Flow Matching on General Manifolds via Pulling Back Geodesic Convex Latent Manifolds

Neil He
neilhe2@illinois.edu
University of Illinois Urbana-Champaign

Ge Liu
geliu@illinois.edu
University of Illinois Urbana-Champaign

Abstract

Generative models that construct conditional paths transforming a source distribution into a target distribution have seen substantial success in modeling scientific data. However, traditional methods typically assume Euclidean geometry and fail to adhere to data manifolds, falling short on tasks such as trajectory inference. Recent methods on general manifolds often rely on approximating geodesics without closed forms, suffer from limited theoretical and implementation grounding, or trade generation quality to learn velocity fields. As a result, we propose GCL-FM, a novel simulation-free framework that enables interpolation on unknown data manifolds by explicitly enforcing a **G**eodesic **C**onvex **L**atent submanifold for **F**low **M**atching. Our theoretical analysis shows that GCL-FM possesses desirable guarantees for modeling manifold transport dynamics, including robustness to data noise. We demonstrate that GCL-FM achieves superior performance on trajectory inference for multiple biological and computational fluid dynamics datasets, while requiring significantly less computational cost compared to prior methods, providing a principled and efficient route to high-fidelity manifold-aware conditional generation for real-world transport dynamics in high dimensions.

1. Introduction

Rapid growth of high-dimensional scientific data has created unprecedented opportunities for scientific discovery (Villegas-Morcillo et al., 2021; Zhang et al., 2023; Urdapilleta et al., 2015; Shi et al., 2013; Sun et al., 2024; Moon et al., 2019). One particular problem is reconstructing the temporal dynamics of a multi-body system, a problem present in single-cell RNA sequencing (Macosko et al., 2015), geophysics (Evensen, 2003), or neural population dynamics (Pandarinath et al., 2018), where data is typically sparse in these fields due to expensive and difficult data collection. This problem could be formulated as a *trajectory inference problem*, where the goal is, given population-level snapshots of noisy data, to generate the full trajectory dynamics of the evolving system.

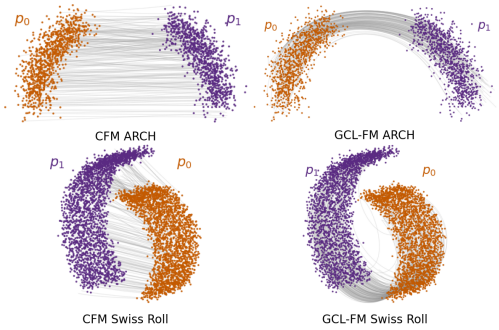

Figure 1. Trajectory inference on the Arch and partial Swiss Roll dataset for CFM and GCL-FM. Grey lines show learned trajectory going from source (orange) to target (purple).

Methods for trajectory inference typically learn objective-matching dynamics with neural networks, including diffusion models (Song et al., 2021b), flow matching (Lipman et al., 2023), and Schr"odinger bridge matching (Shi et al., 2023). Among these, conditional generative models (Tong et al., 2024a) are particularly applicable. The model constructs probabilistic paths via interpolants that connect samples from a source distribution to a target distribution. However, scientific data often lie near a low-dimensional manifold (Fefferman et al., 2016), whereas these methods typically assume Euclidean geometry, causing interpolants to deviate from the data manifold. Recent works adopted these methods to general manifolds, but still suffer from several limitations that hinder their ability to model real-world data. Approaches that approximate the manifold metric (Sun et al., 2025; Kapuśniak et al., 2024) often require additional networks or heuristics to learn geodesics, or incur substantial overhead

Accepted to FoGen 2026: Foundations of Deep Generative Models: Understanding Memorization, Generalization, and Reasoning, an ICML 2026 workshop (non-archival).

from Jacobian computations. Methods that pull back geometry from latent manifolds (de Kruiff et al., 2024) require high-dimensional NeuralODEs and suffer theoretical issues such as instability on positively curved manifolds and the method itself not adhering to required theoretical properties. Methods that approximate velocity fields (Petrović et al., 2025) can degrade generation quality. FlatVI (Palma et al., 2025) also studies trajectory inference, but is tailored to count single-cell data that are discrete and also does not enforce geodesic convex, whereas our focus is transport on unknown continuous manifolds. Figure 1 shows visualization of trajectory inference on toy datasets (Swiss roll and ARCH (Tong et al., 2020)), where the Euclidean methods fail to reconstruct the curved dynamics from the source (orange) data snapshot to the target (purple) snapshot, while our method GCL-FM generates the missing dynamics.

In light of the limitations of prior Euclidean and Riemannian approaches, we propose to solve trajectory inference in a learned latent submanifold that is explicitly enforced to be geodesically convex, and then pull back the latent transport dynamics through a simple diffeomorphic parameterization. This avoids the need to separately learn data-manifold geodesics, reduces the excessive computation incurred by high-dimensional NeuralODEs or Jacobian computations, and enables learning manifold velocity fields while maintaining high generation quality. Compared to prior pullback and manifold flow matching methods, this yields a simple latent geometry together with fully anisotropic pullback metrics, while remaining computationally efficient. Comparisons are given in Table 1. **Our contributions** is summarized as:

(1.) We propose GCL-FM, a novel flow matching framework on unknown manifolds by learning a latent submanifold that is explicitly learned to be geodesically convex through appropriately designed oracles, and pulls back the latent transport dynamics to the data manifold via a simple diffeomorphic parametrization. Compared to prior works, GCL-FM is the only manifold-based method that (i) avoids a separate geodesic learner, (ii) enforces geodesic convexity, (iii) allows fully anisotropic metrics via pullback, and (iv) avoids Jacobian/NeuralODE overhead.

(2.) We provide several novel theoretical results on GCL-FM's optimality: (i) robustness of the pullback optimal transport paths against data noise; (ii) effectiveness of training objective in learning a geodesic convex submanifold with appropriate interpolation indicator oracles; (iii) a novel framework that provably alleviates the instability on manifolds with positive curvature where prior works suffered (de Kruiff et al., 2024; Diepeveen, 2024).

(3.) We validate GCL-FM on multiple datasets across single-cell and computational fluid dynamics, showing superior performance while being significantly faster and more memory efficient than existing methods.

2. Related Works

Geometric Generative Models. Diffusion, flow matching, and Schrödinger bridge models have seen strong empirical success (Croitoru et al., 2023; Song et al., 2021a; Ho et al., 2020; Tong et al., 2024a; Lipman et al., 2023; Shi et al., 2023; Tong et al., 2024b). Many works extend these frameworks from Euclidean space to Riemannian manifolds (Yim et al., 2023; Gemici et al., 2016; Lou et al., 2024; De Bortoli et al., 2022; Huang et al., 2022; Mangoubi et al., 2025; Zhu et al., 2025; Chen & Lipman, 2024), and pullback approaches have also been explored (Collas et al., 2025). Rather than generating on unknown manifolds, these methods typically assume the underlying manifold geometry is known.

Table 1. Comparison of manifold-based methods, with GCL-FM offering comprehensive computational and theoretical advantages. G.C. stands for geodesic convexity.

Property	MFM	GAGA	PFM	GCL-FM (Ours)
No separate geodesic learner	✗	✗	✔	✔
Enforced G.C.	N.A.[1]	✗	✗	✔
Full anisotropic metric	✗	✔	✔	✔
Compute efficiency	✔	✗ (Jacobian)	✗ (NeuralODE)	✔

Trajectory Inference. Reconstructing transport dynamics across time points or states is central in the natural sciences (Macosko et al., 2015; Evensen, 2003; Pandarinath et al., 2018; Jasche & Wandelt, 2013). Recent methods are largely based on conditional flow matching (Tong et al., 2024a; Lipman et al., 2023) or bridge matching (Shi et al., 2023), but usually assume Euclidean geometry. Subsequent manifold-aware approaches (Tong et al., 2024b; Shi et al., 2023; Chen & Lipman, 2024; Neklyudov et al., 2024) either assume the manifold metric is known or rely on surrogate metrics. CurlyFM (Petrović et al., 2025) instead matches instantaneous velocity fields, but can trade off generation quality. On unknown manifolds, MFM (Kapuśniak et al., 2024) uses a diagonal metric and therefore cannot capture full anisotropic structure; GAGA (Sun et al., 2025) uses Jacobians and a separate geodesic learner, which requires significant computational resources; and PFM (de Kruiff et al., 2024), built on (Diepeveen, 2024), requires high-dimensional NeuralODEs, does not enforce the geodesic convexity assumed by the theory, and can be sensitive to noise on positively curved manifolds. FlatVI (Palma et al., 2025) instead targets discrete single-cell count data and also does not enforce geodesic convexity. In contrast, GCL-FM learns a

geodesically convex latent submanifold and performs flow matching there, avoiding separate geodesic learning and the heavy Jacobian or NeuralODE overhead of prior pullback methods.

Geometric Encoding and Interpolation. Prior methods also explored geometric embedding not in trajectory inference setting (Duque et al., 2023; 2020; Fasina et al., 2023; Lee et al., 2022),however, they are not formulated as diffeomorphisms and are not designed for generative trajectory inference under sparse observations. Likewise, interpolation methods on data manifolds (Oring et al., 2021; Mi et al., 2021; Michelis & Becker, 2021) usually rely on straight lines in a Euclidean latent space, which can distort curved manifolds and miss intermediate regions when data are sparse.

3. Preliminaries

3.1. Conditional Flow Matching

Let ρ_0, ρ_1 be distributions on $\mathbb{R}^d$. *Continuous Normalizing Flow* (Chen et al., 2018) learns a map ψ_1 such that $[\psi_1]_\# \rho_0 = \rho_1$, where ψ_t is a time-dependent diffeomorphism $\mathbb{R}^d \to \mathbb{R}^d$ with $\psi_0 = \mathrm{Id}$. We say ψ_t is generated by a vector field u_t if $\frac{d\psi_t}{dt} = u_t(\psi_t)$ and denote $\rho_t = [\psi_t]_\# \rho_0$. We may then learn a neural network $v_{t,\theta}$ that generates ψ_1 by matching its output to the vector field u_t. However, since the density paths ρ_t and the vector field u_t is intractable, *Conditional Flow Matching* (CFM) (Tong et al., 2024a) assumes that the density paths ρ_t is a mixture of conditional path $\rho_t = \int \rho_t(x|z)\nu(z)dz$ where z is sampled from the joint distribution $\nu = (\rho_0, \rho_1)$. The matching objective is $L_{CFM} = \mathbb{E}_{(x_0,x_1)\sim\nu, t\sim U(0,1)} \left\| v_{t,\theta}(x_t) - \frac{dx_t}{dt} \right\|_2^2$ where $x_t = \psi(x_0|x_0, x_1)$ are interpolations from x_0 to x_1.

3.2. Riemannian Geometry and Flow Matching

Riemannian Geometry. A smooth n-dimensional manifold $\mathcal{M}$ is a locally Euclidean topological space. Each point $x \in \mathcal{M}$ has an associated tangent space $T_x\mathcal{M}$, an n-dimensional vector space. A Riemannian metric $\mathfrak{g}$ assigns to each x a smoothly varying inner product $\mathfrak{g}_x : T_x\mathcal{M} \times T_x\mathcal{M} \to \mathbb{R}$, inducing norms, curvature, and distance $d_\mathcal{M}$. A Riemannian manifold is the pair $(\mathcal{M}, \mathfrak{g})$. Geodesics $\gamma_{x,y}(t)$ are locally distance-minimizing smooth curves. Under suitable assumptions one can define the exponential map $\exp_x : T_x\mathcal{M} \to \mathcal{M}$ and its inverse $\log_x : \mathcal{M} \to T_x\mathcal{M}$.

