Geometry-Aware Generative Autoencoders for Warped Riemannian Metric Learning and Generative Modeling on Data Manifolds

Xingzhi Sun[*♡] **Danqi Liao**[*♡] **Kincaid MacDonald**[♡] **Yanlei Zhang**[♢] **Guillaume Huguet**[♢]
Guy Wolf[♢] **Ian Adelstein**[♡†] **Tim G. J. Rudner**[♣†] **Smita Krishnaswamy**[♡♢†]
[*]Equal contribution. [†]Corresponding authors.
[♡]Yale University [♠]New York University [♢]Mila - Quebec AI Institute and Universite de Montréal

Abstract

Rapid growth of high-dimensional datasets in fields such as single-cell RNA sequencing and spatial genomics has led to unprecedented opportunities for scientific discovery, but it also presents unique computational and statistical challenges. Traditional methods struggle with geometry-aware data generation, interpolation along meaningful trajectories, and transporting populations via feasible paths. To address these issues, we introduce Geometry-Aware Generative Autoencoder (GAGA), a novel framework that combines extensible manifold learning with generative modeling. GAGA constructs a neural network embedding space that respects the intrinsic geometries discovered by manifold learning and learns a novel *warped* Riemannian metric on the data space. This warped metric is derived from both the points on the data manifold and negative samples off the manifold, allowing it to characterize a meaningful geometry across the entire latent space. Using this metric, GAGA can uniformly sample points on the manifold, generate points along geodesics, and interpolate between populations across the learned manifold. GAGA shows competitive performance in simulated and real-world datasets, including a 30% improvement over SOTA in single-cell population-level trajectory inference.

1 INTRODUCTION

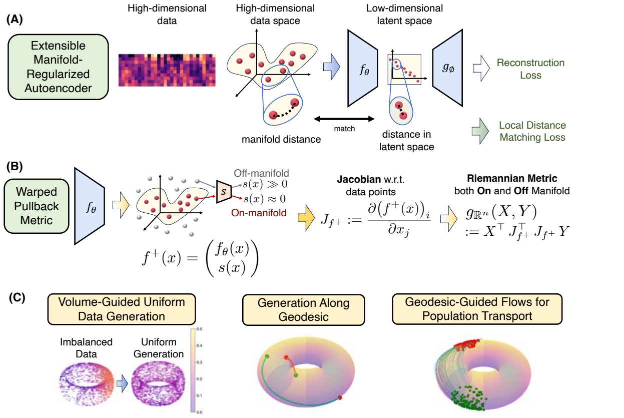
(A)
High-dimensional data
High-dimensional data space
Low-dimensional latent space
Extensible Manifold-Regularized Autoencoder
f_θ
g_ϕ
Reconstruction Loss
manifold distance
match
distance in latent space
Local Distance Matching Loss
(B)
Warped Pullback Metric
f_θ
Off-manifold
$s(x) \gg 0$
$s(x) \approx 0$
On-manifold
s
$f^+(x) = \begin{pmatrix} f_\theta(x) \\ s(x) \end{pmatrix}$
Jacobian w.r.t. data points
$J_{f^+} := \frac{\partial (f^+(x))_i}{\partial x_j}$
Riemannian Metric both **On** and **Off** Manifold
$g_{\mathbb{R}^n}(X, Y)$
$:= X^\top J_{f^+}^\top J_{f^+} Y$
(C)
Volume-Guided Uniform Data Generation
Imbalanced Data
Uniform Generation
Generation Along Geodesic
Geodesic-Guided Flows for Population Transport
0.5
0.4
0.3
0.2
0.1
0.0

Figure 1: The Geometry-Aware Generative Autoencoder (GAGA) framework. **(A)** Training the networks. **(B)** Obtaining the warped pullback metric. **(C)** Challenging applications enabled by GAGA.

Recent scientific discoveries are increasingly driven by the analysis of high-dimensional data across various fields, including single-cell RNA sequencing (scRNA-seq), spatial genomics, and many others (Jindal et al., 2018; Van de Sande et al., 2023; Wang et al., 2022; Zhao et al., 2022; Sun et al., 2024). These high-dimensional datasets offer unprecedented opportunities to explore complex physical and biological systems, but they also pose unique computational and statistical challenges.

❶ First, it is difficult to generate new data points that faithfully follow the underlying data geometry (for example, to combat inconsistent or undersampling in parts of the data manifold) in the absence of explicit analytical forms describing the data, especially when data imbalance complicates the process (Krawczyk, 2016). ❷ Second, interpolating between two samples along a meaningful trajectory, which is valuable for understanding transitions such as developmental progressions, remains challenging due to the complex and nonlinear structure of the data (Aggarwal et al., 2001). ❸ Third, aligning or transporting populations across different experimental conditions, time points, or biological states is a fundamental challenge, as tradi-

Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258. Copyright 2025 by the author(s).

tional matching methods often fail to capture the complex dependencies and interactions inherent in high-dimensional spaces (Martínez-Minaya et al., 2018).

When working with high-dimensional data, it is useful to consider the manifold hypothesis, which posits that such data often reside on a lower-dimensional manifold embedded within the high-dimensional data space (Fefferman et al., 2016). Building on this foundation, we propose a novel framework called the Geometry-Aware Generative Autoencoder (GAGA) to simultaneously address all three challenges.

