TRAJECTORY INFERENCE VIA ACCELERATION MATCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Trajectory inference is a fundamental problem in many scientific domains: given a collection of unpaired snapshots of observations at discrete time points, the goal is to generate smooth trajectories that best resemble and interpolate the data. Existing numerical approaches to this problem have various computational burdens, such as explicitly computing spline constructions or simulating trajectories during training. In order to overcome these limitations, we propose a new algorithm called Acceleration Matching (AM). Our approach consists of lifting the original interpolation problem to phase space and then regressing onto an explicit conditional acceleration field that induces random trajectories with smooth positional paths that, under suitable assumptions, agree with the prescribed marginals. Importantly, our resulting training algorithm only requires positional data, avoids trajectory simulation during training, and uses direct Gaussian conditioning to sample training states and targets. We provide ample numerical evidence suggesting that AM is competitive with or superior to existing algorithms on several benchmark problems from the existing literature.

1 INTRODUCTION

We investigate the trajectory inference problem: given snapshots of unpaired observations collected at multiple time points $0 = t_0 < t_1 < \cdots < t_J = 1$, the goal is to generate trajectories that fit the given marginal data. In probabilistic terms, we want to find a measure-valued curve $t \in [0,1] \mapsto \mu_t \in \mathcal{P}(\mathbb{R}^d)$ such that

$$\mu_{t_j} = \rho_{t_j} \quad \forall j \in \{0, 1, \ldots, J\}\,, \tag{1}$$

for the given (empirical) probability distributions $\{\rho_{t_j}\}_{j=0}^J$. This problem appears in a variety of data-driven research domains, including astronomy, economics, and single-cell RNA sequencing, among others (Franzke et al., 2015; Kazakevičius et al., 2021; Yeo et al., 2021; Bunne et al., 2022; Kubica et al., 2007; Liounis et al., 2020). In each of these cases, data are obtained over time in the form of unpaired observations. For example, the destructive nature of single-cell RNA sequencing makes it impossible to have repeated measurements of the same cell over time as the population (of cells) continues to evolve. As a result, the nature of the data renders standard procedures (based on, for example, time-series modeling) unhelpful. These unique challenges have led to several mathematical and algorithmic developments for the problem; see Lavenant et al. (2024); Hong et al. (2025); Bunne et al. (2023); Schiebinger et al. (2019), among many others.

A naive approach is to simply construct $(\mu_t)_{t\in[0,1]}$ in a piecewise manner, learning J independent segments of the probability path of interest. This can be accomplished by separately learning transport maps between each ρ_{t_j} and $\rho_{t_{j+1}}$ and then stitching the paths together. For instance, one can learn optimal transport maps (Manole et al., 2024; Pooladian & Niles-Weed, 2021; Deb et al., 2021) or instead pass through the popular flow matching framework (Lipman et al., 2023; Albergo & Vanden-Eijnden, 2022; Liu et al., 2022).

Depending on the method, the learn-then-stitch approach unfortunately requires, in the worst case, solving J separate learning problems. Moreover, without a smooth interpolation construction, this naive approach can produce trajectories that lack global smoothness, leading to non-smooth, potentially unrealistic trajectories. The left side of Figure 1 displays trajectories given by training a

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

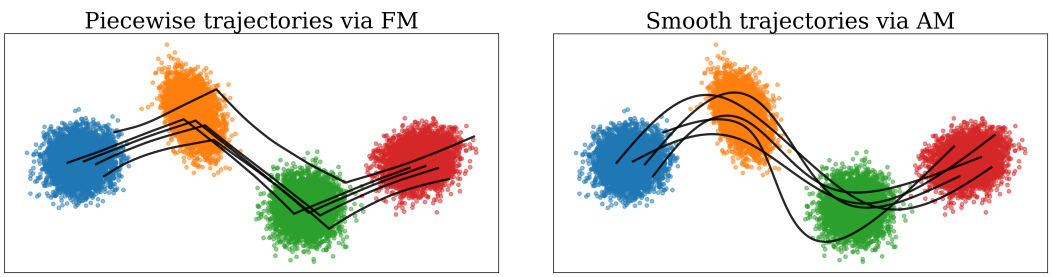


Figure 1: AM outputs smoother trajectories **(right)** than piecewise flow matching **(left)**.

simple vector field using the flow matching (FM) framework; the trajectories are visibly non-smooth due to the abrupt discontinuity in the velocity field as it switches between the marginals.

Our goal, by contrast, is to generate smooth positional trajectories by construction. To this end, we lift the original problem and consider trajectories given by stochastic differential equations (SDEs) on phase space (see Section 2.2 for more background), which are of the form

$$\mathrm{d}X_t = V_t\,\mathrm{d}t\,, \qquad \mathrm{d}V_t = a_t(X_t, V_t)\,\mathrm{d}t + \sqrt{\varepsilon}\,\mathrm{d}B_t\,, \tag{2}$$

where $a_t : \mathbb{R}^{2d} \to \mathbb{R}^d$ is an acceleration field and $(B_t)_{t\in[0,1]}$ is standard Brownian motion with diffusion parameter $\varepsilon > 0$. These dynamics generate smooth trajectories by design, as the velocities now have continuous sample paths and thus the positional paths are continuously differentiable. Our approach then consists of defining and targeting an appropriate acceleration field that yields dynamics which satisfy the marginal constraints $\mathrm{Law}(X_{t_j}) = \rho_{t_j}$ for all $j \in \{0, 1, \ldots, J\}$.

While we are not the first to consider dynamics of the form (2), related prior work uses trajectory simulation or iterative conditional-velocity propagation within their training pipelines that render their methods computationally intensive (Chen et al., 2023b; Theodoropoulos et al., 2025); see Section 4.3 for a thorough comparison.

Contributions. In this work, we present *Acceleration Matching* (AM), a novel approach for trajectory inference. Our algorithm is based on the Markovianization of kinetic Brownian bridges in phase space (see Lemma 2 and Theorem 1 in Section 3.1 for more precise details) and can thus be viewed as an extension of the flow matching framework (Lipman et al., 2023; Albergo & Vanden-Eijnden, 2022; Liu et al., 2022) for learning second-order dynamics in the presence of multiple marginal constraints. Concretely, we jointly sample velocities conditioned on all position knots, then sample intermediate bridge states and explicit acceleration targets. This approach yields a training objective that only assumes access to positional data and requires no numerical trajectory rollouts, yet enjoys a notion of smoothness along the entire trajectory. Importantly, under the stated regularity assumptions, the population acceleration field satisfies the marginals constraints defined by (1) (see Theorem 1). From the right-hand side of Figure 1, we see that trajectories generated by AM are visibly smoother than those generated by piecewise FM. We demonstrate the practical viability of AM on several standard datasets from the literature in Section 4, where our approach achieves competitive or superior held-out marginal errors, with performance varying across datasets and couplings.

2 Background

2.1 Notation

Let $\mathcal{P}(\mathbb{R}^d)$ denote the set of probability measures over $\mathbb{R}^d$. For a collection of probability distributions $\{P_i\}_{i=1}^M \subset \mathcal{P}(\mathbb{R}^d)$, let $\Pi(P_1, \ldots, P_M)$ denote the set of couplings between the measures i.e., for $\pi \in \Pi(P_1, \ldots, P_M)$, $\pi \in \mathcal{P}((\mathbb{R}^d)^M)$ such that for every $i \in \{1, \ldots, M\}$, the i^{th} marginal of π is P_i. Let $\mathcal{C}^d := C([0,1]; \mathbb{R}^d)$ be the space of continuous functions from the unit interval to $\mathbb{R}^d$ for $d \in \mathbb{N}$, endowed with the uniform topology and the corresponding Borel σ-field $\mathcal{B}(\mathcal{C}^d)$. The set of probability measures over the resulting measurable space is denoted by $\mathcal{P}(\mathcal{C}^d)$. We use bold-font vectors (e.g., $\boldsymbol{t}, \boldsymbol{x}, \boldsymbol{v}$) to denote a concatenated vector; for example, $\boldsymbol{t} = (t_0, t_1, \ldots, t_J) \in \mathbb{R}^{J+1}$ and $\boldsymbol{x} = (x_0, x_1, \ldots, x_J) \in \mathbb{R}^{d(J+1)}$ (and similarly for $\boldsymbol{v}$). For a set A, let $\mathbf{1}_A$ denote the 0-1 indicator

function over A. For a smooth function $(x, v) \mapsto F(x, v) \in \mathbb{R}$, we write $\nabla_x F(x, v)$ and $\nabla_v F(x, v)$ to explicitly denote with respect to which argument the gradient is taken.

2.2 Evolution of second-order probability flows

In this section, we briefly review probability paths on phase space, the joint space of positions $x \in \mathbb{R}^d$ and velocities $v \in \mathbb{R}^d$ of a given physical system. Throughout, we write phase space as $\mathbb{R}^{2d}$.

Our approach is based on processes defined as solutions of second-order SDEs of the form

$$\mathrm{d}X_t^a = V_t^a \,\mathrm{d}t\,, \quad \mathrm{d}V_t^a = a_t(X_t^a, V_t^a)\,\mathrm{d}t + \sqrt{\varepsilon}\,\mathrm{d}B_t\,, \quad t \in [0,1]\,, \tag{3}$$

with initial conditions (X_0^a, V_0^a), where $(B_t)_{t\in[0,1]}$ is d-dimensional standard Brownian motion, $\varepsilon > 0$, and $(t, x, v) \in [0,1] \times \mathbb{R}^{2d} \mapsto a_t(x, v) \in \mathbb{R}^d$. We then naturally interpret $(a_t)_{t\in[0,1]}$ as an inhomogeneous acceleration field that governs the underlying dynamics.

As an example, if $a_t(x, v) = -\gamma v$ for a parameter $\gamma > 0$, the resulting process is called *kinetic underdamped Brownian motion*, denoted by $\mathsf{R}^{\gamma,\varepsilon} \in \mathcal{P}(\mathcal{C}^{2d})$. If $\gamma = 0$ (and thus $a_t \equiv 0$), then we simply refer to $\mathsf{R}^{0,\varepsilon}$ as kinetic Brownian motion. Whenever there is no risk of confusion, we omit the explicit dependence on $\gamma \geq 0$ and $\varepsilon > 0$ and simply write R for the reference process.

For any $t \in [0,1]$, let $\mu_t^a \coloneqq \mathrm{Law}(X_t^a, V_t^a)$ where $(X_t^a, V_t^a)_{t\in[0,1]}$ evolves according to (3). Under appropriate conditions on $(a_t)_{t\in[0,1]}$ (see e.g., Bogachev et al. (2022)), it can be shown that the pair $(\mu_t^a, a_t)_{t\in[0,1]}$ satisfies the *kinetic Fokker–Planck equation* (also called the Vlasov–Fokker–Planck equation), which is written as

$$\partial_t \mu_t^a + \langle \nabla_x \mu_t^a, v\rangle + \nabla_v \cdot (a_t \mu_t^a) = \frac{\varepsilon}{2}\Delta_v \mu_t^a\,. \tag{KFP}$$

3 Acceleration Matching

In this section, we present our proposed algorithm for trajectory inference called Acceleration Matching (AM).

Our main technical observations and contributions appear in Section 3.1, where we construct a suitable interpolation process, the acceleration field corresponding to the second-order flow, and the Markovianized process. Our complete algorithm appears in Section 3.2, which includes a detailed description of our simulation-free training objective. Finally, we describe the inference step of our method in Section 3.3.

3.1 Conditional acceleration fields and Markovianization

Recall our second-order dynamics of interest given by (3). Our goal is to define a particular family of acceleration fields that are easy to learn from positional data in the form of marginals $\{(t_j, \rho_{t_j})\}_{j=0}^J$. Importantly, we want the resulting dynamics to yield marginals that agree with (1), i.e., $\mathrm{Law}(X_{t_j}^a) = \rho_{t_j}$.

To this end, take an arbitrary coupling $\pi \in \Pi(\rho_{t_0}, \ldots, \rho_{t_J})$; for simplicity, consider the independent coupling of the marginals. Let $(X_t, V_t)_{t\in[0,1]} \sim \mathsf{R}$ be a kinetic (underdamped) Brownian motion for some fixed $\gamma \geq 0$ and $\varepsilon > 0$. We define our interpolation process $\mathsf{P}^I \in \mathcal{P}(\mathcal{C}^{2d})$ in two steps:

1. For $\boldsymbol{t} = (t_0, \ldots, t_J) \in [0,1]^{J+1}$, sample $X_{\boldsymbol{t}} \sim \pi$
2. Sample the remainder of the path from the regular conditional distribution $\mathsf{R}_{|\boldsymbol{t}}$,

where a formal definition of $\mathsf{R}_{|\boldsymbol{t}}$ is given in Appendix A. Mathematically, our interpolation process is written as

$$\mathsf{P}^I = \int \mathsf{R}_{|\boldsymbol{t}}(\boldsymbol{x}, \cdot)\,\pi(\mathrm{d}\boldsymbol{x})\,, \tag{4}$$

where $\mathsf{P}^I \in \mathcal{P}(\mathcal{C}^{2d})$. As we assume query access to π (at least in the case of independent marginals), it remains to efficiently sample from our conditional distribution $\mathsf{R}_{|\boldsymbol{t}}$. Thankfully, it turns out that

this remaining step corresponds to sampling from a suitable Gaussian distribution. Indeed, our next result shows that if the conditional distribution of V_0 given X_0 is Gaussian, then $V_{\boldsymbol{t}}$ given $X_{\boldsymbol{t}}$ is also Gaussian.

