Single-cell transcriptomics meets lineage tracing. *Cell Stem Cell*, 23 (2):166–179, August 2018. ISSN 19345909. doi: 10.1016/j.stem.2018.04.014.

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

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. Mapping cells through time and space with moscot. *bioRxiv preprint 2023.05.11.540374*, 2023.

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

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

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. Eleven grand challenges in single-cell data science. *Genome Biology*, 21(1):31, 2020. ISSN 1474-760X. doi: 10.1186/s13059-020-1926-6.

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

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

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

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

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

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

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

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

Mallasto, A., Gerolin, A., and Ha Quang, M. Entropy-regularized 2-Wasserstein distance between Gaussian measures. *Information Geometry*, 5, 07 2022.

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

Moon, K. R., Stanley, J. S., Burkhardt, D., van Dijk, D., Wolf, G., and Krishnaswamy, S. Manifold learning-based methods for analyzing single-cell rna-sequencing data. *Current Opinion in Systems Biology*, 7:36–46, 2018. ISSN 24523100. doi: 10.1016/j.coisb.2017.12.008.

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. Visualizing structure and transitions in high-dimensional biological data. *Nature Biotechnology*, 37(12):1482–1492, 2019.

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

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

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. Brassinosteroid gene regulatory networks at cellular resolution in the arabidopsis root. *Science*, 379 (6639):eadf4721, 2023.

Peluchetti, S. Diffusion bridge mixture transports, Schrödinger bridge problems and generative modeling. *arXiv preprint 2304.00917*, 2023.

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

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

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

Rozen, N., Grover, A., Nickel, M., and Lipman, Y. Moser flow: Divergence-based generative modeling on manifolds. *Neural Information Processing Systems (NeurIPS)*, 2021.

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

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

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. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, 176(4):928–943.e22, 2019.

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

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

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

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

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

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

Somnath, V. R., Pariset, M., Hsieh, Y.-P., Martinez, M. R., Krause, A., and Bunne, C. Aligned diffusion Schrödinger bridges. *arXiv preprint 2302.11419*, 2023.

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

Song, K.-U. Applying regularized Schrödinger-bridge-based stochastic process in generative modeling. *arXiv preprint 2208.07131*, 2022.

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

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

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

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

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

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

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

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

Tong, A., Malkin, N., Huguet, G., Zhang, Y., Rector-Brooks, J., Fatras, K., Wolf, G., and Bengio, Y. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint 2302.00482*, 2023b.

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

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

Wagner, D. E. and Klein, A. M. Lineage tracing meets single-cell omics: Opportunities and challenges. *Nature Reviews Genetics*, 21(7):410–427, July 2020. ISSN 1471-0056, 1471-0064. doi: 10.1038/s41576-020-0223-2.

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

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. Broadly applicable and accurate protein design by integrating structure prediction networks and diffusion generative models. *bioRxiv preprint 2022.12.09.519842*, 2022.

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

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

Yang, K. D., Damodaran, K., Venkatachalapathy, S., Soylemezoglu, A. C., Shivashankar, G. V., and Uhler, C. Predicting cell lineages using autoencoders and optimal transport. *PLOS Computational Biology*, 16:1–20, 04 2020. URL https://doi.org/10.1371/journal.pcbi.1007828.

# Appendix

## A. Proofs of theorems

**Theorem 3.1.** *Under mild regularity conditions, the ODE $dx = u_t^\circ(x)\,dt$ generates the probability path (12) from initial conditions $p_0(x)$, and the score of $p_t$ is given by (13).*

*Proof of Theorem 3.1.* The first statement is Theorem 3.1 from Tong et al. (2023b), 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 (13), 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. □

**Theorem 3.2.** *If $p_t(x) > 0$ for all $x \in \mathbb{R}^d$ and $t \in [0,1]$, then $\mathcal{L}_{\text{[SF]}^2\text{M}}(\theta)$ and $\mathcal{L}_{\text{U[SF]}^2\text{M}}(\theta)$ are equal up to a constant, and $\nabla_\theta\mathcal{L}_{\text{U[SF]}^2\text{M}}(\theta) = \nabla_\theta\mathcal{L}_{\text{[SF]}^2\text{M}}(\theta)$, where $\mathcal{L}_{\text{U[SF]}^2\text{M}}(\theta)$ is the unconditional score and flow matching loss (6).*

*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. (2023b).)