GAGA combines the power of extensible manifold learning with generative modeling. It first learns a generalizable neural network embedding space that respects the geometries discovered by non-linear dimensionality reduction techniques (Figure 1 panel A). Then, it derives a novel *warped* pullback metric on the original data space (Figure 1 panel B). Uniquely, this metric is created as much by points not in the dataset as by points that are in the data. The warped metric is learned by embedding negative samples **off** the manifold and points **on** the manifold far away from each other in the latent space. This creates an implicit penalty for data generation and geodesic computations, effectively nudging geodesics to stay within the data density, and generated points to stay within dimensions of the data.

Using this learned warped Riemannian metric, GAGA can ❶ generate data across the data manifold guided by local volume, ❷ interpolate between two points along the manifold geodesics, and ❸ transport populations along these geodesics. These applications are illustrated in Figure 1 panel C, and are described in Sections 3.2, 3.3, and 3.4, respectively. In this way, GAGA effectively addresses the challenges of geometry-aware data generation, interpolation, and population transport within a unified framework.

In summary, our main contributions are as follows:

1. Designing a geometry-aware generative autoencoder that combines manifold learning with generative modeling.
2. Proposing a novel *warped* pullback metric to create a meaningful geometry on the entire data space, allowing GAGA to stay on the manifold when generating points.
3. Introducing a new generative method that leverages the learned Riemannian pullback metric to achieve uniform sampling from the data manifold, interpolating data along geodesics, and transporting populations along geodesic paths.
4. Demonstrating that the proposed methods work well on both simulated and real biological data.

2 BACKGROUND

Manifold Learning The *Manifold Hypothesis* states that data often lie *on* or *near* a low-dimensional manifold within high-dimensional space (Fefferman et al., 2016). Manifold learning methods such as Diffusion Maps (Coifman and Lafon, 2006), PHATE (Moon et al., 2019), DSE (Liao et al., 2024), DYMAG (Bhaskar et al., 2023), CUTS (Liu et al., 2024a), and HeatGeo (Huguet et al., 2024) use diffusion probabilities to recover the geometry of the manifold despite the sparsity and noise in the data. For details, see Appendix A.

Riemannian Manifolds and Metrics An n-dimensional manifold $\mathcal{N}$ is a space locally resembling $\mathbb{R}^n$, and a Riemannian metric g endows each tangent space $T_x\mathcal{N}$ with an inner product $g_x(X,Y) = X^T g(x) Y$, with $g(x)$ an $n \times n$ matrix. The length of a tangent vector X is $\|X\| = \sqrt{g_x(X,X)}$, and for a smooth curve $c : [0,T] \to \mathcal{N}$ the length is $L(c) = \int_0^T \sqrt{g_{c(t)}\big(\dot{c}(t), \dot{c}(t)\big)}\, dt$. If $\mathcal{N}$ is parametrized by $f(z)$, $z \in \mathcal{D}$, its volume is given by $\int_{\mathcal{D}} \sqrt{\det g(x)}\, dx$.

A key component of our method is the *Riemannian pullback metric*. Given a map $f : \mathcal{M} \to (\mathcal{N}, g)$, its differential $df_x : T_x\mathcal{M} \to T_{f(x)}\mathcal{N}$ allows us to define $f^*g(X,Y) = g(df_xX, df_xY)$, which equips $\mathcal{M}$ with the geometry inherited from $(\mathcal{N}, g)$. For a detailed discussion, see Appendix B.

3 METHODS

In this section, we will describe the autoencoder and derive the Riemannian pullback metric (Section 3.1). Then, we will show solutions to the three challenges: geometry-aware data generation (Section 3.2), interpolation along meaningful trajectories (Section 3.3), and population transport (Section 3.4). Proofs for all lemmas and propositions are provided in Appendix E.

3.1 Geometry-Aware Encoding for Both On-Manifold and Off-Manifold Points

We first train an autoencoder to learn a latent space whose local Euclidean distances correspond to the data manifold distances. These distances can be obtained from many existing manifold-learning techniques, including PHATE and HeatGeo. We then derive a *warped* metric on data space that allows us to produce a pullback Riemannian metric on the data manifold and impose large distances for points off the manifold. This warped metric enables us to compute on-manifold geodesics for data generation in later sections.

The following result from Riemannian geometry states that by matching data manifold distances in latent space (i.e., learning a local isometry), we construct the desired pullback metric on the data manifold.

Proposition 3.1. *For Riemannian manifolds $(\mathcal{M}, g_{\mathcal{M}}), (\mathcal{N}, g_{\mathcal{N}})$ and diffeomorphism $f : \mathcal{M} \to \mathcal{N}$, if f is a local isometry, i.e., there exists $\epsilon > 0$, such that for any $x_0, x_1 \in \mathcal{M}, d_{\mathcal{M}}(x_0, x_1) < \epsilon \implies d_{\mathcal{M}}(x_0, x_1) = d_{\mathcal{N}}(f(x_0), f(x_1))$, then we have $g_{\mathcal{M}} = f^* g_{\mathcal{N}}$.*

To implement this construction, we define an autoencoder consisting of an encoder f_θ and a decoder h_ϕ, both parameterized by neural networks. The autoencoder is jointly optimized with a reconstruction objective (Eqn. (1)) and a local distance matching objective (Eqn. (2)).

