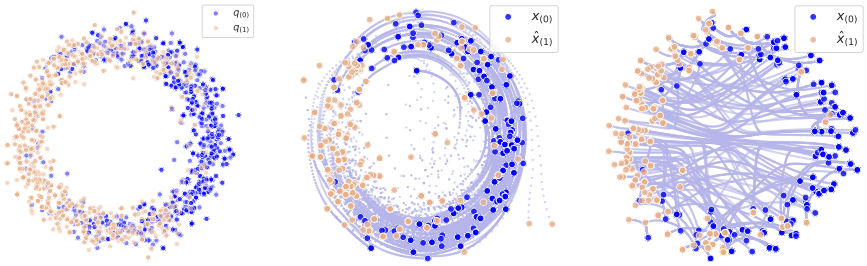
Curly Flow Matching for Learning Non-gradient Field Dynamics

Katarina Petrović[1]*, **Lazar Atanackovic**[2,3,4], **Viggo Moro**[1], **Kacper Kapuśniak**[1], **İsmail İlkan Ceylan**[5,6,1], **Michael Bronstein**[1,6], **Avishek Joey Bose**[1,7]†, **Alexander Tong**[6,7,8]†
[1]University of Oxford, [2]Broad Institute of MIT and Harvard, [3]University of Toronto, [4]Vector Institute, [5]TU Wien, [6]AITHYRA, [7]Mila – Quebec AI Institute, [8]Université de Montréal

Abstract

Modeling the transport dynamics of natural processes from population-level observations is a ubiquitous problem in the natural sciences. Such models rely on key assumptions about the underlying process in order to enable faithful learning of governing dynamics that mimic the actual system behavior. The de facto assumption in current approaches relies on the principle of least action that results in gradient field dynamics and leads to trajectories minimizing an energy functional between two probability measures. However, many real-world systems, such as cell cycles in single-cell RNA, are known to exhibit non-gradient, periodic behavior, which *fundamentally* cannot be captured by current state-of-the-art methods such as flow and bridge matching. In this paper, we introduce CURLY FLOW MATCHING (CURLY-FM), a novel approach that is capable of learning non-gradient field dynamics by designing and solving a Schrödinger bridge problem with a non-zero drift reference process—in stark contrast to typical zero-drift reference processes—which is constructed using *inferred velocities* in addition to population snapshot data. We showcase CURLY-FM by solving the trajectory inference problems for single cells, computational fluid dynamics, and ocean currents with approximate velocities. We demonstrate that CURLY-FM can learn trajectories that better match both the reference process and population marginals. CURLY-FM expands flow matching models beyond the modeling of populations and towards the modeling of known periodic behavior in physical systems. Our code repository is accessible at: https://github.com/kpetrovicc/curly-flow-matching.git.



(a) Asymmetric `circles` (b) CURLY-FM (c) CFM

Figure 1: Particle trajectories generated between samples drawn from asymmetric `circles` distribution at $t = 0$ and $t = 1$ and respective to underlying reference velocity field $f_t(x_t)$. Traditional flow-based models such as OT-CFM and CFM *cannot* capture cyclical patterns in physical systems. CURLY-FM is capable of *learning non-gradient field dynamics* behavior in the underlying data.

*Correspondence to: {katarina.petrovic}@cs.ox.ac.uk
†Equal advising

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

 https://doi.org/10.52202/085713-4053

1 Introduction

Understanding the temporal evolution of multi-body systems remains a central challenge across many applications in life sciences [Lange et al., 2024, Pan and Zhang, 2024], as such systems are often characterized by complex dynamics governing their evolutionary behavior. For example, biological tissues are complex systems that evolve through tissue differentiation characterized by cell divisions and deaths as well as morphological changes. Learning evolutionary behavior in such systems can be formulated as a *trajectory inference* problem [Hashimoto et al., 2016, Lavenant et al., 2024], where the goal is to recover full trajectories of particles given partial and noisy population-level snapshots.

The dominant paradigm in solving the trajectory inference problems in scientific applications involves leveraging tools from computational optimal transport (OT) [Peyré and Cuturi, 2019] to learn neural dynamical systems, e.g. NeuralODE [Chen et al., 2018], such that sampled trajectories under the model optimize a notion of likeliness of being observed [Bunne et al., 2024b]. For instance, in single-cell trajectory inference, such methods broadly follow a pipeline that first infers "optimal" cell trajectories that follow the gradient of some potential function (often termed a Waddington Landscape [Waddington, 1942]), before searching for important regulators of a biological process in development [Schiebinger et al., 2019, Shahan et al., 2022] or disease [Tong et al., 2023, Klein et al., 2025]. Despite the ability to produce approximately optimal trajectories w.r.t. the energy landscape, current methods are limited in their ability to model only gradient-field dynamics. Consequently, trajectories inferred under the model are not realistic and fail to model crucial system dynamics such as periodic behavior that arises in natural systems, e.g., cell cycling [Riba et al., 2022] where periodic behavior is observed at the scale of months, days, hours, and minutes. These behaviors cannot be addressed by current OT-based methods, as gradient field dynamics cannot capture periodic behavior.

Present work. In this paper, we tackle the modeling of systems governed by non-gradient field dynamics. We introduce CURLY FLOW MATCHING (CURLY-FM), a novel approach for learning non-gradient field dynamics by solving a Schrödinger bridge problem with a designed reference process that induces the learning of periodic behavior. Specifically, we consider reference processes with non-zero drift—in stark contrast to zero drift processes in approaches such as Diffusion Schrödinger Bridges [De Bortoli et al., 2021b, Shi et al., 2024]. Such a modification elevates the established (entropic optimal) mass transport problem to a new class of problems that require matching the reference drift while also transporting mass between time marginals associated with observations. As a result, solutions to this Schrödinger bridge problem are capable of learning non-gradient field dynamics and exhibit behaviors such as periodicity as found in cell cycling.

In addition to conventional population snapshot data, we design CURLY-FM by leveraging approximate velocity information that is used to construct the drift of a reference process. Consequently, to model periodic dynamics CURLY-FM solves the Schrödinger bridge problem by decomposing it into a two-stage algorithm. The first stage learns a neural path interpolant by regressing against the drift of our constructed reference process. Unlike straight paths in optimal transform conditional flow matching trajectories, the neural path interpolant exhibits cyclic behavior due to the optimization objective of matching a constructed reference drift. In stage two, we learn, in a simulation-free manner, to construct the generative process that solves the mass transport problem as a mixture of conditional bridges built using optimal transport-based couplings that minimize the length of the velocity field of the neural path interpolant. The combination of our two-stage approach enables CURLY-FM to learn dynamics that *do not* affect the population data, but do affect individual particles, which include observed periodic non-gradient field dynamics, unlike other methods (see table 1).

We instantiate CURLY-FM on modeling a suite of problems in natural sciences that exhibit known non-gradient field dynamics, such as cell cycle systems found in scRNA-seq data with RNA-velocity [La Manno et al., 2018], ocean currents, and computational fluid dynamics for PDEs simulated using the Lagrangian particle discretizations [Toshev et al., 2023]. In each case, the system under consideration is under the influence of a *known* reference process, e.g., RNA-velocity gives an approximation of the instantaneous velocity of a cell, which must be adhered to when solving the trajectory inference problem. More precisely, using the reference process drift and population snapshots CURLY-FM solves the Schrödinger bridge problem by searching for a bridge that matches the reference process as closely as possible, but also matches the marginal distributions at both timepoints of the system dynamics. Consequently, we show that previous flow matching approaches fail to model this cyclic behavior as they are not able to take advantage of the additional reference process information. Furthermore, while there exist simulation-based methods that are in principle able to learn the correct dynamics [Tong et al., 2020], we show that in practice CURLY-FM performs

significantly better both in terms of accuracy to match the reference drift—enabling modeling of cyclic behavior (c.f. fig. 1)—while being computationally cheaper due to its simulation-free training nature.

2 Background and preliminaries

Given two distributions ρ_0 and ρ_1, the *distributional matching problem* seeks to find a push-forward map $\psi : \mathbb{R}^d \to \mathbb{R}^d$ that transports the initial distribution to the desired endpoint, $\rho_1 = [\psi]_\#(\rho_0)$. Such a problem setup is pervasive in many areas of machine learning and notably encompasses the standard generative modeling and optimal transport settings [De Bortoli et al., 2021a, Peyré et al., 2019]. In this paper, we consider the setting where each distribution ρ_0 and ρ_1 is an empirical distribution that is accessible through a dataset of observations $\{x_0^i\}_{i=1}^N \sim \rho_0(x_0)$ and $\{x_1^j\}_{j=1}^N \sim \rho_1(x_1)$. Thus the modeling task is to learn the (approximate) transport map ψ.

2.1 Continuous normalizing flows and flow matching

One common choice for modeling ψ is as a deterministic dynamic system with a time-dependent generator $\psi_t : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$. The solution to this dynamical system is an ordinary differential equation (ODE) and the learned transport map is known as a *continuous normalizing flow* (CNF). A CNF is a time-indexed neural transport map ψ_t, for all time $t \in [0,1]$, that is trained to push forward samples from prior μ_0 to a desired target μ_1. Specifically, a CNF models the ODE $\frac{d}{dt}\psi_t(x) = f_t\left(\psi_t(x)\right)$ with initial conditions $\psi_0(x_0) = x_0$ and $f_t : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ being the time-dependent vector field associated with the ODE and transports samples from $\mu_0 \to \mu_1$.

The most scalable way to train CNFs is to utilize a *simulation-free* training objective which regresses a learned neural vector field $v_{t,\theta}(x_t) : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ to the desired target vector field $f_t(x_t)$ for all time. This technique is commonly known as flow-matching [Liu, 2022, Albergo and Vanden-Eijnden, 2023, Lipman et al., 2023, Tong et al., 2024a] and has the neural transport map $\psi_{t,\theta}$ which is obtained through a neural differential equation [Chen et al., 2018] $\frac{d}{dt}\psi_{t,\theta}(x) = v_{t,\theta}\left(\psi_{t,\theta}(x)\right)$. Specifically, flow-matching regresses $v_{t,\theta}(x_t)$ to the target *conditional* vector field $f_t(x_t|z)$ associated to the target flow $\psi_t(x_t|z)$. We say that this conditional vector field $f_t(x_t|z)$, *generates* the target density $\mu_1(x_1)$ by interpolating along the probability path $\mu_t(x_t|z)$ in time. We often do not have closed-form access to the generating marginal vector field $f_t(x_t)$. Still, with conditioning, e.g., $z = (x_0, x_1)$, we can obtain a simple analytic expression of a conditional vector field that achieves the same goals. The conditional flow-matching (CFM) objective can then be stated as a simple simulation-free regression,

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,q(z),\mu_t(x_t|z)} \|v_{t,\theta}(t, x_t) - f_t(x_t|z)\|_2^2. \tag{1}$$

The conditioning distribution $q(z)$ can be chosen from any valid coupling, for instance, the independent coupling $q(z) = \mu_0(x_0)\mu_1(x_1)$. To generate samples and their corresponding log density according to the CNF, we may solve the following flow ODE numerically with initial conditions $x_0 = \psi_0(x_0)$ and $c = \log \mu_0(x_0)$, which is the log density under the prior:

$$\frac{d}{dt}\begin{bmatrix} \psi_{t,\theta}(x_t) \\ \log \mu_t(x_t) \end{bmatrix} = \begin{bmatrix} v_{t,\theta}(t, x_t) \\ -\nabla \cdot v_{t,\theta}(t, x_t) \end{bmatrix}. \tag{2}$$

In the next section, we outline a different methodology to build a transport map leveraging *stochastic* dynamics. This allows us to frame the mass transport problem as a Schrödinger bridge, which is well suited to modeling noisy measurements in applications such as single-cell evolution.