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

$$\nabla_\theta\mathbb{E}_{z\sim q(z),x\sim p_t(x|z)}\left[\|w_\theta(t,x) - w_t^\circ(x|z)\|^2\right] = \nabla_\theta\mathbb{E}_{x\sim p_t(x)}\left[\|w_\theta(t,x) - w_t^\circ(x)\|^2\right]. \tag{17}$$

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 $\mathbb{E}_{z\sim q(z),x\sim p_t(x|z)}$.

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

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

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

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 $z$.) □

**Theorem 3.3** ([SF]$^2$M solves the generative modeling problem)**.** *If $q(x_0, x_1) \in U(q_0, q_1)$ and $v_\theta^*, s_\theta^*$ globally minimize $\mathcal{L}_{[\mathrm{SF}]^2\mathrm{M}}(\theta)$ then if $p_t$ is the marginal probability induced by an SDE with drift $[v_\theta^* + \frac{1}{2} g(t)^2 s_\theta^*]$ and diffusion $g$, and initial conditions $p_0$, then $p_t(x) = q_t(x)$ for all $t$ and $x$.*

*Proof of Theorem 3.3.* By Theorem 3.2, assuming sufficient regularity, optimization of $\mathcal{L}_{[\mathrm{SF}]^2\mathrm{M}}(\theta)$ is equivalent to optimization of $\mathcal{L}_{\mathrm{U}[\mathrm{SF}]^2\mathrm{M}}(\theta)$, which is globally miminized 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 $p_0$. □

*Proof of Proposition 3.4.* This is an immediate consequence of Theorem 3.3, which shows that the SDE learned by [SF]$^2$M generates the desired marginals at all $t$, and of (11), which characterizes the marginal of the Schrödinger bridge as that of a mixture of Brownian bridges with mixture weights given by the entropic OT plan. □

## B. 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**
  $u_t \leftarrow v_\theta(t, x_t) + \frac{g(t)^2}{2} s_\theta(t, x_t)$
  $x_{t+\Delta t} \sim \mathcal{N}(x + u_t \Delta t, g(t)^2 \Delta t)$
**return** Samples $x_1$

### B.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).

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 B.1** (minibatch transport plan (Fatras et al., 2020))**.** Consider $\alpha_n$ and $\beta_n$ two empirical probability distributions. For each $A = \{a_1, \dots, a_m\} \in \mathcal{P}_m(\alpha_n)$ and $B = \{b_1, \dots, b_m\} \in \mathcal{P}_m(\beta_n)$ 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*:

$$\Pi^m(\alpha_n, \beta_n) = \binom{n}{m}^{-2} \sum_{A \in \mathcal{P}_m(\alpha_n)} \sum_{B \in \mathcal{P}_m(\beta_n)} \Pi_{A,B}. \tag{18}$$