$$\mathcal{L}_{\text{Recon}}(\theta, \phi) = \frac{1}{N} \sum_{i=1}^{N} ||x_i - h_\phi(f_\theta(x_i))||_2^2 \tag{1}$$

$$\mathcal{L}_{\text{Dist}}(\theta) = \frac{1}{N} \sum_{i<j} e^{-\zeta d(x_i, x_j)} \ell_{\text{SE}}(x_i, x_j, \theta), \tag{2}$$

where

$$\ell_{\text{SE}}(x_i, x_j, \theta) = \left(||f_\theta(x_i) - f_\theta(x_j)||_2 - d(x_i, x_j) \right)^2, \tag{3}$$

$x_1, \ldots, x_N$ are the data samples, and $d(x_i, x_j)$ is the manifold distance between points x_i and x_j obtained via selected manifold-learning methods. The hyperparameter $\zeta > 0$ and the term $e^{-\zeta d(x_i, x_j)}$ weigh the penalty towards the more important local geometry of the data manifold.

In summary, we minimize the following objective (Eqn. (4)) with respect to encoder and decoder parameters θ and ϕ to obtain geometry-aware embeddings.

$$\mathcal{L}(\theta, \phi) = \lambda_1 \mathcal{L}_{\text{Dist}}(\theta) + \lambda_2 \mathcal{L}_{\text{Recon}}(\theta, \phi) \tag{4}$$

This objective balances distance matching and reconstruction with hyperparameters λ_1, λ_2. It results in an embedding that matches the data geometry and retains the information needed to reconstruct the data.

Pullback metric Next we show how to compute the pullback metric via the Jacobian of the encoder. The pullback (via the encoder) of the Euclidean metric from latent space yields a non-Euclidean data space metric, capturing local distances on the data manifold.

Definition 3.1. *The pullback of the Euclidean metric from latent space to the data manifold $\mathcal{M}$ is defined by $g_{\mathcal{M}}(X, Y) := X^\top J_f^\top J_f Y$, where $X, Y \in T_x\mathcal{M}$ are tangent vectors at $x \in \mathcal{M}$, $J_f := \partial f_\theta(x)_i / \partial x_j$ is the Jacobian of f_θ at x.*

Warping the Local Euclidean Metric Although the construction above produces a pullback metric on the entire data space, it is only accurate near the training data, i.e., along the data manifold. For points off of the manifold, we the local Euclidean metric to create large distances between on-and-off manifold points. In order to achieve this, we create a special embedding for both on-manifold points x_i and off-manifold points $\check{x}_i$. These points are embedded in a latent space with an auxiliary dimension, where the value of that dimension represents the deviation from the manifold: it is nearly zero for on-manifold points and large for off-manifold points.

Suppose we have a function $s(x)$ such that $s(x) \approx 0$ for x on the manifold, and $s(x)$ increase as x moves away from the manifold. Let

$$f^+(x) = \begin{pmatrix} f_\theta(x) \\ \beta s(x) \end{pmatrix}, \text{where } \beta \text{ is a hyperparameter} \tag{5}$$

Definition 3.2. *The pullback of the warped local Eulidean metric on the full space $\mathbb{R}^n$ is defined by $g_{\mathbb{R}^n}(X, Y) := X^\top J_{f^+}^\top J_{f^+} Y$, where $X, Y \in T_x\mathbb{R}^n$ are tangent vectors at $x \in \mathbb{R}^n$, $J_{f^+} := \partial(f^+(x))_i / \partial x_j$ is the Jacobian of f^+ at x.*

Points off the manifold, where $s(x)$ is large, are placed into an extended dimension of latent space, far from the on-manifold points. Formally, we have:

Lemma 3.2. *If there exists $\alpha \in \mathbb{R}$ such that for any $x, \check{x}, \alpha||x - \check{x}|| \leq |s(x) - s(\check{x})|$. Then for any $x, \check{x}, ||f^+(x) - f^+(\check{x})|| \geq \alpha\beta||x - \check{x}||$. Furthermore, denoting $\mathcal{D}_{\mathcal{M}}(y) := \inf_{x \in \mathcal{M}} ||x - y||$ and $\mathcal{D}_{f^+(\mathcal{M})}(y) := \inf_{x \in \mathcal{M}} ||f^+(x) - f^+(y)||$, then for any $\check{x}$, we have $\mathcal{D}_{f^+(\mathcal{M})}(\check{x}) \geq \alpha\beta\mathcal{D}_{\mathcal{M}}(\check{x})$.*

This lemma assumes that s satisfies a Lipschitz condition, meaning it grows moderately. In our framework, s is a Wasserstein-GAN-style discriminator, and we enforce Lipschitz continuity via weight clipping and spectral normalization (see Appendix C.1). This approach is supported by WGAN (Arjovsky et al., 2017, Section 3), spectral normalization (Miyato et al., 2018, Section 2.1), and Lipschitz discriminators (Tong et al., 2022, Section 3). In practice, we obtain such function $s(x)$ by training a discriminator with negative sampling. See Appendix C for details.

Note that $g_{\mathcal{M}}$ is defined only on the tangent space of $\mathcal{M}$, whereas the warping allows $g_{\mathbb{R}^n}$ to be defined on the tangent space of the entire data space $\mathbb{R}^n$.