3 Schrödinger Bridge with non-zero reference field

The complex nature of particle dynamics can be captured as a mass transport problem under a prescribed reference process. Specifically, we model particle evolution using a parametrized stochastic differential equation (SDE), with drift $v_{t,\theta} : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$, diffusion coefficient $g_t > 0$:

$$\mathrm{d}\mathbf{X}_t = v_{t,\theta}(\mathbf{X}_t)\,\mathrm{d}t + g_t \mathrm{d}\mathbf{B}_t, \quad \mathbf{X}_0 \sim \mu_0, \mathbf{X}_1 \sim \mu_1, \tag{3}$$

where $\mathbf{B}_t$ is a standard Brownian motion and by convention time $t \in [0,1]$ flows from $t = 0$ to $t = 1$ such that marginal distribution at the endpoints are ρ_0 and ρ_1. These endpoints are provided as empirical distributions and represent endpoint observations along a transport trajectory. The SDE in eq. (3) induces a path measure in the space of Markov path measures $(\mathbb{P}_{t,\theta})_{t\in[0,1]} \in \mathcal{P}(C[0,1], \mathbb{R}^d)$

https://doi.org/10.52202/085713-4053

such that the marginal density p_t evolves according to the following Fokker-Plank equation:

$$\frac{\partial p}{\partial t} = -\nabla \cdot (v_{t,\theta}(\mathbf{X}_t), p_t(\mathbf{X}_t)) + \frac{g_t^2}{2}\Delta p_t(\mathbf{X}_t), \quad p_0 = \rho_0, p_1 = \rho_1. \tag{4}$$

In addition, our modeling of particle dynamics is informed by a reference process which is defined by the following SDE with corresponding drift $f_t : \mathbb{R}^d \to \mathbb{R}^d$ and diffusion coefficient $g_t > 0$:

$$\mathrm{d}\mathbf{X}_t = f_t(\mathbf{X}_t)\,\mathrm{d}t + g_t d\mathbf{B}_t. \tag{5}$$

We denote the induced path measure of eq. (5) as $(\mathbb{Q}_t)_{t\in[0,1]}$.

Note further that we assume the diffusion coefficient, g_t, to be the same for both processes $\mathbb{P}_t$ and $\mathbb{Q}_t$ to simplify the setting and facilitate easier exposition of the setting considered in this paper.

Schrödinger bridge with zero-drift. We now state the Schrödinger bridge problem, which finds an optimal path measure $\mathbb{P}^*$ that is the solution to the following KL-divergence minimization problem:

$$\mathbb{P}^* = \operatorname*{argmin}_{\theta} \left[\mathrm{KL}\left(\mathbb{P}_\theta || \mathbb{Q}\right) : \mathbb{P}_0 = \mu_0, \mathbb{P}_1 = \mu_1\right] \tag{6}$$

In settings where eq. (5) is zero-drift and with constant diffusion coefficient—i.e. $\mathrm{d}\mathbf{X}_t = g_t\mathbf{B}_t$—the Schrödinger bridge problem [Schrödinger, 1932] devolves into the *Diffusion Schrödinger Bridge* problem [De Bortoli et al., 2021a, Bunne et al., 2023]. In this special case, the Schrödinger bridge problem admits a unique solution and is linked to the entropic optimal transport plan through the seminal result of Föllmer [1988]. Specifically, $\mathbb{P}^*$ is a mixture of conditional Brownian bridges $\mathbb{Q}_t(\cdot|x_0, x_1)$ weighted by the entropic OT-plan $\pi^* \in \Pi(\rho_0 \otimes \rho_1)$ which is a valid coupling in the product measure $\rho_0 \otimes \rho_1$, in other words $\int \pi(x_0, \cdot) = \mu_0(x_0), \int \pi(\cdot, x_1) = \mu(x_1)$,

$$\mathbb{P}^* = \int \mathbb{Q}_t(\cdot|x_0, x_1)\, d\pi^*(x_0, x_1) \tag{7}$$

$$\pi^*(\mu_0, \mu_1) = \operatorname*{argmin}_{\pi\in\Pi(\mu_0\otimes\mu_1)} \int c(x_0, x_1) d\pi(x_0, x_1) + 2\sigma^2 \mathrm{KL}(\pi||\rho_0 \otimes \rho_1). \tag{8}$$

This holds when $g_t = \sigma$ for some constant σ (i.e. g_t is "fixed"), which assume for the remainder of this work. In this case, operationally, the conditional Brownian bridges take the form of a Normal distribution $\mathbb{Q}_t(\cdot|x_0, x_1) = \mathcal{N}(x_t; tx_1 + (1-t)x_0, t(1-t)\sigma^2)$ with the mean given as an interpolation between two endpoints. Furthermore, when $\sigma \to 0$, the entropic OT problem reduces to the regular OT problem. We note that this Schrödinger bridge problem can be reinterpreted as a stochastic optimal control problem where the control cost is the drift $v_{t,\theta}$. That is the stochastic optimal control perspective minimizes *average kinetic energy*[3] of the learned process which leads to the following optimization problem:

$$v_\theta^* = \left\{ \min_\theta \int \mathbb{E}_{\mathbb{P}_t}\left[\frac{1}{2}\|v_{t,\theta}(\mathbf{X}_t)\|_2^2\right] dt : \mathrm{d}\mathbf{X}_t = v_{t,\theta}(\mathbf{X}_t)\,\mathrm{d}t + g_t \mathrm{d}\mathbf{B}_t, \mathbb{P}_0 = \rho_0, \mathbb{P}_1 = \rho_1 \right\}. \tag{9}$$

We approximate $\mathbb{P}^*$ using mini-batch OT [Fatras et al., 2020, 2021] and simulation-free matching algorithms [Tong et al., 2024a,b, Pooladian et al., 2023], iterative proportional and Markov fitting [De Bortoli et al., 2021a, Shi et al., 2024], and generalized Schrödinger bridge matching [Liu et al., 2023a].

3.1 Schrödinger Bridges with non-zero drift

We now consider the more general case where the drift of the reference process $\mathbb{Q}$ is non-zero. In this case, existing computational approaches, which rely on gradient field dynamics, no longer apply. As detailed in table 1, we consider the case of a constant $g_t > 0$ and tailor our approach to small g_t as previous work has shown this to be preferred empirically [Tong et al., 2024b]. While it is possible to also learn g_t as in GSBM [Liu et al., 2023a], this is computationally expensive; in this work, we focus on the setting of fixed or small g_t and vectorfields with Curl, as motivated by applications in our experiments section 4. In this case, we can approximate the

Table 1: Overview of the properties of different approaches.

Method	$\mathbb{P}_\theta$	$\mathbb{Q}$		Models Curl
	v_t	f_t	g_t	
DSBM	✓	✗	fixed	✗
OT-CFM	✓	✗	$\lim_{g_t\to 0}$	✗
GSBM	✓	✓	learned	✗
CURLY-FM (ours)	✓	✓	fixed	✓

[3] Schrödinger bridges minimize the relative entropy w.r.t. to $\mathbb{Q}$ and kinetic energy in the deterministic case.

marginal process using a mixture of conditional bridges, albeit not necessarily Brownian. Specifically, we consider the case modeling the marginal process $\mathbb{P}_t = \int \mathbb{Q}_t(\cdot|x_0, x_1)d\pi(x_0, x_1)$, where we use $\mathbb{Q}_t(\cdot|x_0, x_1)$ to denote the stochastic bridge pinned at x_0, x_1 at times 0 and 1 respectively and π is a valid coupling. With this decomposition, it remains to specify the parameterization of $\mathbb{Q}_t(\cdot|x_0, x_1)$ and $\pi(x_0, x_1)$. As a modeling choice, we parameterize $\mathbb{Q}_t(\cdot|x_0, x_1)$ using a mixture of Brownian bridges with learnable parameters η:

$$\mathbb{Q}_{\eta,t} = f_{t,\eta}(x_{t,\eta}|x_0, x_1)dt + g_t \mathrm{d}\mathbf{B}_t, \tag{10}$$

with the idea that,

$$f^*_\eta = \left\{ \min_\eta \int \mathbb{E}_{\mathbb{P}_t} \left[\frac{1}{2} \|f_{t,\eta}(x_t) - f_t(x_t)\|_2^2 \right] : \mathrm{d}\mathbf{X}_t = f_{t,\eta}(\mathbf{X}_t)\,\mathrm{d}t + g_t \mathrm{d}\mathbf{B}_t, \mathbb{Q}_0 = \rho_0,\, \mathbb{Q}_1 = \rho_1 \right\}.$$

We approximate the mean of $\mathbb{Q}_t$ by designing a neural path interpolant $\varphi_{t,\eta}$ with parameters η:

$$\mu_{t,\eta} := \int_0^t f_{s,\eta}(\mu_s)ds = tx_1 + (1-t)x_0 + t(1-t)\varphi_{t,\eta}(x_0, x_1). \tag{11}$$

We optimize $\varphi_{t,\eta}$ by minimizing the following simulation-free objective of the relative kinetic energy:

$$\mathcal{L}(\eta) = \mathbb{E}_{t\sim\mathcal{U}[0,1], x_0\sim\rho_0, x_1\sim\rho_1} \left[\left\| \frac{\partial \mu_{t,\eta}}{\partial t} - f_t(\mu_{t,\eta}) \right\|_2^2 \right], \quad f_t(\mu_{t,\eta}) = \kappa(\mu_{t,\eta}, x_0) f_0(x_0).$$

Here $\kappa_t(\mu_{t,\eta}, x_0)$ is any smooth function, e.g. a nearest neighbor based distance kernel $\kappa_t(\mu_{t,\eta}, x_0) = \|\mu_{t,\eta} - x_0^i\|_2 / \sum_i^N \|\mu_{t,\eta} - x_0^i\|_2$. Then we use a kernel definition of f_t based on data as it is uncommon to have access to f_t for the practical applications we consider. We further discuss our assumptions for κ_t in the appendix §G.4. We also note $\partial\mu_{t,\eta}/\partial t$ can be computed using automatic differentiation:

$$\frac{\partial \mu_{t,\eta}}{\partial t} = f_{t,\eta}(\mu_t) = x_1 - x_0 + t(1-t)\frac{\partial \varphi_{t,\eta}}{\partial t}(x_0, x_1) + (1-2t)\varphi_{t,\eta}(x_0, x_1). \tag{12}$$

The pseudocode for learning the neural path is presented in algorithm 1. To approximate $\mathbb{P}_t$ we next learn to approximate the optimal mixture of conditional bridges $\mathbb{P}^*_t = \mathbb{E}_{x_0,x_1\sim\pi^*(x_0,x_1)}[x_{t,\eta}]$. However, this necessitates the feasibility of computing the OT-plan, which is defined below:

$$\pi^*(\mu_0, \mu_1) = \underset{\pi\in\Pi(\mu_0\otimes\mu_1)}{\operatorname{argmin}} \int c(x_0, x_1)d\pi(x_0, x_1), \quad c(x_0, x_1) = \int_0^1 \left\| \frac{\partial \mu_{t,\eta}}{\partial t} - f_t(\mu_{t,\eta}) \right\|_2^2 dt$$
$$\text{s.t.} \int \pi(x_0, \cdot) = \rho_0(x_0), \int \pi(\cdot, x_1) = \rho(x_1).$$

Indeed, this optimal transport cost $c(x_0, x_1)$ can be computed through simulating the entire trajectory. However, we opt for using a stochastic estimator of the cost with K samples:

$$c(x_0, x_1) = \mathbb{E}_{t\sim\mathcal{U}[0,1]} \left[\left\| \frac{\partial x_{t,\eta}}{\partial t} - f_t(x_{t,\eta}) \right\|_2^2 \right] = \frac{1}{K} \sum_i^K \left\| \frac{\partial x^i_{t,\eta}}{\partial t} - f_t(x^i_{t,\eta}) \right\|_2^2. \tag{13}$$

This cost ensures that we choose a coupling which minimizes the total cost in eq. (10). We highlight that the optimal plan π^* is intractable and we instead use a biased minibatch approximation of the plan (see §G.3). We use the cost in eq. (13) to estimate a transport plan $\pi(x_0, x_1)$ to construct the approximated mixture of conditional bridges $\mathbb{P}_t$ which is needed to learn the drift $v_{t,\theta}(x_t)$ of eq. (3),

$$\mathcal{L}_{\text{flow}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,1], (x_0,x_1)\sim\pi(x_0,x_1)} \left[\frac{1}{2} \left\| v_{t,\theta}(x_{t,\eta}) - \left(\frac{\partial \mu_{t,\eta}}{\partial t} \right) \texttt{.detach()} \right\|_2^2 \right].$$

The procedure for this marginal (flow) matching objective is presented in algorithm 2. In the case that $g_t := \sigma > 0$, for a constant σ, we also need to learn the marginal score $s_{t,\theta} \approx \nabla \log p_t(x_t)$. This can be learned using a conditional score matching objective where $\lambda_t = 2\sqrt{t(1-t)}/\sigma$ and

https://doi.org/10.52202/085713-4053

$x_{t,\eta} = \mu_{t,\eta} + \sigma\epsilon\sqrt{t(1-t)}$ with $\epsilon \sim \mathcal{N}(0,1)$ [Tong et al., 2024b],

$$\mathcal{L}_{\text{score}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}[0,1],(x_0,x_1)\sim\pi(x_0,x_1)}\left[\frac{1}{2}\left\|\lambda_t s_{t,\theta}(x_{t,\eta}) + \epsilon\right\|_2^2\right].$$

In totality, the combined loss is given by thus $\mathcal{L}(\theta) = \mathcal{L}_{\text{flow}}(\theta) + \mathcal{L}_{\text{score}}(\theta)$.