Pullback Geometry. Let $(M, \mathfrak{g}_M)$ and $(N, \mathfrak{g}_N)$ be d-dimensional Riemannian manifolds. A *diffeomorphism* $\varphi : N \to M$ is a smooth bijection with a continuously differentiable inverse. When φ exists, geometric objects on N are induced from M, e.g. $d_N(x, y) = d_M(\varphi(x), \varphi(y))$, $\gamma_{x,y}(t) = \varphi^{-1}(\gamma_{\varphi(x),\varphi(y)}(t))$, and similarly for exp, log, and in particular, optimal transport paths (Lee, 2013). Thus, one may choose a reference manifold M and transfer its geometry to N through φ. In particular, geometric objects on N are obtained by mapping points into M, computing the corresponding distances, geodesics, or transport paths there, and then pulling them back through φ^{-1}.

Prior pullback-based works (Sun et al., 2025; Palma et al., 2025; de Kruiff et al., 2024) assume that the encoder's output is already a manifold, which is not automatically guaranteed. We thus relax global diffeomorphism by requiring only $\varphi : N \to \varphi(N)$ to be a diffeomorphism to pull back the geometry of $(\varphi(N), \mathfrak{g}_M|_{\varphi(N)})$, where $\varphi(N)$ is geodesically convex: for any $x, y \in S$, the $\mathfrak{g}_M$-geodesic satisfies $\gamma_{x,y}(t) \in \varphi(N)$ for all t. Wehn M is a symmetric manifolds and φ is a distant isometry, deviation of pullback geodesics under data noise can be bounded (Diepeveen, 2024):

$$\|\gamma_{z,y}(t) - \gamma_{x,y}(t)\|_2^2 \le \beta_{1-t}(\gamma)\|z - x\|_2 + o(\|z - x\|_2); \qquad \|\gamma_{x,z}(t) - \gamma_{x,y}(t)\|_2^2 \le \beta_t(\gamma)\|z - y\|_2 + o(\|z - y\|_2)$$

for $x, y, z \in N$, where β_t is a function that depends on position and the curvatures of M. For non-positive curvature, β is linear or sub-linear in time and the pullback is robust; for positive curvature, β can induce **exponential sensitivity to noise, an inherent instability in prior work** (Diepeveen, 2024; de Kruiff et al., 2024).

Riemannian Flow Matching. Riemannian Flow Matching (RFM) (Chen & Lipman, 2024) generalizes CFM to Riemannian manifolds, where the interpolation x_t from x_0 to x_1 is given as $x_t = \gamma_{x_0,x_1}(\kappa(t)) = \exp_{x_1}(\kappa(t)\log_{x_1}(x_0))$, where $\kappa(t)$ is a time-scheduler. The RFM loss is $L_{RFM} = \mathbb{E}_{(x_0,x_1)\sim\nu, t\sim U(0,1)} \left[\left\| v_{t,\theta}(x_t) - \frac{dx_t}{dt} \right\|_\mathfrak{g}^2 \right]$.

4. Method

To obtain robust *geodesic* pullbacks, we aim to learn a map $\varphi : D \to M$ that (1) is a ***diffeomorphism***, (2) has ***globally bounded Lipschitz constants***, and (3) has a ***geodesically convex image*** (Diepeveen, 2024). With these requirements in

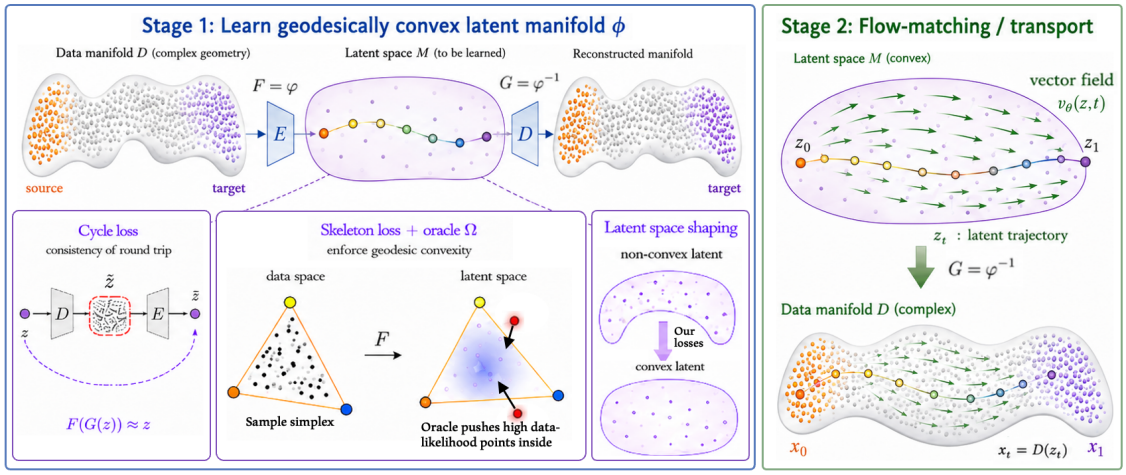

Figure 2. Illustration of GCL-FM. In stage 1, we learn a diffeomorphism $\varphi : D \to \varphi(D) \subset M$. An interpolation indicator oracle Ω enforces geodesic convexity in the learned latent submanifold by correcting the model whenever $\varphi(D)$ is not geodesic convex. In the second stage, we learn the transport dynamics by approximating the vector field with $v_{\theta,t}$. During inference, data are first encoded with φ, the model then computes latent dynamics with $v_{\theta,t}$ and pulls back with φ^{-1}. GCL-FM enables direct geodesics, shown as yellow lines, and transport dynamics, shown as green arrows, computation on D learning only φ.

mind, the high-level idea of GCL-FM, shown in Figure 2, is a two-stage process on unknown data manifolds. In stage one, we learn the latent geometry with an autoencoder pair (F, G). We show later that if one can satisfy (3), then using reconstruction loss L_{rec} and the isometry loss L_{iso} terms enables φ to satisfy (1) and (2). To satisfy (3), we propose the geodesic k-skeleton loss L^k_{skel} and the geodesic k-skeleton cycle consistency loss L^k_{cyc}, which together encourage interpolated latent points to behave like valid intermediate states rather than drifting toward only the observed samples. These losses utilize an oracle Ω, which we also define later, serves as the key mechanism that scores whether a latent interpolant behaves like a genuine intermediate point on the data manifold. In stage two, we solve the RFM objective on the embedded data $\varphi(D)$, and for sampling we obtain the data-space path as the pushforward $\rho_t = [\varphi^{-1}]_{\#}\mu_t$, where μ_t is the learned vector field. The final generated samples are thus the decoded trajectories.

Below, we first introduce a simple approach that avoids modeling φ with high-dimensional NeuralODEs or invertible ResNets, as in prior work (de Kruiff et al., 2024; Diepeveen, 2024). We then define the losses and oracle above and explain how together they enforce geodesic convexity in the latent manifold. Finally, we provide guarantees for GCL-FM, including robustness to data noise and show how to mitigate instabilities when the data manifold contains regions of positive curvature. Proofs and full statements for our theoretical results can be found in Appendix A.

4.1. Efficient Modeling of φ.

Prior work learns ϕ as a diffeomorphism with globally bounded Lipschitz constants using NeuralODEs or invertible ResNets (de Kruiff et al., 2024; Diepeveen, 2024). This forces the latent dimension to match the data's ambient dimension, which is inefficient when the data lie on a much lower-dimensional manifold.

To take full advantage of the low-dimensional nature of D, we propose to model φ with an autoencoder pair (F, G) that learns a low-dimensional latent submanifold directly. For now, we assume that we have somehow satisfied requirement (3). Then, leveraging a classic result in Riemannian geometry known as the Myers-Steenrod theorem (Myers & Steenrod, 1939), it suffices to learn φ as a distance-preserving bijection without explicitly modeling smoothness as in prior work, if the encoder F already has learned a geodesic convex image. Thus, we utilize the reconstruction loss $L_{rec} = \mathbb{E}_{x\in D}\|x - G(F(x))\|_2^2$ encourages a bijection between D and $\varphi(D)$. To enforce distance preservation, we use the isometry loss $L_{iso} = \mathbb{E}_{x_i,x_j\in D} e^{\tau d_D(x_i,x_j)}(d_M(F(x_i), F(x_j)) - d_D(x_i, x_j))^2$, where d_D is a data-manifold distance from manifold learning methods and $\tau > 0$ controls the emphasis on local geometry. These losses together ensure requirement (1). For requirement (2), we apply SpectralNorm (Miyato et al., 2018) to control Lipschitz constants. We empirically show that this formulation outperforms the NeuralODE formulation (de Kruiff et al., 2024) on both synthetic and real-world datasets, while achieving significant speedups (see Section 5). To ensure the output of F is on M, we parametrize F as $F = \exp_p \circ F'$, where $p \in M$ and F' is an unrestricted neural network. However, *note that in order for the above formulation to be valid, one needs* (F, G) *to also satisfy requirement (3)*, further demonstrating the importance of learning a geodesic convex image. In the following section, we discuss a general framework to do so.

4.2. Learning a Geodesic Convex Latent Manifold

In this section, we introduce a general framework for learning requirement (3), i.e., a geodesically convex subset in the latent manifold. Geodesic convexity is **particularly important for trajectory inference**, since scientific datasets are often sparse and contain missing regions that the model attempts to generate. Conceptually, the goal is to make the learned latent image close to geodesically convex, so that geodesics between encoded data points remain inside the latent image rather than leaving it. To do this, we introduce ***skeleton and consistency losses*** below that encourage latent interpolants to stay consistent with true geodesic interpolations, instead of only fitting the observed samples pointwise. Accordingly, given $D \subset \mathbb{R}^d$ with intrinsic dimension d', we aim to learn $\varphi : \mathbb{R}^d \to M$ into a d'-dimensional symmetric Riemannian manifold M such that $\varphi(D) \subset M$ is geodesically convex with respect to the restricted metric $\mathfrak{g}M|\varphi(D)$.

Closure under Geodesic Skeletons. Prior methods for learning convex latent spaces typically operate in Euclidean settings by enforcing that straight-line interpolations in latent space decode and re-encode consistently, often using a discriminator (Sainburg et al., 2018; Oring et al., 2021) that encourages decoded interpolations to fool a critic. A natural extension to Riemannian manifolds is to replace straight lines with geodesic interpolants. However, under data scarcity, ensuring $\varphi(D)$ only covers pairwise interpolants can still leave holes in latent space—for example, geodesics between intermediate interpolated points may remain unmodeled. Moreover, in trajectory inference, portions of the data manifold are often missing, so L_{adv} may instead push interpolations toward observed samples rather than the correct intermediate (potentially unobserved) regions. To address issues caused by data scarcity, we propose to enforce that $\varphi(D)$ is closed under interpolation within geodesic triangles, and more generally within geodesic k-simplices. While natural choices are geometric barycenters, these are typically intractable to compute (Ramsay & Richtmyer, 2013). Instead, we use the local approximations:

Definition 4.1. Let $\{z_0, \dots, z_k\} \in M$, the *geodesic k-simplex* is the proxy for geodesic hull points: $\sigma(z_1, \dots, z_k) = \left\{ \exp_{z_0}\left(\sum_{i=1}^{k} u_i \log_{z_0}(z_i) \right) : u_i \geq 0, \sum_{i=1}^{k} u_i \leq 1 \right\}$. We further denote $z \in \sigma(z_1, \dots, z_k)$ as $z_{(z_i, u_i)}$, parametrized by the vertices and weights. The *geodesic k-skeleton* is the union of all such convex hulls for finite vertex set $Z \subset M$, denoted as $S_k(Z) = \bigcup_{j=0}^{k} \bigcup_{(z_i)_i \in Z^{j+1}} \sigma(z_0, \dots, z_j)$.

The **geodesic k-skeleton cycle consistency loss** is then the following for $\hat{x}^{(z)} = G(z)$, where the intuition is that a latent interpolant should remain stable under the encoder and decoder: $L_{cyc}^k = \mathbb{E}_{\substack{x_0,\dots,x_k \in D \\ z \in \sigma(F(x_0),\dots,F(x_k))}} \left[\left\| G(F(\hat{x}^{(z)})) - \hat{x}^{(z)} \right\|_2^2 \right]$.

