MULTI-MARGINAL STOCHASTIC FLOW MATCHING FOR ALIGNMENT OF HIGH-DIMENSIONAL SNAPSHOT DATA AT IRREGULAR TIME POINTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Modeling the evolution of high-dimensional systems from limited snapshot observations at irregular time points poses a significant challenge in quantitative biology and related fields. Traditional approaches often rely on dimensionality reduction techniques, which can oversimplify the dynamics and fail to capture critical transient behaviors in non-equilibrium systems. We present a novel extension of simulation-free score and flow matching methods to the multi-marginal setting, enabling the alignment of high-dimensional data measured at non-equidistant time points without reducing dimensionality. The use of measure-valued splines enhances robustness to irregular snapshot timing, and score matching prevents overfitting in high-dimensional spaces. We validate our framework on several synthetic and benchmark datasets and apply it to single-cell perturbation data from melanoma cell lines and gene expression data collected at uneven time points.

1 INTRODUCTION

Understanding cellular responses to perturbations is a fundamental challenge in quantitative biology, with significant implications for fields such as developmental biology, cancer research, and drug discovery (Altschuler & Wu, 2010; Saeys et al., 2016). Modeling these responses requires capturing complex stochastic dynamics in high-dimensional cellular states, which evolve over time under the influence of both deterministic and random factors. Developing generative models that accurately represent these dynamics is crucial for simulating cellular behavior and predicting responses to new perturbations. A common approach to modeling such systems is through stochastic differential equations (SDEs), particularly the Langevin equation as an Itô SDE (Gardiner, 1985; Risken, 1996); The evolution of the cellular state $X(t) \in \mathbb{R}^d$ can be described by

$$dX(t) = u_t(X(t))\,dt + g(t)\,dW(t), \tag{1}$$

where $u_t(x)$ is the drift term representing deterministic dynamics, $g(t)$ is the diffusion coefficient capturing stochastic fluctuations, and $W(t)$ is a Wiener process modeling random noise. At the population level, the corresponding probability density function $p(x,t)$ evolves according to the Fokker-Planck equation (Risken, 1996):

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

where $p_t(x)$ is shorthand for $p(x,t)$, $\nabla\cdot$ denotes the divergence operator, and Δ is the Laplacian operator.

In practice we only observe the system through snapshot measurements at discrete, possibly irregular time points $t_0 < t_1 < \cdots < t_M$, providing samples from the marginal distributions $\rho_i = p_{t_i}(x)$ (Schofield et al., 2023). Therefore, we lack trajectory data that would reveal how individual states evolve between these snapshots due to the destructive nature of single-cell measurements. This raises a fundamental question: *among the infinitely many stochastic processes that could connect these observed marginals (Weinreb et al., 2018), which one is the most likely?*

1.1 LEAST ACTION PRINCIPLE

To address this problem, we turn to the theory of *Optimal Transport* (OT) (Villani, 2009), which seeks the most efficient way to transform one probability distribution into another. In the simplest case of two marginals ρ_0 and ρ_1, OT aims to find a transport map T that minimizes the cost functional:

$$\min_T \int \|x - T(x)\|^2 \, d\rho_0(x) \quad \text{subject to } T_{\#}\rho_0 = \rho_1, \tag{3}$$

where $T_{\#}\rho_0$ denotes the pushforward of ρ_0 under T. The Kantorovich's generalized formulation of (3) is a linear programming problem over the set of joint probability distributions, leading to the definition of the Wasserstein-2 distance:

$$W_2^2(\rho_0, \rho_1) = \min_{\pi \in \Pi(\rho_0, \rho_1)} \int \|x - y\|^2 \, d\pi(x, y), \tag{4}$$

where $\Pi(\rho_0, \rho_1)$ is the set of joint distributions with marginals ρ_0 and ρ_1. While OT provides a deterministic model based on the principle of least action—finding the shortest path or geodesic in the space of probability distributions—it does not account for the inherent stochasticity of biological systems (Horowitz & Gingrich, 2020). Cells are subject to both extrinsic noise, such as variations in initial conditions and environmental inputs (Hilfinger & Paulsson, 2011), and intrinsic noise arising from the thermodynamic uncertainty in biochemical reactions (Mitchell & Hoffmann, 2018).
To incorporate stochasticity and identify the most likely stochastic process connecting the observed marginals, we consider the entropic-regularized optimal transport problem, a particular case of the *Schrödinger Bridge Problem* (SBP) (Schrödinger, 1931; Léonard, 2014). The SBP seeks the stochastic process that minimally deviates from a prior—typically a Brownian motion—while matching the observed marginals. It can be considered a general statistical inference and model improvement methodology in which one updates the probability of a hypothesis based on the most recent observations while making the fewest possible assumptions beyond the available information (Pavon et al., 2021). This approach aligns with Occam's razor principle and aims to find the simplest stochastic process that explains the data with minimal adjustment to our prior belief.

Extension to Multiple Marginals: Extending this rationale to the multi-marginal (MMOT) case with arbitrary time points $t_0, t_1, \ldots, t_M$, we pose the same question: among all possible stochastic processes that could connect the observed marginal distributions $\{\rho_i\}_{i=0}^M$, which one is the most probable given our prior knowledge? This leads us to formulate the problem as finding the drift $u_t(x)$ that minimizes the cumulative transport cost and provides the smoothest and most efficient flow connecting the observed distributions over time, while ensuring robustness against overfitting. In summary, we require:

- Robustness Against Overfitting: By minimizing the total transport cost across all time intervals, we introduce only essential adjustments to match the observed marginals, preventing the model from overfitting to limited observations and ensuring that the inferred dynamics generalize well beyond the training data.
- Insensitivity to Timing of Snapshots: The formulation inherently accommodates arbitrary and irregular time points t_i, making it robust to the choice of measurement times. By focusing on the minimal action path that passes through the observed marginals, we capture the system's evolution without being constrained by the timing of data collection.
- Scalability in High Dimensions: While directly solving high-dimensional transport problems is computationally challenging (Benamou & Brenier, 2000; Peyré & Cuturi, 2019), our approach efficiently approximates the solution. By leveraging advances in simulation-free score and flow matching methods, we model the high-dimensional stochastic process directly in the ambient space, avoiding dimensionality reduction that could obscure important dynamical features.

1.2 LITERATURE REVIEW

Direct learning of the high-dimensional partial differential equation (2) is computationally prohibitive due to the complexity of integration and divergence computations in high-dimensional spaces (Benamou & Brenier, 2000; Peyré & Cuturi, 2019). Hence, current approaches typically

consider reduced-dimensional data representations with gradient-based drifts originating from developmental biology Weinreb et al. (2018); Schiebinger et al. (2019) where the focus is primarily on slow time scales and the assumption of low-dimensional manifold dynamics is often useful. In this context, dimensionality reduction tools such as t-SNE (Van der Maaten & Hinton, 2008), UMAP (McInnes et al., 2018), and PHATE (Moon et al., 2019) are extensively used to simplify the modeling. However, these techniques can obscure critical faster-scale dynamical information, introduce artifacts (Kiselev et al., 2019), and result in the loss of important biological information in the reduced, folded space.

Neural Ordinary Differential Equations (Neural ODEs) have emerged as a powerful tool for modeling continuous-time dynamics and connecting probability measures over time Chen et al. (2018a). This approach offers an alternative method by parameterizing the time derivative of the hidden state with a neural network, which is trained to approximate the drift term in the Fokker-Planck equation (2). While this method has been successfully applied Tong et al. (2020); Huguet et al. (2022) for modeling cellular dynamics and trajectory inference, it still operates primarily in reduced-dimensional spaces.

Recent multi-marginal approaches have attempted to handle multiple time points simultaneously. Chen et al. (2024) developed a deep multi-marginal momentum Schrödinger bridge approach that, while capable of working in high dimensions, requires expensive flow integration and memory-intensive caching of trajectories during training. Similarly, Albergo et al. (2023) proposed stochastic interpolants for multi-marginal modeling but still relies on ODE/SDE integration and marginal distributions as supervision signals, which becomes computationally challenging in high dimensions. These approaches share common limitations: they either require dimension reduction to handle computational complexity, or they depend on expensive numerical integration and trajectory generation during training.

Alternative approaches using generative models attempt to transform a simple distribution to an arbitrary target distribution. Variational Autoencoders (VAEs) (Kingma, 2013) learn an encoder-decoder pair, $q(z \mid x)$ and $p(x \mid z)$, such that the decoder can generate $x \sim \rho_1$ given samples $z \sim \rho_0$. Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) employ a generator-discriminator framework, where the generator $G(z)$ produces $x = G(z)$ with $x \sim \rho_1$ for $z \sim \rho_0$. While successful, these methods are limited by the simplicity of the source distribution ρ_0, often chosen to be uniform or normal for analytical convenience. Moreover, these models represent static transformations with no notion of time and cannot generate intermediate states at arbitrary time points, making them unsuitable for modeling dynamic processes where temporal evolution is crucial.

Although diffusion models (Ho et al., 2020; Song & Ermon, 2019) incorporate a time component by learning a denoising Markovian reverse process, their notion of "time" corresponds to a noise schedule rather than physical time. This limitation prevents them from capturing actual temporal dynamics or generating data at arbitrary time points not specified during training.

Our Approach We introduce Multi-Marginal Stochastic Flow Matching (MMSFM) to address these limitations by adapting recent developments in simulation-free approaches (Lipman et al., 2022; Tong et al., 2023a;b) to our setting. These methods learn p_t directly in the ambient space without dimensionality reduction or explicit simulation. However, their direct application to our multi-marginal setting requires careful adaptation to principles described in Section 1.1 to ensure robust learning and prevent overfitting.

Our key innovation lies in learning continuous spline measures through overlapping windows of consecutive marginals during training. Specifically, we process overlapping triplets $(\rho_i, \rho_{i+1}, \rho_{i+2})$ in a rolling fashion, where we demonstrate in Section 2.3 that a window size of two strikes an optimal balance between enforcing smoothness constraints and computational efficiency. This approach enables us to capture local dynamics across uneven time intervals, maintain consistency between overlapping windows, and generate intermediate states between observed snapshots, effectively creating a "motion picture" of the system's evolution. The overlapping nature of these learned flows ensures robustness against the specific choice of measurement times while preserving the high-dimensional structure of the data.