Algorithm 1 Training algorithm for neural path interpolant network

Require: Marginals $\rho_0(x_0)$ and $\rho_1(x_1)$, network φ_η, reference drift $f_{t,\eta}$

while Training **do**
- Sample $(x_0, x_1, t) \sim \rho_0(x_0)\rho_1(x_1)\mathcal{U}(0,1)$
- $\mu_{t,\eta} = (1-t)x_0 + tx_1 + t(1-t)\varphi_{t,\eta}(x_0, x_1)$
- $\frac{\partial\mu_{t,\eta}}{\partial t} = x_1 - x_0 + t(1-t)\frac{\partial\varphi_{t,\eta}(x_0,x_1)}{\partial t} + (1-2t)\varphi_{t,\eta}(x_0,x_1)$
- $\mathcal{L}(\eta) = \left\|\frac{\partial\mu_{t,\eta}}{\partial t} - f_{t,\eta}(\mu_{t,\eta})\right\|_2^2$
- $\eta \leftarrow \text{Update}(\eta, \nabla_\eta\mathcal{L}(\eta))$

return $\varphi_{t,\eta}$

Algorithm 2 Marginal Score and Flow Matching

Require: Marginals $\rho_0(x_0)$ and $\rho_1(x_1)$, network φ_η, vector field network $v_{t,\theta}$.

while Training **do**
- Sample $(x_0, x_1, t_{ij}) \sim \rho_0(x_0)\rho_1(x_1)\mathcal{U}(0,1)$
- $C_\eta^{ij}(x_0^i, x_1^j) = \mathbb{E}_t\left[\left\|\frac{\partial\mu_{t,\eta}}{\partial t} - f_{t,\eta}(\mu_{t,\eta})\right\|_2^2\right]$
- $x_0, x_1 \sim \pi(x_0, x_1) \leftarrow \text{OT}(x_0, x_1, C_\eta)$
- $t \sim \mathcal{U}(0,1), \epsilon \sim \mathcal{N}(0,1)$
- $\mathcal{L}(\theta) = \mathcal{L}_{\text{flow}}(\theta) + \mathcal{L}_{\text{score}}(\theta)$
- $\theta \leftarrow \text{Update}(\theta, \nabla_\theta\mathcal{L}(\theta))$

return v_θ

Remark 1. *We highlight that while $\mathcal{L}(\theta)$ seeks to match $v_{t,\theta}$ to velocity of the neural path-interpolant $\frac{\partial\mu_{t,\eta}}{\partial t}$ and the optimal velocity $v_t^* \neq f_{t,\eta}$ since the reference process $\mathbb{Q}_\eta$ does not necessarily transport ρ_0 to ρ_1. More precisely, $\mathbb{Q}_\eta$ does not have constraints at the endpoints, that $\mathbb{Q}_0 = \rho_0$ and $\mathbb{Q}_1 = \rho_1$, which are required from our learned process $\mathbb{P}_\theta$ and its drift $v_{t,\theta}$.*

4 Experiments

We investigate the application of CURLY-FM on multiple applications which exhibit non-gradient field dynamics including a simple toy example, an ocean currents modeling application, a computational fluid mechanics dataset, and an application to single-cell trajectory inference. We benchmark CURLY-FM against both simulation-free flow matching approaches: Conditional flow matching (CFM) [Liu et al., 2023b, Peluchetti, 2023, Lipman et al., 2023, Albergo et al., 2023], optimal transport conditional flow matching (OT-CFM) [Tong et al., 2024a] and when possible metric flow matching [Kapuśniak et al., 2024] which cannot model non-zero drift dynamics, as well as simulation-based methods in TrajectoryNet and SBIRR [Shen et al., 2025] which can model non-zero drift dynamics but are much slower [Tong et al., 2020] and numerically unstable.

We evaluate CURLY-FM using metrics both on held out samples (2-Wasserstein ($\mathcal{W}_2$)) as well as metrics which directly measure how well the learned drift f_θ field matches the reference drift (Cosine distance and L_2 cost). We note that in many cases, it is not possible to match the reference drift exactly as the model is forced to match the marginals.

4.1 Synthetic Experiments

We start our experimental study of learning cyclical patterns from population-level observed populations by considering a synthetic example. We construct source and target distributions on asymmetrically arranged `circles` (fig. 1a), each with higher particle population density on one side. Given a circular reference velocity field $f_t(x_t, \omega)$ with constant rotational speed, the goal is to learn the velocity-field $v_{t,\theta}(x_t)$ and trajectories $\psi_{t,\theta}(x_t)$ for $t \in [0,1]$. We find that previous flow matching methods with zero-reference field f_t^* result in straight paths between source and target distributions, thereby failing to capture cycling patterns in the underlying data (see 1b and fig. 1c).

4.2 Modeling Ocean Currents

We model ocean currents in the Gulf of Mexico using a resolution of 1 km of bathymetry data from HYbrid Coordinate Ocean Model (HYCOM), which allows us to obtain a reference field. We follow the data processing pipeline of Shen et al. [2025] and observe 111 particles per time-point (see §D for exact dataset details). We report our quantitative results in table 2 and observe that across the left-out time points, CURLY-FM obtains the best results for the majority of the reported metrics,

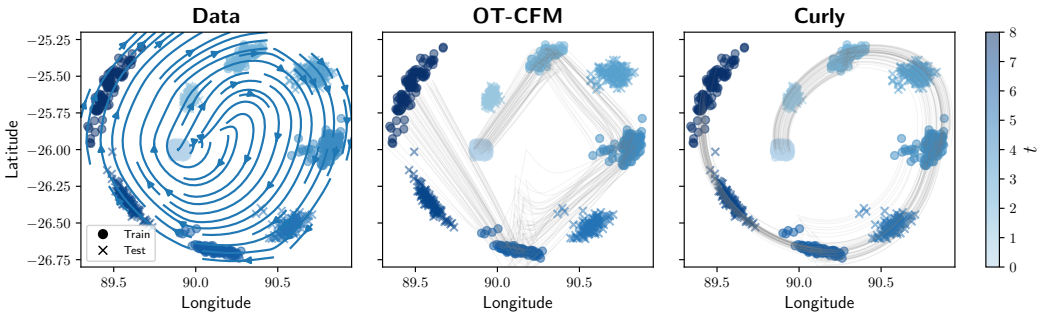


Figure 2: Visualization of ground truth data and vectorfield (left), OT-CFM predicted trajectories (center), and CURLY-FM predictions (right). Curly fits the vortex much better than OT-CFM.

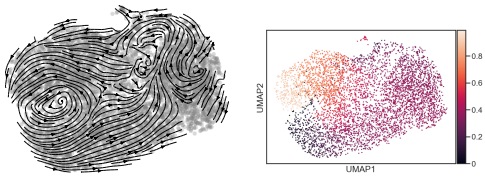
Table 2: Quantitative metrics on left out test timepoints for oceans. * numbers taken from Shen et al. [2025]

Metric	Method	t_2	t_4	t_6	t_8
EMD[1]	OT-CFM	0.148 ± 0.004	0.227 ± 0.008	0.191 ± 0.012	0.250 ± 0.018
	MFM	0.107 ± 0.014	0.056 ± 0.014	0.052 ± 0.011	0.070 ± 0.021
	Vanilla-SB*	0.270 ± 0.058	0.300 ± 0.056	0.420 ± 0.056	0.410 ± 0.048
	SBIRR Shen et al. [2025]*	0.073 ± 0.020	0.072 ± 0.012	0.120 ± 0.029	0.094 ± 0.023
	CURLY-FM	$\mathbf{0.019 \pm 0.003}$	$\mathbf{0.045 \pm 0.005}$	$\mathbf{0.027 \pm 0.001}$	$\mathbf{0.030 \pm 0.006}$
Cos. Dist.	OT-CFM	0.229 ± 0.004	0.121 ± 0.008	0.034 ± 0.005	0.067 ± 0.007
	MFM	$\mathbf{0.179} \pm 0.010$	$\mathbf{0.011} \pm 0.001$	$\mathbf{0.002} \pm 0.001$	0.004 ± 0.002
	CURLY-FM	0.231 ± 0.004	0.017 ± 0.001	$\mathbf{0.002} \pm 0.000$	$\mathbf{0.002} \pm 0.000$
L_2 cost	OT-CFM	0.167 ± 0.004	0.144 ± 0.014	0.095 ± 0.005	0.250 ± 0.023
	MFM	0.203 ± 0.011	$\mathbf{0.067} \pm 0.011$	$\mathbf{0.101} \pm 0.015$	$\mathbf{0.141} \pm 0.018$
	CURLY-FM	$\mathbf{0.151} \pm 0.004$	0.098 ± 0.001	0.135 ± 0.010	0.178 ± 0.017

and also outperforms the previous state-of-the-art SBIRR [Shen et al., 2025] on the EMD metric. Moreover, we note that CURLY-FM is computationally fast and achieves these results in minutes compared to 4hrs for the simulation-based SBIRR. These findings are also substantiated in fig. 2, where we see trajectories that look more natural at modeling periodic behavior than OT-CFM.

4.3 Experiments on Single-Cell Data

To show that CURLY-FM is effective in learning dynamic behavior in single-cell data, we leverage two biologically rich datasets consisting of cell cycles in human cell fibroblasts [Riba et al., 2022] and erythroblast development in mouse [Pijuan-Sala et al., 2019]. We aim to learn cell state trajectories and development paths considering the respective RNA-velocity fields, providing information about cell cycling, lineage bifurcation, and transcriptional dynamics.



(a) RNA-Velocity Field (b) Cell Cycles

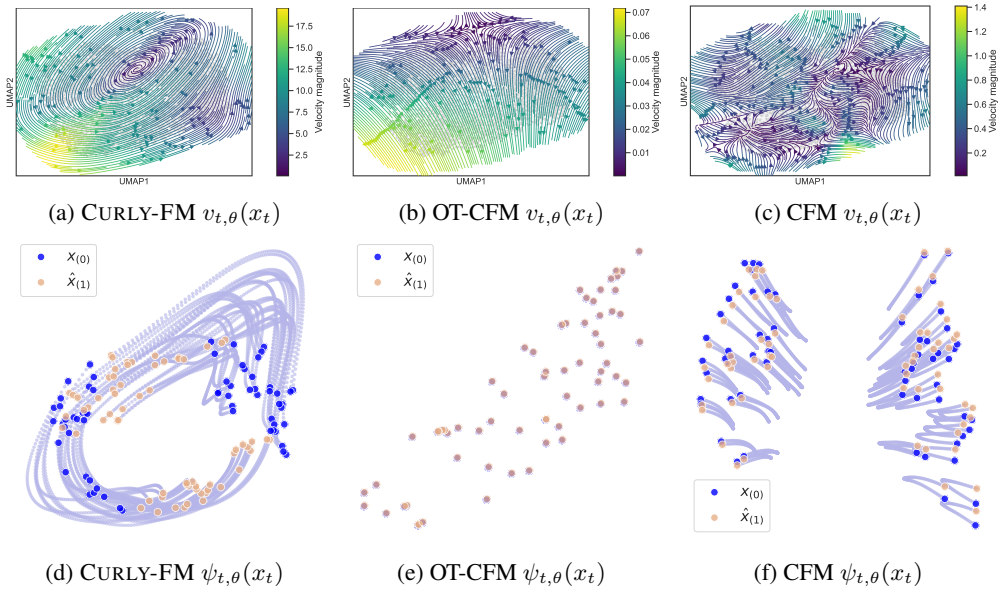
Figure 3: Ground truth data.

Cell cycle dynamics in human fibroblasts. We study the nature of cell cycling in human fibroblasts and reconstruct cyclical patterns in spliced-unspliced RNA space for single genes. We leverage RNA-velocities in figure 3a to construct cell state transition paths in figure 3 by estimating RNA velocity field between marginals using k-nn algorithm. Further dataset details are included in §E.1. Figure 4 shows learned velocity fields $v_{t,\theta}(x_t)$ and trajectories $\psi_{t,\theta}(x_t)$ between cell cycle distributions at $t = 0$ and $t = 1$. In table 3, we show results on the trajectory inference task comparing CURLY-FM to CFM, OT-CFM, and TrajectoryNet. Given the underlying cell cycle process, the aim is to learn circular trajectories resulting from a divergence-free velocity field. While traditional methods are successful in generating end points near ground truth, they fail at learning cyclic patterns, as shown in figures 4f.