Lemma 1. *Assume that if $(X_0, V_0) \sim \mathsf{R}_0$, the conditional distribution of V_0 given X_0 is Gaussian almost surely with conditional mean $x_0 \mapsto m_0(x_0)$ and covariance matrix $x_0 \mapsto \Sigma_0(x_0)$. Then, the conditional distribution of $V_{\boldsymbol{t}}$ given $X_{\boldsymbol{t}}$ is also Gaussian with explicit conditional mean and covariance; see* (11)*.*

In brief, given only position knots $X_{\boldsymbol{t}}$, we can easily sample velocities $V_{\boldsymbol{t}} \mid X_{\boldsymbol{t}}$ in closed-form. The proof is an easy consequence of Lemma A.1 and is deferred to Appendix A.1.

The next step is to complete the sampling procedure for the *entire path* $(X_t, V_t)_{t\in[0,1]}$ given the $J+1$ knots $(X_{\boldsymbol{t}}, V_{\boldsymbol{t}})$ in phase space. In the case of standard Brownian motion, this would be immediate, as it reduces to simulating independent Brownian bridges. Remarkably, an analogous characterization holds for the kinetic damped Brownian motion in phase space. For the remainder of this section, we focus on the case $\gamma = 0$, deferring the general case to the appendix.

Lemma 2 (Kinetic Brownian bridges)**.** *Fix $t_j < t_{j+1} \in [0,1]$, $x_j, x_{j+1} \in \mathbb{R}^d$, $v_j, v_{j+1} \in \mathbb{R}^d$, and let R be kinetic Brownian motion with $\varepsilon > 0$. In the interval $[t_j, t_{j+1})$, the bridge of the reference process, i.e., the conditional distribution of $(X_t, V_t)_{t\in[t_j,t_{j+1}]}$ given $(X_{t_j}, V_{t_j}) = (x_j, v_j)$ and $(X_{t_{j+1}}, V_{t_{j+1}}) = (x_{j+1}, v_{j+1})$ is the unique weak solution starting from (X_{t_j}, V_{t_j}) to*

$$\mathrm{d}X_t = V_t\,\mathrm{d}t\,, \quad \mathrm{d}V_t = a_t^j(X_t, V_t; X_{t_{j+1}}, V_{t_{j+1}})\,\mathrm{d}t + \sqrt{\varepsilon}\,\mathrm{d}B_t^j\,, \tag{5}$$

on $[t_j, t_{j+1})$, where $\{(B_t^k)_{t\in[t_k,t_{k+1}]} : k \in \{0,\dots,J-1\}\}$ are independent d-dimensional Brownian motions and for any $x, v, x_{j+1}, v_{j+1} \in \mathbb{R}^d$ and $t \in [t_j, t_{j+1}]$,

$$a_t^j(x, v; x_{j+1}, v_{j+1}) = \frac{6(x_{j+1} - x)}{(t_{j+1} - t)^2} - \frac{2(v_{j+1} + 2v)}{(t_{j+1} - t)}\,. \tag{6}$$

The proof of Lemma 2 is based on an application of Doob's h-transform on phase space; see Appendix A.2. An important corollary of Lemma 2 is that the non-Markovian path measure P^I can now be represented as a solution to a sequence of stochastic differential equations: given $(X_{\boldsymbol{t}}^I, V_{\boldsymbol{t}}^I)$, Lemma 2 shows that for any $j \in \{0,\dots,J-1\}$, $(X_t^I, V_t^I)_{t\in[t_j,t_{j+1}]}$ is the unique weak solution starting from (X_{t_j}, V_{t_j}) to

$$\mathrm{d}X_t^I = V_t^I\,\mathrm{d}t\,, \quad \mathrm{d}V_t^I = a_t^j(X_t^I, V_t^I; X_{t_{j+1}}^I, V_{t_{j+1}}^I)\,\mathrm{d}t + \sqrt{\varepsilon}\,\mathrm{d}B_t^j\,, \tag{7}$$

over the interval $t \in [t_j, t_{j+1}]$.

Based on this observation, it becomes transparent that $(X_t^I, V_t^I)_{t\in[0,1]}$ need not be Markov. However, we can find a diffusion process $(X_t^M, V_t^M)_{t\in[0,1]}$ sharing the same marginals, replacing the acceleration field in (7) with a suitable Markovianization that will be easy to learn using a parametric model. This is formalized by our next result.

Theorem 1. *Let $(X_t^I, V_t^I)_{t\in[0,1]} \sim \mathsf{P}^I$ be an interpolated process associated with a coupling $\pi \in \Pi(\rho_{t_0}, \dots, \rho_{t_J})$ and a kinetic Brownian distribution R for $\varepsilon > 0$ fixed. Consider an acceleration field $(a_t^M)_{t\in[0,1]}$ satisfying almost surely*

$$a_t^M(X_t^I, V_t^I) \coloneqq \sum_{j=0}^{J-1} \mathbf{1}_{[t_j,t_{j+1})}(t)\,\mathbb{E}[a_t^j(X_t^I, V_t^I; X_{t_{j+1}}^I, V_{t_{j+1}}^I)|(X_t^I, V_t^I)]\,,$$

where a_t^j is given by (6)*. In addition, suppose that a_t^M is Lipschitz in all variables for any $t \in [0,1]$. For $(X_0^M, V_0^M) \sim \mathsf{P}_0^I$, consider $(X_t^M, V_t^M)_{t\in[0,1]}$ the diffusion process solution over $[0,1]$*

$$\mathrm{d}X_t^M = V_t^M\,\mathrm{d}t\,, \quad \mathrm{d}V_t^M = a_t^M(X_t^M, V_t^M)\,\mathrm{d}t + \sqrt{\varepsilon}\,\mathrm{d}B_t\,.$$

Then, for any $t \in [0,1]$, it holds that $\mathrm{Law}(X_t^M, V_t^M) = \mathrm{Law}(X_t^I, V_t^I)$, and therefore the positional marginals match the prescribed constraints.

In brief, Theorem 1 states that the marginal laws of the original non-Markovian interpolant can be recovered through a suitably defined acceleration field. In particular, under the stated regularity assumptions, this projection preserves the phase-space distribution at each time while simultaneously defining a Markovian law for the generated trajectories. This construction should be viewed as a direct *lifted* analogue of, e.g., flow matching (Albergo & Vanden-Eijnden, 2022; Lipman et al., 2023; Liu et al., 2022) in the two-marginal setting, corresponding to $J = 1$ in our framework.

Remark 3.1. We briefly mention the work of Chen et al. (2024), which also derives phase-space kinetic bridges. Their resulting acceleration field, which can be recovered as a particular case of the kinetic bridge acceleration derived above in Equation (6), is then approximated by regression. A separate work by Chen et al. (2025) extends this framework by augmenting the space to arbitrary N-variables which recovers Flow Matching for $N = 1$ and phase space dynamics for $N = 2$. Importantly, neither work addresses our setting where multiple marginals are present, and the goal of their work is to develop more efficient samplers for generative modeling of a single distribution of image data.

3.2 Loss function and training

We now describe a simple algorithm for learning the Markovian acceleration field from Theorem 1. Let a_θ denote a parametrized neural network such that $a_\theta : [0,1] \times \mathbb{R}^{2d} \to \mathbb{R}^d$. Given that a_t^M is a (piecewise) conditional expectation, it is natural to consider the following population loss

$$\mathcal{L}_{\mathtt{AM}}(a_\theta) = \sum_{j=0}^{J-1} \int_{t_j}^{t_{j+1}} \mathbb{E}\|a_\theta(t, X_t, V_t) - a_t^j(X_t, V_t; X_{t_{j+1}}, V_{t_{j+1}})\|^2 \,\mathrm{d}t\,, \tag{8}$$

where the expectation is under P^I. At each time $t \in (t_j, t_{j+1})$, it is easy to see that a_t^M is the unique minimizer to the conditional squared-error loss, and so it is omitted.[1]

We now describe a practical implementation of (8) in more detail (see also Algorithm 1). Recall that access to samples of positional data X_{t_j} is readily available from the marginals $\{\rho_{t_j}\}_{j=0}^J$. We then sample $V_{\boldsymbol{t}} \mid X_{\boldsymbol{t}}$ efficiently by Lemma 1 where, under the reference law R, we take $\mathrm{Law}(V_0 \mid X_0) = \mathbf{N}(0, \sigma_v^2 I)$ for $\sigma_v^2 > 0$ for simplicity. We sample $t \sim \mathrm{Unif}(0,1)$ and find the j^{th}-interval such that $t \in [t_j, t_{j+1})$. Also by Gaussian conditioning formulae, we can sample the kinetic Brownian bridge at the sampled time t; see Appendix B.2. Given all of these ingredients, we can now simply regress our acceleration field onto the explicit bridge acceleration target given by (6).

We stress that unlike prior work, our loss does not merely decompose into a learning problem on disjoint intervals, as the velocities used to train a_θ are jointly sampled across time conditioned on all positional knots. This allows us to learn smooth dynamics purely from the formulation of the learning problem instead of explicitly imposing spline-like dynamics into the training pipeline.

Remark 3.2. (On the choice of coupling.) The flexibility of our framework allows us to choose any coupling we want so long as it is used consistently throughout the process. In other words, we require the same coupling when training the acceleration field as when we learn the initial velocity law. With this in mind, we investigate the use of the pairwise OT coupling in addition to the independent coupling, as used in prior work.

3.3 Generating trajectories

To generate trajectories, we draw $X_0 \sim \rho_0$ and require access to the interpolant's initial law of $V_0 \mid X_0$. Although the initial joint law can be sampled by drawing all position knots and their conditional Gaussian velocities, this requires access to the training marginals and, when used, the coupling plans. To circumvent this requirement at inference time and amortize the cost, we instead learn a conditional sampler $V_0 \sim q_\phi(\cdot \mid X_0)$, where q_ϕ approximates the initial law of interest. We learn this sampler using conditional normalizing flows (NF) by obtaining initial position-velocity pairs via Lemma 1; see Appendix B.1 for more information. Then, given an initial position, one only needs the learned models, without retaining the later training marginals or coupling plans. Given the initial position and velocity, we then integrate the learned phase-space SDE using Euler–Maruyama, as specified in Algorithm 2.

[1] In practice, we use cutoffs and observe times in the window $[t_j + \delta, t_{j+1} - \delta]$.

Algorithm 1 Training algorithm for Acceleration Matching

Require: Times $0 = t_0 < \cdots < t_J = 1$; marginals $\{\rho_{t_j}\}_{j=0}^J$; coupling $\pi \in \Pi(\rho_{t_0}, \ldots, \rho_{t_J})$

Initialize network $a_\theta : [0,1] \times \mathbb{R}^{2d} \to \mathbb{R}^d$
while not converged **do**
  Sample t uniformly from $\mathrm{Unif}(0,1)$
  Find $j \in \{0, \ldots, J-1\}$ such that $t_j \le t < t_{j+1}$
  Sample all positional knots $\boldsymbol{x} \sim \pi$
  Sample all velocity knots $\boldsymbol{v} \mid \boldsymbol{x}$
  Sample (x_t, v_t) conditioned on (x_{t_j}, v_{t_j}) and $(x_{t_{j+1}}, v_{t_{j+1}})$
  Compute $\widehat{\mathcal{L}}_{\texttt{AM}}(a_\theta) = \|a_\theta(t, x_t, v_t) - a_t^j(x_t, v_t; x_{t_{j+1}}, v_{t_{j+1}})\|^2$
  Update $\theta \leftarrow \texttt{optim}(\theta, \nabla_\theta \widehat{\mathcal{L}}_{\texttt{AM}}(a_\theta))$
end while
Return a_θ

Algorithm 2 Trajectory generation via Acceleration Matching

Require: Learned conditional sampler $\widehat{q}_\phi$; learned acceleration $\widehat{a}_\theta$; noise scale ε; number of steps K

Initialize $x_0 \sim \rho_0$, draw $v_0 \sim q_\phi(\cdot \mid x_0)$
Set $h = 1/K$
for $k = 0, \ldots, K-1$ **do**
  Update position $x_{(k+1)h} \leftarrow x_{kh} + h\, v_{kh}$
  Draw $\xi_k \sim \mathcal{N}(0, I)$
  Update velocity $v_{(k+1)h} \leftarrow v_{kh} + h\, \widehat{a}_\theta(kh, x_{kh}, v_{kh}) + \sqrt{\varepsilon h}\, \xi_k$
end for
Return $(x_{kh})_{k=0}^K$

Remark 3.3. (Initial velocity ablation) Tables 1 and 2 use a conditional normalizing flow (NF) to generate samples from $V_0 \mid X_0$. Appendix D.1 compares this choice with a conditional flow-matching (FM) sampler under both independent and OT position couplings. The sampler configurations and training settings are collected in Appendix D.

4 Numerical experiments

We now present a set of experiments comparing AM to other state-of-the-art methods for the task of trajectory inference. Our main comparisons are with 3MSBM (Theodoropoulos et al., 2025) (which is also based on a lifting to phase space) and simulation-free flow-matching methods such as MMFM (Rohbeck et al., 2025), OT-CFM (Tong et al., 2024; Pooladian et al., 2023), and OT-MFM (Kapusniak et al., 2024).