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

$$\Pi^k(\alpha_n, \beta_n) := k^{-1} \sum_{(A,B) \in D_k} \Pi_{A,B} \tag{19}$$

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

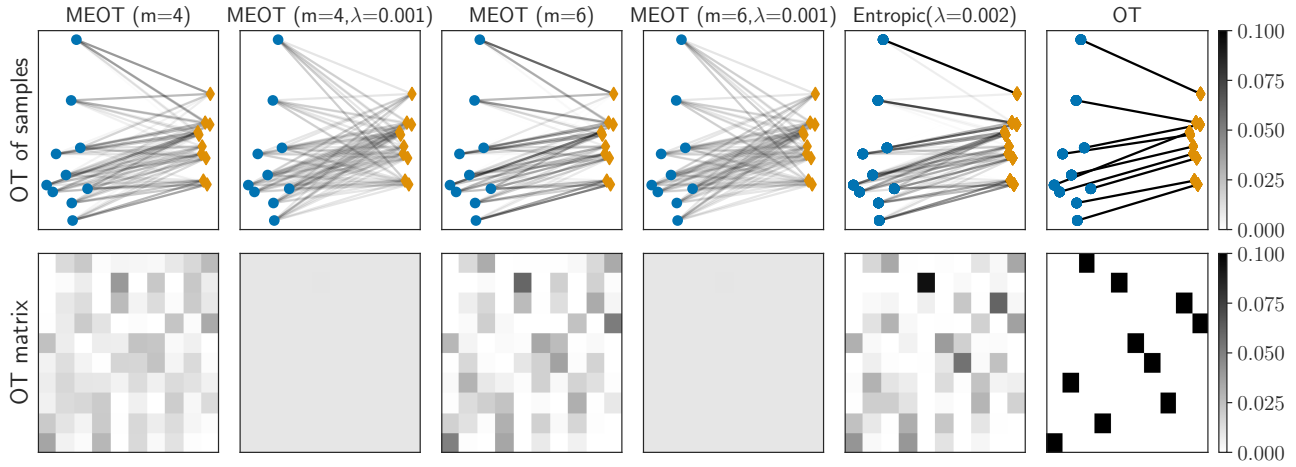


Figure S1: Optimal transport couplings for different OT costs and batch sizes on a 2D example. The top row represent the OT matching between samples while the bottom row represent the minibatch OT plan. We can see that coupling entropic OT with minibatches lead to a uniform plan contrary to using only entropic regularization or minibatch approximation.

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 Figure S1 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 nonoptimal connections with OT variants (Fatras et al., 2021a) it would have added extra hyperparameters that would have led to more compute.

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

## D. Further discussion of related work

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

### D.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. (2023b); Pooladian et al. (2023), trains with the *conditional flow matching* loss,

$$\mathcal{L}_{CFM}(\theta) = \mathbb{E}_{t\sim\mathcal{U}(0,1), z\sim q(z), x\sim p_t(x|z)} \|v_\theta(t,x) - u_t^\circ(x|z)\|^2. \tag{20}$$

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

Table S1: 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]$^2$M-Sinkhorn | [SF]$^2$M-Exact | [SF]$^2$M-Exact-Looped |
|---|---|---|---|---|
| $\mathrm{KL}(p_1, q_1)$ | 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 $\mathrm{KL}(p_t, q_t)$ | 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** |

et al., 2023), action matching (Neklyudov et al., 2022), stochastic interpolants (Albergo & Vanden-Eijnden, 2023; Albergo et al., 2023), the 1-rectified flow (Liu, 2022), optimal transport conditional flow matching (OT-CFM) (Tong et al., 2023b) 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., 2023b).

**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., 2023b; Pooladian et al., 2023). However, recent (Liu et al., 2023a) and concurrent work (Shi et al., 2023) has noted advantages of stochastic dynamics which we also observe (see Table 1 and Table 2), 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 & van Oudenaarden, 2018; Wagner & Klein, 2020; Klein et al., 2023), these are biologically complex, and we leave their analysis to future work.

### D.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. Concurrent work extends this to markovian bridges (Shi et al., 2023; Peluchetti, 2023), which greatly improves performance.