Our results show that considering a non-zero reference field and velocity inference captures non-gradient dynamics in data. Figure 4a and fig. 4d show learned behavior using CURLY-FM. We observe that the trajectory $\psi_{t,\theta}(x_t)$ inferred with CURLY-FM closely matches expected cycling patterns in the fibroblast dataset, in contrast to trajectories inferred using CFM and OT-CFM. This

https://doi.org/10.52202/085713-4053

is quantified in table 3, where we can see the cosine distance to the reference field is significantly lower for CURLY-FM.



(a) CURLY-FM $v_{t,\theta}(x_t)$ (b) OT-CFM $v_{t,\theta}(x_t)$ (c) CFM $v_{t,\theta}(x_t)$

(d) CURLY-FM $\psi_{t,\theta}(x_t)$ (e) OT-CFM $\psi_{t,\theta}(x_t)$ (f) CFM $\psi_{t,\theta}(x_t)$

Figure 4: Vectorfields (top) and trajectory traces (bottom) learned using CURLY-FM (left) OT-CFM (center) CFM (right). CURLY-FM is the only method able to learn the cell cycle.

Table 3: Quantitative results for cell cycle trajectory inference task. We report the mean result for a metric with standard deviation over three seeds. CURLY-FM performs the best across matching inferred velocity field to the reference process (cosine distance) while maintaining comparable predictive quality.

Datasets →	$d = 2$		$d = 10$		$d = 20$	
Algorithm ↓	$\mathcal{W}_2$ ↓	Cos. Dist ↓	$\mathcal{W}_2$ ↓	Cos. Dist ↓	$\mathcal{W}_2$ ↓	Cos. Dist ↓
CFM	0.294 ± 0.030	1.065 ± 0.080	0.606 ± 0.059	1.001 ± 0.037	1.227 ± 0.013	1.007 ± 0.010
OT-CFM	$\mathbf{0.248 \pm 0.030}$	0.800 ± 0.309	$\mathbf{0.586 \pm 0.041}$	1.008 ± 0.039	$\mathbf{1.183 \pm 0.015}$	0.978 ± 0.125
TrajectoryNet	0.531 ± 0.021	1.077 ± 0.031	0.853 ± 0.059	0.979 ± 0.064	–	–
CURLY-FM (Ours)	1.199 ± 0.177	$\mathbf{0.295 \pm 0.040}$	0.930 ± 0.024	$\mathbf{0.300 \pm 0.058}$	1.261 ± 0.077	$\mathbf{0.249 \pm 0.024}$

Reconstructing cell differentiation in mouse Erythroid development.. Mouse erythroid cells develop in a curved trajectory over time. We show that CURLY-FM can adhere to this developmental path purely based on velocity data for the first time. Earlier works have used manifold-based penalties to follow curved structures. We show that this is no longer necessary with clever usage of velocity information. We observe 9,815 erythroid cells undergoing differentiation and partition the data into three temporal snapshots, withholding the central marginal to assess trajectory inference (see §E.2).

We visualize trajectories in figure 5 showing that CURLY-FM clearly follows the developmental pathway of mouse erythroid cells, whereas OT-CFM fails to capture dynamics between marginals. To assess CURLY-FM performance, we measure cosine-distance and L_2 norm between learnt and ground truth velocities as well as $\mathcal{W}_2$ distance between points on left-out marginal. Our quantitative results show that CURLY-FM outperforms OT-CFM at reconstructing the underlying RNA-velocity field and cell trajectories in majority of selected dimensions. MFM continues to achieve lower $\mathcal{W}_2$, indicating stronger adherence to the underlying manifold. Conversely, CURLY-FM attains superior cosine similarity to the ground-truth velocity field, consistent with its objective emphasizing faithful velocity alignment which contributes to a challenge *exactly* matching end-point marginals.

Table 4: Erythroid dataset results across dimension.

Metric	OT-CFM	MFM	CURLY-FM (Ours)
Dimension $d = 2$			
Cos. Dist	0.146 ± 0.001	0.014 ± 0.001	$\mathbf{0.009 \pm 0.000}$
L_2	2.704 ± 0.019	1.999 ± 0.014	$\mathbf{1.663 \pm 0.293}$
$\mathcal{W}_2$	0.646 ± 0.006	$\mathbf{0.269 \pm 0.004}$	0.369 ± 0.090
Dimension $d = 20$			
Cos. Dist	0.489 ± 0.001	0.495 ± 0.001	$\mathbf{0.488 \pm 0.001}$
L_2 ($\times 10^3$)	1.885 ± 0.020	$\mathbf{1.627 \pm 0.040}$	1.721 ± 0.035
$\mathcal{W}_2$	6.103 ± 0.074	$\mathbf{4.855 \pm 0.052}$	6.124 ± 0.027
Dimension $d = 50$			
Cos. Dist	0.490 ± 0.000	0.494 ± 0.000	$\mathbf{0.489 \pm 0.000}$
L_2 ($\times 10^3$)	2.215 ± 0.022	$\mathbf{1.971 \pm 0.023}$	2.045 ± 0.073
$\mathcal{W}_2$	7.969 ± 0.029	$\mathbf{6.727 \pm 0.022}$	7.729 ± 0.046

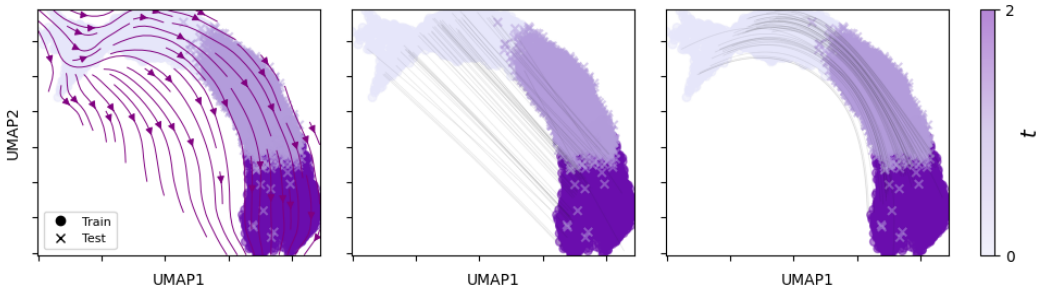


Figure 5: Visualization of ground truth data and vectorfield (left), OT-CFM predicted trajectories (center) and CURLY-FM predictions (right). Curly fits the ground truth much better than OT-CFM.

Table 5: Quantitative results for the CFD trajectory inference task. Metrics are reported on held-out particles from the test set for all marginals. Error bars show standard deviation.

Method	Cos. Dist. ↓	MSE ↓	Prec.@5 ↑	Prec.@10 ↑	Prec.@25 ↑
CFM	0.254 ± 0.003	$\mathbf{0.085 \pm 0.002}$	0.079 ± 0.004	0.164 ± 0.006	0.337 ± 0.016
OT-CFM	0.248 ± 0.011	0.095 ± 0.001	0.303 ± 0.002	0.388 ± 0.004	0.496 ± 0.001
CURLY-FM	$\mathbf{0.189 \pm 0.027}$	0.095 ± 0.003	$\mathbf{0.489 \pm 0.010}$	$\mathbf{0.522 \pm 0.009}$	$\mathbf{0.628 \pm 0.010}$

4.4 Experiments on Computational Fluid Mechanics Data

We evaluate CURLY-FM on a particle-based PDE dataset generated by a Lagrangian solver. Unlike grid-based (Eulerian) methods, Lagrangian approaches discretize the fluid as a set of particles that move with the flow. These particles evolve under the dynamics of the PDE which provides the particles' positions over time; other quantities of interest, such as velocity or energy, are then computed from these positions. We use data from LagrangeBench [Toshev et al., 2023], specifically the two-dimensional decaying Taylor-Green vortex (2DTGV) dataset(see §F).

In table 5, we report quantitative results on left-out particles for each marginal from a test set. We evaluate performance using (i) the cosine distance between the learned velocity field and the reference field; (ii) mean squared error (MSE) between the predicted particle positions at marginal $t+1$ and the ground truth positions, using the known coupling (ordering) between particles across marginals; and (iii) precision@k, measuring how often the predicted position is among the k nearest neighbors of the corresponding ground truth particle. CURLY-FM outperforms baselines in terms of cosine distance and precision@k while matching or outperforming the baseline methods on MSE. The smaller cosine distance for CURLY-FM shows that CURLY-FM produces velocity fields that better align with the reference field. Equal or lower MSE paired with higher precision@k shows that CURLY-FM more accurately recovers the true particle coupling and generates more faithful trajectories (c.f. fig. 10).

4.5 Further analysis of CURLY-FM performance

On higher stochasticity. We extend our discussion in section 3.1 on stochasticity levels σ to consider an ablation where $\sigma > 0$. Despite our work being motivated in the little to no stochasticity regime, we demonstrate that considering $\sigma > 0$ does not impact CURLY-FM efficiency. We find, similar to previous work [Tong et al., 2024b], low values of σ perform the best on all metrics. As a result, we recommend setting σ to zero unless some reference σ value is known. Therefore, all of our experiments are performed under $\sigma = g_t = 0$ assumption. We consider ocean currents dataset and find that larger stochasticity monotonically decreases performance on our tasks, thus justifying our choice of σ for our empirical work.

Table 6: Ablation on stochasticity σ.

σ	Cos. Dist. ↓	L_2 ↓	$\mathcal{W}_2$ ↓
0.01	0.061 ± 0.003	0.141 ± 0.009	0.028 ± 0.066
0.10	0.062 ± 0.002	0.145 ± 0.011	0.066 ± 0.008
1.00	0.145 ± 0.009	0.474 ± 0.058	0.871 ± 0.048

On computational efficiency. We further provide the computational cost in wall clock time for TrajectoryNet, SBIRR and CURLY-FM in table 7 (see §H.2 for further baseline comparison). We observe that CURLY-FM completes the Ocean currents problem in minutes with higher accuracy in trajectory and velocity field inference task, while SBIRR and TrajectoryNet are in the order of multiple hours and unlike CURLY-FM are simulation-based.

https://doi.org/10.52202/085713-4053

5 Related Work

Flow matching. Flow matching [Lipman et al., 2023], also known as rectified flows [Liu, 2022, Liu et al., 2023b] or stochastic interpolants [Albergo and Vanden-Eijnden, 2023, Albergo et al., 2023], has emerged as the default method for training continuous normalizing flow (CNF) models [Chen et al., 2018, Grathwohl et al., 2019]. However, FM can lead to unnatural dynamics less, and therefore many works attempt to derive methods for using minimum energy [Tong et al., 2024a, Pooladian et al., 2023] and more flexible conditional paths [Neklyudov et al., 2024, Kapuśniak et al., 2024].

Table 7: Compute cost.

Method	Hours
TrajectoryNet	7.44
SBIRR	4.67
CURLY-FM (Ours)	**0.06**

Schrödinger bridges with deep learning. To tackle the Schrödinger bridge problem in high dimensions many methods propose simulation-based [De Bortoli et al., 2021b, Chen et al., 2022, Koshizuka and Sato, 2023, Liu et al., 2022] and simulation-free [Shi et al., 2024, Tong et al., 2024b, Pooladian and Niles-Weed, 2023, Liu et al., 2023a] set-ups with various additional components incorporating variable growth rates [Zhang et al., 2025, Pariset et al., 2023, Sha et al., 2024], stochasticity, and manifold structure [Huguet et al., 2022] proposed based on neural ODE and neural SDE [Li et al., 2020, Kidger et al., 2021] frameworks. However, very few methods are able to incorporate approximate velocity data, and either match marginals using simulation [Tong et al., 2020], or do not attempt to match marginals [Qiu et al., 2022]. Finally, Schrödinger bridges with non-zero reference field have also been considered by Bartosh et al. [2024] and concurrently by Bartosh et al. [2025] and Shen et al. [2025], however, they do not employ a two-stage simulation-free approximation as CURLY-FM. We include further details on related work comparison in appendix §B.

RNA-velocity methods on discrete manifolds. A common strategy to regularize and interpret RNA-velocity [La Manno et al., 2018, Bergen et al., 2020] is to restrict it to a Markov process on a graph of cells representing a discrete manifold or compute higher-level statistics on it [Qiu et al., 2022]. However, these approaches are not equipped to match the marginal cell distribution over time. CURLY-FM can be seen as a method that unites these approaches with marginal-matching approaches.