To mitigate inappropriate attraction toward existing samples, we introduce a more general interpolation-indicator oracle to enforce geodesic convexity over simplices. Let Ω be an oracle that, given vertices $(z_i)_i$ and weights $(u_i)_i$, assigns lower scores to true interpolated points $z = z(z_i, u_i)$. We then define the **geodesic k-skeleton loss** as $L_{\text{skel}}^k = \mathbb{E}_{x_0,\dots,x_k \in D \; z \in \sigma(F(x_0),\dots,F(x_k))} \left[f(\Omega(z)) \right]$, where $f : \mathbb{R} \to \mathbb{R}$ is a monotonically increasing function that generalizes losses that use the same oracle. Here L_{skel}^k enforces geometric plausibility of the interpolated path. We discuss choices of Ω in later sections with empirical validation in Section 5.

Theoretical Analysis. During training, the vertex set is given as $Z = \varphi(X)$, where $X = \{X_i\}_i$ are the observed samples. Given the above definition, straight line interpolation corresponds to enforcing $S_1(Z) \subset \varphi(D)$. Intuitively, ensuring that $S_k(Z) \subset \varphi(D)$ would be a strict improvement over $S_{k-1}(Z)$ since $S_{k-1}(Z) \subseteq S_k(Z)$, and letting k be sufficiently large recovers $\varphi(D)$ as a geodesic subspace. We make this notion rigorous below.

Definition 4.2. For bounded $S \subseteq M$, we say S is *geodesic ϵ-convex* if $\sup_{z \in G_S} \inf_{s \in S} d_M(z, s) \leq \epsilon$ where G_S is the smallest geodesic convex set containing S w.r.t. Definition 4.2.

With a slight abuse of notation, we let $S_k(S)$ to be the union of all $S_k(V)$ for all finite $V \subset S$ when S is infinite and denote by $\delta_k = \sup_{z \in S_{k+1}(S)} \inf_{s \in S} d_M(z, s)$ the model error in learning $S_{k+1}(S) \subset S$.

Proposition 4.3. *(Informal) Let M be a manifold with unique minimizing geodesic or a strongly geodesic convex subspace Let $S \subset M$ be bounded. Then $\sup_{z \in G_S} \inf_{s \in S} d_M(z, s) \leq \delta_k + O\left(\frac{\alpha_{S,M}}{\sqrt{k+1}} \right)$ when $k < \dim(S)$. where $\alpha_{S,M}$ is a constant that depends on the radius of S and curvatures of M. Moreover, with reasonably accurate Ω, the bound becomes a function of the quality of Ω and L_{skel}.*

Increasing from k to $k+1$ then reduces the discrepancy between the model training loss and geodesic "convexity-ness" of $\varphi(D)$ by a factor of $\sqrt{\frac{k+1}{k+2}}$, and $\varphi(D)$ is exactly geodesic δ_k-convex when $k \geq \dim(S)$ by a standard Carathéodory type application. Hence as k increases, the learned latent space is nearly geodesic δ_k-convex by minimizing the skeleton loss L^k_{skel}, where the consistency loss ensures that the model still satisfies requirement (1).

4.3. Robustness Against Noisy Data

Robustness on Positively Curved Manifolds. Prior work that uses pullback geometry to model data manifolds (de Kruiff et al., 2024; Diepeveen, 2024) relies on robustness of the learned geodesics to dataset noise: when M has non-positive sectional curvature near the data, this noise is damped in the pullback geodesics. In contrast, positive sectional curvature can amplify noise, causing large deviations even in simple cases such as learning a 1D circle (de Kruiff et al., 2024). Thus, **prior methods can be impractical for data that exhibit positive curvature**. In this section, we propose a framework with theoretical robustness guarantees to data noise when learning representations of positively curved data manifolds.

Proposition 4.4. *(Informal) Let M have positive sectional curvature, and let $\varphi : \mathbb{R}^n \to M$ satisfy that $\varphi|_D$ is a diffeomorphism $D \to \varphi(D)$ with $\varphi(D)$ geodesically convex. Suppose further that $\varphi(D) \subset B_r(p)$ for some $p \in M$, where $B_r(p)$ is a geodesic ball whose sectional curvatures are upper bounded by $\kappa > 0$, and $r \leq \frac{\pi}{4\sqrt{\kappa}}$. Then the pullback geodesic instability induced by M is at worst the Euclidean case. If instead $r \leq \frac{\pi}{2\sqrt{\kappa}} - \epsilon$ for some $\frac{\pi}{4\sqrt{\kappa}} > \epsilon > 0$, then the instability is worse than the Euclidean case by a factor of $O(\mathrm{poly}(1/\epsilon))$.*

Proposition 4.4 implies robustness to data noise when φ maps the data manifold into a small geodesic ball with bounded curvature, so we aim to constrain $\varphi(D) \subset B_r(p)$. We do this by parametrizing the encoder as $F(x) = \exp_p\left(\psi(\|f(x)\|_2)\frac{f(x)}{\|f(x)\|_2}\right)$, where f is a neural network, $\exp_p$ is the exponential map at p, and $\psi : [0,\infty) \to [0,r)$ is smooth and monotone (e.g., $\tanh$ or $\mathrm{sigmoid}$). The map ψ radially contracts $f(D) \subset T_pM$ into $B_r(0) \subset T_pM$. Since the sectional curvatures in $B_r(p)$ are upper bounded by κ, the injectivity radius satisfies $\mathrm{inj}(p) \geq \frac{\pi}{\sqrt{\kappa}}$; choosing r as in Proposition 4.4 ensures that $\exp_p$ is a diffeomorphism $B_r(0) \subset T_pM \to B_r(p) \subset M$.

Robust Optimal Transport Paths. We further show that under the framework of GCL-FM, the OT paths learned by the model are robust to data noise, which is a theoretical result missing in prior attempts (de Kruiff et al., 2024).

Proposition 4.5. *(Informal, OT Path Stability) Let (M, g) be a d-dimensional symmetric Riemannian manifold and $\varphi : \mathbb{R}^d \to M$ a diffeomorphism onto a geodesic convex subset $\varphi(\mathbb{R}^d) \subset M$. Let $\mu, \nu \in P_2(\mathbb{R}^d)$ be probability distributions of finite second moment, let π be an optimal coupling w.r.t. the measure induced by d_φ from the pullback metric $g_\varphi = \varphi^* g$. Let $\tau \in P_2(\mathbb{R}^d)$ be a probability distribution s.t. $\tau \to \mu$. Then assuming absolute continuity of distributions and mild regularity of φ, $W_2(\rho_{\nu,\mu}(t), \rho_{\nu,\tau}(t))^2 = O(W_2(\tau,\mu)^{1/6})$ with similar coefficients to the geodesic case. Further, it is enough to assume the same regularity on a large enough closed subset of the support along the OT paths to obtain the same bound.*

Thus, when learning on noisy data, the deviation of the learned OT paths from the ground truth is bounded by a polynomial function of the deviation caused by the target distribution's noise, where the bound is equal to the current best bound in the manifold OT path deviation under any target noise (Kitagawa et al., 2025)

4.4. Overall Framework.

Here we describe GCL-FM's overall framework. The model proceeds with two stages. First, we train an autoencoder pair (F, G) with the architecture specified in Section 4.1. The loss is given by $L_{ae} = \lambda_{rec}L_{rec} + \lambda_{iso}L_{iso} + \lambda_{cyc}L^k_{cyc} + \lambda_{skel}L^k_{skel}$. Here we specify two choices for Ω. The first is a critic $\mathcal{C} : \mathbb{R}^d \to [0,1]$ that outputs higher scores for real samples. Then $\Omega(z_{(z_i,u_i)}) = -\log(\mathcal{C}(z_{(z_i,u_i)}))$. The critic is updated along with the autoencoder. The second choice is distance-based, where we approximate the distance from the decoded interpolation to each vertex x_i via a one-hop neighborhood estimation: $d^*(G(z_{(z_i,u_i)}), x_i) = \min_{x^* \in D^K} \|G(z_{(z_i,u_i)}), x^*\|_2 + d_D(x^*, x_i)$, where D^K denote the K closest neighbors to z in D. For efficiency, we cluster D based on distance first and pick one representative per cluster to compute D^K. The oracle is then $\Omega(z_{(z_i,u_i)}) = \sum_{i=0}^{k}\left(d_M(z_{(z_i,u_i)}, z_i) - d^*(G(z_{(z_i,u_i)}), x_i)\right)^2$. Both oracles are tested in Section 5. In the second stage, we train a neural network $v_{t,\theta}$ for the RFM objective on the encoded manifold $F(D)$. The final model generates samples $G(z_t)$ where $z_t \sim \mu_t$ and μ_t is the density path generated by $v_{t,\theta}$.

Table 2. W1 distance between generated cells and held-out cells on the Cite-seq and Multiome datasets. Best performance is bolded while second ot best is underscored. GCL-FM achieves the best performance across both datasets for 50 and 100 dimensions.

Data Dimension	**50**		**100**	
Alg.↓ Dataset→	**Cite**	**Multi**	**Cite**	**Multi**
DSBM (Shi et al., 2023)	53.81 ± 7.74	66.43 ± 14.39	58.99 ± 7.62	70.75 ± 14.03
I-CFM (Tong et al., 2024a)	41.83 ± 3.28	49.78 ± 4.43	48.28 ± 3.28	57.26 ± 3.86
OT-CFM (Tong et al., 2024a)	38.76 ± 0.40	47.58 ± 6.62	45.39 ± 0.42	54.81 ± 5.86
$[\text{SF}]^2$M-Exact (Tong et al., 2024b)	40.01 ± 0.78	45.34 ± 2.83	46.53 ± 0.43	52.89 ± 1.99
$[\text{SF}]^2$M-Geo (Tong et al., 2024b)	38.52 ± 0.29	44.80 ± 1.91	44.50 ± 0.42	52.20 ± 1.96
WLF-SB (Neklyudov et al., 2024)	39.24 ± 0.07	47.79 ± 0.11	46.18 ± 0.08	55.72 ± 0.06
WLF-OT (Neklyudov et al., 2024)	36.17 ± 0.03	38.74 ± 0.06	42.86 ± 0.04	47.37 ± 0.05
WLF-UOT (Neklyudov et al., 2024)	34.16 ± 0.04	$\underline{36.13} \pm 0.02$	41.08 ± 0.04	45.23 ± 0.01
OT-MFM (Kapuśniak et al., 2024)	36.39 ± 1.87	45.16 ± 4.96	41.78 ± 1.02	50.91 ± 4.623
GAGA[2] (Sun et al., 2025)	$\underline{28.77} \pm 0.89$	43.84 ± 1.59	$\underline{37.8} \pm 2.89$	$\underline{40.74} \pm 2.22$
GCL-FM (Ours)	$\mathbf{19.98} \pm 0.55$	$\mathbf{20.69} \pm 0.41$	$\mathbf{23.93} \pm 0.48$	$\mathbf{25.49} \pm 0.28$

5. Experiments

We evaluate GCL-FM on single-cell and computational fluid dynamics (CFD) datasets. Single-cell data consist of sparse, noisy population snapshots evolving according to biological constraints, while CFD data provide particle trajectories governed by fluid dynamics. As a baseline implementation, we use $M = \mathbb{R}^n$ for simplicity. On mouse erythroid data, we also implement GCL-FM with $M = S^n$ to test whether a more suitable latent manifold improves performance and to validate Proposition 4.4. We use the critic-based oracle for Cite-seq and Multiome and the distance-based oracle for mouse erythroid and CFD. Finally, we perform ablation studies in Section 5.4 and provide runtime analysis, with additional experimental details and ablations are defferred to Appendix Section B

5.1. Trajectory Inference on CITE-seq and Multiome

We validate GCL-FM's performance for single-cell trajectory inference on the CITE-seq and Multiome datasets (Burkhardt et al., 2022). We follow the setup in GAGA (Sun et al., 2025), where we perform the leave-one-timepoint-out cellular dynamics experiment in which points at one timepoint are excluded, and the goal is to infer the left-out points by interpolating between the remaining timesteps. The results are shown in Table 2, GCL-FM achieves significantly lower W1 distance to the ground-truth data than previous method, demonstrating its ability to solve the trajectory inference problem to reconstruct the left-out cells.