On low-dimensional data, we benchmark against the official MMFM repository, with all default parameters, which we call "MMFM (prior)". For both low- and high-dimensional datasets, we implement MMFM and CFM (cubic and linear interpolation, respectively) with constant variance per the official code (note that this induces Gaussian smoothing at the marginal knots ρ_{t_j} rather than matching the marginals exactly; see Section 4.3). These methods always use pairwise OT couplings for their dynamics. We use the published 3MSBM W_2 results, which do not report standard deviations. For the remaining baselines, we will report the results from the corresponding papers, denoted by OT-CFM* and OT-MFM*.

As in prior work, we consider datasets of the form $\{(t_j, \widehat{\rho}_{t_j})\}_{j=0}^J$ where each $\widehat{\rho}_{t_j}$ is an empirical measure consisting of n_{t_j} atoms in $\mathbb{R}^d$. For each dataset, we train on a subset of the available marginals and evaluate whether the generated trajectories match held-out marginals. Following the conventions of prior work, we use the 2-Wasserstein metric as the evaluation metric for the low-dimensional datasets and the 1-Wasserstein metric for the high-dimensional single-cell datasets. Unless otherwise stated, we average results over five trials. To evaluate our methodology, we assess trajectory inference by measuring our ability to interpolate on held-out marginal data, and include visualizations of the generated paths.

Table 1: Average training and holdout W_2 errors for GoM and LV datasets; lower is better.

	GoM		LV	
Method	Holdout W_2	Train W_2	Holdout W_2	Train W_2
3MSBM*	0.135	—	0.230	—
OT-CFM	0.2664 ± 0.0270	0.1366 ± 0.0478	0.6357 ± 0.0681	0.4230 ± 0.0530
MMFM (prior)	0.209 ± 0.012	0.104 ± 0.015	0.427 ± 0.031	0.400 ± 0.052
MMFM	0.1911 ± 0.0181	0.1253 ± 0.0188	0.5294 ± 0.0356	0.4225 ± 0.0694
AM (indep.)	0.1727 ± 0.0217	0.1034 ± 0.0231	0.3742 ± 0.0584	0.2675 ± 0.1060

4.1 Low-dimensional datasets

We first evaluate AM on two low-dimensional datasets widely used in the literature: ocean-current trajectories in the Gulf of Mexico (GoM), and Lotka–Volterra (LV) dynamics (Shen et al., 2024). For the GoM and LV datasets, $J = 8$ (thus nine total marginals) and we train on the even-time marginals, holding out the odd times (t_1, t_3, t_5, t_7). At inference time, we sample initial positions from $\widehat{\rho}_{t_0}$, simulate trajectories forward in time, and compare the generated samples with both training and held-out marginals using the 2-Wasserstein distance. In Figures 6 and 2, we denote the held-out marginals by t_j^*.

Table 1 summarizes our results for the low-dimensional datasets. Under the reported configurations, our method has lower mean errors than MMFM and OT-CFM, though the reported 3MSBM holdout errors remain lower. Figure 2 contains representative generated trajectories for LV, with GoM trajectories deferred to Figure 6 in Appendix C.1. In all cases, we overlay the generated trajectories on ground-truth marginals, with hold-out times indicated on the legend.

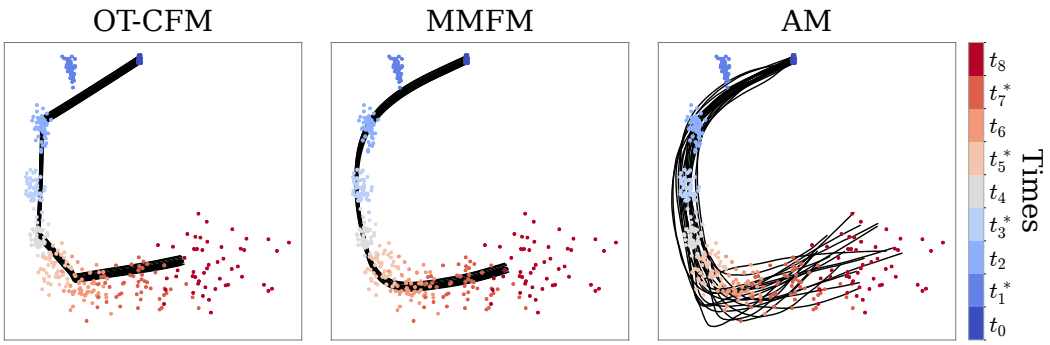


Figure 2: Visualization of trajectories generated for the LV dataset.

4.2 High-dimensional datasets

We next evaluate AM on higher-dimensional datasets that arise from single-cell biology. Following prior work, we use PCA representations of developing embryoid bodies (EB) (Moon et al., 2019) and CITE-seq data (CITE) (Burkhardt et al., 2022), reducing the dimensionality to 5 for EB5 and CITE5, 50 for CITE50, and 100 for EB100. For EB5, CITE5, and CITE50, we use a leave-one-marginal-out (LOO) protocol, always keeping the initial and final marginals in the training set, and report the average held-out W_1 error. For EB5, there are 5 marginals and we leave out one of t_1, t_2, or t_3, and average the resulting errors; for CITE5 and CITE50, we leave out either t_1 or t_2, and average the resulting errors. For EB100, we instead train on t_0, t_2, t_4 and hold out t_1, t_3 together, averaging the two held-out errors.

The high-dimensional results are summarized in Table 2. AM with pairwise-OT couplings has the lowest mean EB5 error among our implementations. For EB100, AM with independent coupling has the lowest reported mean error. On CITE5 and CITE50, OT coupling improves AM over independent coupling, although OT-CFM and OT-MMFM achieve lower mean errors.

Table 2: Holdout W_1 error across different datasets, averaged across five seeds; lower is better.

Method	EB5	EB100	CITE5	CITE50
OT-CFM*	0.790 ± 0.068	–	0.882 ± 0.058	38.756 ± 0.398
OT-MFM*	0.713 ± 0.039	–	0.724 ± 0.070	36.394 ± 1.886
OT-CFM	0.8367 ± 0.0087	13.2210 ± 0.0445	0.4912 ± 0.0036	39.1685 ± 0.2996
OT-MMFM	0.7738 ± 0.0209	12.9282 ± 0.0515	0.5321 ± 0.0092	41.2779 ± 0.4041
AM (OT)	0.7408 ± 0.0151	13.1696 ± 0.1187	0.5417 ± 0.0089	42.5220 ± 0.1630
AM (indep.)	0.8183 ± 0.0548	12.8698 ± 0.0903	0.5752 ± 0.0270	46.0090 ± 0.5974

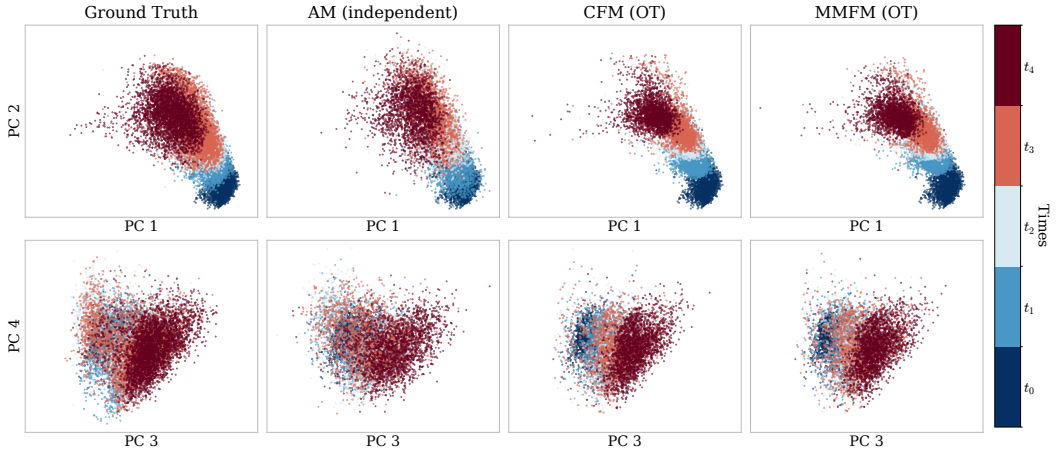


Figure 3: Observed and generated marginals for EB5, with t_1 held out.

4.3 Related work

Multi-sample flow matching and related algorithms

Flow matching (Lipman et al., 2023; Albergo & Vanden-Eijnden, 2022; Liu et al., 2022) is a simulation-free approach for generating probability paths between two marginals. Multi-marginal flow matching (MMFM) is a simulation-free approach proposed by Rohbeck et al. (2025), which uses ideas reminiscent of other works that incorporate data-dependent couplings (Tong et al., 2024; Pooladian et al., 2023; Albergo et al., 2023) combined with using cubic splines. Similar to the proposal by Chewi et al. (2021), they first compute pairwise optimal transport matchings between the marginals. For datasets of moderate size, these matchings can be obtained using a standard optimal transport package (Flamary et al., 2021) where, writing $n = \max_j n_{t_j}$ as the maximum number of points per marginal, dense pairwise cost or coupling matrices require $\mathcal{O}(Jn^2)$ storage; and standard solvers yield a runtime of $\mathcal{O}(Jn^3)$. Then, given such a pairwise aligned coupling, they sample data $X_{0:J} \coloneqq (X_0, \dots, X_J) \sim \text{pairwise}(\rho_0, \dots, \rho_J)$, and approximate the following velocity field

$$\mathbb{E}[\partial_t \,\text{spline}(t; X_{0:J}) + \sigma_t' \xi \,|\, \text{spline}(t; X_{0:J}) + \sigma_t \xi = z]\,,$$

where $\text{spline}(t; X_{0:J})$ is a cubic spline between the drawn knots, $\sigma_t \geq 0$ is the noise coefficient, and $\xi \sim \mathcal{N}(0, I)$. Upon learning the resulting vector field, they simulate the resulting ordinary differential equation to obtain the trajectories. Rohbeck et al. (2025) were also interested in learning conditional velocity fields via classifier-free guidance, which we do not pursue here.

The code provided by Rohbeck et al. (2025) mostly uses a constant variance, i.e., $\sigma_t' = 0$, and so our implementation of MMFM does the same. We note that this choice of constant variance induces positional marginals which are Gaussian-smoothed versions of the prescribed marginal knots.

Optimal transport conditional flow matching (OT-CFM) uses minibatch optimal transport couplings between the marginals when fitting a velocity field (Tong et al., 2024; Pooladian et al., 2023).

Kapusniak et al. (2024) proposed OT-Metric Flow Matching (OT-MFM), which incorporates a Riemannian metric from the data in addition to learning a vector field. The Riemannian metrics take the form of kernel density estimators (e.g., radial basis functions).

MULTI-MARGINAL SCHRÖDINGER BRIDGE METHODS

Many neural-network-based algorithms for solving the trajectory inference problem are based on multi-marginal Schrödinger bridges. For instance, multi-marginal stochastic flow matching (MMSFM) (Lee et al., 2025) is a variant of MMFM that incorporates some ideas from Shi et al. (2024), and uses OT couplings between pairs or triplets of marginals. Schrödinger bridge with iterative reference refinement (SBIRR) (Shen et al., 2024) requires some degree of trajectory simulation as part of the algorithm and has non-smooth positional trajectories as they use standard SDEs, not lifted ones, for generating trajectories.

The work of Theodoropoulos et al. (2025) also passes to phase space (which they refer to as "momentum"), where their goal is to learn a Markovian acceleration field. Apart from this similarity, our proposed method is quite distinct from theirs. They regress on a different conditional acceleration field, one which depends on *several* future positions in a non-trivial way. Their proposed conditional acceleration field is defined as

$$a_t^{\mathrm{3MSBM}}((x_t, v_t) \,|\, \{\boldsymbol{x}_{j+1} \,:\, t_{j+1} \geq t\}) = C_1^j(t)(x_t - \boldsymbol{x}_{j+1}) + C_2^j(t)v_t + C_3^j(t)\textstyle\sum_{\ell=j+1}^{J} \lambda_\ell \boldsymbol{x}_\ell \,,$$

for $t \in [t_j, t_{j+1})$, where $\{\boldsymbol{x}_j\}_{j=0}^J$ denotes the sampled knots, $\{\{C_k^j(t)\}_{k=1}^3\}_{j=0}^J$ are time-varying coefficients corresponding to each segment, and $\{\lambda_j\}_{j=0}^J$ are also computed. Their training pipeline iteratively propagates conditional dynamics to obtain conditional velocities. This allows them to avoid backpropagating through trajectories, though simulation is still costly. Our construction instead directly samples the velocity knots by simple Gaussian conditioning, followed by direct sampling of intermediate bridge states and the explicit target in (6).

5 CONCLUSION

We propose Acceleration Matching as a principled algorithm for trajectory inference. To learn continuously differentiable positional trajectories, we lift the learning problem to phase space, with the goal of learning an appropriate Markovianized acceleration field. This perspective yields an explicit regression objective for learning dynamics with multiple marginal constraints, as all arguments and targets can be directly computed via simple Gaussian conditioning. Our AM has the flexibility of avoiding pairwise-OT preprocessing by using the independent coupling, and we use a conditional initializer to amortize the velocities sampled at inference time. A limitation of our method is its sensitivity to hyperparameter choices, which may require problem-specific tuning to achieve strong performance. Several natural directions remain beyond the scope of this work, such as investigating the role of the reference process, improving overall algorithmic complexity, extending our framework to instance-specific Riemannian manifolds (Chen & Lipman, 2023; Yim et al., 2023; Haviv et al., 2025), quantifying discretization guarantees in the spirit of Chen et al. (2023a) and Conforti et al. (2025), and applying our method to more computationally intensive settings where either J or d (or both) are large.