6 Conclusion

In this work, we introduced CURLY-FM, a method capable of learning non-gradient field dynamics by solving a Schrödinger bridge problem with a non-zero reference process drift. In contrast to prior work, CURLY-FM is simulation-free, greatly improving numerical stability and efficiency. We showed the utility of this method in learning more accurate dynamics in a cell cycle system with known periodic behavior, computational fluid dynamics under Lagrangian solvers, and ocean currents. CURLY-FM opens up the possibility of moving beyond modeling population dynamics with simulation-free training methods and towards reconstructing the underlying governing dynamics [Xing, 2022]. Nevertheless, CURLY-FM is currently limited in its ability to discover the underlying dynamics by accurate inference of the reference field, which is an inherently difficult problem, especially over longer timescales. Exciting directions for future work involve additional verification of trajectories through lineage tracing [McKenna and Gagnon, 2019, Wagner and Klein, 2020], and improved modeling across non-stationary populations with the additional incorporation of unbalanced transport or multiomics datatypes [Baysoy et al., 2023].

Acknowledgments

The authors acknowledge funding from UNIQUE, CIFAR, NSERC, Intel, and Samsung. The research was enabled in part by computational resources provided by the Digital Research Alliance of Canada (`https://alliancecan.ca`), the Province of Ontario, companies sponsoring the Vector Institute (`http://vectorinstitute.ai/partners/`), Mila (`https://mila.quebec`), and NVIDIA. KK is supported by the EPSRC CDT in Health Data Science (EP/S02428X/1). AJB and LA are partially supported by NSERC Post-doc fellowships. LA is supported by the Eric and Wendy Schmidt Center at the Broad Institute of MIT and Harvard. This research is partially supported by EP- SRC Turing AI World-Leading Research Fellowship No. EP/X040062/1 and EPSRC AI Hub on Mathematical Foundations of Intelligence: An "Erlangen Programme" for AI No. EP/Y028872/1. We thank Renato Berlinghieri, author of SBIRR, for valuable discussions and for generously sharing his ocean currents code and data, which made our ocean current experiments possible.

References

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

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

L. Atanackovic, X. Zhang, B. Amos, M. Blanchette, L. J. Lee, Y. Bengio, A. Tong, and K. Neklyudov. Meta flow matching: Integrating vector fields on the wasserstein manifold. *International Conference on Learning Representations (ICLR)*, 2025.

V. Bansal, S. Roy, P. Sarkar, and A. Rinaldo. On the wasserstein convergence and straightness of rectified flow. *arXiv preprint*, 2025.

G. Bartosh, D. Vetrov, and C. A. Naesseth. Neural flow diffusion models: Learnable forward process for improved diffusion modelling. *Neural Information Processing Systems (NeurIPS 2024)*, 2024.

G. Bartosh, D. Vetrov, and C. A. Naesseth. Sde matching: Scalable and simulation-free training of latent stochastic differential equations. In *International Conference on Machine Learning (ICML)*, 2025.

A. Baysoy, Z. Bai, R. Satija, and R. Fan. The technological landscape and applications of single-cell multi-omics. *Nature Reviews Molecular Cell Biology*, 24(10):695–713, 2023.

V. Bergen, M. Lange, S. Peidli, F. A. Wolf, and F. J. Theis. Generalizing rna velocity to transient cell states through dynamical modeling. *Nature biotechnology*, 38(12):1408–1414, 2020.

C. Bunne, Y.-P. Hsieh, M. Cuturi, and A. Krause. The schrödinger bridge between gaussian measures has a closed form. In *International Conference on Artificial Intelligence and Statistics*, pages 5802–5833. PMLR, 2023.

C. Bunne, Y. Roohani, Y. Rosen, A. Gupta, X. Zhang, M. Roed, T. Alexandrov, M. AlQuraishi, P. Brennan, D. B. Burkhardt, A. Califano, J. Cool, A. F. Dernburg, K. Ewing, E. B. Fox, M. Haury, A. E. Herr, E. Horvitz, P. D. Hsu, V. Jain, G. R. Johnson, T. Kalil, D. R. Kelley, S. O. Kelley, A. Kreshuk, T. Mitchison, S. Otte, J. Shendure, N. J. Sofroniew, F. Theis, C. V. Theodoris, S. Upadhyayula, M. Valer, B. Wang, E. Xing, S. Yeung-Levy, M. Zitnik, T. Karaletsos, A. Regev, E. Lundberg, J. Leskovec, and S. R. Quake. How to build the virtual cell with artificial intelligence: Priorities and opportunities. 187(25):7045–7063, 2024a.

C. Bunne, G. Schiebinger, A. Krause, A. Regev, and M. Cuturi. Optimal transport for single-cell and spatial omics. *Nature Reviews Methods Primers*, 4(1):58, 2024b.

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

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

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

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

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

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

H. Föllmer. Random fields and diffusion processes. *Lect. Notes Math*, 1362:101–204, 1988.

 https://doi.org/10.52202/085713-4053

M. Gao, C. Qiao, and Y. Huang. Unitvelo: temporally unified rna velocity reinforces single-cell trajectory inference. *Nature Communications*, 13(1):6586, 2022.

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

T. Hashimoto, D. Gifford, and T. Jaakkola. Learning population-level diffusions with generative rnns. In *International Conference on Machine Learning (ICML)*, 2016.

D. Haviv, A.-A. Pooladian, D. Pe'er, and B. Amos. Wasserstein flow matching: Generative modeling over families of distributions. In *International Conference on Machine Learning (ICML)*, 2025.

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

K. Kapuśniak, P. Potaptchik, T. Reu, L. Zhang, A. Tong, M. Bronstein, A. J. Bose, and F. D. Giovanni. Metric flow matching for smooth interpolations on the data manifold. *Neural Information Processing Systems (NeurIPS)*, 2024.

P. Kidger, J. Foster, X. Li, H. Oberhauser, and T. Lyons. Neural SDEs as Infinite-Dimensional GANs. *International Conference on Machine Learning*, 2021.

D. Klein, G. Palla, M. Lange, M. Klein, Z. Piran, M. Gander, L. Meng-Papaxanthos, M. Sterr, L. Saber, C. Jing, et al. Mapping cells through time and space with moscot. *Nature*, pages 1–11, 2025.

T. Koshizuka and I. Sato. Neural lagrangian schrödinger bridge: Diffusion modeling for population dynamics. *Internationcal Conference on Learning Representations (ICLR)*, 2023.

G. La Manno, R. Soldatov, A. Zeisel, E. Braun, H. Hochgerner, V. Petukhov, K. Lidschreiber, M. E. Kastriti, P. Lönnerberg, A. Furlan, et al. Rna velocity of single cells. *Nature*, 560(7719):494–498, 2018.

M. Lange, Z. Piran, M. Klein, B. Spanjaard, D. Klein, J. P. Junker, F. J. Theis, and M. Nitzan. Mapping lineage-traced cells across time points with moslin. *Genome Biology*, 25(1):277, 2024.

H. Lavenant, S. Zhang, Y.-H. Kim, and G. Schiebinger. Towards a mathematical theory of trajectory inference. *The Annals of Applied Probability*, 34(1A):428–500, 2024.

X. Li, T.-K. L. Wong, R. T. Q. Chen, and D. Duvenaud. Scalable gradients for stochastic differential equations. *International Conference on Artificial Intelligence and Statistics*, 2020.

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

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

G.-H. Liu, Y. Lipman, M. Nickel, B. Karrer, E. A. Theodorou, and R. T. Chen. Generalized schr\" odinger bridge matching. *arXiv preprint arXiv:2310.02233*, 2023a.

G.-H. Liu, Y. Lipman, M. Nickel, B. Karrer, E. A. Theodorou, and R. T. Q. Chen. Generalized schröodinger bridge matching. *International Conference on Learning Representations (ICLR)*, 2024.

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

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

A. McKenna and J. A. Gagnon. Recording development with single cell dynamic lineage tracing. *Development*, 146(12):dev169730, 2019.

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

K. Neklyudov, R. Brekelmans, A. Tong, L. Atanackovic, Q. Liu, and A. Makhzani. A computational framework for solving wasserstein lagrangian flows. *International Conference on Machine Learning (ICML)*, 2024.

X. Pan and X. Zhang. Studying temporal dynamics of single cells: expression, lineage and regulatory networks. *Biophysical Reviews*, 16(1):57–67, 2024.

M. Pariset, Y.-P. Hsieh, C. Bunne, A. Krause, and V. D. Bortoli. Unbalanced diffusion schrödinger bridge. *Uncertainty in Artificial Intelligence (UAI)*, 2023.

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

G. Peyré and M. Cuturi. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 2019.

G. Peyré, M. Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

B. Pijuan-Sala, J. A. Griffiths, C. Guibentif, T. W. Hiscock, W. Jawaid, F. J. Calero-Nieto, C. Mulas, X. Ibarra-Soria, R. C. Tyser, D. L. L. Ho, et al. A single-cell molecular map of mouse gastrulation and early organogenesis. *Nature*, 566(7745):490–495, 2019.

A.-A. Pooladian and J. Niles-Weed. Plug-in estimation of schrödinger bridges. *Internationcal Conference on Learning Representations (ICLR)*, 2023.

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

X. Qiu, Y. Zhang, J. D. Martin-Rufino, C. Weng, S. Hosseinzadeh, D. Yang, A. N. Pogson, M. Y. Hein, K. H. J. Min, L. Wang, et al. Mapping transcriptomic vector fields of single cells. *Cell*, 185 (4):690–711, 2022.

A. Riba, A. Oravecz, M. Durik, S. Jiménez, V. Alunni, M. Cerciat, M. Jung, C. Keime, W. M. Keyes, and N. Molina. Cell cycle gene regulation dynamics revealed by rna velocity and deep-learning. *Nature communications*, 13(1):2865, 2022.

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

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

E. Schrödinger. 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.

Y. Sha, Y. Qiu, P. Zhou, and Q. Nie. Reconstructing growth and dynamic trajectories from single-cell transcriptomics data. *Nature Machine Intelligence*, 6(1):25–39, 2024.

R. Shahan, C.-W. Hsu, T. M. Nolan, B. J. Cole, I. W. Taylor, L. Greenstreet, S. Zhang, A. Afanassiev, A. H. C. Vlot, G. Schiebinger, et al. A single-cell arabidopsis root atlas reveals developmental trajectories in wild-type and cell identity mutants. *Developmental cell*, 57(4):543–560, 2022.

Y. Shen, R. Berlinghieri, and T. Broderick. Multi-marginal schrödinger bridges with iterative reference refinement, 2025.

Y. Shi, V. De Bortoli, A. Campbell, and A. Doucet. Diffusion schrödinger bridge matching. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

 https://doi.org/10.52202/085713-4053

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

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

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

A. Tong, N. Malkin, K. Fatras, L. Atanackovic, Y. Zhang, G. Huguet, G. Wolf, and Y. Bengio. Simulation-free schrödinger bridges via score and flow matching. *AISTATS*, 2024b.

A. Toshev, G. Galletti, F. Fritz, S. Adami, and N. A. Adams. Lagrangebench: A lagrangian fluid mechanics benchmarking suite. In *Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

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

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

J. Xing. Reconstructing data-driven governing equations for cell phenotypic transitions: integration of data science and systems biology. *Physical Biology*, 19(6):061001, 2022.

Z. Zhang, T. Li, and P. Zhou. Learning stochastic dynamics from snapshots through regularized unbalanced optimal transport. In *International Conference on Representation Learning (ICLR)*, 2025.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claim is to introduce a novel generative modelling framework for learning non-gradient field dynamics in natural sciences by designing and solving a Schrödinger bridge problem with a non-zero reference drift process. Abstract and introduction clearly state our contributions, confirming and empirically evaluating these in the remainder of the main text.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation, as long as it is clear that these goals are not attainable by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We describe limitations of our work in Section 6.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: This paper does not provide and include any concrete theoretical results and/or proofs.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have disclosed full details of our experimental set-up in Section 4, supplementary materials, and throughout the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

 https://doi.org/10.52202/085713-4053

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in the supplemental material?