However, directly learning the high-dimensional partial differential equation (2) is computationally prohibitive due to the complexity of integration and divergence computations in high-dimensional spaces (Benamou & Brenier, 2000; Peyré & Cuturi, 2019). Consequently, many existing methods assume that the dynamics occur on a low-dimensional manifold and employ dimensionality reduction techniques to simplify the modeling. Techniques, such as t-SNE (Van der Maaten &

Hinton, 2008), UMAP (McInnes et al., 2018), and PHATE (Moon et al., 2019), are commonly used for visualization and unsupervised clustering, but they can obscure critical dynamical information and introduce artifacts (Kiselev et al., 2019). Aggressive dimensionality reduction may result in the loss of important biological information and can complicate the dynamics in the reduced, folded space. This challenge motivates the development of methods that can model stochastic processes bridging multiple marginal distributions directly in high-dimensional spaces, without relying on dimensionality reduction.

The problem we address involves modeling the evolution of complex, high-dimensional data over time, where observations are made at discrete, potentially irregularly spaced time points. At each of these time points, we have snapshot measurements capturing the system's state, but we lack information about the trajectories of individual data points between these snapshots. Our objective is to learn a continuous probability flow that matches the observed data at the given time points, while meeting several key requirements. First, we make no assumptions about the underlying distributions at each time point, allowing for flexibility in capturing diverse system states. Second, we model the dynamics directly in the original high-dimensional space, avoiding dimensionality reduction that could obscure dynamical features of the process. Finally, our method is robust against the choice of time points for data collection, allowing for accurate alignment even when time points are irregular or unevenly spaced. This approach enables us to capture the full complexity of the system's evolution without sacrificing the richness of the high-dimensional data or being constrained by the timing of data collection. This setting deviates from classical approaches, which often consider transformations from a simple source distribution to a target distribution, typically involving analytic forms and assuming access to trajectory data (Tritschler et al., 2019). However, in many practical scenarios, such as single-cell data collection, trajectory data of individuals within the snapshots is missing because the measurement process is destructive. Similarly, in observational studies on wildlife, animals may not be tagged and cannot be uniquely identified over time, effectively masking the trajectory data (Schofield et al., 2023).

Generative models approach the problem of transforming a simple distribution to an arbitrary target distribution. Variational Autoencoders (VAEs) (Kingma, 2013) learn an encoder-decoder pair, $q(z \mid x)$ and $p(x \mid z)$, such that the decoder can generate $x \sim \rho_1$ given samples $z \sim \rho_0$. Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) propose a generator-discriminator framework, where the generator $G(z)$ produces $x = G(z)$ with $x \sim \rho_1$ for $z \sim \rho_0$. Although successful, these methods are limited by the simplicity of the source distribution ρ_0, often chosen to be uniform or normal for analytical convenience. However, these models are static transformations with no notion of time and cannot generate intermediate states at arbitrary time points. This limitation makes them unsuitable for modeling dynamic processes where temporal evolution is crucial. While diffusion models (Ho et al., 2020; Song & Ermon, 2019) incorporate a time component by learning a denoising Markovian reverse process, the "time" in these models corresponds to a noise schedule rather than physical time. Consequently, they cannot capture the actual temporal dynamics of the system or generate data at arbitrary time points not specified during training.

To address these limitations, we propose a method to learn p_t in a stochastic and simulation-free manner while remaining in the ambient or high-dimensional space $\mathbb{R}^d$. Our approach is robust and independent of the specific time points chosen for data collection, allowing us to generate samples at arbitrary time points t, including those not present in the training data. This capability enables the generation of intermediate states between static frames, effectively creating a "motion picture" of the system's evolution. Our methodology extends Conditional Flow Matching (Lipman et al., 2022) and builds upon the works of Tong et al. (2023a;b), by generalizing them to multi-marginal settings with irregular time points. Specifically, we learn flows for overlapping subsets (e.g., triplets) $(\rho_i, \rho_{i+1}, \rho_{i+2})$ in a rolling window fashion. By learning multiple overlapping flows, which may differ based on control points and the choice of spline algorithm, the overall flow becomes more robust and stable by capturing local dynamics across uneven time intervals. This approach enables the alignment of high-dimensional distributions at multiple time points without aggressive dimensionality reduction.

In summary, our contributions are: i) We develop a framework for aligning multimarginal high-dimensional single-cell data with non-equidistant time points, while remaining in high-dimensional space. ii) We extend Conditional Flow Matching in Lipman et al. (2022) and build

upon the works of Tong et al. (2023b) to learn flows for overlapping triplets, enhancing robustness and stability in multi-marginal settings. iii) By incorporating stochasticity through score matching, we improve robustness and avoid overfitting in the ambient spaces that allows for further regulation if more information about the evolution of measure becomes available. iv) We demonstrate the effectiveness of our method on several synthetic and benchmark datasets and apply it to high-dimensional single-cell perturbation data from melanoma cell lines collected at uneven time points, capturing complex cellular dynamics without dimensionality reduction. This work opens new avenues for modeling cellular responses to perturbations, providing a computationally efficient and biologically accurate framework that handles the complexities of high-dimensional, stochastic biological systems. Moreover, our method can be used as a generative modeling technique to generate intermediate states between static frames at different time points, effectively creating a continuous 'motion picture' of the system's evolution.

2 Problem Formulation and Methodology

Formally, let $0 = t_0 < t_1 < \cdots < t_M = 1$ denote a sequence of time points, and let ρ_i be the probability distribution of the system state at time t_i in $\mathbb{R}^d$. Our data consists of snapshot measurements $X_{t_i} = \{x^{(j)} : x \sim \rho_i\}_{j=1}^{N_i}$, at these timepoints. The goal is to learn a continuous probability flow $p_t(x)$ for $t \in [0, 1]$, satisfying $p_{t_i} = \rho_i$ for all i, which describes the evolution of the system over time.

2.1 Dynamic formulation of the Wasserstein distance and Wasserstein Splines

Optimal Transport (OT) theory provides a mathematical framework for comparing and transforming probability distributions (Villani, 2009). The Wasserstein distance, also known as the Kantorovich distance, measures the minimal cost of transporting mass from one distribution to another. For two probability measures μ and ν on $\mathbb{R}^d$, the Wasserstein-2 distance is defined as:

$$W_2^2(\mu, \nu) = \inf_{\pi \in \Pi(\mu,\nu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\|^2 \, d\pi(x, y), \tag{5}$$

where $\Pi(\mu, \nu)$ denotes the set of all couplings (joint distributions) with marginals μ and ν. Benamou and Brenier (Benamou & Brenier, 2000) introduced a dynamic formulation of the Wasserstein distance, connecting OT with fluid dynamics:

$$W_2^2(\mu, \nu) = \inf_{p_t, u_t} \left\{ \int_0^1 \int_{\mathbb{R}^d} \|u_t(x)\|^2 \, p_t(x) \, dx \, dt \;\middle|\; \frac{\partial p_t}{\partial t} + \nabla \cdot (p_t u_t) = 0, \; p_0 = \mu, \; p_1 = \nu \right\}. \tag{6}$$

Traditionally, OT has focused on transporting mass between a source and a target distribution. However, in our setting, we need to interpolate between multiple distributions ρ_i at times t_i. While MMOT extends this framework to multiple distributions, computing MMOT plans becomes computationally challenging in high dimensions. Prior work (Chen et al., 2018b; Benamou et al., 2019) examined the formulation

$$\inf_{X_t} \int_0^1 \mathbb{E}\left[\left\| \ddot{X}_t \right\|^2 \right] dt, \tag{7}$$

termed P-splines by Chewi et al. (2021). Unfortunately this does not fit our needs because X_t here is considered to be a stochastic process whereas we need a deterministic flow. Moreover, these formulations are still quite computationally expensive given that we need to solve this problem within the training loop. Instead, Chewi et al. (2021) proposed *transport splines* as a method to efficiently obtain deterministic maps that smoothly interpolate between multiple distributions. The key idea is to sample points from the distributions ρ_i and apply a Euclidean interpolation algorithm between these points. Consider samples $\{X_{t_i}\}_{i=0}^M$. We aim to construct interpolated paths through all X_{t_i} by applying spline interpolation to the sampled points. The specific spline algorithm is left as a design choice for the user. Options include the natural cubic spline interpolation which minimizes the integral of the squared acceleration

$$\inf_{\gamma_t} \int_0^1 \mathbb{E}\left[\|\ddot{\gamma}_t\|^2 \right] dt, \tag{8}$$

where γ_t denotes a curve in space the second derivative (acceleration) of X_t with respect to time $\frac{d^2}{dt^2}\gamma_t$, and the cubic Hermite spline (Hermite & Borchardt, 1878) which represents each interval (x_i, x_{i+1}) as a the third-degree polynomial:

$$X(t) = (2t^3 - 3t^2 + 1)x_i + (t^3 - 2t^2 + t)x'_i + (-2t^3 + 3t^2)x_{i+1} + (t^3 - t^2)x'_{i+1} \tag{9}$$

where x_0, x_1 are the boundary constraints, and x'_0, x'_1 are the derivatives w.r.t. time at those points.

As an alternative perspective, transport splines can be viewed as compositions of optimal transport plans. In practice, we consider transport splines as compositions of OT plans. Let π be the MMOT plan over $(x_{t_0}, x_{t_1}, \ldots, x_{t_M})$. By applying a first-order Markov approximation, we decompose π into conditional plans

$$\pi(x_{t_0}, \ldots, x_{t_M}) \approx \pi(x_{t_0}, x_{t_1}) \prod_{i=2}^{M} \pi(x_{t_i} \mid x_{t_{i-1}}), \tag{10}$$

where $\pi(x_{t_i} \mid x_{t_{i-1}})$ specifies how to transport a point $x_{t_{i-1}} \sim \rho_{i-1}$ to the distribution ρ_i. By applying the transport spline procedure to batches of vectors $(X_{t_i})_{i=0}^M$, the conditional plans act as alignment operators, allowing us to construct Euclidean splines through optimally coupled points $(X^\star_{t_i})_{i=0}^M$. However, Euclidean interpolation may not capture the true geometry of the probability distributions, especially in high-dimensional, nonlinear settings. Additionally, the choice of spline algorithm affects the interpolation quality, particularly when the time intervals are uneven.

2.2 Simulation-Free Score and Flow Matching

We aim to model the stochastic process bridging the multiple distributions ρ_i by learning the underlying dynamics of the system in Equation (1) and the associated Fokker-Planck equation (2). Tong et al. (2023b) introduced a reparameterization of the drift $u_t(x)$ as

$$u_t(x) = u_t^\circ(x) + \frac{g^2(t)}{2} \nabla \log p_t(x), \tag{11}$$

where $u_t^\circ(x)$ is the deterministic component, and $\nabla \log p_t(x)$ is the score function of the density $p_t(x)$.