5.2. Experiments on Mouse Erythroid Data

We further validate GCL-FM on modeling cell differentiation in mouse erythroid development over time. We follow the CurlyFM setup (Petrović et al., 2025), partition the data into three temporal snapshots, and withhold the middle one for testing. We report cosine distance and L_2 distance for the learned velocity field, and $\mathcal{W}_2$ distance for generated samples. We additionally evaluate GCL-FM with $M = S^n$, using both a naive spherical model and a projected variant based on Proposition 4.4. Results are shown in Table 3, with the full table including standard deviations deferred to Appendix B and Table 11. Overall, GCL-FM attains smaller L_2 error and competitive or better $\mathcal{W}_2$, while the projected spherical variant consistently improves over the Euclidean one. In contrast, the naive spherical variant performs worse, illustrating the positive-curvature instability discussed in prior work (de Kruiff et al., 2024; Diepeveen, 2024)

Table 3. Quality of learned vector field and generated points on mouse erythroid data, comparing baselines and GCL-FM variants. GCL-FM achieves significantly lower L2 error with better or comparable W_2 distance, with the projection significantly improving the spherical variants.

Metric	**OT-CFM**	**MFM**	**CurlyFM**	**GCLFM** ($\mathbb{S}^n$ **+ Proj.**)	**GCLFM** ($\mathbb{S}^n$ **w/o Proj.**)	**GCLFM** ($\mathbb{R}^n$)
Dimension $d = 2$						
Cos. Dist	0.146	0.014	0.009	**0.003**	**0.003**	0.005
L_2	2.704	1.999	1.663	**1.302**	1.902	1.312
$\mathcal{W}_2$	0.646	0.269	0.369	**0.181**	0.361	1.815
Dimension $d = 20$						
Cos. Dist	0.489	0.495	**0.488**	0.498	0.496	0.496
L_2 ($\times 10^3$)	1.885	1.627	1.721	**0.158**	0.248	0.167
$\mathcal{W}_2$	6.103	4.855	6.124	**4.307**	4.927	4.454
Dimension $d = 50$						
Cos. Dist	0.490	0.494	0.489	**0.498**	0.498	0.497
L_2 ($\times 10^3$)	2.215	1.971	2.045	**0.160**	0.262	0.188
$\mathcal{W}_2$	7.969	**6.727**	7.729	7.167	7.691	7.317

5.3. Experiments on Computational Fluid Dynamics

We evaluate GCL-FM on computational fluid dynamics data represented as particles advected by a flow over time, where the positions are also used to compute a reference velocity field. We follow CurlyFM(Petrović et al., 2025) and hold out a subset of particles and evaluate on these unseen particles at each marginal. We measure position prediction accuracy using (i) mean squared error (MSE) between predicted positions and the ground-truth positions, and (ii) precision@k, which is how often the prediction lies among the k nearest neighbors of the corresponding ground-truth particle. We also evaluate velocity-field reconstruction by the cosine distance between the learned velocity field and the reference field. The results are shown in Table 4. GCL-FM outperforms all baselines in MSE distance and Prec.@k, demonstrating GCL-FM's ability to more accurately recover the true particle coupling and trajectories. GCL-FM also outperforms the Euclidean baselines in cosine distance while achieving comparable performance as CurlyFM, showing its ability to learn the underlying velocity field. Notably, GCL-FM achieves the velocity field alignment without sacrificing the generation quality, which is not the case for CurlyFM.

Table 4. Velocity field alignment (Cos.Dist.) and particle position prediction accuracy (MSE and Prec.@k) on CFD data. GCL-FM achieves higher prediction accuracy while aligning with the reference vector field. Standard deviations (STD) are small and are omitted due to visibility. See Appendix Section B for full table.

Method	Cos. Dist. ↓	MSE ↓	Prec.@5 ↑	Prec.@10 ↑	Prec.@25 ↑
CFM	0.254	0.085	0.079	0.164	0.337
OT-CFM	0.248	0.095	0.303	0.388	0.496
CurlyFM	**0.189**	0.095	0.489	0.522	0.628
GCL-FM (ours)	0.197	**0.077**	**0.499**	**0.603**	**0.683**

5.4. Ablation and Additional Studies

We perform ablation studies to study the effects of varying k and provide runtime analysis. Additional ablations studies are shown in Appendix B, including oracle ablation and robustness to hyperparameters. We also include an additional comparison with FlatVI (Palma et al., 2025) in the appendix, omitted here as FlatVI and the task are tailored to discrete generative modeling, whereas our focus here is transport on continuous manifolds.

Ablation on k. To assess the effect of varying k in L^k_{skel} and L^k_{cyc} on GCL-FM 's performance, we evaluate GCL-FM on single-cell data with $k \in \{0, 1, 2, 3\}$. We also test our diffeomorphism formulation against PFM and d'-PFM (de Kruiff et al., 2024), which parametrizes the diffeomorphism as a NeuralODE. Results are shown in Table 5, where k-GCL-FM denotes the value of k used in both L^k_{skel} and L^k_{cyc} (0 indicates that these losses are removed). Overall, $k = 3$ performs best, and larger k generally improves performance. The largest gain occurs going from $k = 0$ to $k = 1$, highlighting the effectiveness of the losses. Nonetheless, 0-GCL-FM still outperforms the baselines in Table 2 and PFM, demonstrating the effectiveness of our diffeomorphism parametrization and that the model can sometimes infer geodesic convexity from the data.

Table 5. W_1 distance on the Cite-seq and Multiome datasets. k indicates the corresponding L^k_{cyc}, L^k_{skel} losses.

Data Dimension	**50**		**100**	
Alg.↓ Dataset→	**Cite**	**Multi**	**Cite**	**Multi**
PFM	26.08 ± 0.59	39.31 ± 0.55	34.42 ± 0.99	78.71 ± 2.22
d'-PFM	56.87 ± 1.35	125.35 ± 3.11	38.42 ± 1.77	178.96 ± 3.54
0-GCL-FM	21.53 ± 0.67	22.00 ± 0.24	24.42 ± 0.80	26.13 ± 0.80
1-GCL-FM	20.37 ± 0.74	20.98 ± 0.54	24.29 ± 0.69	$\mathbf{25.41} \pm 0.21$
2-GCL-FM	20.85 ± 0.45	$\mathbf{20.23} \pm 0.48$	24.27 ± 0.41	$\mathbf{25.43} \pm 0.22$
3-GCL-FM	$\mathbf{19.98} \pm 0.55$	20.69 ± 0.41	$\mathbf{23.93} \pm 0.48$	$\mathbf{25.49} \pm 0.28$

Runtime Comparisons. We compare the runtime and memory footprint of GCL-FM to prior methods. In Figure 3, we show total training time and peak GPU memory for reproducing the results in Table 2 on the 100D Multiome dataset. GCL-FM, PFM, and GAGA are trained for the same number of epochs, while MFM follows (Kapuśniak et al., 2024) and is trained longer due to slower convergence. Overall, GCL-FM uses substantially less compute and memory.

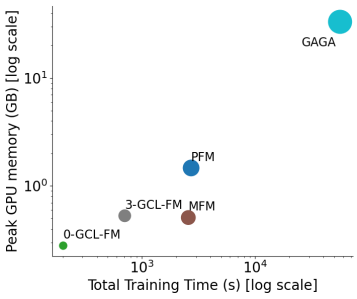

Figure 3. Total runtime (s) and peak GPU memory (GB) for manifold-based methods on the 100d Multiome data.

6. Conclusion and Limitation

We introduce GCL-FM, a flow matching framework for learning transport dynamics on unknown data manifolds. We provide theoretical guarantees that GCL-FM is robust to data noise and can recover geodesic convex latent submanifolds, and empirically demonstrate improved trajectory inference across single-cell and computational fluid dynamics data. GCL-FM also resolves instabilities that prior pullback methods face on positively curved manifolds. While we tested two oracles in this work, we have not explored broader oracle design, especially tailoring to downstream tasks. Further, although our experiments focus on data embedded in Euclidean space, GCL-FM naturally extends to non-Euclidean ambient spaces. Finally, systematically studying the choice of latent manifold is another promising direction.

References

Barman, S. Approximating nash equilibria and dense bipartite subgraphs via an approximate version of carathéodory's theorem. In *STOC*, pp. 361–369, 2015.

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

Chen, R. T. Q. and Lipman, Y. Flow matching on general geometries. In *ICLR*, 2024.

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. Neural ordinary differential equations. In *NeurIPS*, 2018.

Collas, A., Ju, C., Salvy, N., and Thirion, B. Riemannian flow matching for brain connectivity matrices via pullback geometry. In *NeurIPS*, 2025.

Croitoru, F.-A., Hondru, V., Ionescu, R. T., and Shah, M. Diffusion models in vision: A survey. *TPAMI*, 2023.

De Bortoli, V., Mathieu, E., Hutchinson, M., Thornton, J., Teh, Y. W., and Doucet, A. Riemannian score-based generative modelling. *NeurIPS*, 35:2406–2422, 2022.

de Kruiff, F., Bekkers, E., Öktem, O., Schönlieb, C.-B., and Diepeveen, W. Pullback flow matching on data manifolds. *arXiv preprint arXiv:2410.04543*, 2024.

Diepeveen, W. Pulling back symmetric riemannian geometry for data analysis. *arXiv preprint arXiv:2403.06612*, 2024.

Duque, A. F., Morin, S., Wolf, G., and Moon, K. R. Extendable and invertible manifold learning with geometry regularized autoencoders. In *2020 IEEE International Conference on Big Data (Big Data)*, pp. 5027–5036. IEEE, December 2020.

Duque, A. F., Morin, S., Wolf, G., and Moon, K. R. Geometry regularized autoencoders. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):7381–7394, 2023.

Evensen, G. The ensemble kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics*, 53: 343–367, 2003.

Fasina, O., Huguet, G., Tong, A., Zhang, Y., Wolf, G., Nickel, M., Adelstein, I., and Krishnaswamy, S. Neural FIM for learning Fisher information metrics from point cloud data. In *ICML*, 2023.

Fefferman, C., Mitter, S., and Narayanan, H. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016.

Gemici, M. C., Rezende, D., and Mohamed, S. Normalizing flows on riemannian manifolds. *arXiv preprint arXiv:1611.02304*, 2016.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020.

Huang, C.-W., Aghajohari, M., Bose, J., Panangaden, P., and Courville, A. C. Riemannian diffusion models. *NeurIPS*, 35:2750–2761, 2022.

Jasche, J. and Wandelt, B. D. Bayesian physical reconstruction of initial conditions from large-scale structure surveys. *Monthly Notices of the Royal Astronomical Society*, 432(2):894–913, 2013.

Kapuśniak, K., Potaptchik, P., Reu, T., Zhang, L., Tong, A., Bronstein, M., Bose, A. J., and Di Giovanni, F. Metric flow matching for smooth interpolations on the data manifold. In *NeurIPS*, 2024.

Kitagawa, J., Letrouit, C., and Mérigot, Q. Stability of optimal transport maps on riemannian manifolds. In *arXiv:2504.05412*, 2025.

Lee, J. M. Smooth manifolds. In *Introduction to Smooth Manifolds*, pp. 1–31. Springer, 2013.

Lee, Y., Yoon, S., Son, M., and Park, F. C. Regularized autoencoders for isometric representation learning. In *ICLR*, 2022.

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. In *ICLR*, 2023.

Lou, A., Xu, M., Farris, A., and Ermon, S. Scaling riemannian diffusion models. *NeurIPS*, 36, 2024.

Macosko, E. Z., Basu, A., Satija, R., Nemesh, J., Shekhar, K., Goldman, M., Tirosh, I., Bialas, A. R., Kamitaki, N., Martersteck, E. M., Trombetta, J. J., Weitz, D. A., Sanes, J. R., Shalek, A. K., Regev, A., and McCarroll, S. A. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell*, 161(5):1202–1214, May 2015.