   Answer: [Yes]

   Justification: We have provided open access to the data and code in the supplementary materials as well as references to the datasets used in the paper which are all publicly available for full reproducibility of our results.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We have provided full training and test details in supplementary materials.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report 1-sigma standard deviation across three chosen seeds across all experimental results in the main text and supplementary materials.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have provided full details on used compute resources and model configurations in supplementary materials.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

https://doi.org/10.52202/085713-4053

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We confirm that our research adheres NeurIPS code of ethics. We rely exclusively on publicly accessible datasets with permissible licenses, acknowledge all third-party resources, and uphold fairness, transparency, and reproducibility throughout our model development and evaluation.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We include a broader impacts section thoroughly assessing positive and negative societal impacts of our work in supplementary materials.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our work does not pose high risks of misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have provided citations and credits to all authors of publicly used datasets and baselines that are used to empirically validate our work in main text and supplementary materials.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: No new assets are released in this research.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

https://doi.org/10.52202/085713-4053

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our work does not include any form of crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our work does not include any form of crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: Core method of our research has not been developed with use of LLMs.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

Appendix

A Complementary Orthogonal Work on Single Cell

There are many recent works tackling the single-cell trajectory inference problem on single-cell RNA-sequencing data. In this work we focus on incorporating a reference velocity field and learning non-gradient field dynamics. There are many other related works that may be used in conjunction with ideas in this paper. Here we detail some of these complementary works and how they could be combined with ideas in CURLY-FM. Specifically a number of areas have been identified as improving how cells are modeled by flow-based networks.

- **Optimal transport / minimal energy**: Cells are modeled by optimal transport over short enough time scales and is therefore desirable in almost all applications of single-cell trajectory inference [Hashimoto et al., 2016, Schiebinger et al., 2019, Tong et al., 2020, 2024a].
- **Density or manifold assumptions**: Cells are also known to lie on a low dimensional manifold within gene space [Moon et al., 2018]. This knowledge has been exploited in a number of works that are both require simulation during training [Tong et al., 2020, Huguet et al., 2022] and more recent methods which do not [Neklyudov et al., 2024, Kapuśniak et al., 2024]
- **Unbalanced transport (modeling cell birth and death)**: By default, flow models assume conservation of mass over time through the continuity equation. Over long time scales this is not a good model of a population of cells as [Zhang et al., 2025]
- **Stochasticity**: Cells move stochastically based on unobserved factors. This has led to a number of methods that attempt to model cells with stochastic dynamics [Koshizuka and Sato, 2023, Tong et al., 2024b, Schiebinger, 2021], and in particular with Schrödinger bridges, as the stochastic extension of dynamic optimal transport. While there have been many works on learning Schrödinger bridges simulation-free there is little work on efficient learning of general reference drift functions which we tackle here.
- **Velocity estimates**: RNA-velocity [La Manno et al., 2018] exploits particularities about the RNA collection data to measure both older and newer RNA transcripts at a single timepoint. This allows the approximation of RNA-velocity— the approximate instantaneous change of the RNA expression of each cell. First used in Tong et al. [2020], velocity estimates are relatively underutilized in the trajectory inference problem as data is more scarce and more difficult to process. To our knowledge, CURLY-FM is the first method to provide a simulation-free method for incorporating velocity information into a learned flow.
- **Distribution conditioning** Where most flow-based frameworks model cells as a non-interacting point cloud, recent work has also considered flows which include terms for the interaction between cells [Atanackovic et al., 2025, Haviv et al., 2025]. This allows the modeling of more complex interactions between cells which is an extremely prevelent dynamic in real cell systems.

While CURLY-FM focuses on the incorporation of optimal transport, stochasticity, and in particular *velocity*, it does not address problems of unbalanced transport or manifold structure of the single-cell datatype. Future work will incorporate ideas from CURLY-FM in combination with existing ideas on how to model density and unbalanced transport for more expressive, accurate, and useful models of cell dynamics towards virtual cells [Bunne et al., 2024a].

B Further Details on Related Work

Iterative Algorithms. Shen et al. [2025] propose an iterative bi-level algorithm, first solving a Schrödinger bridge and estimating the forward-backward drift given a current guess of the reference drift. Given the solution to step 1, trajectories are simulated to obtain an updated estimate of the reference drift. This bi-level approach is considered since the reference drift belongs to a family of possible reference drifts rather than a single prescribed one. In contrast, CURLY-FM employs a two-stage algorithm that is not iterative and also does not search over a family of reference drifts. More critically, CURLY-FM is also simulation-free, which is achieved by making the modeling assumption that the conditional mixture of bridges is modelled as Brownian bridges. In practice, this means that

https://doi.org/10.52202/085713-4053

CURLY-FM is significantly more efficient to train as shown in table 15. Finally, note, both Shen et al. [2025] and CURLY-FM can model non-gradient field dynamics, but only the latter is simulation-free.

Latent SDEs. Bartosh et al. [2024] and Bartosh et al. [2025] consider Latent SDEs for learning stochastic dynamics with simulation-free training without explicitly solving Schrödinger Bridge problem as considered by CURLY-FM framework. This means that unlike CURLY-FM, work on Latent SDEs does not consider minimizing KL divergence to find an optimal path measure with respect to the reference process. Further, SDEs in CURLY-FM are not Latent SDEs. Whilst building a computational approach to solving the Schrödinger Bridge problem with latent SDEs may be possible, this is an orthogonal research direction to our work.

C Algorithmic Details

In this section we detail python code for training and inference for reproducibility.

```python
for _ in range(n_iter):
    x0 = sample_x0(batch_size)
    x1 = sample_x1(batch_size)
    x0, x1 = coupling(x0, x1)
    t = torch.rand(batch_size).type_as(x0)
    eps = torch.randn_like(x0)
    lambda_t = (2* torch.sqrt(t * (1-t))) / sigma
    xt, xt_dot = get_xt_xt_dot(t, x0, x1, geodesic_model)
    xt = xt + eps * sigma
    vt = drift_model(xt, t)
    st = score_model(xt, t)
    flow_loss = torch.mean((vt - xt_dot) ** 2)
    score_loss = torch.mean((lambda_t * st + eps) ** 2)
    loss = flow_loss + score_loss
```

Listing 1: Python implementation of CurlyFM Score and Flow Training algorithm.

```python
x = torch.randn(batch_size, dim)
for t in torch.linspace(0, 1, 100)[:-1]:
    drift = drift_model(x,t) + score_model(x,t)
    x = drift * dt + sigma * torch.sqrt(dt) * torch.randn_like(x)
```

Listing 2: Python implementation of CurlyFM inference algorithm.

D Ocean Currents Dataset

We assess performance of CURLY-FM on real ocean currents dataset acquired from a HYbrid Coordinate Ocean Model (HYCOM) reanalysis released by US Department of Defense.

Dataset. This data consists of real ocean currents measurements in Gulf of Mexico acquired at 1km bathymetry, providing hourly ocean currents velocity fields for the geographic region between 98E and 77E in longitude and 18N to 32N in latitude at each day since January 1st, 2001.

Experimental Set-up. We leverage experimental set-up as presented in Shen et al. [2025], and focus on a specific time point at 17:00 UTC on June 1st, 2024, extracting ocean surface velocity field that contains a vortex. From a point near its center, we uniformly draw 1,000 initial positions whithin radius 0.05 and evolve them across nine time steps such that $\Delta t = 0.9$, computing velocities by the nearest grid node for ~ 111 observations per time step.

Ablations. Tables 8 and 9 show ablations on using coupling cost in algorithm 2 for trajectory inference between drifter observations in ocean currents experiments. We observe that there is limited effect in integrating coupling cost over increased numbers of time steps between marginals.

Table 8: Comparison of CURLY-FM with and without the coupling in algorithm. 2. Without coupling refers to an independent coupling. The results are averaged over three seeds.

Metric	Method	t_2	t_4	t_6	t_8
Cos. Dist.	CURLY-FM (without coupling)	0.230 ± 0.003	0.017 ± 0.001	0.002 ± 0.000	0.002 ± 0.000
	CURLY-FM	0.231 ± 0.004	0.017 ± 0.001	0.002 ± 0.000	0.002 ± 0.000
L_2 cost	CURLY-FM (without coupling)	0.152 ± 0.003	0.099 ± 0.001	0.132 ± 0.007	0.187 ± 0.012
	CURLY-FM	0.151 ± 0.004	0.098 ± 0.001	0.135 ± 0.010	0.178 ± 0.017

Table 9: Comparison of different numbers of times (n) used to compute the coupling cost in algorithm 2. The times are equispaced except for $n = 1$, where they are drawn uniformly at random. The results are averaged over three seeds.

Metric	Method	t_2	t_4	t_6	t_8
Cos. Dist.	CURLY-FM (n=1)	0.231 ± 0.004	0.017 ± 0.001	0.002 ± 0.000	0.002 ± 0.000
	CURLY-FM (n=3)	0.230 ± 0.002	0.017 ± 0.001	0.002 ± 0.000	0.002 ± 0.000
	CURLY-FM (n=5)	0.229 ± 0.005	0.018 ± 0.001	0.002 ± 0.000	0.002 ± 0.000
	CURLY-FM (n=10)	0.227 ± 0.005	0.017 ± 0.001	0.002 ± 0.000	0.002 ± 0.000
L_2 cost	CURLY-FM (n=1)	0.151 ± 0.004	0.098 ± 0.001	0.135 ± 0.010	0.178 ± 0.017
	CURLY-FM (n=3)	0.153 ± 0.002	0.101 ± 0.002	0.132 ± 0.004	0.184 ± 0.011
	CURLY-FM (n=5)	0.153 ± 0.003	0.104 ± 0.002	0.131 ± 0.001	0.193 ± 0.011
	CURLY-FM (n=10)	0.150 ± 0.004	0.101 ± 0.017	0.130 ± 0.007	0.191 ± 0.014

E Single Cell Datasets

E.1 Human Fibroblasts Dataset

We consider the human fibroblasts dataset [Riba et al., 2022] that contains genomic information about 5,367 cells observed across a fibroblast cell cycle. Cell data further contains information about cycling genes, and more specifically, their RNA velocities, which are used to estimate the reference RNA velocity field. Figure 7 shows a distribution of cell rotations and phases during a cell cycle process.

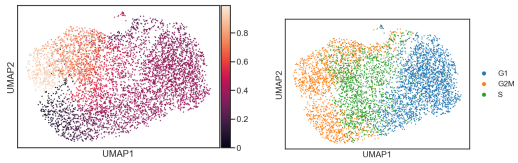
Pre-processing. Data is pre-processed by selecting top d variable genes from the data. Further, we use `scvelo` package [Bergen et al., 2020] to compute imputed unspliced (Mu) and spliced (Ms) expressions as well as velocity graph. We construct a cell-cell k-nn graph using `scv.pp.neighbours(adata)` in the joint spliced and unspliced expression space using default `scvelo` hyperparameters. For each gene and cell, we compute averaged fits moment and the centered second moment of spliced and unspliced genes as well as relared attributes. RNA velocities are computed using `scv.tl.velocity(adata)` fitting the stochastic transcriptional dynamics model [La Manno et al., 2018]. Finally, we compute low-dimensional embeddings for velocities using UMAP with `scv.tl.velocity_embedding(adata)` for our visualizations in figure 4d. For selecting top d highly variable genes, we use `sc.pp.highly_variable_genes(adata, n_top_genes=d)`.

E.2 Mouse Erythroid

We consider a dataset showing mouse gastrulation subset to erythroid lineage [Pijuan-Sala et al., 2019], representing the developmental pathway during which embryonic cells diversify into lineage-specific precursors, evolving into adult organisms. The data consists of 9,815 cells evolving through five lineage stages as shown in Figure 8, and it is available through `scvelo` package API.

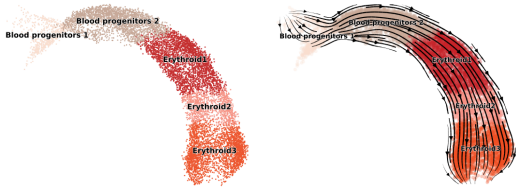
Pre-processing. Data is pre-processed using `scvelo` and `unitvelo` [Gao et al., 2022] package to compute imputed unspliced (Mu) and spliced (Ms) expressions as well as the velocity graph. With `unitvelo`, we construct the cell latent time used to approximate the differentiated time experienced by cells. Mouse erythroid velocity field is computed using `unitvelo.run_model()` following instructions from `unitvelo` documentation.

https://doi.org/10.52202/085713-4053



(a) Cell Rotations γ (b) Cell Phases

Figure 7: Human Fibroblasts Dataset.



(a) Cell Stages (b) RNA Velocity Field

Figure 8: Mouse Erythroid Dataset.