Substituting this reparameterization into the Fokker-Planck equation (2), we recover the continuity equation when $g(t) = 0$:

$$\frac{\partial p_t}{\partial t} = -\nabla \cdot (p_t u_t^\circ). \tag{12}$$

This observation allows us to decouple the learning of the deterministic drift $u_t^\circ(x)$ and the score function $\nabla \log p_t(x)$. Therefore, specifying $u_t^\circ(x)$ and $\nabla \log p_t(x)$ is sufficient to define the SDE drift $u_t(x)$. Tong et al. (2023b) proposed the unconditional score and flow matching objective:

$$\mathcal{L}(\theta) = \mathbb{E}_{t\sim\mathcal{U}(0,1), x\sim p_t(x)} \left[\|v_t(x;\theta) - u_t^\circ(x)\|^2 + \lambda(t)^2 \|s_t(x;\theta) - \nabla \log p_t(x)\|^2 \right], \tag{13}$$

where $v_t(x;\theta)$ and $s_t(x;\theta)$ are neural networks approximating the drift and score functions, respectively, and $\lambda(t)$ is a weighting function. However, $p_t(x)$ is unknown and thus directly computing $u_t^\circ(x)$ and $\nabla \log p_t(x)$ is challenging. To overcome this, Tong et al. (2023a) proposed a conditional formulation of the loss function:

$$\begin{aligned} \mathcal{L}(\theta) =& \mathbb{E}_{t\sim\mathcal{U}(0,1), z\sim q(z), x\sim p_t(x|z)} \left[\|v_t(x;\theta) - u_t^\circ(x|z)\|^2 \right] \\ &+ \lambda(t)^2 \mathbb{E}_{t\sim\mathcal{U}(0,1), z\sim q(z), x\sim p_t(x|z)} \left[\|s_t(x;\theta) - \nabla \log p_t(x|z)\|^2 \right], \end{aligned} \tag{14}$$

where z represents conditioning variables, and $x \sim p_t(x|z)$. In this conditional framework, $u_t^\circ(x|z)$ and $\nabla \log p_t(x|z)$ can be computed analytically or estimated empirically based on the conditional distribution $p_t(x|z)$. We can reconstruct the learned SDE drift using:

$$u_t(x;\theta) = v_t(x;\theta) + \frac{g^2(t)}{2} s_t(x;\theta), \tag{15}$$

and integrate it with given initial conditions x_0 to infer the trajectories that develop from those initial conditions.

2.3 Learning Overlapping Mini-Flows for Multi-Marginal Data

We aim to train a single ODE drift network $v_t(x;\theta)$ and a single score network $s_t(x;\theta)$ to learn *mini-flows* on overlapping $(k+1)$-tuples $(\rho_i, \rho_{i+1}, \ldots, \rho_{i+k})$ for $i = 0, 1, \ldots, M-k$ in a rolling window fashion. We aim to train an ODE drift network $v_t(x;\theta)$ and a score network $s_t(x;\theta)$ to learn an overall flow based on the *mini-flows* on overlapping $(k+1)$-tuples $(\rho_i, \rho_{i+1}, \ldots, \rho_{i+k})$ for $i = 0, 1, \ldots, M-k$ in a rolling window fashion. The resulting overall flow is more robust and stable by expanding the search space and capturing local dynamics within each window. This approach can be considered as a convolution over the mini-flows learned from each window. Because transport splines are ultimately just approximations for the true MMOT, the rolling windows provide a variation of perturbations in the approximated error from any single geodesic spline segment estimate. See Figure 2 for a visual representation of the variation of paths in an interval. Our method handles overlapping trajectories where $u^{\circ}_{t_i}(x) \neq u^{\circ}_{t_j}(x)$ for fixed x and $t_i \neq t_j$, accommodating the possibility that trajectories may cross over a point at different times in multi-marginal settings. In practice, we train using mini-batches of size b.

By incorporating stochasticity through score matching, we improve robustness and avoid overfitting in high-dimensional spaces. The score $\nabla_x \log p_t(x|z)$ allows the model to capture the inherent uncertainty and variability in the data. Using the log derivative trick, we see that the score is equivalent to $\nabla_x p_t(x|z)/p_t(x|z)$, indicating that the score nudges predictions towards more likely regions and thereby implicitly explores the region around the local per-sample flow. This efficiency allows us to remain in the ambient dimension d and sidestep dimensionality reduction strategies which often introduce information loss and additional complexities into the flow dynamics. Moreover, re-projecting the trajectories back into the ambient space introduces undesirable reconstruction artifacts.

The window size k is a hyperparameter balancing bias and variance. One can set $k = M$ to learn a single multi-marginal flow over all ρ_i for a less expensive transport plan but more overfitting risk. For our purposes, we choose $k = 2$ and learn mini-flows over triplets $(\rho_i, \rho_{i+1}, \rho_{i+2})$. Consequently, our conditioning variable z is a 3-tuple (x_i, x_{i+1}, x_{i+2}).

2.3.1 Window Size k and Spline Algorithm

We choose our window size $k = 2$ based on the properties of our chosen euclidean spline algorithm and considerations to the running time. We opt to use monotonic cubic Hermite splines instead of natural cubic splines for four main reasons. First, the guaranteed monotonicity of each piecewise cubic polynomial ensures no overshoot, thus removing overshooting from the conditional ODE flow regression target described in 2.3.3. Second, monotonic cubic Hermite splines by construction do not necessarily have a continuous second derivative. While this is a desired property for smoother curves (and in fact enforced for natural cubic splines), this condition can restrict the curve from taking a more direct path such as from a linear piecewise interpolation. Third, as the probability measure space equipped with the Wasserstein distance metric supports the triangle inequality, the length of a MMOT spline curve between any two distributions is necessarily greater than the length of the geodesic between those same distributions. Thus, we can expect that for a smaller window size, the length of a spline curve should be closer in distance to the sum of the lengths of the geodesic curves than that for a larger window size. Fourth, the specific coefficients describing each piecewise cubic polynomial are efficient to compute, scaling linearly in $\mathcal{O}(k)$ with the $k+1$ points to interpolate. For a window size k and M timepoints, the overlapping window routine computes $(M-k)k$ splines resulting in a total complexity of $\mathcal{O}((M-k)k)$. This is "maximized" when $k = M/2$ for a complexity of $\mathcal{O}(M^2)$, and "minimized" when $k = 1$ or $k = M-1$ for a complexity of $\mathcal{O}(M)$. As an added bonus, monotonic cubic Hermite splines are highly insensitive to control points that are not immediate neighbors. Thus, choosing a larger window size $k > 2$ does not meaningfully increase the amount of information captured by the spline. We include a more in-depth discussion of splines in Appendix A.1.

2.3.2 Transport Splines Sampling of z and Stratified Sampling of t

We sample z from a MMOT plan π using transport splines by first drawing samples $X_{t_i}, X_{t_{i+1}}, X_{t_{i+2}} \sim \rho_i, \rho_{i+1}, \rho_{i+2}$, where each X_{t_i} is a batch of i.i.d. samples from ρ_i. Then, we compute the MMOT plan given by the first-order Markov approximation (10):

$$\pi(x_{t_i}, x_{t_{i+1}}, x_{t_{i+2}}) = \pi(x_{t_i}, x_{t_{i+1}})\, \pi(x_{t_{i+2}} \mid x_{t_{i+1}}).$$

The initial plan $\pi(x_{t_i}, x_{t_{i+1}})$ is a standard OT plan w.r.t. the squared Euclidean distance $\|x_{t_i} - x_{t_{i+1}}\|^2$ as the cost function. Next, we compute the conditional map $\pi(x_{t_{i+2}} \mid x_{t_{i+1}})$ using

$$\pi(x_{t_{i+2}} \mid x_{t_{i+1}}) = \frac{\pi(x_{t_{i+1}}, x_{t_{i+2}})}{\pi(x_{t_{i+1}})} = \frac{\pi(x_{t_{i+1}}, x_{t_{i+2}})}{\int \pi(x_{t_{i+1}}, x_{t_{i+2}})\, dx_{t_{i+2}}}.$$

By working with empirical samples, we can replace the computationally expensive integration with a summation over $x_{t_{i+2}} \in X_{t_{i+2}}$. Moreover, when using empirical samples, this procedure can be considered as inducing an alignment $(X^\star_{t_i}, X^\star_{t_{i+1}}, X^\star_{t_{i+2}})$ describing a coupling such that each tuple $(X^{\star(j)}_{t_i}, X^{\star(j)}_{t_{i+1}}, X^{\star(j)}_{t_{i+2}})$ outlines the control points for the trajectory of $X^{\star(j)}_{t_i}$ through ρ_{i+1} and ρ_{i+2}, where $X^{\star(j)}_{t_i}$ denotes the jth element of $X^\star_{t_i}$. In practice, we implement this alignment by solving pairwise OT problems between the samples, and we sample from $X_{t_{i+1}}$ and $X_{t_{i+2}}$ with replacement, so $X^\star_{t_{i+1}}$ and $X^\star_{t_{i+2}}$ may contain repeated entries.

In the original source-target distribution pair setting, we sample $t \sim \mathcal{U}(0, 1)$. To accommodate our mini-flow method, we could sample $t \sim \mathcal{U}(t_i, t_{i+2})$ for the ith mini-flow. However, this approach is ineffective for training uneven time intervals—for example, $t_{i+1} - t_i \ll t_{i+2} - t_{i+1}$ or vice versa—leading to insufficient sampling from the smaller interval. To handle this, we adopt a stratified sampling strategy, sampling an equal number of time points from $\mathcal{U}(t_i, t_{i+1})$ and $\mathcal{U}(t_{i+1}, t_{i+2})$ to ensure balanced training across intervals. Specifically, for a total batch size of b, we sample $b/2$ time points from each interval.

2.3.3 Mini-Flow ODE and Score Regression Targets

Theorem 3 of Lipman et al. (2022) and Theorem 2.1 of Tong et al. (2023c) derive the ODE flow regression target for a Gaussian probability path $p_t(x) = \mathcal{N}(x \mid \mu_t, \sigma_t^2)$ a conditional Gaussian probability path $p_t(x \mid z) = \mathcal{N}(x \mid \mu_t, \sigma_t^2)$ as

$$u_t^\circ(x \mid z) = \frac{\sigma_t'}{\sigma_t}(x - \mu_t) + \mu_t' \tag{16}$$

where μ_t and σ_t are respectively the time-varying mean and standard deviation of the flow conditioned on z. The prime notation ($'$) denotes differentiation w.r.t. time t. We set $\mu_t = \mu_{i:i+2}(t)$ for a transport spline $\mu_{i:i+2}(t) : [t_i, t_{i+2}] \to \mathbb{R}^d$, constructed through the points in z via Euclidean spline interpolation. While transport splines are theoretically agnostic to the choice of spline algorithm, the selected algorithm significantly impacts performance. In practice, natural cubic splines may perform poorly when consecutive time intervals are large, due to constraints on the second derivative that prevent rapid changes in acceleration and curvature. This can result in loops–to slow down or speed up–and substantial deviations from the linear piecewise path. Applying Occam's Razor, we argue that the "optimal" spline path should closely approximate the linear piecewise path. Monotonic cubic Hermite splines (Fritsch & Carlson, 1980) satisfy this criterion by avoiding restrictions on second derivatives and ensuring the spline path between each control point pair $[x_{t_i}, x_{t_{i+1}}]$ is either entirely non-increasing or non-decreasing.