3.2 Using the Learned Pullback Metric to Generate Uniformly on the Manifold

Tackling Challenge 1: *Volume-Guided Generation.*

Here we present a method for sampling uniformly across the data manifold. Notice that this method

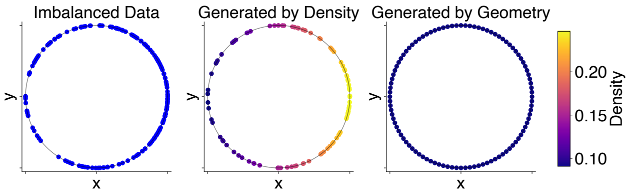

Figure 2: Density-based vs geometry-based generation. Left: Data has sampling imbalance. Middle: Density-based methods, e.g. Diffusion Model and Flow Matching, maintain this bias. Right: Geometry-aware generation alleviates imbalance by generating points uniformly across the manifold.

of generation is **markedly different** from generative methods that match distributions (and practically mainly the modes of the distribution) such as GANs and diffusion models. Here, rather than sampling from a *probability distribution*, we sample from the *geometry* or the shape of the data evenly. To do this, we utilize the pullback metric that we defined in the previous section to create a volume element that is useful for generation. By utilizing this learned metric, GAGA enables us to correct for sampling biases and imbalances, ensuring uniform coverage of the manifold during data generation. See Figure 2 for an illustration of this difference.

We begin by defining the volume distribution function, which represents a uniform distribution on the manifold based on its intrinsic geometry.

Definition 3.3. *Let $g_{\mathcal{M}}$ be the Riemannian metric, of the manifold, define the* volume distribution function $p_{vol}(x) = \frac{1}{Z}\sqrt{\det g_{\mathcal{M}}(x)}$*, where $Z = \int_{x \in \mathcal{M}} \sqrt{\det g_{\mathcal{M}}(x)} dx$, as the normalized volume element normalized to sum to 1. The corresponding probability distribution is defined as the uniform distribution on the manifold.*

The intuition behind Definition 3.3 can be illustrated with the example shown in Figure 3. Consider a spiral, which is a one-dimensional manifold. In this case, points are uniformly distributed along the spiral such that the curve lengths between adjacent points are equal. This is achieved by placing more points where the curve length (i.e., volume) is larger, ensuring that the point density remains consistent along the entire manifold. Essentially, the number of points per unit curve length remains constant, which makes the point density proportional to the volume element.

Next, we propose an algorithm for generating uniformly on the manifold using Langevin dynamics, combined with the pullback metric learned by GAGA. Our approach leverages Langevin dynamics to sample points while following the volume distribution function derived from the pullback metric, ensuring that generated points remain faithful to the manifold's intrinsic geometry. Specifically, we solve the following stochastic differential equation (SDE).

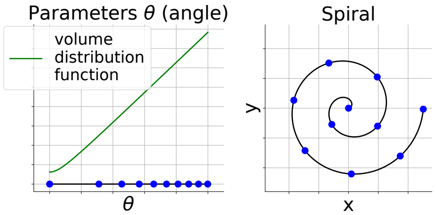

Figure 3: Demonstration of uniform sampling on a spiral (a 1D manifold). Left: In the space parameterized by polar angle, data (blue points) are distributed with density proportional to the volume distribution function (green curve), and may appear non-uniform. Right: In fact, corresponding data on the manifold (blue points) are equally spaced w.r.t. geodesic distance, and are therefore "uniformly distributed".

$$
\begin{aligned}
dX_t &= -\nabla f_{\text{target}}(X_t)dt + \sqrt{2}dW_t \\
f_{\text{target}}(x) &= \lambda s(x) - \log(f_{vol}(x)),
\end{aligned} \tag{6}
$$

where W_t represents Brownian motion. $f_{vol}(x) := \left|\prod_{i=1}^{d} \sigma_i(x)\right|$, and $\sigma_i(x), i = 1, \ldots, d$ are the singular values of the Jacobian matrix $J_f(x)$. This corresponds to the volume distribution function defined in Definition 3.3 (up to a normalization factor). By multiplying the d singular values, we obtain the square root of the pseudo-determinant, since J_f has rank d, thereby avoiding degeneracy. The function s, as used in Eqn. (5), is designed to be close to 0 on the manifold and increases as x moves away from the manifold, and its gradient will pull the generated points towards the manifold. In practice, we use a Gaussian process to obtain s, as described in Appendix C.2. The hyperparameter $\lambda > 0$ controls the balance between the volume distribution function and the manifold constraint. In practice, we discretize this process using the Unadjusted Langevin Algorithm (ULA).

Proposition 3.3. *Suppose $f_{\text{target}}(x) = \lambda s(x) - \log(f_{vol})(x)$ is α-strongly convex for some constant $\alpha > 0$, i.e. $\nabla^2 f(x) \succeq \alpha I$, then the distribution of X in Eqn. (6) converges exponentially fast in Wasserstein distance to a distribution supported on the data manifold, whose restriction on the manifold is proportional to the volume distribution function.*

Proposition 3.3 demonstrates an exponential convergence rate of volume-guided generation in Wasserstein distance. Additionally, in Appendix F, we present a proposition establishing exponential convergence rates in total variation distance.