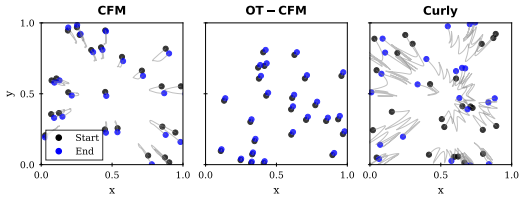


Figure 9: Trajectories of particles for CFD for different methods.

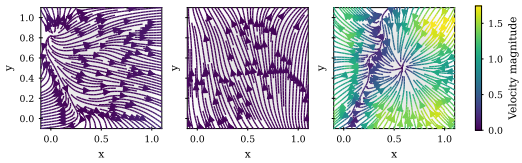


Figure 10: Learned CFD velocity field for CFM (left), OT-CFM (middle), and CURLY-FM (right).

Filtering RNA Velocities. To address tail-effects of noisy single-cell data, we filter RNA velocities by down-weighting distant neighbors in the k-NN estimate at x_t and injecting small Gaussian noise. We construct new estimate f_t^* by interpolating between k-nn velocity estimate f_t and noise such that $f_t^* = (1 - w_\gamma(x_t)) * f_t + w_\gamma(x_t) * \mathcal{N}(0, 0.1)$. The weight $w_\gamma(x_t) \in [0, 1]$ is given by a sigmoid of the distance between the k-NN distance and a threshold hyperparameter γ, so that larger distances yield larger $w_\gamma(x_t)$ and thus stronger penalization of distant neighbors.

F Computational Fluid Dynamics Dataset

Experimental details for CFD. We conducted our CFD experiments using the two-dimensional decaying Taylor-Green vortex (2DTGV) dataset provided by LagrangeBench [Toshev et al., 2023]. We subsampled 2000 particles and considered five equispaced marginals (snapshot distributions over particle positions). The goal was to perform trajectory inference from unordered population snapshots. Since instantaneous velocity is not directly observed, we constructed the reference drift field using finite differences. This aligns with how derived physical quantities—such as velocity or energy—are computed from particle positions in Lagrangian PDE solvers.

We considered a dataset split of $[80\%, 20\%]$ across the train and test sets, respectively. For the 2000 particles, this resulted in 1600 being used for training and 400 for testing. All other hyperparameters of CURLY-FM were the same as for the other experiments.

F.1 Ablations on CFD

Figure 9 illustrates the trajectories of 25 particles under CFM, OT-CFM, and CURLY-FM. Both CFM and OT-CFM tend to produce straight, relatively short paths, most notably in the case of OT-CFM, indicating a preference for minimal transport effort. In contrast, CURLY-FM learns longer, more intricate trajectories that better resemble the expected fluid dynamics.

Figure 10 visualizes the velocity fields inferred by each method. While CFM and OT-CFM yield smoother and simpler velocity patterns, CURLY-FM captures a richer and more structured field. This complexity reflects closer alignment with the reference field and suggests improved physical fidelity. The resulting velocity field from CURLY-FM more accurately models the underlying dynamics, adhering to both data-driven transport and the governing reference flow.

Tables 10 and 11 show CURLY-FM without coupling and for different number of times used to evaluate the coupling cost, respectively. In particular, table 10 shows that the coupling based on minimizing the kinetic energy only marginally improves performance for CFD experiments. Furthermore, table 11 shows that there is little to no benefit in using additional times to approximate the coupling cost in algorithm 2.

Table 10: Comparison of CURLY-FM with and without the coupling in algorithm. 2. Without coupling refers to an independent coupling. The results are averaged over five seeds.

Method	Cos. Dist. ↓	MSE ↓
CURLY-FM without coupling	0.214 ± 0.007	0.092 ± 0.007
CURLY-FM	0.189 ± 0.027	0.095 ± 0.003

Table 11: Comparison of different numbers of times (n) used to compute the coupling cost in algorithm 2. The times are equispaced except for $n = 1$, where they are drawn uniformly at random. The results are averaged over five seeds.

Method	Cos. Dist. ↓	MSE ↓
CURLY-FM (n=1)	0.189 ± 0.027	0.095 ± 0.003
CURLY-FM (n=3)	0.185 ± 0.027	0.092 ± 0.003
CURLY-FM (n=5)	0.179 ± 0.020	0.091 ± 0.005
CURLY-FM (n=10)	0.182 ± 0.029	0.095 ± 0.006

G Further Experimental Details

G.1 Single-marginal Set-up

$\varphi_{t,\eta}(x_0, x_1, t)$ and $v_{t,\theta}(x_t, \eta)$ design. Both $\varphi_{t,\eta}(x_0, x_1, t)$ and $v_{t,\theta}(x_t, \eta)$ are designed as MLP models with 3 layers. We select MLP dimensions based on the number of chosen highly variable genes d. For $\varphi_{t,\eta}(x_0, x_1, t)$ we choose $d_{in} = 2 \times d$ and $d_{out} = d$. For $v_{t,\theta}(x_t, \eta)$, we choose dimensions of $d_{in} = d$. The dataset is split as $[80\%, 10\%, 10\%]$ across training, validation, and test.

Training. All CURLY-FM and baseline experiments are run using $lr = 10^{-4}$ learning rate and Adam optimizer with default β_1, β_2, and ϵ values across three seeds and with 1,000 epochs split into 500 epochs to train $\varphi_{t,\eta}$ followed by 500 epochs to train $v_{t,\theta}$.

Baselines. TrajectoryNet was run with 250 epochs with the Euler integrator with 20 timesteps per timepoint. We use 250 epochs to limit the experimental time and the number of function evaluations to roughly $5\times$ that of simulation-free methods. We use a batch size of 256 samples. We use a Dormand-Prince 4-5 (dopri5) adaptive step size ODE solver to sample trajectories with absolute and relative tolerances of 10^{-4}.

Compute. All experiments were conducted using a mixture of CPUs and A10 GPUs.

G.2 Multi-marginal set-up

In the multi-marginal setting, we train by randomly sampling interpolation times t within intervals corresponding to each adjacent pair of marginals—for example these intervals will be [0,1] and [1,2] if our marginals lie at $t \in \{0, 1, 2\}$. For each interval $[t_i, t_{i+1}]$, we sample a random time t, then compute both the neural interpolant x_t and its derivative $\dot{x}_t$. We also compute global time $\hat{t} = t_i + t$, if say $t = 0.5$, the global time for the second marginal pair will be $\hat{t} = 1 + 0.5 = 1.5$. This effectively ensures that neural interpolant and its derivative match the global axis of time. We then connect all the marginals by concatenating neural interpolant, its derivative and global times into single tensors and use these as inputs to train our neural path interpolant in algorithm 1 and the drift in algorithm 2.

$\varphi_{t,\eta}(x_0, x_1, t)$ and $v_{t,\theta}(x_t, \eta)$ design. We design $\varphi_{t,\eta}(x_0, x_1, t)$ and $v_{t,\theta}(x_t, \eta)$ as MLPs as shown in table 12 across all multi-marginal experiments. CFD experiments additionally use four residual connection blocks with no dropout. We select MLP dimensions based on the number of chosen highly variable genes d. For $\varphi_{t,\eta}(x_0, x_1, t)$ we choose $d_{in} = 2 \times d$ and $d_{out} = d$. For $v_{t,\theta}(x_t, \eta)$, we choose dimensions of $d_{in} = d$.

Training. We show training details for multimarginal set-up in table 12 and compute test metrics on left-out marginals. We select $k = 20$ neighbours for ocean currents and CFD experiments and $k = 30$ neighbors for mouse erythroid experiments to compute ground-truth velocities.

Compute. All experiments were conducted using a mixture of CPUs and A10 NVIDIA GPUs.

https://doi.org/10.52202/085713-4053

Table 12: Overview of model design and training hyperparameters across multimarginal experiments.

	Mouse Erythroid		**Ocean Currents**		**CFD**	
	$\varphi_{t,\eta}(x_0, x_1, t)$	$v_{t,\theta}(x_t)$	$\varphi_{t,\eta}(x_0, x_1, t)$	$v_{t,\theta}(x_t)$	$\varphi_{t,\eta}(x_0, x_1, t)$	$v_{t,\theta}(x_t)$
Channels	256	256	64	64	64	64
Batch size	256	256	64	64	256	256
Epochs	2k	3k	5k	3k	1.5k	1.5k
Learning rate	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}

G.3 Cost of OT plan

The cost of the optimal plan π^* is intractable as it would in general require computing costs over stochastic paths. Consequently, we make several approximations to these couplings that enable faster throughput as offered in simulation-free training. In particular, we make three approximations to the cost $c(x_0, x_1)$ between two points, and one on the coupling given this cost:

- Algorithm 1 has learned a path or set of paths for the means of Gaussians in proximity to the optimal $\mathbb{Q}_t(x_t|x_0, x_1)$
- We consider low stochasticity σ setting, and therefore the cost is close to the distance of the means travel
- The integral of the squared length of the curve is approximated empirically using a Monte Carlo estimate

After approximating $c(x_0, x_1)$, we use we use mini-batch OT to approximate the entropic OT problem following set-up in Tong et al. [2024b]. Interestingly, in the low stochasticity setting, Tong et al. [2024b] found that mini-batch OT with no stochasticity was empirically a better approximator of π^* in the Gaussian case. This is because mini-batch transport adds some amount of "entropy" to the plan due to its approximation. We follow this approximation in our setting.

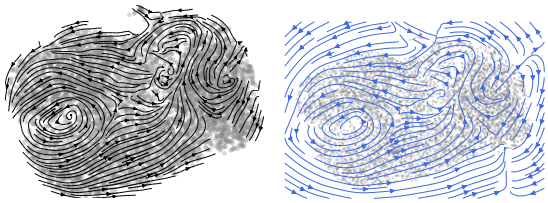
G.4 Further details on ground-truth velocity estimate

On use of kernels. We highlight that in practical settings we investigate, a continuous ground truth reference field does not exist. In our applications, we only have access to the reference field at discrete points in space and time that correspond to ground truth data at the different time marginals. Consequently, we use a kernel to build a continuous reference field $f_t(x_{t,\eta})$.

Reference drift velocity estimate f_t. In our considered setting, we assume access to the ground truth reference drift at samples $x_0 \sim \mu_0$ and $x_1 \sim \mu_1$ as part of the problem setup. Note that in the high-impact application domains we consider, such as trajectory inference in single-cell data, we have access to the RNA velocity [Riba et al., 2022, Bergen et al., 2020], which is assumed to be a reasonable estimate (up to a scaling factor) of the SDE velocity [Tong et al., 2020]. As we need to estimate CURLY-FM everywhere in space and time during training, we construct a smoothed version of the reference drift by using a kernel κ_t. This allows us to construct the reference drift $f_t(\mathbf{x_t})$—using knowledge from the ground truth reference drift in the existing dataset—in places where there are no ground truth samples. Consequently, we take f_t in these intermediate points as our ground truth reference drift. We further highlight that kernel estimate is not used in cases where ground-truth velocities are given on a continuous domain.

Ablation on kernel estimate accuracy. To show that estimate accuracy does not effect our findings, we conduct an ablation study by constructing a noisy reference drift f_t^{noise} for various noise levels $\beta \in [0, 1]$. We show results in table 13 for the Ocean Currents dataset. The noisy reference drift is obtained as a linear combination of the ground truth reference drift f_t and noise from a standard gaussian distribution ($f_t^{\text{noise}} = (1-\beta) * f_t + \beta * \text{noise}$). We find that the performance between $\beta = 0$ (no noise, i.e. regular CURLY-FM) and $\beta = 0.25$ are similar whilst the performance for $\beta = 0.5$ and higher gradually becomes worse, as the noise dominates over the ground truth reference drift in f_t^{noise} . This shows that CurlyFM is robust to moderate amounts of noise added to the ground truth reference drift.

Reconstructed field. We provide comparison to ground truth velocity field and k-nn estimate for the human fibroblasts data. From figure 12 it is clear that our approach faithfully reconstructs ground-truth RNA velocities.



(a) Ground truth velocity (b) k-nn velocity estimate

Figure 12: Ablation and estimate of ground truth velocity.

Table 13: Noisy reference drift ablation.