Often a consistency loss is added to make sure the forward and backwards drift are consistent which increases stability of the training procedure (Song, 2022). The consistency loss can be seen as making sure the forward and backwards flows match the (weighted) score function.

$$\mathcal{L}_{cons} = \mathbb{E}_{t\sim\mathcal{U}(0,1),z\sim q(z),x\sim p_t(x|z)}\left[\|v_\theta^f(t,x) + v_\theta^b(t,x) - g(t)^2\nabla\log p_t(x|z)\|^2\right]/g(t)^2 \tag{21}$$

which should be zero when the forward and backward drifts are consistent. [SF]$^2$M directly parameterizes the flow and score, and recovers the forward and backwards bridges.

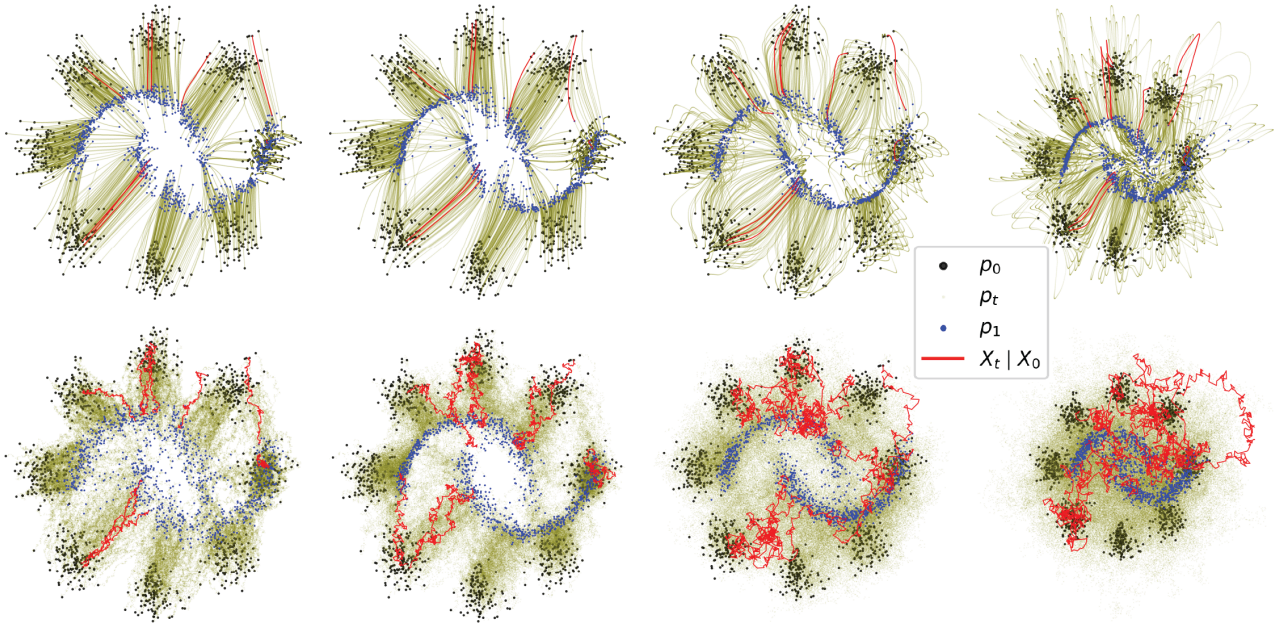


Figure S2: Learned ODE (top) and SDE (bottom) simulations for $\sigma \in [0.1, 1, 2, 3]$ from left to right for trained [SF]$^2$M model. The marginals match regardless of the chosen $\sigma$. Trajectory initializations are matched between runs.

## E. Additional results and ablations

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

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]$^2$M 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]$^2$M is compatible with outer looping. In Table S1 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]$^2$M effectively takes a single outer loop, and may have advantages even over a single outer loop as shown in Tong et al. (2023b); 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 Algorithm 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.