Algorithm 1 Volume-Guided Generation

```
Input: $s(x), f_{vol}(x)$, initial sample $\mathbf{x}_0$, step size $\eta$, number of steps $N$, threshold $\epsilon$
Output: Filtered final sample $\mathbf{x}_{N,\text{filtered}}$
Initialize $\mathbf{x} \leftarrow \mathbf{x}_0$
for $t = 1$ to $N$ do
    Sample Gaussian noise $\boldsymbol{\xi}_t \sim \mathcal{N}(0, I)$
    $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \eta\nabla(\lambda s(x) - \log(f_{vol}(x))) + \sqrt{2\eta} \cdot \boldsymbol{\xi}_t$
end for
$\mathbf{x}_{N,\text{filtered}} \leftarrow \{\mathbf{x} \in \mathbf{x}_N : s(\mathbf{x}) < \epsilon\}$
Return $\mathbf{x}_{N,\text{filtered}}$
```

3.3 Generating along Manifold Geodesics

Tackling Challenge 2: *On-Manifold Interpolation.*

We now turn to the problem of generating the geodesic between a pair of points on the data manifold. This is useful when points in a manifold could represent the time evolution of a system, such as in single cell sequencing. It has been shown that such data usually follow the manifold hypothesis (Moon et al., 2018), and that geodesic generation can model cellular trajectories such as those taken during differentiation.

One could try to find the curve which minimizes length with respect to the metric $g_{\mathcal{M}}$. However, this metric is only accurate on the manifold, and such shortest paths might cut through data space. Indeed, we need to minimize length under the condition that the curve stays on the manifold. The main result of this section shows that this constrained optimization problem is actually solved by minimizing arc length with respect to the warped metric $g_{\mathbb{R}^n}$. Intuitively, this metric imposes large penalties for deviating from the manifold, as off-manifold points are embedded into the dimension-extended latent space, forcing the shortest path onto the manifold.

We begin with a neural-network parameterized interpolation curve. for any $x_0, x_1 \in \mathcal{M}$, we define a neural network-parameterized interpolation curve $c_\eta(x_0, x_1, \cdot) : [0,1] \to \mathbb{R}^n$ satisfying $c_\eta(x_0, x_1, 0) = x_0, c_\eta(x_0, x_1, 1) = x_1$. More details on parameterization are provided in Appendix D. We minimize

$$\mathcal{L}_{\text{Geo}}(\eta, x_0, x_1) = \frac{1}{M}\sum_{m=1}^{M} g_{\mathbb{R}^n}(\dot{c}_\eta, \dot{c}_\eta)(x_0, x_1, t_m) \quad (7)$$

where $0 = t_0 < t_1 < ... < t_M = 1$ are sampled time points. Note that Eqn. (7) is a discretization of the integral $\int_0^1 g_{\mathbb{R}^n}(\dot{c}_\eta, \dot{c}_\eta)(x_0, x_1, t)dt$. In Do Carmo and Flaherty (1992), this is defined as the energy of the curve, and minimizing the energy is equivalent to minimizing the curve length (Do Carmo and Flaherty, 1992, Chapter 9, Proposition 2.5).

The following proposition demonstrates that geodesic computation on $\mathcal{M}$ can be achieved by minimizing arc length with respect to the metric $g_{\mathbb{R}^n}$.

Lemma 3.4. *Assume that the ω-thickening of the manifold $\mathcal{M} \subset \mathbb{R}^n$, defined as $\mathcal{M}^\omega := \{x \in \mathbb{R}^n : \inf_{m \in \mathcal{M}} d(x, m) < \omega\}$, (i.e., the set of points whose distance from $\mathcal{M}$ is less than ω) maps into a subset of the ϵ-thickening of $f(\mathcal{M})$. Here, the ϵ-thickening is defined analogously, with ϵ chosen such that for every $x \in f(\mathcal{M})$, the ball $B_\epsilon(x) := \{y \in \mathbb{R}^n : \|y - x\| < \epsilon\}$ intersects $f(\mathcal{M})$ in exactly one connected component.*

Then, for any smooth curve $c : [0,1] \to \mathbb{R}^n$ connecting x_0 and x_1 (i.e., $c(0) = x_0$ and $c(1) = x_1$), there exists a smooth curve $c' : [0,1] \to \mathcal{M}$ lying entirely ***on the manifold*** *(with $c'(0) = x_0$ and $c'(1) = x_1$) such that $\mathcal{L}_{Geo}(c') \leq \mathcal{L}_{Geo}(c) - \alpha^2\beta^2\frac{1}{M}\sum_{m=1}^{M}\big(\mathcal{D}_{\mathcal{M}}(c(t_m)) - \mathcal{D}_{\mathcal{M}}(c(t_{m-1}))\big)^2 + \xi$. Here, α is defined as in Lemma 3.2, $\mathcal{D}_{\mathcal{M}}$ denotes the distance from a point to $\mathcal{M}$ (also as in Lemma 3.2), and ξ is a fixed positive constant independent of x_t and β.*