For σ_t we consider the case of Brownian bridges with constant diffusion $g(t) = \sigma$ and set $\sigma_t = \sigma\sqrt{t(1-t)}$ along the global time $t \in [0, 1]$. Alternatively, we can set σ_t based on the Brownian bridge of the mini-flow from $t_i = a$ to $t_{i+2} = b$, reparameterizing as $\sigma_t = \sigma\sqrt{r(t)(1 - r(t))}$, where $r(t) = \frac{t-a}{b-a}$. In this case, the derivative σ_t' must take into account the reparameterization, yielding $\sigma_t' = \frac{d\sigma_t}{dr} \cdot \frac{dr}{dt} = \frac{d\sigma_t}{dr} \cdot \frac{1}{b-a}$. Because $\mu_t, \sigma_t, \mu_t', \sigma_t'$ can be expressed analytically, we can directly compute these quantities and efficiently compute the regression target u_t° using Equation (16).

Given our Gaussian probability path, we can easily derive the score regression target as $\nabla \log p_t(x \mid z) = \frac{\mu_t - x}{\sigma_t^2}$, or alternatively $-\frac{\epsilon}{\sigma_t}$ for $\epsilon \sim \mathcal{N}(0, I)$.

2.3.4 MINI-FLOW SCORE MATCHING TARGET

Given our Gaussian probability path, we can easily derive the score regression target as $\nabla \log p_t(x \mid z) = \frac{\mu_t - x}{\sigma_t^2}$, or alternatively $-\frac{\epsilon}{\sigma_t}$ for $\epsilon \sim \mathcal{N}(0, I)$. However, this direct formulation does not protect against numerical instability when σ_t is small. We follow the approach used by Tong et al. (2023b) and take advantage of the user-defined weighting schedule $\lambda(t)$ to cancel out the division and learn the scaled target $\frac{g(t)^2}{2} \nabla \log p_t(x \mid z)$. By rewriting the inside of the expectation of the scaled score loss as

$$\lambda(t)^2 \left\| \hat{s}_t(x;\theta) - \frac{g(t)^2}{2} \nabla \log p_t(x|z) \right\|^2 = \left\| \lambda(t) \hat{s}_t(x;\theta) + \lambda(t) \frac{g(t)^2 \epsilon}{2\sigma_t} \right\|^2,$$

we can see that when setting $\lambda(t) = \frac{2\sigma_t}{g(t)^2}$, the score loss becomes

$$\left\| \lambda(t) \hat{s}_t(x;\theta) + \epsilon \right\|^2 \qquad \epsilon \sim \mathcal{N}(0, I).$$

This approach allows us to reconstruct the mini-flow SDE drift as the sum of the mini-flow ODE drift and the scaled score network output:

$$u_t(x;\theta) = v_t(x;\theta) + \hat{s}_t(x;\theta). \tag{17}$$

We summarize our method in Algorithm 1. Once we have the trained networks $v_t(x;\theta), s_t(x;\theta)$, we can construct the SDE drift $u_t(x;\theta)$ using (15), and generate trajectories from given initial conditions x_0 by an SDE integration with drift $u_t(x;\theta)$ and diffusion σ.

Algorithm 1 MMSFM Training

procedure TRAIN MMSFM
  Initialize networks $v_t(x;\theta), s_t(x;\theta)$
  Set diffusion term $g(t) \leftarrow \sigma$
  Set time-varying variance $\sigma_t \leftarrow \sigma\sqrt{t(1-t)}$ or $\sigma\sqrt{r(t)(1-r(t))}$
  Set weights $\lambda(t) \leftarrow \dfrac{2\sigma_t}{g(t)^2}$ ▷ From scaling strategy
  Set window size k
  while Training **do**
    for $i = 0$ to $M - k$ **do** ▷ Rolling window
      Sample mini-batches $X_{t_i}, \ldots, X_{t_{i+k}} \sim \rho_i, \ldots, \rho_{i+k}$
      Compute OT plans $\pi(X_{t_i}, X_{t_{i+1}}), \pi(X_{t_{i+2}} \mid X_{t_{i+1}}), \ldots$
      Generate aligned samples $z \leftarrow (X^\star_{t_i}, \ldots, X^\star_{t_{i+k}})$ using π
      Compute transport splines $\mu_t \leftarrow \mu_{i:i+k}(t)$ ▷ Batch computation
      Set $p_t(x \mid z) \leftarrow \mathcal{N}(x \mid \mu_t, \sigma_t^2)$
      Sample times t using stratified sampling over $[t_i, t_{i+k}]$
      Sample $x \sim p_t(x \mid z)$
      Compute $u_t^\circ(x \mid z) \leftarrow \dfrac{\sigma_t'}{\sigma_t}(x - \mu_t) + \mu_t'$
      Compute $\nabla \log p_t(x \mid z) \leftarrow \dfrac{\mu_t - x}{\sigma_t^2}$
      Compute loss:
      $$\mathcal{L}(\theta) \leftarrow \left\| v_t(x;\theta) - u_t^\circ(x \mid z) \right\|^2 + \lambda(t)^2 \left\| s_t(x;\theta) - \nabla \log p_t(x \mid z) \right\|^2$$
      Update θ using $\mathcal{L}(\theta)$
    end for
  end while
  return Trained networks $v_t(x;\theta), s_t(x;\theta)$
end procedure

3 RESULTS

We briefly describe our data and setup below, and also include a more detailed experimental setup description in Appendix B. We summarize our results in Tables 1 and 2.

3.1 EXPERIMENTAL SETUP

We applied our rolling window framework to three synthetic datasets, one single-cell dataset from COLO858 melanoma cells, and two RNA gene expression datasets. From our framework, we use the $k = 1$ (pairwise) and $k = 2$ (triplet) mini-flow settings. We approximate the MMOT plan with transport splines computed on mini-batch OT given the smaller computation cost and asymptotic convergence properties (Fatras et al., 2019; 2021). We additionally use MIOFlow (Huguet et al., 2022) for a third set of flows on the synthetic datasets and COLO858 in order to examine the difference in performance for ambient-space and latent-space models. The initial conditions for the generated trajectories are from a held-out set of samples from the source distribution ρ_0. Evaluations are computed by leaving out a timepoint marginal during training and calculating the Wasserstein metrics W_1 and W_2^2 (using Euclidean distance as the cost function), the maximum mean discrepancy with a mixture kernel (MMD(M)), and the maximum mean discrepancy using a Gaussian kernel (MMD(G)) at the left-out timepoint.

3.2 SYNTHETIC DATA

Our three synthetic datasets are the S-shaped Gaussians, the α-shaped Gaussians, and the DynGen synthetic scRNA dataset (Cannoodt et al., 2021). The S and α-shaped Gaussians both consist of 7 marginal distributions in $\mathbb{R}^2$. We select these two datasets because S-shaped Gaussians involve learning a flow with changing curvature, and the α shaped Gaussians have a cross-over point for some x where the flow $u_{t_i}(x) \neq u_{t_j}(x)$ and $i \neq j$. We evaluate both datasets on two different timepoint labels. We test our method on equidistant timepoints $\mathcal{T}_1 = (0, 0.17, 0.33, 0.5, 0.67, 0.83, 1)$, and arbitrary timepoints $\mathcal{T}_2 = (0, 0.08, 0.38, 0.42, 0.54, 0.85, 1)$. The DynGen dataset has 5 marginal distributions on equidistant timepoints $\mathcal{T} = (0, 0.25, 0.5, 0.75, 1)$, and introduces a bifurcating flow.

3.3 REAL DATA

We also apply our method to three real datasets. The COLO858 dataset contains single-cell snapshot data from the AP-1 transcription factor network in COLO858 melanoma cells. The AP-1 network integrates signals from the upstream MAPK pathway, linking signal transduction to transcription and driving cellular plasticity, epigenetic reprogramming, and resistance to MAPK inhibitors in melanoma (Kong et al., 2017; Johannessen et al., 2013; Shah et al., 2010; Maurus et al., 2017; Fallahi-Sichani et al., 2015; Ramsdale et al., 2015; Comandante-Lou et al., 2022). The dataset consists of high-dimensional measurements of 15 AP-1 transcription factors from the FOS, JUN, and ATF families, collected at eight non-equidistant timepoints $\mathcal{T} = (0, 0.5, 2, 6, 15, 24, 72, 120)$ measured in hours following BRAF/MEK inhibitor treatment (Comandante-Lou et al., 2022), which we normalize to $\mathcal{T} = (0, 0.004, 0.017, 0.05, 0.125, 0.2, 0.6, 1)$.

To visualize the high-dimensional dynamics, we employ a GAE with a Gaussian kernel (Huguet et al., 2022), embedding the 15-dimensional data into a two-dimensional latent space. Unlike methods such as t-SNE or UMAP, the GAE provides a consistent representation, allowing new data to be embedded into a shared coordinate system. We plot snapshots of the inferred trajectories at various timepoints in Figure 1.

Finally, we consider gene expression data from the Multiome and CITEseq datasets published as part of a NeurIPS competition (Burkhardt et al., 2022). Measurements are taken at $\mathcal{T} = (2, 3, 4, 7)$ days, which again normalize to $\mathcal{T} = (0, 0.2, 0.4, 1)$. We follow the procedure in (Tong et al., 2023b) and preprocess the data into the first 50 and 100 principal components, along with the top 1000 highly variable genes (Satija et al., 2015; Stuart et al., 2019; Zheng et al., 2017).

Table 1: Comparison of the inferred distributions generated by MIOFlow and our method using pairwise and triplet mini-flows at the held-out timepoint. For the equidistant timepoints $\mathcal{T}_1$, we hold out $t_5 = 0.83$ and $t_4 = 0.67$ respectively for the S-shaped and α-shaped data. We do the same for the arbitrary timepoints $\mathcal{T}_2$, holding out $t_5 = 0.85$ and $t_4 = 0.54$ for the S-shaped and α-shaped data. From Dyngen, we hold out $t_1 = 0.25$ and from COLO858 $t_3 = 0.05$. The best results are in bold; lower is better.