β	Cos. Dist. ↓	L_2 ↓	$\mathcal{W}_2$ ↓
0.00	0.062 ± 0.003	0.143 ± 0.010	0.034 ± 0.006
0.25	0.057 ± 0.033	0.021 ± 0.036	0.051 ± 0.030
0.50	0.087 ± 0.047	0.301 ± 0.085	0.091 ± 0.046
0.75	0.261 ± 0.123	0.381 ± 0.120	0.145 ± 0.062
1.00	0.428 ± 0.157	0.445 ± 0.121	0.237 ± 0.079

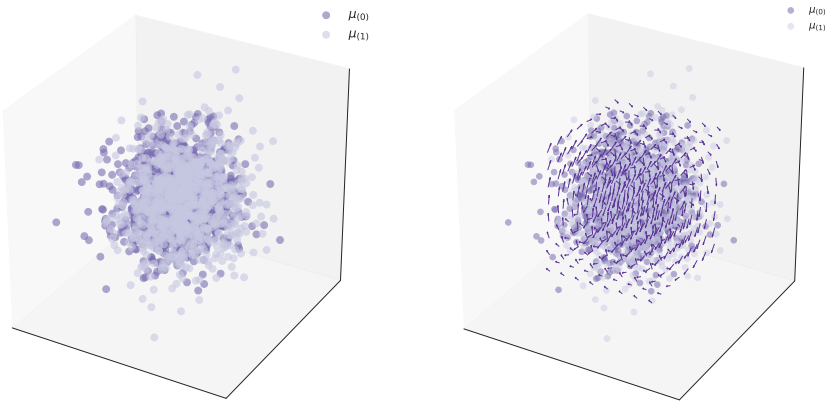
G.5 On the tradeoff between directly computing OT plan vs. iterative refinement

Two main strategies exist for approximating the static (possibly regularized) optimal transport coupling. In this work we primarily use the mini-batch approximation. However, there are also iterative-refinement type approaches that are unbiased in the infinite limit like those presented in De Bortoli et al. [2021b], Shi et al. [2024]. These methods work by creating an "outer loop" where the bridge is simulated in one direction, then matched in the other to create an iterative refinement to match marginals. While this is possible to incorporate into our framework, it is not as suitable for our application domain, where we assume small stochasticity level. Iterative approaches are not suitable for small stochasticity levels because the number of iterations of iterative proportional fitting or iterative Markov fitting (IPF / IMF) for suitable convergence depends is inversely proportional to the stochasticity [Tong et al., 2024b, Shi et al., 2024]. Indeed, at the zero noise limit, the IMF approach collapses to a rectified flow, which only solve the OT problem in limited domains e.g. 1D [Liu et al., 2023b], and a few select Gaussian settings [Bansal et al., 2025]. For low noise levels the convergence rate is significantly slower.

H Supplementary Results

H.1 Additional Synthetic Experiments

CURLY-FM robustness. To further assess CURLY-FM robustness, we designed a toy experiment with an analytical Schrödinger Bridge solution and non-gradient dynamics based on [Tong et al., 2024b, De Bortoli et al., 2021b, Shi et al., 2024]. We bridge two Gaussians in the presence of a spiral reference field across various dimensions d and stochasticity levels $\sigma = g_t = 0$. Specifically, we initialize two Gaussians in 20 dimensions centered at $\mu_0 = [-0.1, 0, 0, \ldots, 0]$ and $\mu_1 = [0.1, 0, 0, \ldots, 0]$ with standard deviations $\sigma_0 = \sigma_1 = [1, \ldots, 1]$. We also define a ground truth transport field, which unlike the standard OT field, has an additional rotational component. We note that this has equivalent marginal probability distributions, but also rotates around the origin in the second and third dimensions. This allows us to test how well our method works in a simple toy setting.



(a) Source and target distribution (b) Velocity field

Figure 13: Example of a chosen synthetic setting in case of 3-dimensions.

https://doi.org/10.52202/085713-4053

Table 14: CURLY-FM ablation in synthetic setting and comparison to OT-CFM, SF2M and SB-CFM.

(a) CURLY-FM

Dim	σ	$\mathrm{KL}(p_1, q_1)$	Mean KL	Cos. Dist.	$W_2(p_1, q_1)$
3	0.0	0.043	0.024	0.001	0.192
3	0.1	0.040	0.017	0.038	0.610
3	1.0	0.049	0.049	0.519	0.613
5	0.0	0.021	0.017	0.010	0.871
5	0.1	0.041	0.018	0.018	0.880
5	1.0	0.065	0.038	0.522	0.887
20	0.0	0.258	0.178	0.022	3.887
20	0.1	0.250	0.169	0.029	3.891
20	1.0	0.358	0.234	0.541	3.749

(b) OT-CFM

Dim	σ	$\mathrm{KL}(p_1, q_1)$	Mean KL	Cos. Dist.	$W_2(p_1, q_1)$
3	0.0	0.033	0.024	0.999	0.369
3	0.1	–	–	–	–
3	1.0	–	–	–	–
5	0.0	0.031	0.021	1.000	0.879
5	0.1	–	–	–	–
5	1.0	–	–	–	–
20	0.0	0.201	0.165	0.997	3.921
20	0.1	–	–	–	–
20	1.0	–	–	–	–

(c) SF2M

Dim	σ	$\mathrm{KL}(p_1, q_1)$	Mean KL	Cos. Dist.	$W_2(p_1, q_1)$
3	0.0	–	–	–	–
3	0.1	0.017	0.017	0.480	0.602
3	1.0	0.024	0.038	0.509	0.603
5	0.0	–	–	–	–
5	0.1	0.038	0.040	0.516	1.118
5	1.0	0.108	0.058	0.518	1.127
20	0.1	–	–	–	–
20	0.1	0.537	0.481	0.515	4.194
20	1.0	0.572	0.525	0.510	4.347

(d) SB-CFM

Dim	σ	$\mathrm{KL}(p_1, q_1)$	Mean KL	Cos. Dist.	$W_2(p_1, q_1)$
3	0.0	–	–	–	–
3	0.1	0.029	0.012	0.999	0.428
3	1.0	0.028	0.015	0.998	0.418
5	0.0	–	–	–	–
5	0.1	0.026	0.012	1.000	0.902
5	1.0	0.024	0.017	0.997	0.878
20	0.0	–	–	–	–
20	0.1	0.233	0.156	0.995	3.850
20	1.0	0.264	0.165	0.998	3.876

Table 15: Computational efficiency comparison

Method	DM-SB	Vanilla-SB	TrajectoryNet	SBIRR	CURLY-FM **(ours)**
Hours	15.44	0.43	7.44	4.67	**0.06**

```python
def psi_xt(xt):
    rot_speed = 1 * np.pi
    velocity = torch.stack([
        (mu_1 - mu_0) * torch.ones_like(xt[..., 0]),
        rot_speed *  xt[...,2],
        rot_speed * -xt[...,1]
    ], dim=-1)
    if xt.shape[-1] > 3:
        extra_zeros = torch.zeros_like(xt)[..., 3:]
        velocity = torch.cat([velocity, extra_zeros], dim=-1)
    return velocity
```

Listing 3: Python implementation of ground truth vector field.

We compare against the closed-form solution and OT-CFM, measuring KL divergence, Wasserstein distance, and the cosine distance to the ideal rotational angle. Results in table 14 confirm CURLY-FM performs best in low-dimensional, low-stochasticity settings, achieving high cosine similarity.

Comparison to baselines in synthetic setting. We further compare CURLY-FM robustness in synthetic setting to SF2M and SB-CFM baselines. We find that CURLY-FM outperforms both SF2M and SB-CFM on stochasticity $\sigma = 0.1$ on cosine distance. In case of $\sigma = 0.1$, CURLY-FM outperforms SB-CFM, and achieves similar cosine distance with SF2M while providing better $\mathcal{W}_2$ metrics.

H.2 Computational Efficiency

We provide comparison across DM-SB, Vanilla-SB, TrajectoryNet and SBIRR baselines in terms of computational efficiency evaluated on the ocean currents experiments (shown in table 15). We further report results for the 2D and 10D settings for TrajectoryNet and CURLY-FM in table 16. We find that TrajectoryNet is considerably slower than CURLY-FM and is not a scalable approach. This is evident in that training TrajectoryNet takes, on average, 11 times longer than CURLY-FM in the 2D case, and 17 times longer in the 10D case. Moreover, as dimensionality increases from 2D to 10D, CURLY-FM incurs an increase in computational cost by 1.7x, whereas TrajectoryNet incurs

Table 16: Computational efficiency comparison

Method	$d = 2$ (seconds)	$d = 10$ (seconds)
TrajectoryNet	17005 ± 110	$^{*}43080 \pm 65$
CURLY-FM	$\mathbf{1429 \pm 31}$	$\mathbf{2471} \pm 10$

Table 17: GSBM Control Points and Efficiency

Method	Cos. Dist. ↓	L_2 ↓	$\mathcal{W}_2$ ↓	Train (s)
GSBM 2	0.279 ± 0.006	0.395 ± 0.008	0.337 ± 0.009	37
GSBM 10	0.158 ± 0.166	0.130 ± 0.060	0.247 ± 0.144	68
GSBM 15	0.075 ± 0.017	0.134 ± 0.030	0.248 ± 0.045	68
GSBM 30	0.088 ± 0.034	0.077 ± 0.023	0.225 ± 0.085	68
GSBM 60	0.269 ± 0.126	0.200 ± 0.124	0.400 ± 0.065	68
CURLY-FM	0.062	0.143	0.034	176

at least a 2.5x increase in computational cost. We use (*) to denote runs that did not finish within allotted resource allocation time.

H.3 Further Baseline Comparisons

H.3.1 Generalized Schrödinger Bridge Matching

In this section we provide a more extensive comparison of CURLY-FM to Generalized Schrödinger Bridge Matching (GSBM) [Liu et al., 2024].

Experimental Set-up. We make two modifications to improve GSBM in our setting for the fairest comparison to CURLY-FM: We use our loss instead of GSBM loss in equation 6a to control splines and to incorporate the signal from a reference drift directly. We fix σ to the target σ as we assume a constant σ and this is the optimum for each bridge. We keep the iterative algorithm as found in the original GSBM paper. We follow the formulation in Tong et al. [2024b] which separately learns the deterministic flow and stochastic score functions, which then can be added together to calculate the stochastic drift or used separately to integrate deterministically. We note that we do this so that we can experiment with different stochasticity and simulate forward or backwards in time.

Number of control points. We initially set the number of control points to 2. We provide an additional ablation over the number of control points in GSBM, noting that the default number in GSBM is 15. We also find 15 provides the best tradeoff. We find that more or fewer control points do not improve performance. Otherwise we keep the same hyperparameters as CURLY-FM in terms of iterations, kernels, batch size, learning rate, models, etc. for fair comparison. Table 17 compares CURLY-FM to GSBM on the Ocean currents dataset with varying number of control points

On loss used in baseline comparison. We clarify that the loss (6a) in GSBM cannot take into account a non-zero reference drift directly. Loss 6a only considers potential functions $\mathbf{V}_t(x_t)$ and the kinetic energy of u_t, and therefore is theoretically problematic in our setting. We therefore use CURLY-FM loss function, which we believe is a more fair comparison. For completeness we further include a GSBM baseline with loss 6a found in the GSBM paper. We find that CURLY-FM is able to outperform both variations of GSBM on our oceans dataset across the three main quantitative metrics of interest. We further find our modification of GSBM's loss allows it to better match the non-zero reference drift as evidenced by a lower cosine distance, while also performing slightly better in $\mathcal{W}_1$ distance.

Table 18: GSBM loss comparison

Method	Cos. Dist. ↓	L_2 ↓	$\mathcal{W}_2$ ↓
GSBM (our loss)	0.075 ± 0.017	0.134 ± 0.030	0.248 ± 0.045
GSBM (6a)	0.083 ± 0.037	0.134 ± 0.029	0.201 ± 0.058
CURLY-FM	0.070 ± 0.001	0.107 ± 0.003	0.052 ± 0.004

https://doi.org/10.52202/085713-4053

H.3.2 Metric Flow Matching

For completeness of our analysis, we additionally report results for Metric Flow Matching (MFM) [Kapuśniak et al., 2024]. MFM introduces a geometric bias by enforcing interpolations that remain close to the underlying data manifold, effectively learning smooth geodesic paths that reflect intrinsic geometric structure. The two-stage learning strategy in CURLY-FM is directly adapted from MFM — replacing the manifold-constrained interpolations with regression against the reference non-gradient dynamics. In other words, CURLY-FM can be viewed as introducing an alternative inductive bias on trajectories: whereas MFM constrains paths to lie on the manifold defined by data, CURLY-FM enforces a bias on *velocities*, aiming to learn reference-consistent vector fields.

Since the core algorithmic structure of CURLY-FM (Algorithm 1) mirrors that of MFM, the two formulations are in fact compatible and can be combined—leveraging manifold-constrained interpolants together with velocity-based biases. We leave this promising direction for future work.