AI USE STATEMENT

AI tools were used to assist with implementing methods, provide feedback on experimental methodology, interpret results, and edit the manuscript to improve readability and presentation. The authors take responsibility for the final content of the paper, including its claims, proofs, and experimental results.

REFERENCES

Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

Michael S Albergo, Mark Goldstein, Nicholas M Boffi, Rajesh Ranganath, and Eric Vanden-Eijnden. Stochastic interpolants with data-dependent couplings. *arXiv preprint arXiv:2310.03725*, 2023.

Dominique Bakry, Ivan Gentil, and Michel Ledoux. *Analysis and geometry of Markov diffusion operators*, volume 103. Springer, 2014.

Vladimir I Bogachev, Nicolai V Krylov, Michael Röckner, and Stanislav V Shaposhnikov. *Fokker–Planck–Kolmogorov Equations*, volume 207. American Mathematical Society, 2022.

Charlotte Bunne, Laetitia Papaxanthos, Andreas Krause, and Marco Cuturi. Proximal optimal transport modeling of population dynamics. In *International Conference on Artificial Intelligence and Statistics*, pp. 6511–6528. PMLR, 2022.

Charlotte Bunne, Stefan G Stark, Gabriele Gut, Jacobo Sarabia del Castillo, Mitch Levesque, Kjong-Van Lehmann, Lucas Pelkmans, Andreas Krause, and Gunnar Rätsch. Learning single-cell perturbation responses using neural optimal transport. *Nature methods*, 20(11):1759–1768, 2023.

D. Burkhardt, M. Luecken, A. Benz, P. Holderrieth, J. Bloom, C. Lance, A. Chow, and R. Holbrook. Open problems - multimodal single-cell integration. `https://kaggle.com/competitions/open-problems-multimodal`, 2022. Kaggle.

Ricky TQ Chen and Yaron Lipman. Flow matching on general geometries. *arXiv preprint arXiv:2302.03660*, 2023.

Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru R. Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. In *The Eleventh International Conference on Learning Representations*, 2023a.

Tianrong Chen, Guan-Horng Liu, Molei Tao, and Evangelos Theodorou. Deep momentum multi-marginal Schrödinger bridge. *Advances in Neural Information Processing Systems*, 36:57058–57086, 2023b.

Tianrong Chen, Jiatao Gu, Laurent Dinh, Evangelos A. Theodorou, Joshua Susskind, and Shuangfei Zhai. Generative modeling with phase stochastic bridges. In *International Conference on Learning Representations*, 2024. URL `https://arxiv.org/abs/2310.07805`.

Tianrong Chen, Huangjie Zheng, David Berthelot, Jiatao Gu, Joshua Susskind, and Shuangfei Zhai. TADA: Improved diffusion sampling with training-free augmented DynAmics. In *Advances in Neural Information Processing Systems*, volume 38, pp. 130031–130060, 2025. doi: 10.52202/085713-3911. URL `https://arxiv.org/abs/2506.21757`.

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.

Giovanni Conforti, Alain Durmus, and Marta Gentiloni Silveri. KL convergence guarantees for score diffusion models under minimal data assumptions. *SIAM Journal on Mathematics of Data Science*, 7(1):86–109, 2025.

Nabarun Deb, Promit Ghosal, and Bodhisattva Sen. Rates of estimation of optimal transport maps using plug-in estimators via barycentric projections. *Advances in Neural Information Processing Systems*, 34:29736–29753, 2021.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. In *International Conference on Learning Representations*, 2017. URL `https://openreview.net/forum?id=HkpbnH9lx`.

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. POT: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021. URL `http://jmlr.org/papers/v22/20-451.html`.

Christian LE Franzke, Terence J O'Kane, Judith Berner, Paul D Williams, and Valerio Lucarini. Stochastic climate theory and modeling. *Wiley Interdisciplinary Reviews: Climate Change*, 6(1): 63–78, 2015.

Narendra S Goel, Samaresh C Maitra, and Elliott W Montroll. On the Volterra and other nonlinear models of interacting populations. *Reviews of modern physics*, 43(2):231, 1971.

Doron Haviv, Aram-Alexandre Pooladian, Dana Pe'Er, and Brandon Amos. Wasserstein flow matching: Generative modeling over families of distributions. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 22238–22258, 2025.

Wanli Hong, Yuliang Shi, and Jonathan Niles-Weed. Trajectory inference with smooth Schrödinger bridges. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 23722–23750. PMLR, 13–19 Jul 2025.

HYCOM Consortium and Center for Ocean-Atmospheric Prediction Studies (COAPS). HYCOM-TSIS 1/100° Gulf of Mexico Reanalysis (GOMb0.01). Dataset, 2024. URL https://www.hycom.org/data/gomb0pt01/gom-reanalysis. Accessed 2026-07-03.

Kacper Kapusniak, Peter Potaptchik, Teodora Reu, Leo Zhang, Alexander Tong, Michael M. Bronstein, Joey Bose, and Francesco Di Giovanni. Metric flow matching for smooth interpolations on the data manifold. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=fE3RqiF4Nx.

Rytis Kazakevičius, Aleksejus Kononovicius, Bronislovas Kaulakys, and Vygintas Gontis. Understanding the nature of the long-range memory phenomenon in socioeconomic systems. *Entropy*, 23(9):1125, 2021.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Jeremy Kubica, Larry Denneau, Tommy Grav, James Heasley, Robert Jedicke, Joseph Masiero, Andrea Milani, Andrew Moore, David Tholen, and Richard J Wainscoat. Efficient intra- and inter-night linking of asteroid detections using kd-trees. *Icarus*, 189(1):151–168, 2007.

Hugo Lavenant, Stephen Zhang, Young-Heon Kim, and Geoffrey Schiebinger. Toward a mathematical theory of trajectory inference. *The Annals of Applied Probability*, 34(1A):428–500, 2024.

Justin Lee, Behnaz Moradijamei, and Heman Shakeri. Multi-marginal stochastic flow matching for high-dimensional snapshot data at irregular time points. *arXiv preprint arXiv:2508.04351*, 2025.

Andrew J Liounis, Jeffrey L Small, Jason C Swenson, Joshua R Lyzhoft, Benjamin W Ashman, Kenneth M Getzandanner, Michael C Moreau, Coralie D Adam, Jason M Leonard, Derek S Nelson, et al. Autonomous detection of particles and tracks in optical images. *Earth and Space Science*, 7 (8):e2019EA000843, 2020.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

Tudor Manole, Sivaraman Balakrishnan, Jonathan Niles-Weed, and Larry Wasserman. Plugin estimation of smooth optimal transport maps. *The Annals of Statistics*, 52(3):966–998, 2024.

K. R. Moon, D. van Dijk, Z. Wang, S. Gigante, D. B. Burkhardt, W. S. Chen, K. Yim, Antonia van den Elzen, M. J. Hirn, R. R. Coifman, N. B. Ivanova, G. Wolf, and S. Krishnaswamy. Visualizing structure and transitions in high-dimensional biological data. *Nature Biotechnology*, 37: 1482–1492, 2019. doi: 10.1038/s41587-019-0336-3. URL https://doi.org/10.1038/s41587-019-0336-3.

K. Neklyudov, R. Brekelmans, A. Tong, L. Atanackovic, Q. Liu, and A. Makhzani. A computational framework for solving Wasserstein lagrangian flows. In R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 37461–37485. PMLR, 2024. URL https://proceedings.mlr.press/v235/neklyudov24a.html.

Aram-Alexandre Pooladian and Jonathan Niles-Weed. Entropic estimation of optimal transport maps. *arXiv preprint arXiv:2109.12004*, 2021.

Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky T. Q. Chen. Multisample flow matching: Straightening flows with minibatch couplings. In *Proceedings of the 40th International Conference on Machine Learning*. PMLR, 23–29 Jul 2023.

Martin Rohbeck, Edward De Brouwer, Charlotte Bunne, Jan-Christian Huetter, Anne Biton, Kelvin Y Chen, Aviv Regev, and Romain Lopez. Modeling complex system dynamics with flow matching across time and conditions. In *The Thirteenth International Conference on Learning Representations*, 2025.

Geoffrey Schiebinger, Jian Shu, Marcin Tabaka, Brian Cleary, Vidya Subramanian, Aryeh Solomon, Joshua Gould, Siyan Liu, Stacie Lin, Peter Berube, et al. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, 176(4):928–943, 2019.

Yunyi Shen, Renato Berlinghieri, and Tamara Broderick. Multi-marginal Schrödinger bridges with iterative reference refinement. *arXiv preprint arXiv:2408.06277*, 2024.

Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion Schrödinger bridge matching. *Advances in Neural Information Processing Systems*, 36, 2024.

Panagiotis Theodoropoulos, Augustinos Saravanos, Evangelos Theodorou, and Guan-Horng Liu. Momentum multi-marginal Schrödinger bridge matching. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 85885–85922, 2025.

A. Tong, J. Huang, G. Wolf, D. van Dijk, and S. Krishnaswamy. Trajectorynet: A dynamic optimal transport network for modeling cellular dynamics. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9526–9536. PMLR, 2020. URL https://proceedings.mlr.press/v119/tong20a.html.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=CD9Snc73AW. Expert Certification.

D. Wang, Y. Jiang, Z. Zhang, X. Gu, P. Zhou, and J. Sun. Joint velocity-growth flow matching for single-cell dynamics modeling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=aXAkNlbnGa.

Christina Winkler, Daniel Worrall, Emiel Hoogeboom, and Max Welling. Learning likelihoods with conditional normalizing flows. *arXiv preprint arXiv:1912.00042*, 2019. URL https://arxiv.org/abs/1912.00042.

Grace Hui Ting Yeo, Sachit D Saksena, and David K Gifford. Generative modeling of single-cell time series with prescient enables prediction of cell trajectories with interventions. *Nature communications*, 12(1):3222, 2021.

Jason Yim, Brian L. Trippe, Valentin De Bortoli, Emile Mathieu, Arnaud Doucet, Regina Barzilay, and Tommi Jaakkola. SE(3) diffusion model with application to protein backbone generation. In *Proceedings of the 40th International Conference on Machine Learning*. PMLR, 23–29 Jul 2023.

A SUPPLEMENT OF SECTION 3

We begin with some additional notation.

We denote by $(X_t)_{t\in[0,1]}$ the canonical process, i.e., $X_t(h) = h_t \in \mathbb{R}^d$ for any $h \in \mathcal{C}^d$ and any $t \in [0,1]$. For any t, Q_t denotes the t-marginal $(X_t)_\sharp \mathsf{Q}$ for $\mathsf{Q} \in \mathcal{P}(\mathcal{C}^d)$. By a slight abuse of notation, we denote by $(X_t, V_t)_{t\in[0,1]}$ the canonical process on the phase space $\mathbb{R}^{2d}$, i.e., $X_t(h) = \mathrm{proj}_x(h_t)$ and $V_t(h) = \mathrm{proj}_v(h_t)$ for any $h \in \mathcal{C}^{2d}$ and any $t \in [0,1]$. Here, proj_x and proj_v denote the projections onto the first and last d components, respectively.

We also recall the dynamics associated with kinetic underdamped Brownian motion, denoted $\mathsf{R} = \mathsf{R}^{\gamma,\varepsilon}$

$$\begin{cases} \mathrm{d}X_t = V_t \,\mathrm{d}t \\ \mathrm{d}V_t = -\gamma V_t \,\mathrm{d}t + \sqrt{\varepsilon}\,\mathrm{d}B_t \,. \end{cases} \tag{9}$$

For additional shorthand, let $Z_t \coloneqq (X_t, V_t)$ for $t \in [0,1]$.

We now provide a formal definition of $\mathsf{R}_{|\boldsymbol{t}}$; this is a Markov kernel on $(\mathbb{R}^d)^{J+1} \times \mathcal{B}(\mathcal{C}^{2d})$ satisfying for any measurable sets $A \in \mathcal{B}(\mathcal{C}^{2d})$ and $B \in \mathcal{B}((\mathbb{R}^d)^{J+1})$, $\mathsf{R}(A \cap \{X_{\boldsymbol{t}} \in B\}) = \mathbb{E}[\mathbf{1}_B(X_{\boldsymbol{t}})\mathsf{R}_{|\boldsymbol{t}}(X_{\boldsymbol{t}}, A)]$. Recall the definition of P^I from (4).