		S-shaped (hold out t_5)			α-shaped (hold out t_4)		
		MIOFlow	Pairwise	Triplets	MIOFlow	Pairwise	Triplets
$\mathcal{T}_1$	W_1	8.16	2.36	**1.83**	21.54	**3.78**	4.54
	W_2^2	66.91	5.87	**3.86**	464.36	**14.56**	21.06
	MMD(G)	7.26	2.29	**1.47**	7.65	**3.96**	4.26
	MMD(M)	66.19	5.24	**3.11**	463.66	**13.92**	20.01
$\mathcal{T}_2$	W_1	9.42	2.12	**1.62**	5.04	8.08	**3.79**
	W_2^2	89.07	4.56	**2.73**	25.85	76.82	**14.73**
	MMD(G)	7.37	2.36	**1.53**	6.46	4.01	**3.77**
	MMD(M)	88.37	4.12	**2.22**	25.35	64.81	**14.07**
		Dyngen (hold out t_1)			COLO858 (hold out t_3)		
	W_1	0.85	**0.74**	0.83	0.48	0.93	**0.42**
	W_2^2	0.98	**0.63**	0.82	0.25	0.92	**0.19**
	MMD(G)	0.53	0.38	**0.22**	1.30	1.08	**0.26**
	MMD(M)	0.51	0.19	**0.10**	0.08	0.74	**0.05**

Table 2: Comparison of the pairwise and triplet methods on the CITEseq and Multiome gene expression datasets. We hold out $t_2 = 0.4$ for both datasets.

		PCA 50		PCA 100		Hi-Var 1000	
		Pairwise	Triplets	Pairwise	Triplets	Pairwise	Triplets
CITEseq	W_1	54.18	**53.98**	62.85	**62.08**	**50.64**	50.71
	W_2^2	3027.28	**3019.89**	4036.41	**3942.08**	**2579.84**	2585.98
	MMD(G)	0.16	0.16	0.16	**0.15**	0.05	0.05
	MMD(M)	339.20	**344.89**	345.09	**331.72**	**48.53**	49.83
Multiome	W_1	61.79	**60.92**	70.72	**70.39**	56.15	**56.10**
	W_2^2	3918.50	**3806.89**	5077.07	**5029.56**	3166.01	**3160.84**
	MMD(G)	0.30	**0.27**	0.25	**0.23**	0.04	0.04
	MMD(M)	793.34	**705.21**	656.86	**621.32**	40.71	**40.29**

3.4 Discussion

Learned flows are visualized in Appendix C. Our method, whether pairwise or triplet, consistently outperformed MIOFlow on the interpolation at the held-out timepoint for the synthetic data. Interestingly, the pairwise model slightly outperformed the triplet model for the α-shaped Gaussians on $\mathcal{T}_1$. We believe that in this specific instance, the held-out marginal and timepoints did not sufficiently challenge the pairwise model. Specifically, the masked timepoint corresponded to an interval where the momentum from the prior interval was enough infer the held-out marginal. In contrast, the α-shaped Gaussians on $\mathcal{T}_2$ show that the triplet model outperformed the pairwise model by a significant margin; in fact, even MIOFlow generally outperformed the pairwise model in this instance. This suggests that the triplet method is more effective for non-equidistant time snapshots especially when capturing complex temporal dynamics because the variation of flows provided by splines in overlapping windows helps learn the held-out marginal. The success of our methods on $\mathcal{T}_2$ demonstrates the robustness and stability of our approach even when handling arbitrary timepoints. Looking at the trajectory plots, we can also confirm that our method is able to handle datasets with varying flow curvatures and flow cross-overs.

The bifurcating flow of Dyngen posed a challenge for our models. Our held-out timepoint

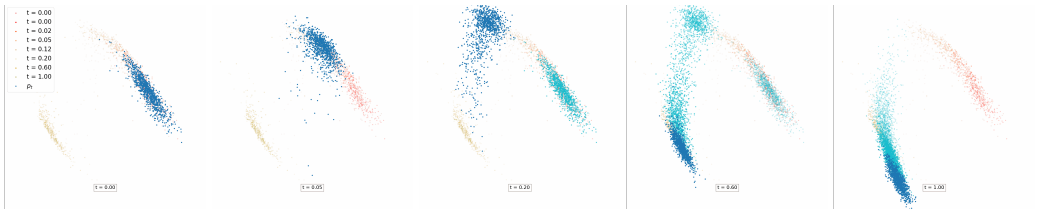

Figure 1: Visualization of transcription factor dynamics in COLO858 melanoma cells following BRAF/MEK inhibitor treatment, using our triplet stochastic flow matching model. The plots show snapshots of the inferred trajectories embedded in a two-dimensional GAE space (Huguet et al., 2022). The original high-dimensional data (15 transcriptiuon factors) is mapped to $\mathbb{R}^2$ for visualization. Trajectories start on the right and follow a mirrored 'N' shape, ending on the left. Blue dots represent model predictions at the current timepoint; cyan dots represent historical trajectories; red dots represent measured data at different timepoints.

of t_1 is when the flows begin to split. Our models outperformed MIOFlow, with the pairwise model scoring best on W_1 and W_2^2, and the triplet model scoring best on MMD(G) and MMD(M). Despite this, our method struggles to handle bifurcating trajectories. Each branch of the split should be isolated, but we occasionally see particles transitioning between the two branches. We suspect that we can trace this behavior to our sampling approach because we use mini-batch OT to create the transport splines. Mini-batch OT, however, does not enforce a consistency constraint on the sampling process, resulting in cases where particles are able to jump between separate branches of the bifurcated flow. We did not explore methods to mitigate this problem and believe this to be an avenue for future work.

The COLO858 trajectories again show our triplet method performs the best, scoring better than the MIOFlow trajectories and significantly outperforming the pairwise method. The overlapping mini-flows of the triplet model greatly stabilize the overall flow in the individual intervals. In addition, remaining in the high-dimensional ambient space without aggressive dimensionality reduction preserves important biological information, leading to more biologically plausible trajectories. The ability to generate samples at arbitrary timepoints allows us to explore the system's behavior beyond the observed data, potentially identifying critical time windows where intervention might be most effective. This has implications for understanding drug resistance mechanisms and designing more effective therapeutic strategies.

In all instances, MIOFlow generated idiosyncratic trajectories which matched the marginals at the specified timepoints but performed poorly between those timepoints. We believe this to be the case because the MIOFlow model operates in the embedding space generated by a GAE. This structure works very well for trajectories in the embedding space but poses a problem when reconstructing the trajectories in the ambient space. Specifically, the GAE is only trained on the data marginals $\{\rho_i\}_{i=0}^M$ at times $\{t_i\}_{i=0}^M$, which means that data points not specified in the data are effectively out-of-distribution w.r.t. the GAE. These out-of-distribution points arise naturally from generating trajectories which spend time traveling between the data distributions. In addition, we notice that all the reconstructed points exhibit high bias and low variance, tending to be bunched very close to each other. This quality perhaps captures the first moment well but not any higher moments.

Finally, we validate the ability for our triplet model to learn flows on high dimensional, noisy data, taken at irregular timepoints. We do so by comparing results from just the pairwise and triplet models on the CITEseq and Multiome datasets. These two datasets contain high dimensional samples with noise inherent to biological measurements, with measurements from non-uniform time intervals. We can see the triplet model successfully outperforms the pairwise model on inferring the distribution at the held out timepoint even in these conditions.

3.5 Experimental Setup

We applied our stochastic flow matching framework to three synthetic datasets and one single-cell dataset from COLO858 melanoma cells. The initial conditions for the generated trajectories are from a held-out set of samples from the source distribution ρ_0. Evaluations are computed by leaving out a timepoint marginal during training and calculating the Wasserstein metrics W_1 and W_2^2 (using Euclidean distance as the cost function), the maximum mean discrepancy with a mixture kernel (MMD(M)), and the maximum mean discrepancy using a Gaussian kernel (MMD(G)) at the left-out timepoint. We summarize our results in Table 1.

From our framework, we use the $k = 1$ (pairwise) and $k = 2$ (triplet) mini-flow settings; we additionally use MIOFlow (Huguet et al., 2022) for a third set of flows. We use monotonic cubic Hermite splines as our spline algorithm for the triplet mini-flows. In order to keep comparisons fair, we set all neural networks to be MLPs with two hidden layers, each with 64 neurons, and the SELU activation function. The input to the networks is the time t concatenated to the state x. All networks are trained for 2500 gradient steps using a learning rate of 1×10^{-4}. We set $\sigma = 0.15$ for our method and likewise as the noise scale in MIOFlow. All models are optimized using the AdamW optimizer. Additionally, MIOFlow requires training a Geodesic Autoencoder (GAE) to embed high-dimensional data into a lower-dimensional space and to then reconstruct trajectories learned in the embedded space. We define the encoder as an MLP with three hidden layers of sizes 128, 64, and 32. This encoder outputs an embedding into $\mathbb{R}^2$. The decoder has the same architecture but in reverse. We use ReLU as the activation function. The GAE is trained for 1000 gradient steps using the AdamW optimizer.

3.6 Synthetic Data

S-Shaped Gaussians. This synthetic dataset contains 7 Gaussians in $\mathbb{R}^2$ outlining an S-shape, each associated with a time t_i in $\{t_i\}_{i=0}^6$. We select this dataset to examine a setting where the curvature of the flow varies. We consider two timepoint assignments: one with equidistant timepoints $\mathcal{T}_1 = (0, 0.17, 0.33, 0.5, 0.67, 0.83, 1)$, and another with arbitrary timepoints $\mathcal{T}_2 = (0, 0.08, 0.38, 0.42, 0.54, 0.85, 1)$. For each $\mathcal{T}$, we hold out t_5, which corresponds to 0.83 and 0.85, respectively. We plot trajectories for $\mathcal{T}_2$ in Figure 3.

α-Shaped Gaussians. This setting also contains 7 Gaussians in $\mathbb{R}^2$ and associated timepoints $\{t_i\}_{i=0}^6$. Now, the flow through the Gaussians traces the shape of the α character. Crucially, this dataset involves a flow cross-over, such that there are data points x where the flow $u_{t_i}(x) \neq u_{t_j}(x)$ for $i \neq j$. We consider the same two assignments of timepoints $\mathcal{T}_1$ and $\mathcal{T}_2$ as in the S-shaped Gaussians, but hold out t_4 instead. This corresponds to holding out $t_4 = 0.67$ from $\mathcal{T}_1$, and $t_4 = 0.54$ from $\mathcal{T}_2$. We plot trajectories for $\mathcal{T}_2$ in Figure 4.