### E.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 S1. 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.

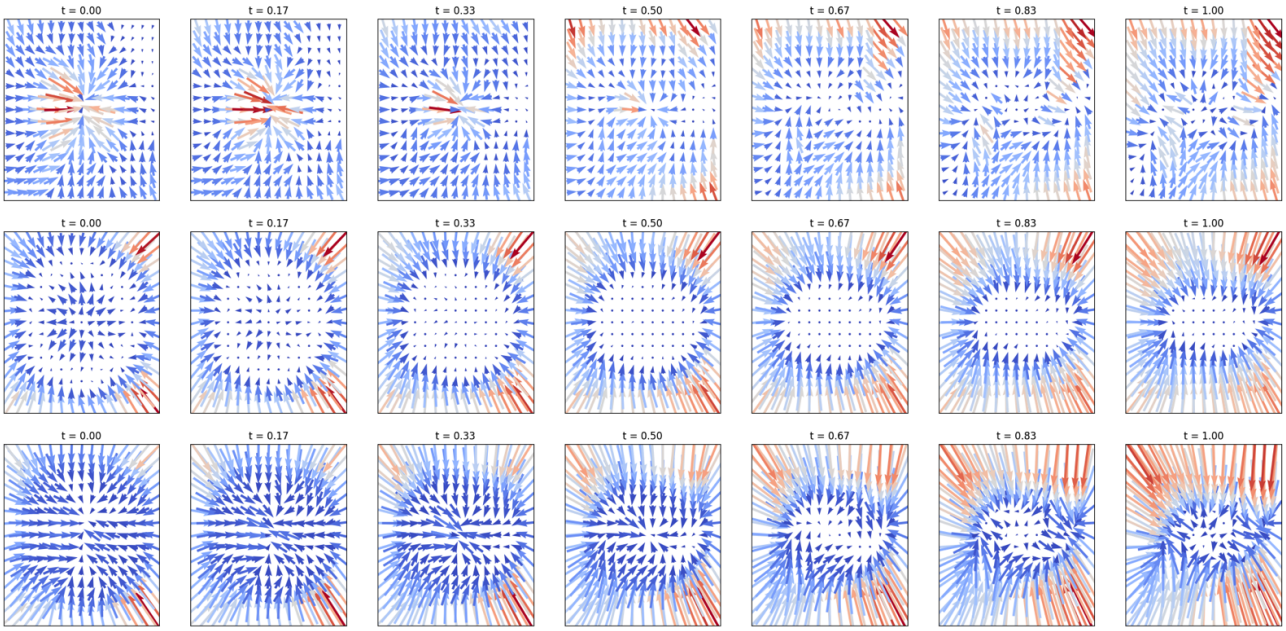


Figure S3: From top to bottom $v_\theta(t, x)$, $s_\theta(t, x)$ and $u_t(x)$ inferred through eq. 4 for the 8-Gaussians to moons dataset.

Table S2: 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]$^2$M | 0.791 ± 0.097 | 0.056 ± 0.005 | — | — | 0.932 ± 0.159 | **0.062 ± 0.006** | — | — |
| NGM-[SF]$^2$M$_{\sigma=0}$ | 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]$^2$M$_{\sigma=0.1}$ | 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]$^2$M$_{\sigma=0.01}$ | 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]$^2$M$_{\sigma=0.001}$ | 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 |

# F. Experimental details

## 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 & 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 Algorithm 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 [SF]$^2$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

**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 $l \in [1, \dots, L]$ **do**
  **for** Inner loop $i \in [1, \dots, I]$ **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 §B.1*
      $x_0, x_1 \sim \pi^{\otimes m}$ ▷ *Resample OT pairs from* $\pi$
    **else**
      $x_0, x_1 \sim \mathbb{T}^{\otimes m}$
    $t \sim \mathcal{U}(0, 1)$
    $p_t(x|x_0, x_1) \leftarrow \mathcal{N}(x; tx_1 + (1-t)x_0, \sigma^2 t(1-t))$
    $x \sim p_t(x|x_0, x_1)$
    $\mathcal{L}_{[\text{SF}]^2\text{M}} \leftarrow \|v_\theta(t, x) - u_t(x|x_0, x_1)\|^2 + \lambda(t)^2 \|s_\theta(t, x) - \nabla_x \log p_t(x|x_0, x_1)\|^2$
    $\theta \leftarrow \text{Update}(\theta, \nabla_\theta \mathcal{L}_{[\text{SF}]^2\text{M}})$
  $\mathbb{T}_f \leftarrow (x_0, \hat{x}_1)^{\otimes n//2}$ ▷ *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** $v_\theta, s_\theta$