Proposition 3.5. *When $\mathcal{L}_{Geo}$ is minimized, $\max_{m=1,\ldots,M} \mathcal{D}_{\mathcal{M}}(c(t_m)) \leq \sqrt{\xi}/(\alpha\beta)$, i.e., for sufficiently large β, $c(t)$ is close to the manifold with a maximum distance of $\sqrt{\xi}/(\alpha\beta)$. Furthermore, let $c'(t)$ be a geodesic between x_0 and x_1 under the metric $g_{\mathcal{M}}$, we have $\frac{1}{M}\sum_{m=1}^{M} g_{\mathcal{M}}(\dot{c}, \dot{c})(x_0, x_1, t_m) \leq \frac{1}{M}\sum_{m=1}^{M} g_{\mathcal{M}}(\dot{c}', \dot{c}')(x_0, x_1, t_m) + \xi'\sqrt{\xi}/(\alpha\beta)$ for some positive constant ξ'. That is, c approximately minimizes the energy (and hence curve length) under $g_{\mathcal{M}}$.*

Lemma 3.4 shows that for any curve connecting two points on the manifold, there exists an alternative curve where the loss difference is controlled by the difference in their distances from the manifold. Proposition 3.5 then uses this result to establish that a necessary condition for minimizing the loss is that the curve remains sufficiently close to the manifold. Moreover, it demonstrates that the minimizer's energy is nearly equal to that of the true geodesic. Thus, the minimizer is (approximately) 1) on the manifold and 2) of minimal length, and is therefore the geodesic.

In summary, minimizing Eqn. (7) yields the geodesic on $\mathcal{M}$ between x_0 and x_1 with respect to the pullback metric $g_{\mathcal{M}}$. This is achieved by minimizing the curve length under the warped pullback metric.

3.4 Population Interpolation along geodesics

Tackling Challenge 3: *Population Transport.*

In the previous section, we achieved point-wise geodesic computation, learning the geodesic between a pair of points. More generally, we aim to generate population-level geodesics. Given two distributions on the manifold, we want to generate geodesics be-

Algorithm 2 Geodesic-Guided Flow Matching

```
    **Input:** Starting and ending populations $\mathcal{X}, \mathcal{Y}$, encoder $f$, dimension-extended encoder $r$, $t = (t_1, ..., t_M)$
    **while** Training **do**
        **Sample batches of size $b$ *i.i.d.* from the datasets**
        Sample $\{x_1, ..., x_l\} \subset \mathcal{X}, \{y_1, ..., y_l\} \subset \mathcal{Y}$
        $\mu \leftarrow \frac{1}{l}\sum_{i=1}^{l} I(x = x_i), \nu \leftarrow \frac{1}{l}\sum_{i=1}^{l} I(x = y_i)$
        $\pi^* = \underset{\pi \sim \Gamma(\mu,\nu)}{\arg\min} \left(\frac{1}{l}\sum_{i=1}^{l} \pi(x_i', y_i')||f(x_i') - f(y_i')||^2\right)^{1/2}$
        Sample $(x_{j_1}, y_{j_1}), ..., (x_{j_l}, y_{j_l}) \overset{i.i.d.}{\sim} \pi^*$
        **Compute geodesic and velocity-matching losses**
        $L \leftarrow \frac{1}{l}\sum_{i=1}^{l}(\lambda_3 \mathcal{L}_{\text{geo}}(\eta, x_{j_i}, y_{j_i}) + \lambda_4 \mathcal{L}_{\text{FM}}(\nu, \eta, x_{j_i}, y_{j_i}))$
        $\eta, \nu \leftarrow \text{GradientDescentUpdate}(\eta, \nu, \nabla L)$
    **end while**
    **Output:** $\nu$
```

tween populations sampled from these distributions, minimizing the expected total length of the geodesics. This equates to solving the dynamical optimal transport problem (Tong et al., 2020; Benamou and Brenier, 2000), where the cost is the curve length on the manifold.

To solve this, we first find the optimal pairing of points from the starting and ending distributions to minimize total geodesic length, then compute those geodesics. To generalize to new points, we learn a vector field matching the time derivatives (speed) of the geodesics. Given a point sampled from the first distribution, we can generate the geodesic by integrating the vector field starting from the point.

Specifically, we define a neural network $v_\nu(x_0, t) \in \mathbb{R}^n$, and the flow matching loss for any joint distribution π and curve c_η as the following.

$$\begin{aligned} &\mathcal{L}_{\text{FM}}(\nu, \eta, x_0, x_1) \\ &= \mathbb{E}_{\pi(x_0,x_1)}||v_\nu(t, x_0) - \frac{d}{dt}c_\eta(t, x_0, x_1)||^2 \end{aligned} \tag{8}$$

When this loss is minimized, v_ν is the vector field that matches the time derivatives of the curves.

In each training step, we sample starting and ending points from the two distributions, and solve the optimal transport problem where the ground distance is the Euclidean distance in the latent space. This optimal transport plan π would minimize the total geodesic length between $(x_0, x_1) \sim \pi$, because GAGA is trained so that the Euclidean distance in the latent space is matched to the geodesic distance on the data manifold. We then parameterize the interpolation curves c_η as in Section 3.3, and minimize the following loss which balances the loss Eqn. (7) that the c_η are the geodesics, and the aforementioned flow matching loss (Eqn. (8)).

$$\begin{aligned} &\mathcal{L}_{\text{GFM}}(\nu, \eta, x_0, x_1) \\ &= \lambda_3 \mathcal{L}_{\text{geo}}(\eta, x_0, x_1) + \lambda_4 \mathcal{L}_{\text{FM}}(\nu, \eta, x_0, x_1) \end{aligned} \tag{9}$$

Further details are provided in Algorithm 2.