Dyngen We use the simulated scRNA-seq dataset with Dyngen (Cannoodt et al., 2021) that used by Huguet et al. (2022) which results in a $\mathbb{R}^5$ ambient space and equidistant timepoints $\mathcal{T} = (0, 0.25, 0.5, 0.75, 1)$. We hold out $t_1 = 0.25$. This dataset increases the difficulty of the flow matching task by introducing a bifurcation of the flow. See Figure 5 for trajectories generated from our experiments.

3.7 Application to COLO858 Melanoma Cells

We apply our stochastic flow matching framework to align single-cell snapshot data from the AP-1 transcription factor network in COLO858 melanoma cells. The AP-1 network integrates signals from the upstream MAPK pathway, linking signal transduction to transcription and driving cellular plasticity, epigenetic reprogramming, and resistance to MAPK inhibitors in melanoma (Kong et al., 2017; Johannessen et al., 2013; Shah et al., 2010; Maurus et al., 2017; Fallahi-Sichani et al., 2015; Ramsdale et al., 2015; Comandante-Lou et al., 2022). The dataset consists of high-dimensional measurements of 15 AP-1 transcription factors from the FOS, JUN, and ATF families, collected at eight non-equidistant timepoints $\mathcal{T} = (0, 0.5, 2, 6, 15, 24, 72, 120)$ measured in hours following BRAF/MEK inhibitor treatment (Comandante-Lou et al., 2022). The data was acquired using the 4i

(Iterative Indirect Immunofluorescence Imaging) technique, allowing multiplexed imaging of protein markers in single cells (Gut et al., 2018). In our experiments, we normalize the timepoints to range from 0 to 1, resulting in $\mathcal{T} = (0, 0.004, 0.017, 0.05, 0.125, 0.2, 0.6, 1)$, and we hold out $t_3 = 0.05$. To visualize the high-dimensional dynamics, we employ a GAE with a Gaussian kernel (Huguet et al., 2022), embedding the 15-dimensional data into a two-dimensional latent space. Unlike methods such as t-SNE or UMAP, the GAE provides a consistent and invertible representation, allowing new data to be embedded in the same coordinate system without distortion. We plot snapshots of the inferred trajectories at various timepoints in Figure 1.

3.8 Discussion

Our method, whether pairwise or triplet, consistently outperformed MIOFlow on the interpolation at the held-out timepoint for the synthetic data. Interestingly, the pairwise model slightly outperformed the triplet model for the α-shaped Gaussians on $\mathcal{T}_1$. However, we believe that this does not indicate a general pattern favoring the pairwise model, as the held-out marginal in this case was not chosen to challenge its limitations. Specifically, the masked timepoints corresponded to intervals where linear interpolation sufficed, and the complexity of the flow was minimal. In contrast, the α-shaped Gaussians on $\mathcal{T}_2$ show that the triplet model outperformed the pairwise model by a significant margin; in fact, even MIOFlow generally outperformed the pairwise model in this instance. This suggests that the triplet method is more effective for non-equidistant time snapshots, especially when capturing complex temporal dynamics. The success of our methods on $\mathcal{T}_2$ demonstrates the robustness and stability of our approach even when handling arbitrary timepoints. Looking at the trajectory plots, we can also confirm that our method is able to handle datasets with varying flow curvatures and flow cross-overs.

The bifurcating flow of Dyngen posed a challenge for our models. Our held-out timepoint of t_1 is when the flows begin to split. Our models outperformed MIOFlow, with the pairwise model scoring best on W_1 and W_2^2, and the triplet model scoring best on MMD(G) and MMD(M). Despite this, our method struggles to handle bifurcating trajectories. Each branch of the split should be isolated, but we occasionally see particles transitioning between the two branches. We suspect that we can trace this behavior to our mini-batch sampling approach because we use batch OT when creating the multi-marginal splines. Batch OT does not enforce a hard constraint on the sampling process, resulting in cases where particles are able to jump between separate flows. We did not explore methods to mitigate this problem and believe this to be an avenue for future work.

The COLO858 trajectories again show our triplet method performs the best, scoring better than the MIOFlow trajectories and significantly outperforming the pairwise method. The overlapping mini-flows of the triplet model greatly stabilize the overall flow in the individual intervals. In addition, remaining in the high-dimensional ambient space without aggressive dimensionality reduction preserves important biological information, leading to more biologically plausible trajectories. The ability to generate samples at arbitrary timepoints allows us to explore the system's behavior beyond the observed data, potentially identifying critical time windows where intervention might be most effective. This has implications for understanding drug resistance mechanisms and designing more effective therapeutic strategies.

In all instances, MIOFlow generated idiosyncratic trajectories which matched the marginals at the specified timepoints but performed poorly between those timepoints. We believe this to be the case because the MIOFlow model operates in the embedding space generated by a GAE. This structure works very well for trajectories in the embedding space but poses a problem when reconstructing the trajectories in the ambient space. Specifically, the GAE is only trained on the data marginals $\{\rho_i\}_{i=0}^M$ at times $\{t_i\}_{i=0}^M$, which means that data points not specified in the data are effectively out-of-distribution w.r.t. the GAE. These out-of-distribution points arise naturally from generating trajectories which spend time traveling between the data distributions. In addition, we notice that all the reconstructed points exhibit high bias and low variance, tending to be bunched very close to each other. This quality perhaps captures the first moment well but not any higher moments.

4 CONCLUSION

We have presented a novel framework for aligning high-dimensional single-cell data in a multi-marginal setting with non-equidistant timepoints, while remaining in the high-dimensional space and avoiding the pitfalls of dimensionality reduction. By expanding the literature of Conditional Flow Matching, we have developed a method that learns flows for overlapping triplets, enhancing robustness and stability in multi-marginal settings. Our application to the COLO858 melanoma single-cell dataset demonstrates the method's effectiveness in capturing complex cellular dynamics while avoiding the need to simulate differential equations during training. We further validate our method's scalability and ability to learn in high dimensional spaces using the CITEseq and Multiome datasets. The incorporation of stochasticity through score matching improves robustness and avoids overfitting, enabling the model to generalize to new conditions. This work opens new avenues for generative algorithms as well as modeling cellular responses to perturbations, providing a computationally efficient and biologically accurate framework capable of handling the complexities of high-dimensional, stochastic biological systems.

REFERENCES

Michael S Albergo, Nicholas M Boffi, Michael Lindsey, and Eric Vanden-Eijnden. Multimarginal generative modeling with stochastic interpolants. *arXiv preprint arXiv:2310.03695*, 2023.

Steven J Altschuler and Lani F Wu. Cellular heterogeneity: do differences make a difference? *Cell*, 141(4):559–563, 2010.

Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.

Jean-David Benamou, Thomas O Gallouët, and François-Xavier Vialard. Second-order models for optimal transport and cubic splines on the wasserstein space. *Foundations of Computational Mathematics*, 19:1113–1143, 2019.

Daniel B Burkhardt, Beatriz P San Juan, John G Lock, Smita Krishnaswamy, and Christine L Chaffer. Mapping phenotypic plasticity upon the cancer cell state landscape using manifold learning. *Cancer Discovery*, 12(8):1847–1859, 2022.

Robrecht Cannoodt, Wouter Saelens, Louise Deconinck, and Yvan Saeys. Spearheading future omics analyses using dyngen, a multi-modal simulator of single cells. *Nature Communications*, 12(1):3942, 2021.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018a.

Tianrong Chen, Guan-Horng Liu, Molei Tao, and Evangelos Theodorou. Deep momentum multi-marginal schrödinger bridge. *Advances in Neural Information Processing Systems*, 36, 2024.

Yongxin Chen, Giovanni Conforti, and Tryphon T Georgiou. Measure-valued spline curves: An optimal transport viewpoint. *SIAM Journal on Mathematical Analysis*, 50(6):5947–5968, 2018b.

Sinho Chewi, Julien Clancy, Thibaut Le Gouic, Philippe Rigollet, George Stepaniants, and Austin Stromme. Fast and smooth interpolation on wasserstein space. In *International Conference on Artificial Intelligence and Statistics*, pp. 3061–3069. PMLR, 2021.

Natacha Comandante-Lou, Douglas G Baumann, and Mohammad Fallahi-Sichani. Ap-1 transcription factor network explains diverse patterns of cellular plasticity in melanoma cells. *Cell reports*, 40(5):111147, 2022.

Mohammad Fallahi-Sichani, Nathan J Moerke, Mario Niepel, Tinghu Zhang, Nathanael S Gray, and Peter K Sorger. Systematic analysis of brafv 600e melanomas reveals a role for jnk/c-jun pathway in adaptive resistance to drug-induced apoptosis. *Molecular systems biology*, 11(3):797, 2015.

Kilian Fatras, Younes Zine, Rémi Flamary, Rémi Gribonval, and Nicolas Courty. Learning with minibatch wasserstein: asymptotic and gradient properties. *arXiv preprint arXiv:1910.04091*, 2019.

Kilian Fatras, Younes Zine, Szymon Majewski, Rémi Flamary, Rémi Gribonval, and Nicolas Courty. Minibatch optimal transport distances; analysis and applications. *arXiv preprint arXiv:2101.01792*, 2021.

Frederick N Fritsch and Ralph E Carlson. Monotone piecewise cubic interpolation. *SIAM Journal on Numerical Analysis*, 17(2):238–246, 1980.

Crispin W Gardiner. *Handbook of stochastic methods*, volume 3. springer Berlin, 1985.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Gabriele Gut, Markus D Herrmann, and Lucas Pelkmans. Multiplexed protein maps link subcellular organization to cellular states. *Science*, 361(6401):eaar7042, 2018.

M Ch Hermite and M Borchardt. Sur la formule d'interpolation de lagrange. *Journal für die reine und angewandte Mathematik (Crelles Journal)*, 1878(84):70–79, 1878.

Andreas Hilfinger and Johan Paulsson. Separating intrinsic from extrinsic fluctuations in dynamic biological systems. *Proceedings of the National Academy of Sciences*, 108(29):12167–12172, 2011.

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

Jordan M Horowitz and Todd R Gingrich. Thermodynamic uncertainty relations constrain non-equilibrium fluctuations. *Nature Physics*, 16(1):15–20, 2020.

Guillaume Huguet, Daniel Sumner Magruder, Alexander Tong, Oluwadamilola Fasina, Manik Kuchroo, Guy Wolf, and Smita Krishnaswamy. Manifold interpolating optimal-transport flows for trajectory inference. *Advances in neural information processing systems*, 35:29705–29718, 2022.

Cory M Johannessen, Laura A Johnson, Federica Piccioni, Aisha Townes, Dennie T Frederick, Melanie K Donahue, Rajiv Narayan, Keith T Flaherty, Jennifer A Wargo, David E Root, and Levi A. Garraway. A melanocyte lineage program confers resistance to map kinase pathway inhibition. *Nature*, 504(7478):138–142, 2013.

Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Vladimir Yu Kiselev, Tallulah S Andrews, and Martin Hemberg. Challenges in unsupervised clustering of single-cell rna-seq data. *Nature Reviews Genetics*, 20(5):273–282, 2019.

Xiangjun Kong, Thomas Kuilman, Aida Shahrabi, Julia Boshuizen, Kristel Kemper, Ji-Ying Song, Hans WM Niessen, Elisa A Rozeman, Marnix H Geukes Foppen, Christian U Blank, and Daniel S. Peeper. Cancer drug addiction is relayed by an erk2-dependent phenotype switch. *Nature*, 550(7675):270–274, 2017.

Christian Léonard. A survey of the Schrödinger problem and some of its connections with optimal transport. *Discrete Contin. Dyn. Syst. A*, 34(4):1533–1574, 2014.

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

K Maurus, A Hufnagel, F Geiger, S Graf, C Berking, A Heinemann, A Paschen, S Kneitz, C Stigloher, E Geissinger, and C Otto. The ap-1 transcription factor fosl1 causes melanocyte reprogramming and transformation. *Oncogene*, 36(36):5110–5121, 2017.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Simon Mitchell and Alexander Hoffmann. Identifying noise sources governing cell-to-cell variability. *Current opinion in systems biology*, 8:39–45, 2018.

Kevin R. Moon, David van Dijk, Zheng Wang, Scott Gigante, Daniel B. Burkhardt, William S. Chen, Kristina Yim, Antonia van den Elzen, Matthew J. Hirn, Ronald R. Coifman, Natalia B. Ivanova, Guy Wolf, and Smita Krishnaswamy. Visualizing structure and transitions in high-dimensional biological data. *Nature Biotechnology*, 37(12):1482–1492, 2019. doi: 10.1038/s41587-019-0336-3. URL https://doi.org/10.1038/s41587-019-0336-3.

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

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

Rachel Ramsdale, Robert N Jorissen, Frederic Z Li, Sheren Al-Obaidi, Teresa Ward, Karen E Sheppard, Patricia E Bukczynska, Richard J Young, Samantha E Boyle, Mark Shackleton, Gideon Bollag, Georgina Long, Eugene Tulchinsky, Helen Rizos Rizos, Richard Pearson, Grant A McArthur, Amardeep Dhillon, and Petranel T Ferrao. The transcription cofactor c-jun mediates phenotype switching and braf inhibitor resistance in melanoma. *Science signaling*, 8(390):ra82–ra82, 2015.

H Risken. The fokker-planck equation, 1996.

Yvan Saeys, Sofie Van Gassen, and Bart N Lambrecht. Computational flow cytometry: helping to make sense of high-dimensional immunology data. *Nature Reviews Immunology*, 16(7):449–462, 2016.

Rahul Satija, Jeffrey A Farrell, David Gennert, Alexander F Schier, and Aviv Regev. Spatial reconstruction of single-cell gene expression data. *Nature biotechnology*, 33(5):495–502, 2015.

Geoffrey Schiebinger, Jian Shu, Marcin Tabaka, Brian Cleary, Vidya Subramanian, Aryeh Solomon, Joshua Gould, Siyan Liu, Stacie Lin, Peter Berube, Lia Lee, Jenny Chen, Justin Brumbaugh, Philippe Rigollet, Konrad Hochedlinger, Rudolf Jaenisch, Aviv Regev, and Eric S. Lander. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, 176(4):928–943, 2019.

Matthew R Schofield, Richard J Barker, William A Link, and Heloise Pavanato. Estimating population size: The importance of model and estimator choice. *Biometrics*, 79(4):3803–3817, 2023.

Erwin Schrödinger. Über die Umkehrung der Naturgesetze. *Sitzungsberichte Preuss. Akad. Wiss. Berlin. Phys. Math.*, 144:144–153, 1931.

Meera Shah, Anindita Bhoumik, Vikas Goel, Antimone Dewing, Wolfgang Breitwieser, Harriet Kluger, Stan Krajewski, Maryla Krajewska, Jason DeHart, Eric Lau, and DM Kallenberg. A role for atf2 in regulating mitf and melanoma development. *PLoS genetics*, 6(12):e1001258, 2010.

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

Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M Mauck, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. Comprehensive integration of single-cell data. *cell*, 177(7):1888–1902, 2019.

Alexander Tong, Jessie Huang, Guy Wolf, David Van Dijk, and Smita Krishnaswamy. Trajectorynet: A dynamic optimal transport network for modeling cellular dynamics. In *International conference on machine learning*, pp. 9526–9536. PMLR, 2020.

Alexander Tong, Nikolay Malkin, Kilian Fatras, Lazar Atanackovic, Yanlei Zhang, Guillaume Huguet, Guy Wolf, and Yoshua Bengio. Simulation-free schr\" odinger bridges via score and flow matching. *arXiv preprint arXiv:2307.03672*, 2023a.

Alexander Tong, Nikolay Malkin, Kilian FATRAS, Lazar Atanackovic, Yanlei Zhang, Guillaume Huguet, Guy Wolf, and Yoshua Bengio. Simulation-free schrödinger bridges via score and flow matching. In *ICML Workshop on New Frontiers in Learning, Control, and Dynamical Systems*, 2023b.

Alexander Tong, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Kilian Fatras, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*, 2023c.

Sophie Tritschler, Maren Büttner, David S Fischer, Marius Lange, Volker Bergen, Heiko Lickert, and Fabian J Theis. Concepts and limitations for learning developmental trajectories from single cell genomics. *Development*, 146(12):dev170506, 2019.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Cédric Villani. *Optimal transport: old and new*, volume 338. Springer, 2009.

Caleb Weinreb, Samuel Wolock, Betsabeh K Tusi, Merav Socolovsky, and Allon M Klein. Fundamental limits on dynamic inference from single-cell snapshots. *Proceedings of the National Academy of Sciences*, 115(10):E2467–E2476, 2018.

Grace XY Zheng, Jessica M Terry, Phillip Belgrader, Paul Ryvkin, Zachary W Bent, Ryan Wilson, Solongo B Ziraldo, Tobias D Wheeler, Geoff P McDermott, Junjie Zhu, et al. Massively parallel digital transcriptional profiling of single cells. *Nature communications*, 8(1):14049, 2017.

A Cubic Splines

Cubic splines are a class of piecewise functions interpolating between control points $(t_0, x_0), \ldots, (t_n, x_n)$, taking the form

$$S(t) = \begin{cases} S_0(t) & t_0 \leq t < t_1 \\ \vdots & \\ S_{n-1}(t) & t_{n-1} \leq t \leq t_n \end{cases}$$

where S_i is the cubic polynomial $S_i(t) = a_i(t-t_i)^3 + b_i(t-t_i)^2 + c_i(t-t_i) + d_i$ for i in $0, \ldots, n-1$. There are 4 coefficients to solve for per equation which results in n equations and $4n$ unknowns.

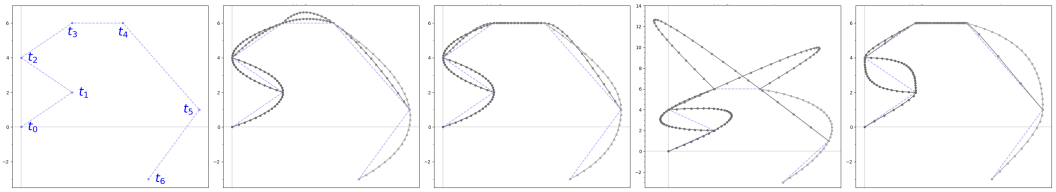

Figure 2: Comparison of euclidean splines on overlapping windows of size $k = 2$, demonstrating the potential for overlapping windows to capture variations of paths though the same intervals. Gray horizontal and vertical lines indicate the xy-axes. The arbitrary time intervals are set to $\mathcal{T} = (0, 0.05, 0.2, 0.27, 0.86, 0.95, 1)$. From left to right: 1) The 7 points to interpolate with time labels $t_0, \ldots, t_6$. 2) Natural cubic splines on equidistant time intervals. 3) Monotonic cubic Hermite splines on equidistant time intervals. 4) Natural cubic splines on arbitrary time intervals. Note the overshooting required to satisfy the continuity of S'' at t_3 and t_4. 5) Monotonic cubic Hermite splines on arbitrary time intervals.

A.1 Natural Cubic Splines

Natural cubic splines solve for the above coefficients a_i, b_i, c_i, d_i by applying four conditions. The first requires the spline to interpolate the data points (t_i, x_i) such that $S(t_i) = x_i$ resulting in $n+1$ constraints. The second requires S to be continuous at the interior points such that $S_i(t_i) = S_{i+1}(t_i)$, resulting in $n-1$ constraints. The third and fourth conditions respectively require S' and S'' to be continuous for a total of $2n-2$ constraints. Finally two boundary conditions are added such that $S''(t_0) = S''(t_n) = 0$. In total, we have constructed a system of equations with $4n$ unknowns and $4n$ constraints. Ultimately, this setup constructs a tridiagonal system of equations which is

efficiently solvable in $\mathcal{O}(n)$ time using a single forward and backward pass. Perhaps reasonably, natural cubic splines are quite local as the influence of neighboring intervals greatly decreases the further away the neighbor is.

A.2 Monotonic Cubic Hermite Splines

Cubic Hermite splines approach the problem differently. Consider a single time interval $[0, 1]$ and corresponding points x_0, x_1. Let the position of x at time t be given by the following cubic polynomial:

$$x_t = at^3 + bt^2 + ct + d.$$

Likewise, let m_t be the velocity of x_t at time t, given by

$$m_t = 3at^2 + 2bt + c.$$

At $t = 0$ and $t = 1$, we can solve for x_0, x_1, m_0, m_1 in terms of a, b, c, d to get the following system of equations:

$$\begin{aligned} x_0 &= d \\ x_1 &= a + b + c + d \\ m_0 &= c \\ m_1 &= 3a + 2b + c. \end{aligned}$$

Solving this system of equations, we get

$$x(t) = (2t^3 - 3t^2 + 1)x_0 + (t^3 - 2t^2 + t)m_0 + (-2t^3 + 3t^2)x_1 + (t^3 - t^2)m_1$$

as the polynomial interpolating $(0, x_0)$ to $(1, x_1)$. All that remains is to specify values for m_0 and m_1. In other words, cubic Hermite spline algorithms are defined by how the velocities m_i are selected.