**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**
  $t \sim \mathcal{U}(0, 1)$
  $x_0, x_1 \sim q_0^{\otimes m}, q_1^{\otimes m}$
  $\pi \leftarrow \text{Sinkhorn}(x_0, x_1, 2\sigma^2)$ ▷ *Or* $\text{OT}(x_0, x_1)$ *see §B.1*
  $x_0, x_1 \sim \pi^{\otimes m}$ ▷ *Resample OT pairs from* $\pi$
  $p_t(x|x_0, x_1) \leftarrow \mathcal{N}(x; tx_1 + (1-t)x_0, \sigma^2 t(1-t))$
  $x \sim p_t(x|x_0, x_1)$
  $\mathcal{L}_{[\text{SF}]^2\text{M}} \leftarrow \|v_\theta(t, x) - u_t(x|x_0, x_1)\|^2 + \lambda(t)^2 \|s_\theta(t, x) - \nabla_x \log p_t(x|x_0, x_1)\|^2$ ▷ *see 8, 14*
  $\theta \leftarrow \text{Update}(\theta, \nabla_\theta \mathcal{L}_{[\text{SF}]^2\text{M}})$
**return** $v_\theta, s_\theta$

$\lambda(t)$ 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:

$$\lambda(t) = \sigma_t = \sigma\sqrt{t(1-t)} \tag{22}$$

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

$$\lambda(t) = \frac{2}{\sigma^2}\sigma_t = \frac{2\sqrt{t(1-t)}}{\sigma} \tag{23}$$

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:

$$\lambda(t)^2 \left\| s_\theta(t,x) - \nabla_x \log p_t(x|x_0,x_1) \right\|^2 = \left\| \lambda(t) s_\theta(t,x) - \lambda(t)\nabla_x \log p_t(x|x_0,x_1) \right\|^2 \tag{24}$$

$$= \left\| \lambda(t) s_\theta(t,x) + \epsilon_t \right\|^2 \tag{25}$$

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 eq. 23 and the simplfied objective in eq. 25.

#### F.1.2. Static optimal transport

As discussed in §B.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. (2023b) 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

$$\mathcal{W}_2 = \left( \min_{\pi \in U(\hat{p}_1, q_1)} \int \|x-y\|_2^2 d\pi(x,y) \right)^{1/2} \tag{26}$$

where $\hat{p}_1$ is sampled via eq. 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 eq. 3). 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

$$NPE(q_0, q_1, v_\theta) = \frac{\left| \mathbb{E}_{x(0)\sim q_0} \int \|v_\theta(t,x_t)\|^2 dt - \mathcal{W}_2^2(q_0,q_1) \right|}{\mathcal{W}_2^2} \tag{27}$$

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

$$q_t(x) = \mathcal{N}\left(x; 0.2t - 0.1, (t(1-t)\sqrt{4+\sigma^4} + (1-t)^2 + t^2)I\right) \tag{28}$$

with the KL divergence.

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

$$\overline{\mathrm{KL}} = \frac{1}{20}\sum_{k=0}^{20} \mathrm{KL}(\tilde{p}_{k/20}||q_{k/20}) \tag{29}$$

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.

$$\mathrm{KL}(\tilde{p}_1||q_1) \tag{30}$$

### 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 Algorithm 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 3, we truncate to the first five dimensions, then whiten each dimension following (Tong et al.,

**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 §B.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** $v_\theta, s_\theta$

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 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.[2] 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 4, 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 & Branson (2022) we can define:

$$v_{\theta_j}(t, x) = \phi(\cdots \phi(\phi(x\theta_j^{(1)})\theta_j^{(2)})\cdots)\theta_j^{(K)}, \qquad j = 1, \ldots, d, \tag{31}$$

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 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 eq. 31.