After training, we generate the geodesics by integrating the vector field v_η. Given an initial point x_0, we can generate points along the geodesics starting from it with $x(t) = x_0 + \int_0^t v_\nu(x_0, \tau)d\tau$.

Finally, the following proposition shows that our method generates desired population-level geodesics.

Proposition 3.6. *Given starting and ending distributions p, q, at the convergence of Algorithm 2,*

$$x(t) = x_0 + \int_0^t v_\nu(x_0, \tau)d\tau, x_0 \sim p, t \in [0, 1]$$

are geodesics between the two distributions following the optimal transport plan that minimizes the total expected geodesic lengths.

4 EMPIRICAL RESULTS

Geometry-aware Autoencoder First, we empirically show that GAGA preserves manifold distances of data in the latent space by evaluating GAGA on Splatter (Zappia et al., 2017), a synthetic single-cell RNA sequence dataset.

Single-cell RNA sequence data are high-dimensional, noisy, and sparse and have been demonstrated to reside on low-dimensional manifolds, making them ideal datasets for evaluating our method (Heimberg et al., 2016; Moon et al., 2018).

The encoder was evaluated by Denoised Embedding Manifold Preservation (DEMaP) described in (Moon et al., 2019) which measures the correlation between Euclidean distances in latent space and ground truth manifold distances in data space.

The results show that our distance matching loss is important for preserving manifold distances, as evidenced by higher DEMaP scores averaged across different noise levels (Table 1).

GAGA can also effectively reconstruct high-dimensional features through the decoder. See Appendix G.1 for results on reconstruction, details on the Splatter dataset, and our evaluation criteria.

Volume-guided Generation on Manifold We assessed the effectiveness of volume-guided generation on both simulated and real data.

We first illustrated our method on three toy datasets: hemisphere saddle, and paraboloid. On these manifolds, the volume element is known, which we used as ground truth. We evaluated the generation by computing the kernel density estimation and comparing it with the ground truth (see Appendix G.2.2 for details).

Table 1: Average DEMaP for Autoencoders (AE) and GAGA on simulated single-cell datasets over different noise settings.

	Objective	Cellular State Space	DEMaP (↑)
AE	$\mathcal{L}_{\text{Recon}}$	Clusters	$0.347_{\pm 0.117}$
GAGA	$\mathcal{L}_{\text{Recon}}, \mathcal{L}_{\text{Dist}}$	Clusters	$\mathbf{0.645}_{\pm 0.195}$
AE	$\mathcal{L}_{\text{Recon}}$	Trajectories	$0.433_{\pm 0.135}$
GAGA	$\mathcal{L}_{\text{Recon}}, \mathcal{L}_{\text{Dist}}$	Trajectories	$\mathbf{0.600}_{\pm 0.191}$

We generate imbalanced data by sampling from Gaussian distribution in the parameter space. In Figure 4 (B,C,D) and Table 2 we show that the densities of the points generated by GAGA are closer to the ground truth volume elements compared to the original data points, indicating that GAGA largely reduces data imbalance. In addition, Figure 4 (A) shows that the generated points stay on the data manifold and cover the sparse regions well in the original data. The complete result figure can be found in Appendix H.3. We also compared our method with Riemannian Flow Matching Chen and Lipman (2023) in Appendix H.4 to demonstrate the faithfulness of generated points to the data geometry.

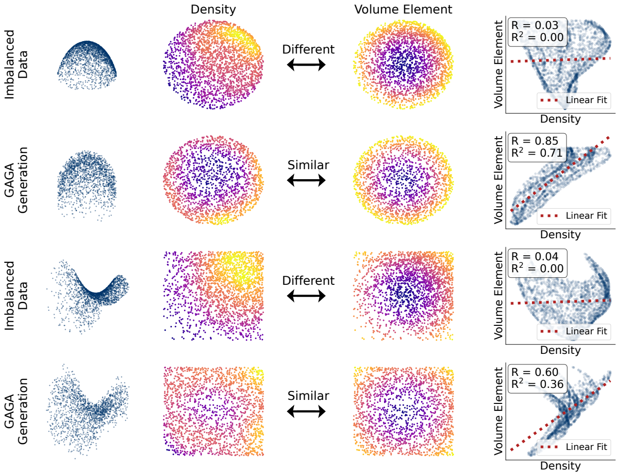

Figure 4: Geometry-aware generation with GAGA on hemisphere and saddle. **(A)** Generated points remain on the manifold, and are more evenly distributed compared to raw data. **(B)** Kernel density estimation. **(C)** Ground truth volume elements computed analytically. **(D)** In raw data, density does not correlate to volume element, indicating data imbalance. GAGA generation corrects the imbalance indicated by higher correlation between volume element and density.

Table 2: Pearson correlation and R^2 score between data density and ground truth volume element. GAGA greatly reduces data imbalance.