A.1 PROOF OF LEMMA 1

Lemma A.1. *Conditionally on (X_0, V_0), the reference process (9) is a Gaussian process with mean and covariance functions given by, for $s, t \in [0,1]$ and $s \le t$,*

$$m_Z(t) = \begin{pmatrix} X_0 + V_0 a_t \\ V_0 b_t \end{pmatrix} = A_t Z_0 \,,$$

where $a_t = \gamma^{-1}(1 - \mathrm{e}^{-\gamma t})$, $b_t = \mathrm{e}^{-\gamma t}$, and

$$A_t = \begin{pmatrix} 1 & a_t \\ 0 & b_t \end{pmatrix} \otimes \mathrm{I}_d \,.$$

Moreover, it holds that for $s \le t$, $\Gamma_Z(s,t) \coloneqq \mathrm{Cov}(Z_s, Z_t \mid Z_0)$ is given by

$$\Gamma_Z(s,t) = \begin{pmatrix} \mathrm{Var}(X_s) + a_{t-s}\mathrm{Cov}(X_s, V_s) & b_{t-s}\mathrm{Cov}(X_s, V_s) \\ \mathrm{Cov}(X_s, V_s) + a_{t-s}\mathrm{Var}(V_s) & b_{t-s}\mathrm{Var}(V_s) \end{pmatrix} \otimes \mathrm{I}_d \,,$$

where

$$\begin{aligned} \mathrm{Var}(V_s) &= (\varepsilon/(2\gamma))(1 - \mathrm{e}^{-2\gamma s}) \,, \\ \mathrm{Var}(X_s) &= (\varepsilon/\gamma^2)\left[s - 2a_s + (1/(2\gamma))(1 - \mathrm{e}^{-2\gamma s})\right] \,, \\ \mathrm{Cov}(X_s, V_s) &= (\varepsilon/2)a_s^2 \,. \end{aligned}$$

In particular, if $\gamma = 0$, then the above simplifies to

$$m_Z(t) = \begin{pmatrix} X_0 + V_0 t \\ V_0 \end{pmatrix} = A_t Z_0 \,, \qquad A_t = \begin{pmatrix} 1 & t \\ 0 & 1 \end{pmatrix} \otimes \mathrm{I}_d \,,$$

and

$$\Gamma_Z(s,t) = \varepsilon \begin{pmatrix} s^2(3t - s)/6 & s^2/2 \\ s^2/2 + s(t-s) & s \end{pmatrix} \otimes \mathrm{I}_d \,.$$

Proof. First, we only show the result for $d = 1$ since given (X_0, V_0) all components are independent and therefore the result tensorizes. Note that for $s, t \in [0,1]$, we can readily solve the SDE for the velocity term to yield

$$V_t = \mathrm{e}^{-\gamma(t-s)} V_s + \eta^V_{s,t} \coloneqq \mathrm{e}^{-\gamma(t-s)} V_s + \sqrt{\varepsilon} \int_s^t \mathrm{e}^{-\gamma(t-u)} \,\mathrm{d}B_u \,,$$

and for $s = 0$, we have that $V_t = \mathrm{e}^{-\gamma t} V_0 + \sqrt{\varepsilon} \int_0^t \mathrm{e}^{-\gamma(t-u)} \, \mathrm{d}B_u$. Solving for the position variable then yields

$$\begin{aligned} X_t = X_s + \int_s^t V_u \, \mathrm{d}u &= X_s + a_{t-s} V_s + \eta_{s,t}^X \\ &\coloneqq X_s + a_{t-s} V_s + \sqrt{\varepsilon} \int_s^t \int_s^u \mathrm{e}^{-\gamma(u-r)} \, \mathrm{d}B_r \, \mathrm{d}u \,, \end{aligned}$$

where in the second step we used that $\int_s^t \mathrm{e}^{-\gamma(u-s)} \, \mathrm{d}u = a_{t-s}$. Therefore, as a solution of a linear SDE, $(X_t, V_t)_{t \in [0,1]}$ is indeed a Gaussian process given initial datum (X_0, V_0). We now proceed and compute the remaining explicit expressions, which make heavy use of the Itô isometry and the Markov property of the Brownian motion.

To start, we compute the double integral as

$$\int_s^t \int_s^u \mathrm{e}^{-\gamma(u-r)} \, \mathrm{d}B_r \, \mathrm{d}u = \int_s^t a_{t-r} \, \mathrm{d}B_r \,,$$

which further simplifies the expression for X_t, which will be useful later.

We readily compute the means of the Gaussian process as

$$\mathbb{E}[V_t \,|\, (X_0, V_0)] = \mathrm{e}^{-\gamma t} V_0 \,, \quad \mathbb{E}[X_t \,|\, (X_0, V_0)] = X_0 + a_t V_0 \,,$$

which implies the representation $m_Z(t) = A_t Z_0$ and the accompanying matrix A_t.

Conditionally on (X_0, V_0), define the centered variables

$$\overline{V}_s = \sqrt{\varepsilon} \int_0^s \mathrm{e}^{-\gamma(s-u)} \, \mathrm{d}B_u \,, \quad \overline{X}_s = \sqrt{\varepsilon} \int_0^s a_{s-u} \, \mathrm{d}B_u \,,$$

and then we readily compute

$$\mathrm{Var}(V_s) = \mathbb{E}\Big[\overline{V}_s^2\Big] = \varepsilon \int_0^s \mathrm{e}^{-2\gamma(s-u)} \, \mathrm{d}u = \frac{\varepsilon}{2\gamma} (1 - \mathrm{e}^{-2\gamma s}) \,,$$

and similarly

$$\begin{aligned} \mathrm{Var}(X_s) = \mathbb{E}\Big[\overline{X}_s^2\Big] = \varepsilon \int_0^s a_{s-u}^2 \, \mathrm{d}u &= \frac{\varepsilon}{\gamma^2} \int_0^s (1 - 2\mathrm{e}^{-\gamma r} + \mathrm{e}^{-2\gamma r}) \, \mathrm{d}r \\ &= \frac{\varepsilon}{\gamma^2} \big(s - 2a_s + (2\gamma)^{-1} (1 - \mathrm{e}^{-2\gamma s}) \big) \,, \end{aligned}$$

and also

$$\mathrm{Cov}(X_s, V_s) = \varepsilon \int_0^s \mathrm{e}^{-\gamma(s-u)} a_{s-u} \, \mathrm{d}u = \frac{\varepsilon}{2} a_s^2 \,.$$

We now compute the covariance between times $s \leq t$. Note that V_t can be written in terms of V_s and an additional Brownian motion term that is independent of V_s (and similarly for X_s). Due to this independence, we can compute the covariance between times as

$$\mathrm{Cov}(V_s, V_t) = \mathrm{Cov}(V_s, \mathrm{e}^{-\gamma(t-s)} V_s + \eta_{s,t}^V) = \mathrm{Cov}(V_s, \mathrm{e}^{-\gamma(t-s)} V_s) = \mathrm{e}^{-\gamma(t-s)} \mathrm{Var}(V_s) \,,$$

and similarly

$$\begin{aligned} \mathrm{Cov}(X_s, V_t) &= \mathrm{Cov}(X_s, \mathrm{e}^{-\gamma(t-s)} V_s) = \mathrm{e}^{-\gamma(t-s)} \mathrm{Cov}(X_s, V_s) \,, \\ \mathrm{Cov}(V_s, X_t) &= \mathrm{Cov}(V_s, X_s + a_{t-s} V_s) = \mathrm{Cov}(X_s, V_s) + a_{t-s} \mathrm{Var}(V_s) \,, \\ \mathrm{Cov}(X_s, X_t) &= \mathrm{Cov}(X_s, X_s + a_{t-s} V_s) = \mathrm{Var}(X_s) + a_{t-s} \mathrm{Cov}(X_s, V_s) \,, \end{aligned}$$

which concludes the proof. □

Recall our notation conventions $0 = t_0 < t_1 < \cdots < t_J = 1$ with $\boldsymbol{t} = (t_0, \ldots, t_J)$. Additionally define $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^{J+1}$ as

$$\boldsymbol{a} \coloneqq a_{\boldsymbol{t}} \coloneqq (a_{t_0}, \ldots, a_{t_J}) \,, \quad \boldsymbol{b} \coloneqq b_{\boldsymbol{t}} \coloneqq (\mathrm{e}^{-\gamma t_0}, \ldots, \mathrm{e}^{-\gamma t_J}) \,,$$

as well as $\boldsymbol{q} \coloneqq (\boldsymbol{a}, \boldsymbol{b}) \in \mathbb{R}^{2(J+1)}$. Additionally, recall

$$V_{\boldsymbol{t}} = (V_0 \quad V_{t_1} \quad \ldots \quad V_{t_J})^\top \in \mathbb{R}^{(J+1)d} \,, \qquad X_{\boldsymbol{t}} = (X_0 \quad X_{t_1} \quad \ldots \quad X_{t_J})^\top \in \mathbb{R}^{(J+1)d} \,.$$

Lemma A.2. *Let $X_0 = x_0 \in \mathbb{R}^d$ be fixed, and assume that*

$$V_0 \mid X_0 = x_0 \sim \mathbf{N}(m_0(x_0), \Sigma_0(x_0))\,.$$

Then conditionally on $X_0 = x_0$, $(X_{\boldsymbol{t}}, V_{\boldsymbol{t}})$ is (jointly) Gaussian with mean $\boldsymbol{m} = (m_{\boldsymbol{X}}, m_{\boldsymbol{V}})$, where the components are

$$m_{\boldsymbol{X}} \coloneqq \mathbf{1}_{J+1} \otimes x_0 + \boldsymbol{a} \otimes m_0(x_0)\,, \quad m_{\boldsymbol{V}} = \boldsymbol{b} \otimes m_0(x_0)\,, \tag{10}$$

and covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{2(J+1)d \times 2(J+1)d}$ given by

$$\boldsymbol{\Sigma} \coloneqq \begin{pmatrix} \Sigma_{\boldsymbol{X}} & \Sigma_{\boldsymbol{XV}} \\ \Sigma_{\boldsymbol{VX}} & \Sigma_{\boldsymbol{V}} \end{pmatrix} \coloneqq (\boldsymbol{q}\boldsymbol{q}^\top \otimes \Sigma_0(x_0)) + \begin{pmatrix} K_{\boldsymbol{X}} & K_{\boldsymbol{VX}}^\top \\ K_{\boldsymbol{VX}} & K_{\boldsymbol{V}} \end{pmatrix} \otimes I_d\,,$$

where for $i, k \in \{0, \dots, J\}$ we define

$$\begin{aligned} [K_{\boldsymbol{X}}]_{ik} &= \mathrm{Var}(X_{t_i \wedge t_k}) + a_{|t_i - t_k|}\mathrm{Cov}(X_{t_i \wedge t_k}, V_{t_i \wedge t_k})\,, \\ [K_{\boldsymbol{V}}]_{ik} &= \mathrm{e}^{-\gamma|t_i - t_k|}\mathrm{Var}(V_{t_i \wedge t_k})\,, \\ [K_{\boldsymbol{VX}}]_{ik} &= \begin{cases} \mathrm{Cov}(X_{t_i}, V_{t_i}) + a_{t_k - t_i}\mathrm{Var}(V_{t_i}) & i \le k\,, \\ \mathrm{e}^{-\gamma(t_i - t_k)}\mathrm{Cov}(X_{t_k}, V_{t_k}) & i > k\,. \end{cases} \end{aligned}$$

Moreover, $V_{\boldsymbol{t}} \mid X_{\boldsymbol{t}}$ is Gaussian with mean and covariance given by

$$\begin{aligned} m_{\boldsymbol{V}|\boldsymbol{X}} &= m_{\boldsymbol{V}} + \Sigma_{\boldsymbol{VX}}\Sigma_{\boldsymbol{X}}^\dagger(X_{\boldsymbol{t}} - m_{\boldsymbol{X}})\,, \\ \Sigma_{\boldsymbol{V}|\boldsymbol{X}} &= \Sigma_{\boldsymbol{V}} - \Sigma_{\boldsymbol{VX}}\Sigma_{\boldsymbol{X}}^\dagger\Sigma_{\boldsymbol{XV}}\,, \end{aligned} \tag{11}$$

where $\Sigma_{\boldsymbol{X}}^\dagger$ denotes the Moore–Penrose pseudoinverse of $\Sigma_{\boldsymbol{X}}$.

Proof. By Lemma A.1, given (X_0, V_0), the vector $(X_{\boldsymbol{t}}, V_{\boldsymbol{t}})$ is jointly Gaussian, so all that remains is to compute the mean and covariance.

Conditional on $X_0 = x_0$, we have by the representations in Lemma A.1 that

$$X_{t_i} = x_0 + a_{t_i}V_0 + \eta_i^X\,, \quad V_{t_i} = b_{t_i}V_0 + \eta_i^V\,, \tag{12}$$

where we recall that we defined $b_t = \mathrm{e}^{-\gamma t}$. We further write the random variable $V_0 \mid X_0 = x_0$ as $V_0 = m_0(x_0) + \xi_0$ for $\xi_0 \sim \mathbf{N}(0, \Sigma_0(x_0))$. Taking expectations leads to the definitions of $m_{\boldsymbol{X}}$ and $m_{\boldsymbol{V}}$ since ξ_0, η_i^X, and η_i^V have mean zero. Then rewriting (12), we have

$$(X_{\boldsymbol{t}}, V_{\boldsymbol{t}}) = (m_{\boldsymbol{X}}, m_{\boldsymbol{V}}) + (\boldsymbol{q} \otimes I_d)\xi_0 + (\eta_{\boldsymbol{t}}^X, \eta_{\boldsymbol{t}}^V)\,,$$

where, to compute the covariance, we separate the two sources of noise (which are independent from one another). We directly compute that

$$\mathrm{Cov}((\boldsymbol{q} \otimes I_d)\xi_0) = (\boldsymbol{q} \otimes I_d)\mathrm{Cov}(\xi_0)(\boldsymbol{q} \otimes I_d)^\top = (\boldsymbol{q}\boldsymbol{q}^\top) \otimes \Sigma_0(x_0)\,,$$

and for $\mathrm{Cov}(\eta_{\boldsymbol{t}}^X, \eta_{\boldsymbol{t}}^V)$, we appeal to the existing formulae in Lemma A.1. The remaining conditional Gaussian formulas follow from standard computations that we omit. □

We end this section by computing a pseudoinverse of $\Sigma_{\boldsymbol{X}}$.