Using BoolODE (Pratapa et al., 2020), we generate simulated single-cell gene expression trajectories for a bifurcating

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

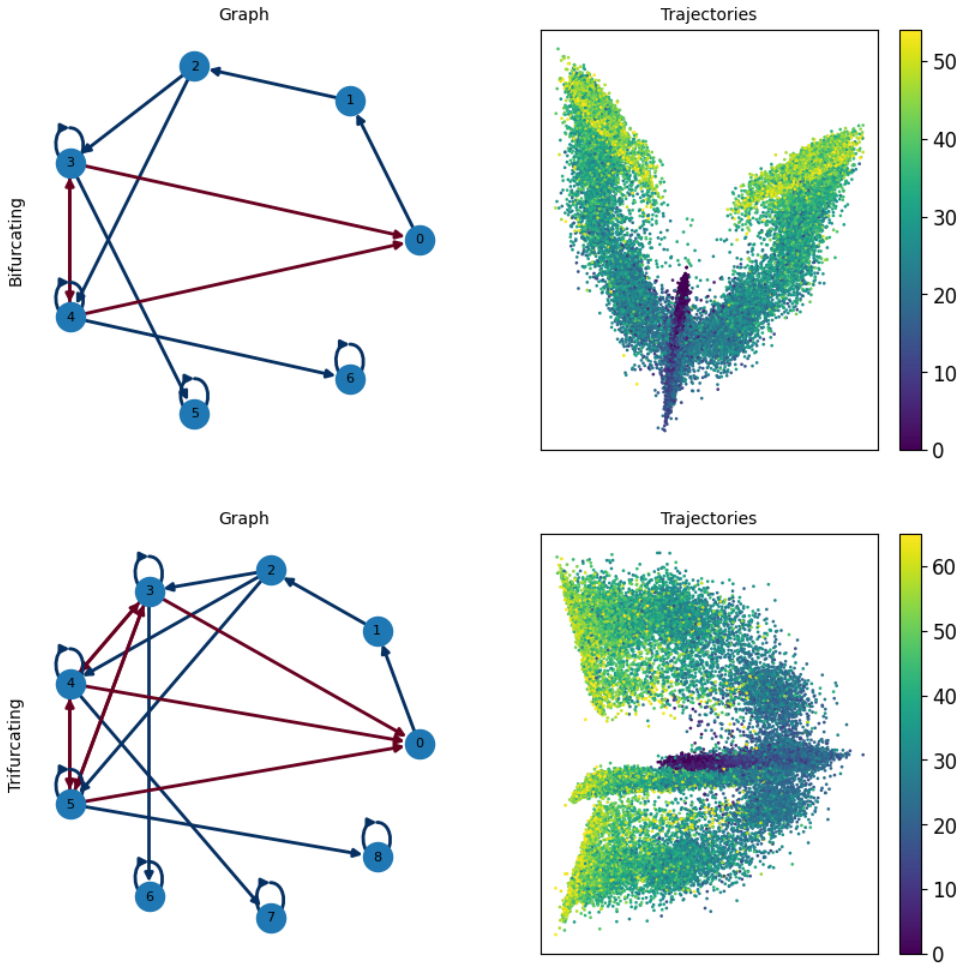


Figure S4: Simulated single-cell trajectories given synthetic GRNs emulating bifurcating (top) and trifurcating (bottom) systems. GRNs contain directed edges of Boolean relationships between genes. For example, a red edge between gene 3 and gene 0 (top left) indicates that the rule for gene 0 is (*not* 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.

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. 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 Figure S4.

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 S2.