Mangoubi, O., He, N., and Vishnoi, N. K. Efficient diffusion models for symmetric manifolds. In *ICML*, 2025.

McCann, R. J. Polar factorization of maps on riemannian manifolds. *Geometric and Functional Analysis*, 11(3): 589–608, 2001.

Mi, L., He, T., Park, C. F., Wang, H., Wang, Y., and Shavit, N. Revisiting latent-space interpolation via a quantitative evaluation framework. In *arXiv preprint*, 2021.

Michelis, M. Y. and Becker, Q. On linear interpolation in the latent space of deep generative models. In *ICLR Workshop on Geometrical and Topological Representation Learning*, 2021.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for generative adversarial networks. In *ICLR*, 2018.

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

Myers, S. and Steenrod, N. The group of isometries of a riemannian manifold. In *Annals of Mathematics*, 1939.

Neklyudov, K., Brekelmans, R., Tong, A., Atanackovic, L., Liu, Q., and Makhzani, A. A computational framework for solving Wasserstein lagrangian flows. In *ICML*, 2024.

Oring, A., Yakhini, Z., and Hel-Or, Y. Autoencoder image interpolation by shaping the latent space. *ICML*, 2021.

Palma, A., Rybakov, S., Hetzel, L., Günnemann, S., and Theis, F. J. Enforcing latent euclidean geometry in single-cell VAEs for manifold interpolation. In *ICML*, 2025.

Pandarinath, C., O'Shea, D. J., Collins, J., Jozefowicz, R., Stavisky, S. D., Kao, J. C., Trautmann, E. M., Kaufman, M. T., Ryu, S. I., Hochberg, L. R., Henderson, J. M., Shenoy, K. V., Abbott, L. F., and Sussillo, D. Inferring single-trial neural population dynamics using sequential auto-encoders. *Nature Methods*, 15:805–815, 2018.

Petrović, K., Atanackovic, L., Moro, V., Kapuśniak, K., Ceylan, İ. İ., Bronstein, M. M., Bose, A. J., and Tong, A. Curly flow matching for learning non-gradient field dynamics. In *NeurIPS*, 2025.

Pijuan-Sala, B., Griffiths, J. A., Guibentif, C., et al. A single-cell molecular map of mouse gastrulation and early organogenesis. *Nature*, 566:490–495, 2019.

Ramsay, A. and Richtmyer, R. D. *Introduction to hyperbolic geometry*. Springer Science & Business Media, 2013.

Sainburg, T., Thielk, M., Theilman, B., Migliori, B., and Gentner, T. Generative adversarial interpolative autoencoding: adversarial training on latent space interpolations encourage convex latent distributions. *arXiv preprint arXiv:1807.06650*, 2018.

Shi, R., Zeng, W., Su, Z., Damasio, H., Lu, Z., Wang, Y., Yau, S.-T., and Gu, X. Hyperbolic harmonic mapping for constrained brain surface registration. In *Proceedings of the IEEE Conference on computer vision and pattern recognition*, pp. 2531–2538, 2013.

Shi, Y., De Bortoli, V., Campbell, A., and Doucet, A. Diffusion schrödinger bridge matching. In *NeurIPS*, volume 36, 2023.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021a.

Song, Z., Yang, X., Xu, Z., and King, I. Graph-based semi-supervised learning: A comprehensive review. *arXiv preprint arXiv:2102.13303*, 2021b.

Sun, X., Xu, C., Rocha, J. F., Liu, C., Hollander-Bodie, B., Goldman, L., DiStasio, M., Perlmutter, M., and Krishnaswamy, S. Hyperedge representations with hypergraph wavelets: Applications to spatial transcriptomics. In *ICASSP*, 2024.

Sun, X., Liao, D., MacDonald, K., Zhang, Y., Liu, C., Huguet, G., Wolf, G., Adelstein, I., Rudner, T. G. J., and Krishnaswamy, S. Geometry-aware generative autoencoders for warped riemannian metric learning and generative modeling on data manifolds. In *AISTATS*, pp. 1018–1026, 2025.

Tong, A., Huang, J., Wolf, G., Van Dijk, D., and Krishnaswamy, S. Trajectorynet: A dynamic optimal transport network for modeling cellular dynamics. In *ICML*, 2020.

Tong, A., Fatras, K., Malkin, N., Huguet, G., Zhang, Y., Rector-Brooks, J., Wolf, G., and Bengio, Y. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024a. ISSN 2835-8856.

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

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

Urdapilleta, E., Troiani, F., Stella, F., and Treves, A. Can rodents conceive hyperbolic spaces? *Journal of the Royal Society Interface*, 12(107):20141214, 2015.

Villegas-Morcillo, A., Sanchez, V., and Gomez, A. M. Foldhsphere: deep hyperspherical embeddings for protein fold recognition. *BMC bioinformatics*, 22:1–21, 2021.

Yim, J., Trippe, B. L., De Bortoli, V., Mathieu, E., Doucet, A., Barzilay, R., and Jaakkola, T. Se (3) diffusion model with application to protein backbone generation. In *ICML*, pp. 40001–40039, 2023.

Zhang, H., Rich, P. D., Lee, A. K., and Sharpee, T. O. Hippocampal spatial representations exhibit a hyperbolic geometry that expands with experience. *Nature Neuroscience*, 26(1):131–139, 2023.

Zhu, Y., Chen, T., Kong, L., Theodorou, E. A., and Tao, M. Trivialized momentum facilitates diffusion generative modeling on lie groups. *ICLR*, 2025.

A. Proofs for Theoretical Results

A.1. Proof for Proposition 4.3

Proposition 4.3. Let M be a complete geodesic d-dimensional Riemannian manifold. Let $S \subset M$ be bounded and let G_S be the smallest geodesic convex set containing S. Suppose $\exists p \in M$ such that for some $r > 0$, the closed geodesic ball $\overline{B_r(p)}$ is strongly geodesic convex and contains G_S. Further, suppose that the sectional curvature on S satisfies. Let $R = \sup_{s \in S} d_M(s, p)$, then for $k < \dim(S)$, $\sup_{z \in G_S} \inf_{s \in S} d_M(z, s) \leq \delta_k + O\left(\frac{\max_{\ell \leq R} \max(1, \beta_{K_{\min}}(\ell))}{\sqrt{k+1}}\right)$, where $K_{\min}$ is the lowerbound of the sectional curvature on $B_r(p)$, and $\beta_\kappa : \mathbb{R} \to \mathbb{R}$ is the function

$$\beta_\kappa(\ell) = \begin{cases} \frac{\sin(\sqrt{\kappa}\ell)}{\sqrt{\kappa}}, \kappa > 0 \\ \ell, \kappa = 0, \\ \frac{\sinh(\sqrt{-\kappa}\ell)}{\sqrt{-\kappa}}, \kappa < 0 \end{cases}$$

In particular, for suitably small R, the bound becomes $\sup_{z \in G_S} \inf_{s \in S} d_M(z, s) \leq \delta_k + O\left(\frac{poly(R, K_{\min})}{\sqrt{k+1}}\right)$

Proof. We utilize the following result from (Barman, 2015)

Theorem A.1 ((Barman, 2015))**.** *Let $X \subset \mathbb{R}^n$ be a finite set and fix $\epsilon > 0$. Then for every $\mu \in \operatorname{conv}(X)$ and $2 < p < \infty$, let γ denote $\max_{x \in X} \|x\|_p$. Then there is an ϵ-approximation μ' supported on at most $m = \frac{4p\gamma^2}{\epsilon^2}$ points of X such that $\|\mu - \mu'\|_p < \epsilon$.*

Then let $V \subset T_pM$ be any finite set such that $\|v\| < R$ for any $v \in V$. Letting $p = 2$, $\gamma = R$ in the above theorem immediately gives that for any x in the convex hull $\operatorname{conv}(V)$ and integer $m \geq 1$, there is some $x' = \sum_{i=1}^{m} \alpha_i v_i$ with $\sum_{i=1}^{m} \alpha_i = 1$, $\alpha_i > 0$, and $v_i \in V$ such that $\|x - x'\| < \frac{2\sqrt{2}R}{\sqrt{m}}$.

Now we proceed with the rest of the proof for Proposition 4.3. Since $\overline{B_r(p)}$ is assumed to be strongly geodesic convex, $\exp_p$ restricts to a diffeomorphism $B_r(0) \subset T_pM \to B_r(p) \subset M$. Let $V = \log_p(S) \subset T_pM, X = \log_p(G_s) \subset T_pM$. Note that by construction of G_S, we have $X \subseteq \operatorname{conv}(V)$. Now fix $x \in G_S$ and let $z = \log_p(x) \in X$. Note that $\|v\| \leq R$ for any $v \in V$ by definition of R. Then applying the theorem from (Barman, 2015) with $m = k + 1$, we get there is some $x' = \sum_{i=0}^{k} \lambda_i v_i$ with $v_i \in V$, $\sum_{i=0}^{k} \lambda_i = 1$ such that $\|x' - x\|_2 \leq \frac{2\sqrt{2}R}{\sqrt{k+1}}$. Let $z' = \exp_p(x')$. By assumption $z' \in \exp(\operatorname{conv}(V)) \subseteq B_r(p)$. Further, note that $z' = \exp_p(\sum_i \lambda_i v_i) = \exp_p(\sum_i \lambda_i \log_i(s_i))$, where we take $s_i = \exp_p(v_i) \in S$. Thus z' is particular instance of a geodesic simplex $\sigma(z_0, \ldots, z_k)$ by taking $z_0 = p$. Thus $z' \in S_{k+1}(S)$. Now note that

$$d_M(z, z') = d_M(\exp_p(x), \exp_p(x')) \leq \Lambda(p, R)\|x - x'\| \leq \frac{2\sqrt{2}\Lambda(p, R)}{\sqrt{k+1}}.$$

where $\Lambda(p, R) = \sup_{\|v\| \leq R} \|d(\exp_p)_v\|$. Since $z' \in S_{k+1}(S)$, $d_M(z', S) \leq \delta_k$ by construction. Thus we have

$$d_M(z, S) \leq d_M(z, z') + d_M(z', S) \leq \delta_k + \frac{2\sqrt{2}\Lambda(p, R)}{\sqrt{k+1}}.$$

Since $z \in G_S$ arbitrary, taking the supremum yields $\sup_{z \in G_S} d_M(z, S) \leq \delta_k + \frac{2\sqrt{2}\Lambda(p,R)}{\sqrt{k+1}}$.

Now let $v \in T_pM$. By assumption let $\|v\| = \ell \leq R$. Let $a \in \mathbb{R}$ and let $w = av \in T_pM$. By Gauss lemma, we have

$$\langle d(\exp_p)_v(v), d(\exp_p)_v(w) \rangle = \langle v, w \rangle = a\|v\|$$

where the inner product and norm are taken over $T_{\exp_p(v)}M$. Thus $\|d(\exp_p)_v(w)\| = |a|\|v\|$.

Now consider $w \perp v$. Let $u = v/\ell$ be a unit vector. Consider the unit speed geodesic $\gamma : [0,1] \to M$ via $\gamma(s) = \exp_p(su)$. Let $J(s) = d(\exp_p)_{su}(sw)$ be the Jacobi field along γ. Now let $t = s/\ell$, then at $t = 1$, we get

$$J(\ell) = d(\exp_p)_{\ell u}(\ell w) = d(\exp_p)_v(w)$$

Now by standard application of Rauch comparison theorem against manifolds with constant curvature less than $K_{\min}$ gives

$$\|J(r)\| = \|d(\exp_p)_v(w)\| \leq \|w\| \frac{\beta_{K_{\min}}(\ell)}{\ell}.$$

Now for arbitrary $w \in T_pM$ decomposed into radial and normal components $w_v \parallel v$ and $w_\perp \perp v$, we have

$$\begin{aligned}
\|d(\exp_p)_v(w)\|^2 &= \|d(\exp_p)_v(w_v + w_\perp)\| \\
&= \|d(\exp_p)_v(w_v) + d(\exp_p)_v(w_\perp)\|^2 \\
&= \|d(exp_p))v(w_v)\|^2 + \|d(exp_p))v(w_\perp)\|^2 \\
&\leq |a|^2\|v\|^2 + \|w_\perp\|^2(\beta_{K_{\min}}(\|v\|)/\|v\|)^2
\end{aligned}$$

On the other hand, $\|w\|^2 = |a|^2\|v\|^2 + \|w_\perp\|^2$, thus

$$\begin{aligned}
\frac{\|d(\exp_p)_v(w)\|^2}{\|w\|^2} &\leq \frac{|a|^2\|v\|^2 + \|w_\perp\|^2(\beta_{K_{\min}}(\|v\|)/\|v\|)^2}{\|w\|^2} \\
&= \frac{|a|^2\|v\|^2 + \|w_\perp\|^2(\beta_{K_{\min}}(\|v\|)/\|v\|)^2}{|a|^2\|v\|^2 + \|w_\perp\|^2} \\
&\leq \max(1, (\beta_{K_{\min}}(\|v\|)/\|v\|)^2)
\end{aligned}$$

giving us the desired bound.