Lemma A.3. *Let $\gamma \ge 0, \varepsilon > 0$ and consider the reference process $\mathsf{R}^{\gamma,\varepsilon}$, and recall the notation and setting of Lemma A.2. Write $L_{\boldsymbol{X}} \in \mathbb{R}^{J \times J}$ such that $[L_{\boldsymbol{X}}]_{ik} = [K_{\boldsymbol{X}}]_{ik}$ for $i, k \in \{1, \dots, J\}$. Then $L_{\boldsymbol{X}}$ is positive definite, and it holds that*

$$(\Sigma_{\boldsymbol{X}})^\dagger = \begin{pmatrix} 0 & 0 \\ 0 & \Theta^{-1} \end{pmatrix}\,, \tag{13}$$

where, for $\tilde{\boldsymbol{a}} = (a_{t_1}, \dots, a_{t_J}) \in \mathbb{R}^J$, Θ is defined as

$$\Theta = \tilde{\boldsymbol{a}}\tilde{\boldsymbol{a}}^\top \otimes \Sigma_0(x_0) + L_{\boldsymbol{X}} \otimes I_d\,,$$

and its inverse is given by

$$\Theta^{-1} = L_{\boldsymbol{X}}^{-1} \otimes I_d - (L_{\boldsymbol{X}}^{-1}\tilde{\boldsymbol{a}}\tilde{\boldsymbol{a}}^\top L_{\boldsymbol{X}}^{-1}) \otimes [\Sigma_0(x_0)(I_d + c_\star\Sigma_0(x_0))^{-1}]\,, \tag{14}$$

for $c_\star = \langle \tilde{\boldsymbol{a}}, L_{\boldsymbol{X}}^{-1}\tilde{\boldsymbol{a}} \rangle$. In particular, if $\Sigma_0(x_0)$ is invertible, then

$$\Theta^{-1} = L_{\boldsymbol{X}}^{-1} \otimes I_d - (L_{\boldsymbol{X}}^{-1}\tilde{\boldsymbol{a}}\tilde{\boldsymbol{a}}^\top L_{\boldsymbol{X}}^{-1}) \otimes (\Sigma_0(x_0)^{-1} + c_\star I_d)^{-1}\,.$$

Proof. We first prove that $L_{\boldsymbol{X}}$ is positive definite. As before, work over $\mathbb{R}$ since the d-dimensional covariance tensorizes with I_d.

From the representation in (12), the centered positional noise at time t_i is

$$\eta_i^X = \sqrt{\varepsilon} \int_0^{t_i} a_{t_i-u} \, \mathrm{d}B_u \,,$$

where $a_r = \gamma^{-1}(1 - \mathrm{e}^{-\gamma r})$ for $\gamma > 0$, with the convention $a_r = r$ when $\gamma = 0$. Hence, for any $c = (c_1, \ldots, c_J) \in \mathbb{R}^J$,

$$\langle c, L_{\boldsymbol{X}} c \rangle = \varepsilon \int_0^1 \left(\sum_{i=1}^J c_i a_{t_i-u} \mathbf{1}_{\{u \le t_i\}} \right)^2 \mathrm{d}u \ge 0 \,.$$

We proceed by contradiction: if $\langle c, L_{\boldsymbol{X}} c \rangle = 0$ for $c \neq 0$, then the integrand would be zero for almost every u. Let m be the largest index such that $c_m \neq 0$; on the interval (t_{m-1}, t_m), all terms with $i < m$ vanish because $u > t_i$, and all terms with $i > m$ have zero coefficient by the definition of m. Thus the integrand reduces to $(c_m a_{t_m-u})^2$, which is strictly positive. This yields a contradiction, so $L_{\boldsymbol{X}}$ is positive definite and thus invertible.

The block form of $(\Sigma_{\boldsymbol{X}})^\dagger$ follows because $t_0 = 0$ makes the first row and column of $\Sigma_{\boldsymbol{X}}$ vanish, while the remaining block is Θ. Since $L_{\boldsymbol{X}}$ is positive definite and $\Sigma_0(x_0)$ is positive semidefinite, Θ is invertible. The displayed expression for Θ^{-1} is then the Sherman–Morrison–Woodbury formula applied to $\Theta = L_{\boldsymbol{X}} \otimes I_d + \tilde{\boldsymbol{a}}\tilde{\boldsymbol{a}}^\top \otimes \Sigma_0(x_0)$. □

A.2 Proof of Lemma 2

Lemma A.4. *For $\gamma \ge 0, \varepsilon > 0$, consider kinetic underdamped Brownian motion $\mathsf{R}^{\gamma,\varepsilon}$, and for $t_j, t_{j+1} \in [0,1]$ with $t_j < t_{j+1}$, fix endpoints (x_j, v_j) and (x_{j+1}, v_{j+1}) respectively. Then the bridge of the reference process conditioned on $(X_{t_j}, V_{t_j}) = (x_j, v_j)$ and $(X_{t_{j+1}}, V_{t_{j+1}}) = (x_{j+1}, v_{j+1})$ is the unique solution to*

$$\begin{cases} \mathrm{d}X_t = V_t \, \mathrm{d}t \\ \mathrm{d}V_t = (\varepsilon \nabla_v \log p_{t_{j+1}-t}(X_t, V_t, x_{j+1}, v_{j+1}) - \gamma V_t) \, \mathrm{d}t + \sqrt{\varepsilon} \, \mathrm{d}B_t \,, \end{cases} \tag{15}$$

on $t \in [t_j, t_{j+1})$, where $p_{t_{j+1}-t}(x, v; x_{j+1}, v_{j+1})$ is the transition density of the reference process.

Proof. Since both the R-bridge and the solution of (15) are Markov processes, it suffices to show that their infinitesimal generators coincide. If we denote by $p_t((x,v),(x',v'))$ the transition density of the reference process R, then the marginal flow of the bridge at time $t \in [t_j, t_{j+1})$ is

$$\mu_t^{\mathsf{br}}(x,v) \propto p_{t-t_j}((x_j,v_j),(x,v)) p_{t_{j+1}-t}((x,v),(x_{j+1},v_{j+1})) \,.$$

Our goal is to understand the evolution of μ_t^{br} in the weak sense. To this end, consider now a smooth test function $\phi : [t_j, t_{j+1}] \times \mathbb{R}^{2d} \to \mathbb{R}$ with compact support; we compute

$$\partial_t \int \phi_t \mu_t^{\mathsf{br}} = \int (\partial_t \phi_t) \mu_t^{\mathsf{br}} + \phi_t (\partial_t p_{t-t_j}) p_{t_{j+1}-t} + \phi_t p_{t-t_j} (\partial_t p_{t_{j+1}-t}) \,.$$

To continue, we require the infinitesimal generator of the kinetic (underdamped) reference process, which we recall is given by

$$\mathcal{L}_\gamma f(x,v) = \langle v, \nabla_x f(x,v) \rangle - \gamma \langle v, \nabla_v f(x,v) \rangle + \frac{\varepsilon}{2} \Delta_v f(x,v) \,,$$

for a test function f. Now, we use the appropriate forward and backward Kolmogorov equations to see that

$$\partial_t p_{t-t_j} = \mathcal{L}_\gamma^* p_{t-t_j} \,, \quad \text{and} \quad \partial_t p_{t_{j+1}-t} = -\mathcal{L}_\gamma p_{t_{j+1}-t} \,,$$

resulting in

$$\begin{aligned} \partial_t \int \phi_t \mu_t^{\mathsf{br}} &= \int (\partial_t \phi_t) \mu_t^{\mathsf{br}} + \phi_t (\mathcal{L}_\gamma^* p_{t-t_j}) p_{t_{j+1}-t} - \phi_t p_{t-t_j} (\mathcal{L}_\gamma p_{t_{j+1}-t}) \\ &= \int (\partial_t \phi_t) \mu_t^{\mathsf{br}} + \mathcal{L}_\gamma (\phi_t p_{t_{j+1}-t}) p_{t-t_j} - \phi_t p_{t-t_j} (\mathcal{L}_\gamma p_{t_{j+1}-t}) \,, \end{aligned} \tag{16}$$

where the last line uses integration by parts with the adjoint $\mathcal{L}_\gamma$. Next, we use the commutation identity for two test functions (Bakry et al., 2014)

$$\mathcal{L}_\gamma(gh) = (\mathcal{L}_\gamma g)h + g(\mathcal{L}_\gamma h) + \varepsilon\langle\nabla_v g, \nabla_v h\rangle\,,$$

to rewrite (16) as

$$\begin{aligned}\partial_t \int \phi_t \mu_t^{\mathsf{br}} &= \int \Big((\partial_t\phi_t)\mu_t^{\mathsf{br}} + (\mathcal{L}_\gamma\phi_t)p_{t_{j+1}-t}p_{t-t_j} + \varepsilon\,\langle\nabla_v\phi_t, \nabla_v p_{t_{j+1}-t}\rangle p_{t-t_j}\Big)\\ &= \int ((\partial_t + \mathcal{L}_\gamma)\phi_t + \varepsilon\,\langle\nabla_v \log p_{t_{j+1}-t}, \nabla_v\phi_t\rangle)\mu_t^{\mathsf{br}}\,,\end{aligned} \tag{17}$$

which implies that μ_t^{br} is a weak solution to the Fokker–Planck equation associated with the generator

$$\mathcal{L}_{\gamma,t}^{\mathsf{br}} := \mathcal{L}_\gamma + \varepsilon\langle\nabla_v \log p_{t_{j+1}-t}, \nabla_v\cdot\rangle\,.$$

□

Lemma A.5. *Let $\mathsf{R}^{\gamma,\varepsilon}$ be the kinetic underdamped Brownian motion reference process for $\gamma > 0$ and $\varepsilon > 0$, and fix $T \in (0,1]$ and $z_T = (x_T, v_T) \in \mathbb{R}^{2d}$. Letting $p_{T-t}((x,v); z_T)$ denote the transition density from time $t \in [0,T)$ to T and $r := T - t$, it holds that*

$$\varepsilon\nabla_v \log p_{T-t}((x,v); z_T) = C_x(r;\gamma)(x_T - x - a_r v) + C_v(r;\gamma)(v_T - b_r v)\,, \tag{18}$$

where

$$\begin{aligned}C_x(r;\gamma) &= \gamma^2(1 - \mathrm{e}^{-\gamma r})/D_r(\gamma)\,,\\ C_v(r;\gamma) &= \gamma(\mathrm{e}^{-2\gamma r} + 2\gamma r\mathrm{e}^{-\gamma r} - 1)/((1 - \mathrm{e}^{-\gamma r})D_r(\gamma))\,,\\ D_r(\gamma) &= \gamma r(1 + \mathrm{e}^{-\gamma r}) + 2\mathrm{e}^{-\gamma r} - 2\,.\end{aligned}$$

In particular, $\gamma \to 0$ recovers (6).

Proof. Writing $r = T - t$, it holds by Lemma A.1 that, conditioning on $Z_t = (x,v)$

$$(X_T, V_T)|(X_t, V_t) = (x,v) \sim \mathbf{N}\Big(\begin{pmatrix} x + a_r v \\ b_r v\end{pmatrix}, Q_r\Big)\,,$$

where $Q_r = \Gamma_Z(r,r)$ (and recall the notation $a_r = \gamma^{-1}(1 - \mathrm{e}^{-\gamma r})$ and $b_r = \mathrm{e}^{-\gamma r}$ from Lemma A.1). Thus up to a constant, we have that

$$\begin{aligned}&\log p_{T-t}((x,v),(x_T,v_T)) =\\ &\quad -\frac{1}{2}\begin{pmatrix} x_T - (x + a_r v)\\ v_T - b_r v\end{pmatrix}^\top Q_r^{-1}\begin{pmatrix} x_T - (x + a_r v)\\ v_T - b_r v\end{pmatrix} + \text{const}\,.\end{aligned}$$

As the matrix Q_r^{-1} is explicit (via block 2-by-2 inversion), one can verify through a tedious calculation that indeed (18) holds, and setting $\gamma = 0$ recovers (6). □

A.3 Proof of Theorem 1

Our proof follows the standard argument that passes through weak solutions of the kinetic underdamped Fokker–Planck equation. Throughout this proof, $\gamma = 0$, as in Theorem 1. Letting $f : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a smooth test function, our goal is to compute

$$\int f\,\partial_t\mu_t^I = \partial_t\Big(\int f\,\mathrm{d}\mu_t^I\Big) = \partial_t\mathbb{E}[f(X_t^I, V_t^I)]\,,$$

where μ_t^I is the density of P_t^I. To proceed, we apply Itô's lemma to first arrive at

$$\begin{aligned}\mathrm{d}f &= \langle\nabla_x f,\,\mathrm{d}X_t^I\rangle + \langle\nabla_v f,\,\mathrm{d}V_t^I\rangle + \frac{1}{2}\langle\mathrm{d}V_t^I, \nabla_v^2 f\,\mathrm{d}V_t^I\rangle\\ &= \big(\langle\nabla_x f, V_t^I\rangle + \langle\nabla_v f, a_t^j\rangle - \gamma\langle\nabla_v f, V_t^I\rangle + \frac{\varepsilon}{2}\Delta_v f\big)\,\mathrm{d}t + \sqrt{\varepsilon}\langle\nabla_v f,\,\mathrm{d}B_t\rangle\,.\end{aligned}$$