Monotonic cubic Hermite splines set m_i using the following strategy. Define $h_k = t_{k+1} - t_k$ and $d_k = \frac{x_{k+1} - x_k}{h_k}$. If the signs of d_k and d_{k-1} do not match or either is 0, then set $m_k = 0$. Otherwise, m_k is given by

$$\frac{w_1 + w_2}{m_k} = \frac{w_1}{d_{k-1}} + \frac{w_2}{d_k}$$

where $w_1 = 2h_k + h_{k-1}$ and $w_2 = h_k + 2h_{k-1}$. We direct the reader to Fritsch & Carlson (1980) for an exact derivation and proof of monotonicity. This formula is also solvable in $\mathcal{O}(n)$ time, but differs from the natural cubic spline in that it is very local. In fact, only the immediate neighboring data points (t_{i-1}, x_{i-1}) and (t_{i+1}, x_{i+1}) influence the curve.

B Experimental Setup

B.1 Training Setup

For all experiments, we used a MLP with an input layer, two hidden layers, an output layer, along with SELU activation functions. All networks were optimized using AdamW. We set $\sigma = 0.15$ for our method and likewise as the noise scale in MIOFlow.

For the S-shaped, α-shaped, DynGen, and COLO858 datasets, we trained for 2500 gradient steps and a learning rate of 1e-4. For the CITEseq and Multiome datasets, we trained for 1000 gradient steps and a learning rate of 1e-5.

MIOFlow is a method to infer "optimal" trajectories on manifolds which correspond to geodesics. As we do not have access to the underlying manifold itself, the authors propose learning it from data using a GAE such that the encoder ϕ is a mapping from the ambient space to the manifold. More specifically, the encoder learns an embedding such that the euclidean distance of two embedded points $\|\phi(x) - \phi(y)\|$ matches some geodesic distance $G(x, y)$ based on a diffusion affinity matrix.

Additionally, MIOFlow requires training a GAE to embed high-dimensional data into a lower-dimensional space and to then reconstruct trajectories learned in the embedded space. We define the encoder as a MLP with three hidden layers of sizes 128, 64, and 32. This encoder outputs an embedding into $\mathbb{R}^2$. The decoder has the same architecture but in reverse. We use ReLU as the activation function. The GAE is trained for 1000 gradient steps using the AdamW optimizer.

B.1.1 SCORE MATCHING IMPLEMENTATION

As noted in Section 2.3.3, we have $\nabla \log p_t(x \mid z) = -\frac{\epsilon}{\sigma_t}$ for $\epsilon \sim \mathcal{N}(0, I)$. However, this direct formulation does not protect against numerical instability when σ_t is small. We follow the approach used by Tong et al. (2023b) and take advantage of the user-defined weighting schedule $\lambda(t)$ to cancel out the division and learn the scaled target $\frac{g(t)^2}{2}\nabla \log p_t(x \mid z)$ based on the Fokker-Planck Equation 2. By rewriting the inside of the expectation of the scaled score loss as

$$\lambda(t)^2 \left\| \hat{s}_t(x;\theta) - \frac{g(t)^2}{2}\nabla \log p_t(x|z) \right\|^2 = \left\| \lambda(t)\hat{s}_t(x;\theta) + \lambda(t)\frac{g(t)^2\epsilon}{2\sigma_t} \right\|^2,$$

we can see that when setting $\lambda(t) = \frac{2\sigma_t}{g(t)^2}$, the score loss becomes

$$\|\lambda(t)\hat{s}_t(x;\theta) + \epsilon\|^2 \qquad \epsilon \sim \mathcal{N}(0, I).$$

This approach allows us to reconstruct the mini-flow SDE drift as the sum of the mini-flow ODE drift and the scaled score network output:

$$u_t(x;\theta) = v_t(x;\theta) + \hat{s}_t(x;\theta). \tag{18}$$

B.2 DYNGEN

We repurpose the DynGen data used in MIOFlow (Huguet et al., 2022) for our experiments. Notably, the data itself is not the raw simulated reads; it is preprocessed into 5 dimensions using PHATE (Moon et al., 2019).

PHATE operates as a dimensionality reduction scheme aiming to preserve both local and global dependency structures. Local structure is learned first by imposing pairwise affinities under a Gaussian kernel. Global structure is inferred by propagating the local affinities via diffusion, effectively learning a statistical manifold based on the information geometry. Finally, metric MDS is used as the dimensionality reduction strategy.

We believe that the GAE used in MIOFlow learns the data manifold for the (PHATE-transformed) DynGen dataset especially well given that, by construction, the DynGen dataset does indeed reside on a manifold equipped with a diffusion-based metric. This matches the prior belief in MIOFlow that diffusion-based affinities can accurately capture the data manifold.

B.3 COLO858

The dataset consists of high-dimensional measurements of 15 AP-1 transcription factors from the FOS, JUN, and ATF families, collected at eight non-equidistant timepoints $\mathcal{T} = (0, 0.5, 2, 6, 15, 24, 72, 120)$ measured in hours following BRAF/MEK inhibitor treatment (Comandante-Lou et al., 2022). The data was acquired using the 4i (Iterative Indirect Immunofluorescence Imaging) technique, allowing multiplexed imaging of protein markers in single cells (Gut et al., 2018).

B.4 CITESEQ AND MULTIOME

These datasets were published as part of a NeurIPS competition for multimodal single-cell integration (Burkhardt et al., 2022). We present a brief overview, and refer the reader to the competition itself for more in-depth descriptions [1]. The data is collected from peripheral CD34+ hematopoietic stem and progenitor cells from healthy human donors. The CITEseq data is measured using 10x Genomics Single Cell Gene Expression with Feature Barcoding technology. The Multiome data is measured using 10x Chromium Single Cell Multiome ATAC + Gene Expression technology.

Technically, both the CITEseq and Multiome datasets are labled, with the former about predicting protein levels given gene expressions, and the latter about predicting gene expressions given ATAC-seq peak counts. We are only interested in the gene expression data, so we only use the CITEseq input data and the Multiome target data. Following Tong et al. (Tong et al., 2023b), we only select

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

cells from the respective datasets from a single donor id `13176`. The gene expression data is already library-size normalized and log1p transformed, so we compute the PCA and top highly variable genes without any further preprocessing step.

C FLOWS VISUALIZATIONS

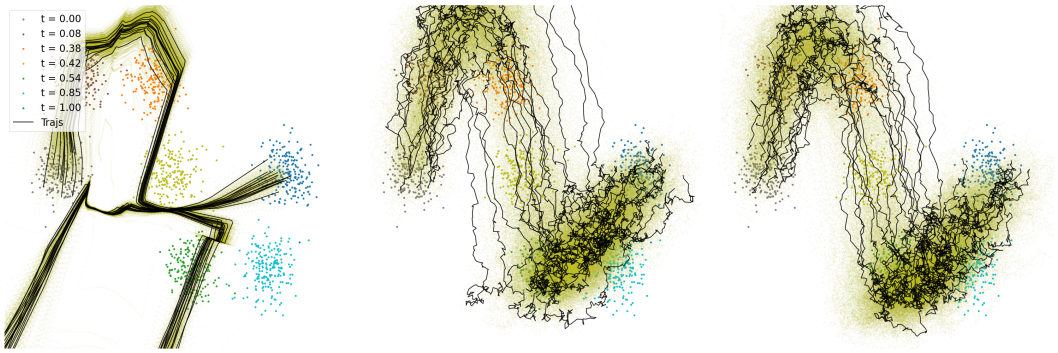

Figure 3: Trajectories for S-shaped Gaussians using arbitrary timepoints $\mathcal{T}_2 = (0, 0.08, 0.38, 0.42, 0.54, 0.85, 1)$ and holding out timepoint $t_5 = 0.85$. Trajectories start from the leftmost point and follow the curve to reach the rightmost point. (left) MIOFlow, (middle) Pairwise, (right) Triplet.

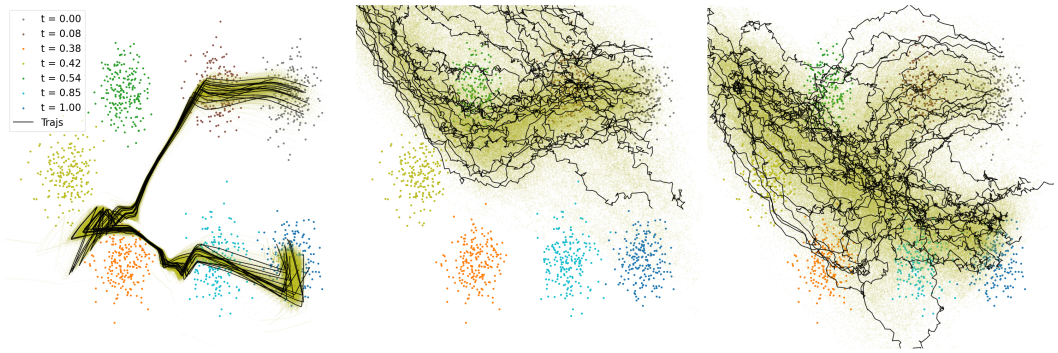

Figure 4: Trajectories for α-shaped Gaussians using arbitrary timepoints $\mathcal{T}_2 = (0, 0.08, 0.38, 0.42, 0.54, 0.85, 1)$ and holding out timepoint $t_4 = 0.54$. Trajectories start from the upper right and loop around to the bottom right. (left) MIOFlow, (middle) Pairwise, (right) Triplet.

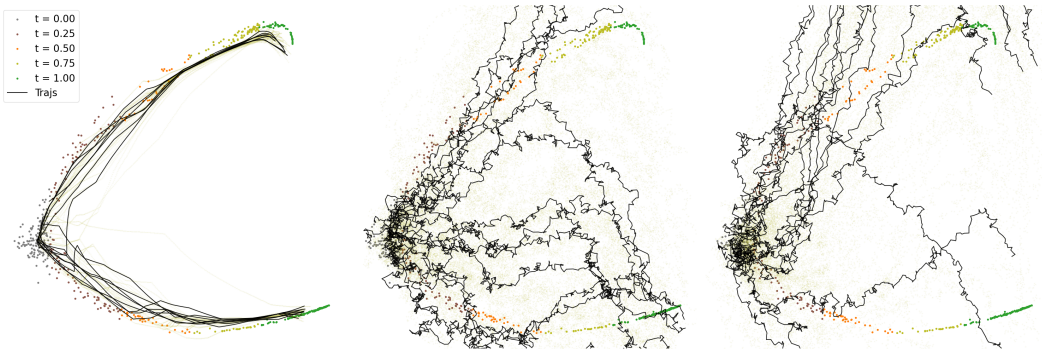

Figure 5: DynGen simulated trajectories. Trajectories start from the leftmost point and quickly bifurcate into the upper and lower right. The trajectories are in $\mathbb{R}^5$, but only the first and second dimensions are shown here. We hold out $t_1 = 0.25$. (left) MIOFlow, (middle) Pairwise, (right) Triplet.