To obtain the polynomial bound $\sup_{z \in G_S} \inf_{s \in S} d_M(z,s) \leq \delta_k + O\left(\frac{poly(R, K_{\min})}{\sqrt{k+1}}\right)$, we consider case work on $K_{\min}$. When $K_{\min} = 0$ the bound is trivial. When $K_{\min} > 0$, using Taylor expansion on $\sin$ and assuming $R \leq 1/\sqrt{K_{\min}}$ gives the bound. When $K_{\min} < 0$, using Taylor expansion on $\sinh$ and assuming $R \leq 1/(2\sqrt{-K_{\min}})$ gives the polynomial bound.

□

The informal version stated in our main text is thus the case where $M = \overline{B_r(p)}$ itself. The following lemma connects the bound to our loss terms and completes the second half of the proposition:

Lemma A.2. *Let M, S be the same as Proposition 4.3, let μ_{k+1} denote the distribution over interpolation points from the sampling procedure in computing L^k_{selk}. Suppose we have access to an oracle Ω and monotonically increasing function f such that $f(\Omega(z)) \geq c d_M(z,S)^p$ for constants $c, p > 0$. Further, suppose that there are $p, r > 0$ such that for any $z \in S_{k+1}(S)$, $\mu_{k+1}(B_r(z)) \geq p$. Then $\delta_k \leq r + \left(\frac{L^k_{skel}}{cp}\right)^{1/p}$.*

Proof. Fix $\epsilon > 0$ and let $z_\epsilon \in S_{k+1}(S)$ be such that $d_M(z_\epsilon, S) > \delta_k - \epsilon$. Note that for any $w \in B_r(z_\epsilon)$, we then have

$$d_M(w,S) \geq d_M(z_\epsilon, S) - d_M(w, z_\epsilon) \geq \delta_k - \epsilon - r$$

Then $f(\Omega(w)) \geq c d_M(w,S)^p \geq c(\delta_k - \epsilon - r)^p$. So we have

$$\begin{aligned}
L^k_{skel} &= \int f(\Omega(z)) d_{\mu_{k+1}}(z) \\
&\geq \int_{B_r(z_\epsilon)} f(\Omega(z)) d_{\mu_{k+1}}(z) \\
&\geq c(\delta_k - \epsilon - r)^p \mu_{p+1}(B_r(z_\epsilon)) \\
&\geq cp(\delta_k - \epsilon - r)^p
\end{aligned}$$

where the last inequality is by assumption. Letting $\epsilon \to 0$ and re-arranging the terms give the desired bound.

□

Discussion. Throughout our proof, we assumed that G_S is contained in a strongly geodesic convex ball. In the case of Hadamard manifold, this assumption is automatically satisfied. In the case of positively curved manifolds, we follow projection method in Proposition Theorem 4.4 and project the data manifold into a strongly geodesic ball. In the general case where $B_r(p)$ is assumed to only be geodesic convex, one may consider a large open set $U \subset B_r(p)$ and cover it by strongly geodesic balls $\{B_{r_i}(p_i)\}$. If G_S is contained entirely within U, then one can prove a similar bound with an analogous proof to the one above, by carefully stitching together paths via overlapping charts. If G_S is not contained within U, the bound incurs an additional error term measuring the distance between the two sets. In this case, one may potentially add an contrastive loss term to L_{skel} to pull G_S into U.

A.2. Proof for Proposition 4.4

Proposition 4.4. Let M be a symmetric Riemannian manifold with positive curvature. Let $\varphi : \mathbb{R}^n \to M$ such that $\varphi|_D$ is a diffeomorphism $D \to \varphi(D)$ with $\varphi(D)$ geodesic convex in M, where $D \subset \mathbb{R}^n$ is a d-dimensional data manifold. Suppose that $\varphi(D) \subset B_r(p)$ for some $p \in M$, where $B_r(p)$ has sectional curvature upperbounded by $\kappa > 0$. Suppose $r \leq \frac{\pi}{4\sqrt{\kappa}}$. Let $x, y, z \in D$. Consider open neighborhoods U_x, U_y on which φ has Lipschitz constant ℓ_x, ℓ_y, and consider open neighborhood $V_{\gamma_{\varphi(x),\varphi(y)}(t)} \subset \varphi(D)$ on which φ^{-1} has Lipschitz constant $h_{\gamma_{\varphi(x),\varphi(y)}(t)}$. Then as $z \to x$,

$$\|\varphi^{-1}(\gamma_{\varphi(x),\varphi(y)}(t)) - \varphi^{-1}(\gamma_{\varphi(z),\varphi(y)}(t))\| \leq \frac{(1-t)\pi}{2} h_{\gamma_{\varphi(x),\varphi(y)}(t)} \ell_x \|z - x\|_2 + o(\|z - x\|_2).$$

And as $z \to y$

$$\|\varphi^{-1}(\gamma_{\varphi(x),\varphi(y)}(t)) - \varphi^{-1}(\gamma_{\varphi(x),\varphi(z)}(t))\| \leq \frac{\pi}{2} h_{\gamma_{\varphi(x),\varphi(y)}(t)} \ell_y \|z - y\|_2 + o(\|z - y\|_2).$$

If $\frac{\pi}{4\sqrt{\kappa}} \leq r \leq \frac{\pi}{2\sqrt{\kappa}} - \epsilon$, then the RHS of the bounds incur an $O(1/\epsilon)$ factor.

Proof. Let $v = \log_{\varphi(x)} \varphi(y) \in T_{\varphi(x)}M$. Then $\|v\| = d_M(\varphi(x), \varphi(y))$. Then for unit operator $\theta \perp v$, we get

$$\begin{aligned}
\langle R^C(\theta, v)v, \theta\rangle &= d_M(\varphi(x), \varphi(y))^2 \langle R^C(\theta, \hat{v})\hat{v}, \theta\rangle \\
&= d_M(\varphi(x), \varphi(y))^2 \frac{d_M(\varphi(x), \varphi(y))^2}{\|\theta\|^2\|\hat{v}\|^2 - \langle\theta, \hat{v}\rangle} \\
&= d_M(\varphi(x), \varphi(y))^2 K(\theta \wedge \hat{v})
\end{aligned}$$

where $\hat{v} = v/\|v\|$ and $K(\theta \wedge \hat{v})$ is the sectional curvature of the plane spanned by $\theta, \hat{v}$. Hence every eigenvalue λ_i of the operator $\theta \to R^C(\theta, v)v$ satisfies $\lambda_i = d_M(\varphi(x), \varphi(y))^2 K'$, for some sectional curvature K'. In particular, we get $\lambda_{\max} \leq 4r^2\kappa$, where r is the uniform bound on $\|v\|$. Now for $t \in [0, 1]$ and by assumption $\lambda_{\max} > 0$ (positive curvature), we get

$$\begin{aligned}
\beta_{\lambda_{\max}}(t) &= \frac{\sin(\sqrt{\lambda_{\max}}t)}{\sin(\sqrt{\lambda_{\max}})} \\
&\leq \frac{\sin(2rt\sqrt{\kappa})}{\sin(2r\sqrt{\kappa})}
\end{aligned}$$

When $r < \frac{\pi}{4\sqrt{\kappa}}$, we get

$$\frac{\sin(2r\sqrt{\kappa}t)}{\sin(2r\sqrt{\kappa})} \leq \frac{\sqrt{\kappa}t}{(2/\pi)\sqrt{\kappa}} = \frac{\pi t}{2}$$

by Jordan's inequality. Then applying the bound to Theorem 3.4 of (Diepeveen, 2024) gives the desired bound for $z \to x$. The case for when $\frac{\pi}{4\sqrt{\kappa}} \leq r \leq \frac{\pi}{2\sqrt{\kappa}} - \epsilon$ follows from $\sin(2r\sqrt{\kappa} = \sin(\pi - \sqrt{\kappa}\epsilon) = \sin(\sqrt{\kappa}s)$ and taking Taylor expansion.

The proof for the bound on $z \to y$ is analogous.

□

A.3. Proof for Proposition 4.5

Proposition 4.5. Let $(M, \mathfrak{g})$ be a d-dimensional symmetric Riemannian manifold. Let $\varphi : \mathbb{R}^d \to M$ such that $\varphi|_D$ is a diffeomorphism $D \to \varphi(D)$ with $\varphi(D)$ is geodesic convex in M, where D is a d-dimensional data manifold. Let $\mu, \nu \in P_2(\mathbb{R}^n)$ be probability distributions of finite second moment. Let $\rho_{\mu,\nu}(t) = (E_t)_\# \pi$ where $E_t(x, y) = \gamma_{x,y}(t)$ is the evaluation map and $\pi \in \Gamma(\mu, \nu)$ is the optimal coupling w.r.t. distance d_φ induced by the pullback metric $\mathfrak{g}_\varphi = \varphi^* \mathfrak{g}$. Let $\tau \in P_2(\mathbb{R}^n)$ such that $\tau \to \mu$. For any $\sigma \in P_2(\mathbb{R}^n)$, let σ^M denote the pushforward $\varphi_\# \sigma$. Suppose there exists a bounded set $X \subset \varphi(D)$ such that X is a Λ-John domain for $\Lambda > 1$, the support for μ^M, ν^M, τ^M are contained in X, and so are the support for their displacement interpolants in M. Further suppose that ν^M is absolutely continuous w.r.t the Riemannian volume measure on M with bounded density. Then there exists a constant $C > 0$, depending only on curvature of M, X, and the Lipchitzness of φ, such that $W_2(\rho_{\nu,\mu}(t), \rho_{\nu,\tau}(t))^2 \leq CtW_2(\mu, \tau)^{1/6}$.

Proof. For notational simplicity, we assume φ is a bi-Lipchitz map w.r.t. the Euclidean metric on D., with Lipchitz constants L_1, L_2. The case where φ and φ^{-1} only has locally bounded Lipchitz constants can be treated similarly to (Diepeveen, 2024), and the bound incurs additional factors of the Lipchitz constants. Nevertheless, these factors does not change the proposition statement.