Taking expectations, as the last term has mean zero by the martingale property, yields

$$\partial_t \mathbb{E}[f(X_t^I, V_t^I)] = \mathbb{E}\Big[\langle \nabla_x f(X_t^I, V_t^I), V_t^I\rangle + \langle \nabla_v f(X_t^I, V_t^I), a_t^j(X_t^I, V_t^I)\rangle - \gamma\langle \nabla_v f(X_t^I, V_t^I), V_t^I\rangle + \frac{\varepsilon}{2}\Delta_v f(X_t^I, V_t^I)\Big],$$

where recall that $a_t^j(X_t^I, V_t^I) = a_t^j(X_t^I, V_t^I; X_{t_{j+1}}^I, V_{t_{j+1}}^I)$. By the tower property of conditional expectations, it holds that

$$\begin{aligned}
&\mathbb{E}[\langle a_t^j(X_t^I, V_t^I; X_{t_{j+1}}^I, V_{t_{j+1}}^I), \nabla_v f(X_t^I, V_t^I)\rangle] \\
&\quad = \mathbb{E}[\mathbb{E}[\langle a_t^j(X_t^I, V_t^I; X_{t_{j+1}}^I, V_{t_{j+1}}^I), \nabla_v f(X_t^I, V_t^I)\rangle \,|\, (X_t^I, V_t^I)]] \\
&\quad = \mathbb{E}[\langle \mathbb{E}[a_t^j(X_t^I, V_t^I; X_{t_{j+1}}^I, V_{t_{j+1}}^I) \,|\, (X_t^I, V_t^I)], \nabla_v f(X_t^I, V_t^I)\rangle] \\
&\quad = \mathbb{E}[\langle a_t^M(X_t^I, V_t^I), \nabla_v f(X_t^I, V_t^I)\rangle],
\end{aligned}$$

where we recall that for an interval $[t_j, t_{j+1})$

$$(t, x, v) \mapsto a_t^M(x, v) \coloneqq \mathbb{E}[a_t^j(X_t^I, V_t^I; X_{t_{j+1}}^I, V_{t_{j+1}}^I) \,|\, (X_t^I, V_t^I) = (x, v)].$$

Therefore, we see that the weak solution to μ_t^I satisfies

$$\partial_t \mathbb{E}[f(X_t^I, V_t^I)] = \mathbb{E}\Big[\langle \nabla_x f(X_t^I, V_t^I), V_t^I\rangle + \langle \nabla_v f(X_t^I, V_t^I), a_t^M(X_t^I, V_t^I)\rangle - \gamma\langle \nabla_v f(X_t^I, V_t^I), V_t^I\rangle + \frac{\varepsilon}{2}\Delta_v f(X_t^I, V_t^I)\Big].$$

Since $\mu_0^M = \mu_0^I$ and, by assumption, the acceleration field implies a unique solution to the kinetic Fokker–Planck equation, we conclude the proof.

B Algorithmic subroutine details

B.1 Learning an initial velocity sampler

Theorem 1 requires that $(X_0^M, V_0^M) \sim \mathsf{P}_0^I$, meaning that the initial position and velocity need to be appropriately sampled from the law of the interpolant at time $t = 0$. As mentioned in Section 3.3, our goal is to learn a conditional sampler for $\mathrm{Law}(V_0 \mid X_0)$. To see how this is possible, recall that we can efficiently sample interpolants

$$v_{0:J} \mid X_{0:J} = x_{0:J},$$

from the Gaussian conditioning formula (11). We then extract the initial position and velocity which correspond to index $j = 0$; repeating this procedure yields a paired dataset $\{(x_0^{(i)}, v_0^{(i)})\}_{i=1}^n$. Then, using this paired data, we can hope to learn a conditional sampler $q_\phi(\cdot \mid x_0)$ that samples from $V_0 \mid X_0$.

Our main experiments use a conditional normalizing flow (NF), trained by conditional maximum likelihood. Table 13 gives its hyperparameters; Appendix D.1 states the NF and conditional flow-matching objectives and compares their results.

Figures 4 and 5 provide a illustration of the learned conditional sampler for our two low-dimensional datasets. We notice that the learned NF qualitatively matches the true conditional bridge reference process in both cases.

B.2 Bridge sampling

For a given $j \in \{0, 1, \ldots, J-1\}$, define $\mathcal{I}_j = [t_j, t_{j+1}]$ and $\delta_j = t_{j+1} - t_j$. We want to sample (a single) intermediate point (position and velocity) along the trajectory given two endpoint constraints

$$(X_t, V_t) | (X_{t_{j+1}}, V_{t_{j+1}}) = (x_{j+1}, v_{j+1}), (X_{t_j}, V_{t_j}) = (x_j, v_j),$$

where $t \in [t_j + c, t_{j+1} - c]$ for $0 < c \ll \delta_j/2$. Again, we can appeal to explicit formulae that arise in the study of Gaussian processes. For simplicity, we restrict our attention to the case $\gamma = 0$.

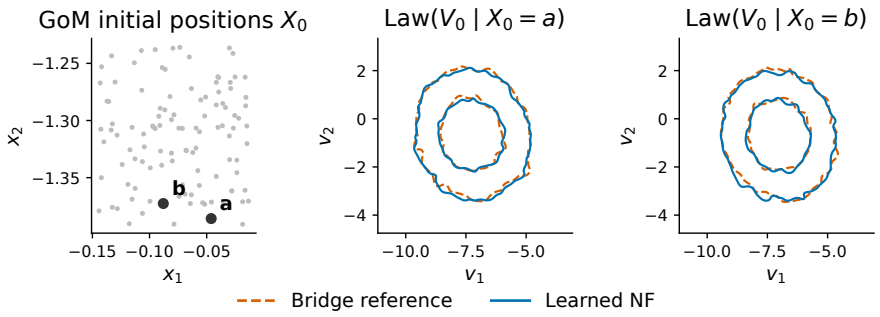


Figure 4: GoM conditional initial velocities at two random source points a and b: learned NF (solid blue) and bridge reference (dashed orange), with 50% and 90% density contours.

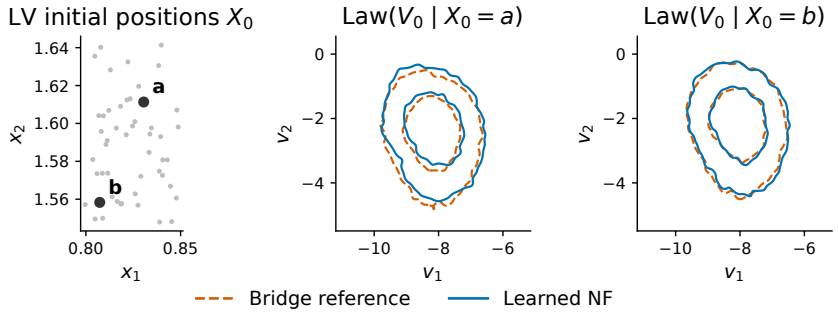


Figure 5: LV conditional initial velocity laws at two random source points a and b.

Define

$$Y_t = \begin{pmatrix} X_{t_j+t} \\ V_{t_j+t} \end{pmatrix},$$

for $t \in [0, \delta_j]$, where the dynamics will again follow the reference process R. Thus

$$Y_t = m_t + \sqrt{\varepsilon}\,\xi_t = \begin{pmatrix} x_j + tv_j \\ v_j \end{pmatrix} + \sqrt{\varepsilon} \begin{pmatrix} \int_0^t B_u \,\mathrm{d}u \\ B_t \end{pmatrix}.$$

Here, the initialization pins down the left endpoint constraints and it remains to enforce the right endpoint. Since $(Y_t; Y_{\delta_j})$ is jointly Gaussian, we can readily compute $\mathrm{Cov}(\xi_t)$, $\mathrm{Cov}(\xi_{\delta_j})$ and $\mathrm{Cov}(\xi_t, \xi_{\delta_j})$ explicitly. By Gaussian conditioning, we have that $Y_t \,|\, Y_{\delta_j} \sim \mathbf{N}(m_{t|j}, C_{t|j})$ with

$$\begin{aligned} m_{t|j} &= m_t + \mathrm{Cov}(\xi_t, \xi_{\delta_j})\mathrm{Cov}(\xi_{\delta_j})^{-1}(Y_{\delta_j} - m_{\delta_j}), \\ C_{t|j} &= \varepsilon\mathrm{Cov}(\xi_t) - \varepsilon\mathrm{Cov}(\xi_t, \xi_{\delta_j})\mathrm{Cov}(\xi_{\delta_j})^{-1}\mathrm{Cov}(\xi_t, \xi_{\delta_j})^\top. \end{aligned}$$

To conclude, we simply substitute $Y_{\delta_j} = (x_{j+1}, v_{j+1})$ in the above expression, and draw a sample. Note that this procedure can be efficiently parallelized.

C Remaining Experimental Details and Results

C.1 Gulf of Mexico

The Gulf of Mexico (GoM) dataset describes the movement of water particles in ocean currents. In our experiments, we work with preprocessed data from Shen et al. (2024). They used data from a HYbrid Coordinate Ocean Model (HYCOM) reanalysis (HYCOM Consortium and Center for Ocean-Atmospheric Prediction Studies (COAPS), 2024) at a specific time snapshot to compute the

velocity field of the ocean currents. Then they generated 999 trajectories in $\mathbb{R}^2$ and randomly sampled each of them at one of the nine time points. With this procedure, each particle is observed only at a single time point and each generated marginal contains $n = 111$ samples. The dataset is divided into 5 training marginals (the even times t_0, t_2, t_4, t_6, t_8) with the remaining 4 times (t_1, t_3, t_5, t_7) used for held-out evaluation. The data are centered and scaled to have zero mean and unit variance.

In Tables 3 and 4, we present the 2-Wasserstein distance on training and held-out marginals, to have a more precise indicator of the performance of the algorithms relative to Table 1. As for the other datasets, precise training information can be found in Table 10. Figures 6 and 2 display 20 generated trajectories.

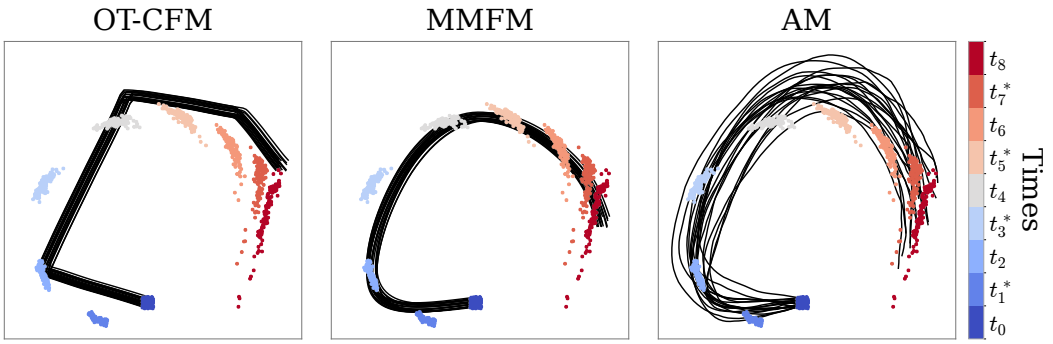


Figure 6: Generated trajectories for the GoM dataset. Stars mark held-out times.

Table 3: Per-timepoint W_2 at training marginals on the Gulf of Mexico dataset.

Method	t_2	t_4	t_6	t_8
OT-CFM	0.0585 ± 0.0085	0.1228 ± 0.0659	0.1869 ± 0.0811	0.3150 ± 0.0958
MMFM (prior)	0.050 ± 0.008	0.089 ± 0.017	0.140 ± 0.028	0.240 ± 0.042
MMFM	0.0713 ± 0.0186	0.1269 ± 0.0404	0.1669 ± 0.0265	0.2615 ± 0.0287
AM	0.0878 ± 0.0068	0.1075 ± 0.0376	0.1503 ± 0.0751	0.1716 ± 0.0691

Table 4: Per-timepoint W_2 at held-out marginals on the Gulf of Mexico dataset.

Method	t_1	t_3	t_5	t_7
3MSBM*	0.20	0.18	0.07	0.09
OT-CFM	0.2686 ± 0.0112	0.3453 ± 0.0246	0.1970 ± 0.0699	0.2546 ± 0.0774
MMFM (prior)	0.332 ± 0.010	0.187 ± 0.037	0.132 ± 0.030	0.187 ± 0.051
MMFM	0.1978 ± 0.0206	0.1920 ± 0.0273	0.1648 ± 0.0501	0.2096 ± 0.0302
AM	0.1805 ± 0.0072	0.1975 ± 0.0114	0.1805 ± 0.0663	0.1323 ± 0.0312

C.2 Lotka–Volterra

This dataset, which simulates the Lotka–Volterra equations of a predator-prey model (Goel et al., 1971), is obtained from Shen et al. (2024). The data live in $\mathbb{R}^2$ and consist of 9 marginals with 50 points each. As with the GoM dataset, the even times are used for training $(t_0, t_2, t_4, t_6, t_8)$ and the remaining 4 for evaluation (t_1, t_3, t_5, t_7). Again, the data are centered and normalized. Tables 5 and 6 respectively contain the training and evaluation errors in W_2 across the times of interest.