By assumption, since ν^M is absolutely continuous w.r.t. the Riemannian volume measure on M with bounded density, on a John domain X, there are unique optimal transport maps $T^M_\tau, T^M_\mu : X \to X$ such that $(T^M_\tau)_\# \nu^M = \tau^M, (T^M_\mu)_\# \nu^M = \mu^M$ (McCann, 2001) (See theorem 8, 9). Further, we let T_μ, T_τ denote the transport maps from $\nu \to \mu, \nu \to \tau$ in D respectively. Invoking Theorem 1.2 of (Kitagawa et al., 2025), we have

$$\int_X d_M(T^M_\mu(x), T^M_\tau(x))^2 d\nu(x) \leq C_1 W_1^M(\mu, \tau)^{1/6} \leq C_1 W_2^M(\mu, \tau)^{1/6}$$

Fix a point $x \in X$ and consider the geodesic $\gamma_{x,T^M_\mu(x)}(t), \gamma_{x,T^M_\tau(x)}(t)$. Now let $\{\Theta^j_x\}_j \subset T_x M, \{\Theta^j_{T^M_\mu(x)}\}_j \subset T_{T^M_\mu(x)} M$ be orthonormal frames that diagonalize the operators

$$\begin{aligned}
\Theta_x &\mapsto R^C(\Theta_x, \log_x(T^M_\mu(x)) \log_x(T^M_\mu(x)) \\
\Theta_{T^M_\mu(x)} &\mapsto R^C(\Theta_{T^M_\mu(x)}, \log_{T^M_\mu(x)}(x) \log_{T^M_\mu(x)}(x)
\end{aligned}$$

with respective eigenvalues $\sigma^j_x, \sigma^j_{T^M_\mu(x)}$. Let $\alpha_\kappa : [0, 1] \to \mathbb{R}$ be defined as

$$\alpha_\kappa(t) = \begin{cases} \frac{\sin(\sqrt{\kappa}\ell)}{\sin(\sqrt{\kappa})}, \kappa > 0 \\ \ell, \kappa = 0, \\ \frac{\sinh(\sqrt{-\kappa}\ell)}{\sinh(\sqrt{-\kappa})}, \kappa < 0 \end{cases}$$

Then by Lemma 3.3 of (Diepeveen, 2024), we have

$$\begin{aligned}
d_M(\gamma_{x,T^M_\mu(x)}(t), \gamma_{x,T^M_\tau(x)}(t))^2 &= d_M\left(T^M_\mu(x), T^M_\tau(x)\right)^2 \sum_{j=1}^d \alpha_{\sigma^j_{T^M_\mu(x)}}(t)^2 \left\langle \frac{\log_{T^M_\mu(x)} T^M_\tau(x)}{\|\log_{T^M_\mu(x)} T^M_\tau(x)\|}, \Theta^j_{T^M_\mu(x)} \right\rangle \\
&\quad + O(d_M\left(T^M_\mu(x), T^M_\tau(x)\right)^3) \\
&\leq d_M\left(T^M_\mu(x), T^M_\tau(x)\right)^2 \alpha_{\sigma_{\max}}(t)^2 + O(d_M\left(T^M_\mu(x), T^M_\tau(x)\right)^3)
\end{aligned}$$

where the last inequality comes from the fact that $\Theta^j_{T^M_\mu(x)}$ is an orthonormal set and $\frac{\log_{T^M_\mu(x)} T^M_\tau(x)}{\|\log_{T^M_\mu(x)} T^M_\tau(x)\|}$ is a unit vector. Now since $\tau \to \mu$, we get the bound

$$d_M(\gamma_{x,T^M_\mu(x)}(t), \gamma_{x,T^M_\tau(x)}(t))^2 \leq d_M\left(T^M_\mu(x), T^M_\tau(x)\right)^2 \alpha_{\sigma_{\max}}(t)^2 + o(d_M\left(T^M_\mu(x), T^M_\tau(x)\right)^2).$$

Table 6. Summary of key hyperparameters used across the main experiments.

Dataset	Oracle	Latent dim.	k	$(\lambda_{rec}, \lambda_{cyc})$	$(\lambda_{dist}, \lambda_{skel})$	Training summary
Toy datasets	Distance	2 / 3	1	$(1.0, 1.0)$	$(0.5, 0.5)$	AE: 100 epochs; VF: 3000 iters
CITE-seq / Multiome	Critic	3	3	$(1.5, 1.5)$	$(0.75, 0.15)$	AE: 100 epochs; VF: 20 epochs
CFD	Distance	2	1	$(1.0, 1.0)$	$(0.5, 0.5)$	AE: 100 epochs; VF: 3000 iters
Mouse erythroid	Distance	2 / 10	1 / 10	$(1.0, 1.0)$ / $(0.5, 0.5)$	$(0.1, 0.1)$ / $(1.0, 1.0)$	AE: 100 epochs; VF: 3000 iters

Then we have

$$\begin{aligned} W_2^M(\rho_{\nu^M,\mu^M}(t), \rho_{\nu^M,\tau^M}(t))^2 &\leq \int_X d_M(\gamma_{x,T_\mu^M(x)}(t), \gamma_{x,T_\tau^M(x)}(t))^2 d\nu(x) \\ &\leq \int_X d_M\left(T_\mu^M(x), T_\tau^M(x)\right)^2 \alpha_{\sigma_{\max}}(t)^2 d\nu(x) \\ &\leq C_2 W_2^M(\mu^M, \tau^M)^{1/6} \end{aligned}$$

where C_2 absorbs the constant of integrating α since ν has bounded density. Then via the pullback we obtain

$$W_2^{d_\varphi}(\rho_{\nu,\mu}(t), \rho_{\nu,\tau}(t))^2 \leq C_2 W_2^M(\mu, \tau)^{1/6}$$

Since by assumption φ is a bi-Lipchitz map w.r.t. to the Euclidean metric on D, we get

$$W_2(\rho_{\nu,\mu}(t), \rho_{\nu,\tau}(t))^2 \leq L_1^{-2} L_2^{1/6} C_2 W_2(\mu, \tau)^{1/6}$$

as desired.

□

B. Experimental and Model Details

In this section we give details of the model (e.g. details on the oracles), experimental details, and additional results. For all experiments, we sample 64 geodesic simplexes per batch and sample 64 points per simplex. All experiments were conducted on a mix of CPUs and NVIDIA A100 GPUs.

B.1. Summary of Key Hyperparameters

For ease of reference, we summarize the main hyperparameters used in our experiments in Table 6.

B.2. Oracle Details

In this section we give more details on the oracles we used for GCL-FM in the experiments.

Critic Oracle. For the critic based oracle, we train a GAN-style critic $\mathcal{C} : \mathbb{R}^d \to [0, 1]$ such that $\mathcal{C}$ assigns higher score to real data points and lower scores to generated data points. We parametrize the critic as a three layer MLP of shape $[256, 128, 64]$ with an additional layer mapping to the score. To help with training the critic, we first train the autoencoder for 30% of the total epochs without L_{skel}^k, L_{cyc}^k and freeze a snapshot of the encoder, obtaining F'. The critic then acts on the encoded data points, i.e. $\Omega(z_{z_i,u_i}) = -\log(\mathcal{C}(F'(z_{z_i,u_i})))$. The critic is trained alongside the autoencoder, where the critic gets a head start for 10 epochs. However, since there are typically a lot more interpolants than real data points in each batch, we additionally append to real data slightly noisy data generated on data from this batch, yielding $x_{real} = \{x_i + \epsilon_i\}$ where ϵ_i is a noisy gaussian vector. The critic is then trained at each epoch along with the autoencoder with the loss

$$-\mathbb{E}[\log(\mathcal{C}(F'(x_{real}))] - \mathbb{E}[\log(1 - \mathcal{C}(F'(z_{u_i,z_i})))]$$

Distance-based Oracle. For the distance-based oracle, we estimated the distance of from the interpolated points in the geodesic skeleton to the vertices. Let $\{x_0, \ldots, x_k\} \in D$, and let $z_i = \varphi(x_i)$. Let $z_{u_i,z_i} \in \sigma(z_0, \ldots, z_k)$. Let D^K denote the K nearest $z \in D$ in terms of $d_D(z, z_{u_i,z_i})$. Then the estimated distance is

$$d^*(G(z_{u,z_i}), x_i) = \min_{x^* \in D^K} \|G(z_{u_i,z_i}), x^*\| + d_D(x^*, x_i)$$

Table 7. W1 distance between generated cells and held-out cells on the Cite-seq and Multiome datasets, viewing the entire trajectory as generated cells. GCL-FM achieves the best performance across both datasets for 50 and 100 dimensions.

Data Dimension	**50**		**100**	
Alg.↓ Dataset→	**Cite**	**Multi**	**Cite**	**Multi**
GAGA (Sun et al., 2025)	23.79 ± 0.83	20.51 ± 1.55	28.43 ± 1.03	28.74 ± 3.15
GCL-FM (Ours)	**18.73** ± 0.48	**18.77** ± 0.49	**22.40** ± 0.21	**24.31** ± 0.32

Note that this is the natural method of adding new points temporary to a shortest graph estimation of manifold distance. In the case of $k = 1$, which is simply a geodesic interpolation, we note that a better distance estimation could be made, namely $d^*(G(z_{u_i,z_i}), x_i) = u_i d_D(x_0, x_1)$. We then match the latent manifold distance to our estimated distance, yielding the oracle $\Omega(z_{u_i,z_i}) = \sum_{i=0}^{k} \big(d_M(z_{(z_i,u_i)}, z_i) - d^*(G(z_{(z_i,u_i)}), x_i)\big)$.

Additional Examples of Oracle. Here we give a few more potential oracles. One such oracle is based on Jensen's inequality, where given a function $g : D \to \mathbb{R}$ that estimates a (geodesic) convex function, the oracle could be $\Omega(z_{u_i,z_i}) = \sum_{i=0}^{k} \max(0, g(z_{u_i,z_i}) - g(z_i)$. Another option is to match instead of instantaneous velocity fields entirely, similar to CurlyFM (Petrović et al., 2025). One could also use a combination of the oracles.

B.3. Visualization on Toy Datasets

Dataset Construction. For the ARCH dataset, we use the construction from (Tong et al., 2020). We sample $x \sim U(0, 1)$ and compute $y = [y_1, y_2]$, where $y_1 = \sin(0.5\pi x) + a_1$ and $y_2 = \sin(0.5\pi x) + a_2$, where $a_1, a_2 \sim \mathcal{N}(0, 0.01)$. We sample a total of $n = 3000$ data points, and hold out the middle 1000 data points for testing/visualizing. For the partial Swiss roll dataset, we sample t uniformly on a grid $t \in [2.5\pi,\ 4.5\pi]$ and drawing an independent height coordinate $h \sim U(0, 10)$. We then compute $x_1 = t\cos t, x_3 = t\sin t$, and form either the 3D data $x = [x_1, h, x_3]$. We scale all coordinates by 0.2 and add i.i.d. Gaussian noise $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$ with $\sigma = 0.1$. Finally, we mean-center the dataset and split the n samples by time order into three contiguous marginals: the first $0.4n$ points, the middle $0.2n$ points (held out), and the last $0.4n$ points. We sample a total of $n = 6000$ data points. The distance on the datasets were computed with isomap.

Model Training. We train an autoencoder in the form of a 3 layer MLP with shape $[256, 128, 64]$, with SpectralNorm, BatchNorm, and SiLU activation. A final projection projecting into latent space of dimension 2 and 3 for the ARCH and Swiss roll datasets respectively. The autoencoder was trained for 100 epochs with learning rate 1e-3 and weight decay 1e-4 on the AdamW optimizer. We use $k = 1$, $\lambda_{dist} = \lambda_{skel} = 0.5$, $\lambda_{rec} = \lambda_{cyc} = 1.0$, and a batch size of 512. The vector field model $v_{\theta,t}$ is trained as a 3 layer MLP with channel size 64, for 3000 iterations with a learning rate of 1e-4 on an Adam optimizer, with a batch size of 256. We used the distance-based oracle to train the autoencoder.

B.4. Experiments on CITE-seq and Multiome Datasets

Dataset Details and Preparation. We use the Cite and Multi datasets from the Multimodal Single-cell Integration Challenge at NeurIPS 2022 (Burkhardt et al., 2022). For preparation of the dataset, we follow (Sun et al., 2025) and evaluate on donor 13176. For the Cite dataset, we combined both train and test inputs to obtain 29394 cells spanning from days 2, 3, 4, 7. For the Multi dataset, we used the train targets to obtain 35396 cells from days 2, 3, 4, 7. For testing, we excluded day 3 and day 4, respectively. The train and test split ratio is 9:1, and the held-out cells were excluded from the training set. We use PHATE (Moon et al., 2019) to compute the distances on the data. Note that in the original setup from (Sun et al., 2025), PHATE was applied to all data points before splitting train and test data. We apply PHATE to only the training data when training our model and reproducing GAGA (Sun et al., 2025).

After selecting the held-out day, we use the remaining days for model fitting and reserve a 9:1 split only within these non-held-out cells for training and validation. The reported W_1 numbers are then computed by comparing generated samples at the held-out time step against the full held-out population for that day. This matches the leave-one-timepoint-out trajectory inference setting studied in the main paper.

Table 8. Additional comparison with FlatVI on two single-cell settings. Lower is better.

Method	EB ↓	ME ↓
FlatVI	41.99 ± 0.04	97.12 ± 0.01
GCL-FM (Ours)	$\mathbf{40.31} \pm 0.41$	$\mathbf{93.99} \pm 0.87$

Table 9. W_1 sensitivity on Cite-seq when varying the common weight on the skeleton regularization terms. Lower is better.

Setting	W_1 ↓
0-GCL-FM (without both losses)	21.53 ± 0.67
without L_{skel}	20.41 ± 0.22
$\lambda_{\text{skel}} = \lambda_{\text{cyc}} = 0.05$	19.80 ± 0.15
$\lambda_{\text{skel}} = \lambda_{\text{cyc}} = 0.15$	19.98 ± 0.55
$\lambda_{\text{skel}} = \lambda_{\text{cyc}} = 0.30$	$\mathbf{19.63} \pm 0.15$
$\lambda_{\text{skel}} = \lambda_{\text{cyc}} = 0.50$	20.13 ± 0.35
$\lambda_{\text{skel}} = \lambda_{\text{cyc}} = 0.75$	20.33 ± 0.21

Model Training. We train an autocoder in the form of a 3 layer MLP with shape $[256, 128, 64]$, with SpectralNorm, BatchNorm, and ReLU activation, and project to a latent space of dimension 3. The autoencoder was trained for 100 epochs with learning rate 1e-3 and weight decay 1e-4 on the AdamW optimizer. We used the critic oracle, where the critic is a 3 layer MLP with batch normalization and ReLU activation, trained with the same learning rate and weight decay as the autoencoder. The model was trained with a batch size of 128. We first train the autoencoder for 30 epochs and then free a snapshot of the encoder for training the critic alongside the autoencoder (see earlier sections for more details). We use a dropout rate of 0.1. We use $\lambda_{rec} = \lambda_{cyc} = 1.5, \lambda_{dist} = 0.75, \lambda_{skel} = 0.15$. For training the vector field model, we use a 3 layer MLP of shape $[256, 128, 64]$ and train for 20 epochs with batch size 128. The model size and number of epochs are identical to (Sun et al., 2025).

Baseline Reproduction. We reproduce GAGA using their official implementation, by computing PHATE on only the training set. For PFM, we again take their official implementation with the same hyperparamaeters and apply the model to this dataset.

Additional Comparison with GAGA. GAGA views the entire sampled trajectory as the generated cells, and computes the W_1 distance with these cells against the held-out set (Sun et al., 2025). To be consistent with prior works, we re-train their model and view the generate cells to be only at the held-out time step. We report these results in Table 2. However, the trajectory view is potentially valid, and we compare this view between GAGA and GCL-FM in Table 7, showing that in this view GCL-FM still achieves better performance.

Additional Comparison with FlatVI. We additionally compare against FlatVI on the two single-cell settings reported in its evaluation protocol. The results in Table 8 show that GCL-FM remains competitive under this additional comparison. As the underlying data in this task is discrete, the generated cells were projected to the nearest integer for each entry. The rest of the hyperparemter settings were identical to the ones used for our single cell experiments, with the only difference being that the latent dimension was set to 10 for consistency with FlatVI.

Sensitivity to Skeleton Loss Weight. We also study the effect of varying the weight assigned to the skeleton regularization terms on Cite-seq. The results in Table 9 show that including these losses consistently improves performance over removing them, and that moderate tuning can further improve over the default setting reported in the main paper.

[1]Geodesic convexity required only for pullback-based methods.

[2]The results reported in (Sun et al., 2025) were evaluated on the entire generated trajectory. To be consistent with the rest of the baselines, we re-train their models and evaluate on the generated data at the held-out timestep. See Appendix B for comparisons with the interpretation that the entire trajectory as the generated cells.

Table 10. Full table with standard deviation. Alignment to reference velocity field (Cos.Dist.) and predicted particle position prediction accuracy (MSE and Prec.@k) on computational fluid dynamics data. GCL-FM achieves higher prediction accuracy while aligning with the reference vector field.

Method	Cos. Dist. ↓	MSE ↓	Prec.@5 ↑	Prec.@10 ↑	Prec.@25 ↑
CFM	0.254 ± 0.003	0.085 ± 0.002	0.079 ± 0.004	0.164 ± 0.006	0.337 ± 0.016
OT-CFM	0.248 ± 0.011	0.095 ± 0.001	0.303 ± 0.001	0.388 ± 0.004	0.496 ± 0.001
CurlyFM	$\mathbf{0.189} \pm 0.027$	0.095 ± 0.003	0.489 ± 0.010	0.522 ± 0.009	0.628 ± 0.010
GCL-FM (ours)	0.197 ± 0.059	$\mathbf{0.077} \pm 0.005$	$\mathbf{0.499} \pm 0.049$	$\mathbf{0.603} \pm 0.031$	$\mathbf{0.683} \pm 0.039$

B.5. Experiments on Computational Fluid Dynamics Data

We show the full dataset with standard deviation included in Table 10.

Dataset Details and Preparation. We use the two-dimensional decaying Taylor-Green vortex (2DTGV) dataset (Toshev et al., 2023). We follow the setup in (Petrović et al., 2025), where we subsampled 2000 particles and considered five evenly spaced out marginals, which are snapshot distributions over particle positions. We used a 80-20 train-test split and computed the distance on the training set using Isomap.

Model Training. We train an autocoder in the form of a 3 layer MLP with shape $[256, 128, 64]$, with SpectralNorm, BatchNorm, and SiLU activation, and project to a latent space of dimension 2. The autoencoder was trained for 100 epochs with learning rate 1e-3 and weight decay 1e-4 on the AdamW optimizer. We used $k = 1, \lambda_{rec} = \lambda_{cyc} = 1.0, \lambda_{dist} = \lambda_{skel} = 0.5$. The model was trained a batch size of 128. Training the vector field model was the same as the one for the toy dataset case, with the only difference being the model was trained for multiple snapshots.

B.6. Experiments on Mouse Erythroid Development Data

Dataset Details and Preparation. We again follow the setup from (Petrović et al., 2025), where a dataset showing mouse gastrulation subset to erythroid lineage was used (Pijuan-Sala et al., 2019). The dataset contains 9,815 cells evolving through five stages, representing the developmental where embryonic cells becomes lineage-specific precursors, evolving into adult organisms. The data was processed using scvelo and unitvelo exactly the same as (Petrović et al., 2025). The velocity fields are computed in the same manner as CurlyFM as well. Given a k-NN velocity estimate f_t at x_t, we form a filtered estimate

$$f_t^* \;=\; \big(1 - w_\gamma(x_t)\big)\, f_t \;+\; w_\gamma(x_t)\, \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, 0.1),$$

where the weight $w_\gamma(x_t) \in [0, 1]$ is a sigmoid function of the gap between the local k-NN distance at x_t and a threshold hyperparameter γ. Distance on the training set is computed using Isomap.

Model Training. For the Euclidean model, we train an autocoder in the form of a 3 layer MLP with shape $[256, 128, 64]$, with SpectralNorm, BatchNorm, and SiLU activation, and project to a latent space of dimension 2 for the $d = 2$ experiment and a latent space of dimension of 10 for the $d = 20, 100$ case. The autoencoder was trained for 100 epochs with learning rate 1e-3 and weight decay 1e-4 on the AdamW optimizer, with a batch size of 512. We use the distance based oracle. For $d = 2$, we use $\lambda_{rec} = \lambda_{cyc} = 1.0, \lambda_{dist} = \lambda_{skel} = 0.1$ and $k = 1$. For $d = 10$, we use $\lambda_{rec} = \lambda_{cyc} = 0.5, \lambda_{dist} = \lambda_{skel} = 1.0$, and $k = 10$. The vector field model $v_{\theta,t}$ is trained as a 3 layer MLP with channel size 64 for $d = 2$ and 256 for $d = 20, 100$, for 3000 iterations with a learning rate of 1e-4 on an Adam optimizer, with a batch size of 256. The number of iteration and size of the vector field model is consistent with CurlyFM.

For the spherical models, the naive model is trained in the exact same way as the Euclidean model. For the project model, we project into the geodesic ball $B_r(p)$ where $r = \frac{\pi}{3\sqrt{\kappa}}$ and $p = [1/\sqrt{\kappa}, 0, \ldots, 0]$, where we use $\kappa \in \{0.25, 0.5, 1.0\}$.

We provide the full mouse erythroid table with standard deviations in Table 11.

Additional Oracle Ablation. We additionally compare the critic-based and distance-based oracles on mouse erythroid data. As shown in Table 12, the distance-based oracle performs better across all three dimensions, supporting its use in the main mouse erythroid experiments.

Table 11. Full mouse erythroid table with standard deviations.

Metric	OT-CFM	MFM	CurlyFM	GCLFM (Spherical + Projection)	GCLFM (Spherical w/o Projection)	GCLFM (Euclidean)
Dimension $d = 2$						
Cos. Dist	0.146 ± 0.001	0.014 ± 0.001	0.009 ± 0.000	$\mathbf{0.003} \pm 0.001$	$\mathbf{0.003} \pm 0.001$	0.005 ± 0.001
L_2	2.704 ± 0.019	1.999 ± 0.014	1.663 ± 0.293	$\mathbf{1.302} \pm 0.192$	1.902 ± 0.177	1.312 ± 0.177
$\mathcal{W}_2$	0.646 ± 0.006	0.269 ± 0.004	0.369 ± 0.090	$\mathbf{0.181} \pm 0.091$	0.361 ± 0.069	1.815 ± 0.021
Dimension $d = 20$						
Cos. Dist	0.489 ± 0.001	0.495 ± 0.001	$\mathbf{0.488} \pm 0.001$	0.498 ± 0.004	0.496 ± 0.007	0.496 ± 0.004
L_2 ($\times 10^3$)	1.885 ± 0.020	1.627 ± 0.040	1.721 ± 0.035	$\mathbf{0.158} \pm 0.022$	0.248 ± 0.088	0.167 ± 0.003
$\mathcal{W}_2$	6.103 ± 0.074	4.855 ± 0.052	6.124 ± 0.027	$\mathbf{4.307} \pm 0.107$	4.927 ± 0.110	4.454 ± 0.071
Dimension $d = 50$						
Cos. Dist	0.490 ± 0.000	0.494 ± 0.000	0.489 ± 0.000	$\mathbf{0.0498} \pm 0.002$	0.498 ± 0.005	0.497 ± 0.002
L_2 ($\times 10^3$)	2.215 ± 0.022	1.971 ± 0.023	2.045 ± 0.073	$\mathbf{0.160} \pm 0.022$	0.262 ± 0.155	0.188 ± 0.011
$\mathcal{W}_2$	7.969 ± 0.029	$\mathbf{6.727} \pm 0.022$	7.729 ± 0.046	7.167 ± 0.098	7.691 ± 0.099	7.317 ± 0.041

Table 12. Oracle ablation on mouse erythroid data using $\mathcal{W}_2$. Lower is better.

Oracle	$d = 2$	$d = 20$	$d = 100$
Critic oracle	0.185 ± 0.040	4.914 ± 0.191	7.921 ± 0.173
Distance oracle	$\mathbf{0.182 \pm 0.021}$	$\mathbf{4.454 \pm 0.071}$	$\mathbf{7.317 \pm 0.041}$

Impact Statement

Generative models that can generate on unknown data manifolds, especially for trajectory inference, can significantly enhance the societal impact of machine learning applications in areas such as biological sciences (e.g. stem cell research), robotics, fluid dynamics, etc. Our method further reduces computational resources required and thereby reducing the carbon footprint. Nonetheless, while our work is mostly focusing on advancing the field of machine learning, generative models in general also presents potential negative risks, including generation of harmful contents or sensitive domains.