C.3 Embryoid body

We use the preprocessed data from Moon et al. (2019), which are formed from the 100-dimensional PCA representation of the scRNA-seq of human embryoid cells collected across multiple time points. The dataset contains five marginals with 2,381, 4,163, 3,278, 3,665, and 3,332 cells (i.e., unpaired data) per marginal, respectively.

Table 5: Per-timepoint W_2 at training marginals on the Lotka–Volterra dataset.

Method	t_2	t_4	t_6	t_8
OT-CFM	0.2090 ± 0.0716	0.2643 ± 0.0716	0.4582 ± 0.0633	1.1834 ± 0.1691
MMFM (prior)	0.179 ± 0.031	0.223 ± 0.039	0.448 ± 0.051	1.150 ± 0.216
MMFM	0.1690 ± 0.0245	0.2082 ± 0.0176	0.4634 ± 0.0680	1.2718 ± 0.2916
AM	0.1562 ± 0.0362	0.2423 ± 0.0630	0.3253 ± 0.1235	0.6135 ± 0.3778

Table 6: Per-timepoint W_2 at held-out marginals on the Lotka–Volterra dataset.

Method	t_1	t_3	t_5	t_7
3MSBM*	0.24	0.17	0.15	0.36
OT-CFM	0.8376 ± 0.0648	0.3537 ± 0.1281	0.4184 ± 0.0882	0.9332 ± 0.0623
MMFM (prior)	0.206 ± 0.039	0.211 ± 0.060	0.351 ± 0.017	0.941 ± 0.109
MMFM	0.5342 ± 0.0294	0.2369 ± 0.0339	0.3640 ± 0.0272	0.9823 ± 0.1324
AM	0.5056 ± 0.0138	0.2470 ± 0.0940	0.2275 ± 0.0719	0.5165 ± 0.1753

For our experiments, we used either the first 5 principal components (resulting in EB5) or all 100 supplied PCs (resulting in EB100). In both cases, we used a coordinatewise standardization to center and scale the training marginals. For EB5, we perform a leave-one-out (LOO) procedure by training on four out of five marginals, following the protocol from Tong et al. (2020). The evaluation is then performed at the held-out time (one of t_1, t_2, t_3) using the 1-Wasserstein distance on normalized data. In contrast, for our experiments on EB100, we train on t_0, t_2, t_4 and hold out marginals at times t_1, t_3 together, following Theodoropoulos et al. (2025), and we report the errors after inverting the normalization. During training, we consider the full marginals and sample minibatches from them. At inference, we consider 2000 points from the first marginal as starting points for the trajectories. Table 7 reports the W_1 distances on the held-out marginals under the leave-one-out protocol.

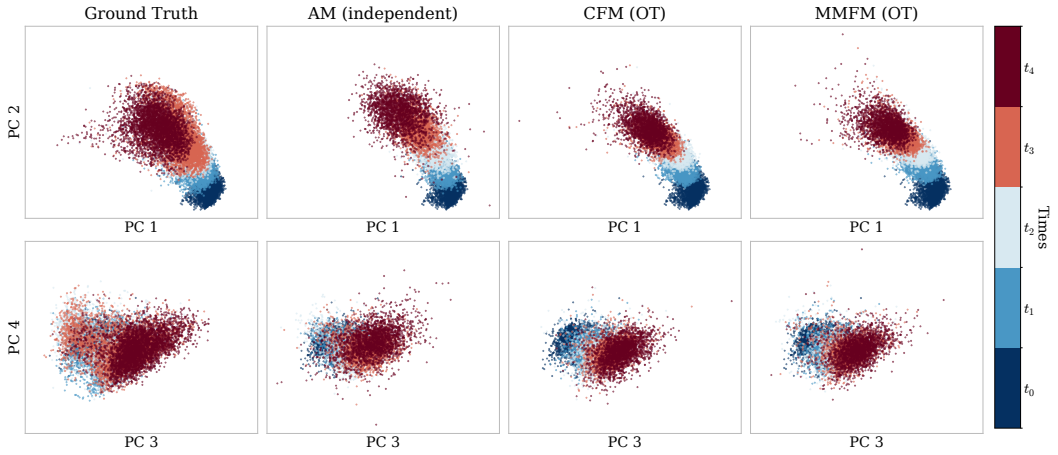


Figure 7: Observed and generated marginals for EB100.

Table 7: Mean and standard deviation of the W_1 error at held-out marginals on the EB-5 dataset.

Method	Coupling	t_1 mean	t_2 mean	t_3 mean	Average $\pm$ seed SD
AM	Independent	0.7614	0.8018	0.8916	0.8183 ± 0.0548
CFM	OT	0.7384	0.8873	0.8844	0.8367 ± 0.0087
MMFM	OT	0.6901	0.8158	0.8156	0.7738 ± 0.0209
AM	OT	0.6865	0.6831	0.8528	0.7408 ± 0.0151

C.4 CITE

This dataset (Burkhardt et al., 2022) consists of single-cell CITE-seq measurements collected at four time points over ten days. We considered both the 5-dimensional (CITE5) and 50-dimensional (CITE50) PCA projections, preprocessed following prior work by Wang et al. (2026). The training data are normalized to zero mean and divided by the maximum per-feature standard deviation, following Neklyudov et al. (2024). We evaluate all methods (considering the normalized space for CITE5 and the original PCA space for CITE50) under the leave-one-out (LOO) protocol as presented in Kapusniak et al. (2024), reporting W_1 on held-out marginals in Tables 8 and 9.

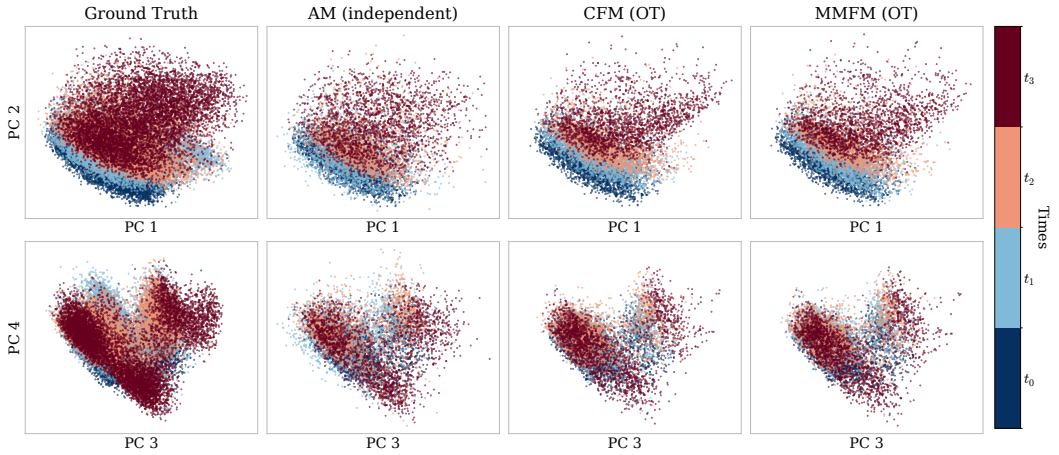


Figure 8: Observed and generated marginals for CITE5, with t_1 held out.

Table 8: Per-timepoint W_1 at held-out marginals on the CITE5 dataset (LOO protocol).

Method	Coupling	t_1 mean	t_2 mean	Average $\pm$ seed SD
AM	Independent	0.5890	0.5614	0.5752 ± 0.0270
CFM	OT	0.5201	0.4623	0.4912 ± 0.0036
MMFM	OT	0.5486	0.5157	0.5321 ± 0.0092
AM	OT	0.5486	0.5348	0.5417 ± 0.0089

Table 9: Per-timepoint W_1 at held-out marginals on the CITE50 dataset (LOO protocol).

Method	Coupling	t_1 mean	t_2 mean	Average $\pm$ seed SD
AM	Independent	45.8693	46.1488	46.0090 ± 0.5974
CFM	OT	39.5653	38.7717	39.1685 ± 0.2996
MMFM	OT	41.1599	41.3959	41.2779 ± 0.4041
AM	OT	41.8725	43.1715	42.5220 ± 0.1630

D TRAINING DETAILS AND HYPERPARAMETERS

Table 10 describes the parameters used to learn the time-dependent acceleration or velocity fields. All models use Adam (Kingma & Ba, 2015) with $(\beta_1, \beta_2) = (0.9, 0.999)$. Errors are averaged over held-out times or folds within each seed, then summarized by the mean and population standard deviation across seeds. The low-dimensional experiments were run on an M2 MacBook Air; EB5, EB100, CITE5, and CITE50 experiments were run on a single NVIDIA H200 GPU.

Table 10: Hyperparameters for training acceleration and velocities.

	GoM	**LV**	**EB5**	**EB100**	**CITE5**	**CITE50**
Architecture	MLP	MLP	MLP	MLP	MLP	MLP
Hidden width	256	256	256	256	256	256
Hidden layers	2	2	5	5	5	4
Batch size	111	50	256	256	256	256
Learning rate	10^{-2}	10^{-2}	10^{-2}	10^{-3}	10^{-3}	10^{-3}
Objective function updates	300	300	5 000	5 000	5 000	5 000
Sampling steps	100	100	2 000	2 000	2 000	2 000

D.1 Conditional velocity samplers

From the discussion in Section 3.3, recall that we are given data of the form $\{(x_0^{(i)}, v_0^{(i)})\}_{i=1}^n$ and we want to generate samples from v given x. We denote an arbitrary conditional sampler as $q_\phi(\cdot \mid x_0)$.

Our first approach is based on conditional normalizing flows (NFs) with a standard Gaussian base and alternating coupling masks (Dinh et al., 2017; Winkler et al., 2019). Such a flow is trained by minimizing the negative log-likelihood

$$\mathcal{L}_{\mathrm{NF}}(\phi) \coloneqq -\mathbb{E}_{(X_0,V_0)}[\log q_\phi(V_0 \mid X_0)] \ . \tag{19}$$

Our second approach uses conditional FM (Lipman et al., 2023). Here, the sampler $q_\phi(\cdot \mid x_0)$ is obtained by numerically integrating (via midpoint discretization) a velocity field u_ϕ, which takes in the concatenated input (v_s, X_0, s). To train the sampler, we set $v_s = (1-s)Z + sV_0$ for $Z \sim \mathcal{N}(0, I)$ and $s \sim \mathrm{Unif}(0,1)$, and minimize

$$\mathcal{L}_{\mathrm{FM}}(\phi) = \frac{1}{d}\mathbb{E}\Big[\|u_\phi(v_s, X_0, s) - (V_0 - Z)\|^2\Big] \ , \tag{20}$$

where (X_0, V_0) are drawn from the training data. Table 13 gives the parameters for learning the relevant networks.

We perform a brief ablation study, comparing the two samplers. Tables 12 and 11 favor the conditional NF configurations overall, which is the one we ultimately decided to use.

Table 11: AM initial-velocity sampler ablation: held-out W_2 (↓)

Coupling	Sampler	GoM	LV
Independent	NF	0.1727 ± 0.0217	0.3742 ± 0.0584
Independent	FM	0.1705 ± 0.0162	0.3599 ± 0.0249
OT	NF	0.1767 ± 0.0101	0.3838 ± 0.0549
OT	FM	0.1536 ± 0.0048	0.4366 ± 0.0792

Table 12: AM initial-velocity sampler ablation: held-out W_1 (↓)

Coupling	Sampler	EB5	EB100
Independent	NF	0.8183 ± 0.0548	12.8698 ± 0.0903
Independent	FM	0.8377 ± 0.0521	13.1508 ± 0.1273
OT	NF	0.7408 ± 0.0151	13.1696 ± 0.1187
OT	FM	0.7464 ± 0.0128	13.3236 ± 0.0737

Table 13: Hyperparameters for training conditional velocity samplers.

	GoM	LV	EB5	EB100	CITE5	CITE50
Reference process						
Reference σ_v^2	50	50	0.005	0.005	0.01	1
Noise amplitude $\sqrt{\varepsilon}$	4	2	0.2	0.2	0.01	0.01
Conditional normalizing flow (NF)						
Coupling layers	6	6	6	6	6	6
Conditioner hidden width	256	256	256	256	256	256
Conditioner hidden layers	2	2	2	2	2	2
Activation	ReLU	ReLU	ReLU	ReLU	ReLU	ReLU
Absolute log-scale bound	2	2	2	2	2	2
Batch size	111	50	256	256	256	256
Learning rate	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}	10^{-3}
Training updates	20 000	20 000	20 000	20 000	20 000	20 000
Sampling	Direct	Direct	Direct	Direct	Direct	Direct
Conditional flow matching (FM)						
Hidden width	256	256	256	256	—	—
Hidden layers	2	2	4	4	—	—
Activation	SiLU	SiLU	SiLU	SiLU	—	—
Batch size	256	256	256	256	—	—
Learning rate	10^{-3}	10^{-3}	10^{-3}	10^{-3}	—	—
Training updates	20 000	20 000	20 000	20 000	—	—
ODE solver	Midpoint	Midpoint	Midpoint	Midpoint	—	—
Sampling steps	32	32	32	32	—	—
Network evaluations	64	64	64	64	—	—