Manifold	Data	R	R^2
Hemisphere	Original Data	0.03	0.00
	GAGA Generation	0.85	0.71
Saddle	Original Data	0.04	0.00
	GAGA Generation	0.60	0.36
Paraboloid	Original Data	0.04	0.00
	GAGA Generation	0.66	0.44

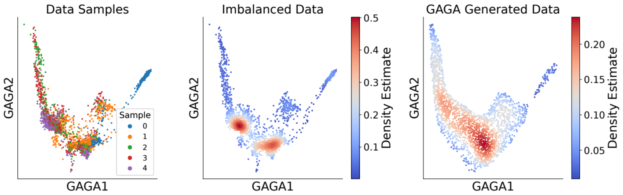

Figure 5: Geometry-aware generation with GAGA on Embryoid Body data. Left: The dataset includes measurements from five experiments. Middle: The data is sparse and imbalanced. Colors indicate density estimation. Right: GAGA reduces sampling imbalance.

Next, we applied volume-guided generation to the Embryoid Body dataset (Moon et al., 2019), a real-world single-cell dataset that captures cellular evolution over the course of 27 days (Figure 5 left). The data is largely imbalanced, with two density peaks, as shown in Figure 5 middle panel. Due to sampling bias, the data points in sample 4 exhibit a very high density, as significantly more data points were measured from this sample. Moreover, there are sparse areas and "holes" in the data manifold.

After volume-guided generation with GAGA, the data imbalance is significantly mitigated. Without deviating from the manifold, the density peaks are less spiky and the "holes" are properly filled in the GAGA-generated data (Figure 5 right panel) compared to the original Embryoid Body data (Figure 5 middle panel).

Generating along Geodesics on Manifold To evaluate GAGA's performance on generating geodesics on data manifold, we started with four toy manifolds: ellipsoid, torus, saddle, and hemisphere in $\mathbb{R}^3$. To make these datasets more challenging, we added Gaussian noise of different scales to the original data and rotate them to higher dimensions using a random rotation matrix. The ground truth geodesic lengths were obtained analytically if the solution is available or by using Dijkstra's algorithm on the noiseless data otherwise. See Appendix G.3 for details.

On the synthetic dataset, we compared our method with Dijkstra's algorithm, and a baseline that directly uses the metric without warping. More baseline comparisons and details are provided in Appendix G.3.

Table 3: Average MSE between predicted geodesic lengths and ground truth on simulated data with different dimensions and noise settings.

Manifold	Djikstra's	No Warping	GAGA
Ellipsoid	$\underline{4.40}$±6.6	143.70±246.5	**3.76**±7.1
Hemisphere	4.83±6.2	43.20±65.7	**0.47**±0.6
Saddle	**1.87**±3.5	55.59±76.8	$\underline{4.11}$±8.8
Torus	$\underline{5.01}$±7.9	271.84±295.3	**4.09**±6.3

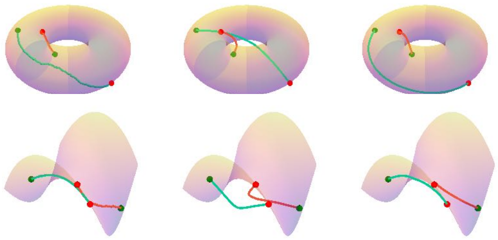

Figure 6: Comparison of ground truth and learned geodesics. From left to right: 1) ground truth, 2) no warping, 3) GAGA.

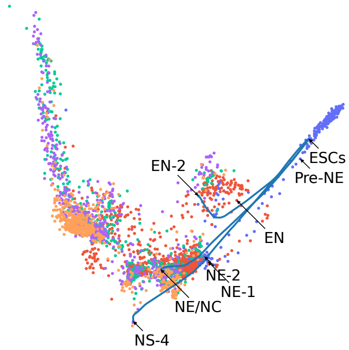

Figure 7: Geodesics learned on Embryoid Body data.

As shown in Table 3, GAGA generally outperforms all other methods except for one case (Djikstra's on saddle). It is worth mentioning that Djikstra's algorithm is only capable of connecting existing points but unable to generate points along the path. Directly using the metric without warping performs the worst by a big margin. We visualized the predicted geodesics on torus and saddle (Figure 6). In general, trajectories generated by GAGA stay on the manifold and are close to the ground truth geodesics, whereas some learned by the metric without the warping either deviate from the ground truth or directly cut through the manifold. More details and results are provided in Appendix H.5 and Appendix H.6.

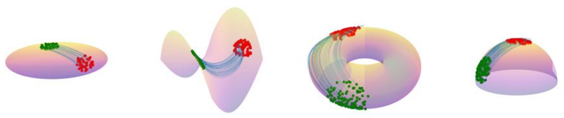

Figure 8: Transporting populations on toy manifolds.

In addition to toy datasets, we also visualized the geodesics learned on the Embryoid Body dataset (Figure 7). The starting points correspond to stem cells, while the ending points are selected at different lineages. The predicted geodesics recover the corresponding differentiation branches, aligning with the biological understanding of the data.

Population Interpolation along geodesics In the final application, we evaluate geodesics-guided population transport on simulated and real data.

For the simulated dataset, GAGA transports the source population to the target population through geodesics, which means that the trajectories remain on the manifold and follow the shortest paths (Figure 8). See Appendix G.4 for details.

Finally, we considered single-cell trajectory inference on the CITE-seq and Multiome datasets from a NeurIPS competition (Burkhardt et al., 2022). We performed 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. GAGA consistently outperforms all other methods by a large margin (Table 4). See details in Appendix H.7.

We have included details of our hyperparameter selection in Appendix G.5 and additional experiment results in Appendix H.

5 RELATED WORK

Geometry-aware encoding Non-linear dimensionality reduction methods such as PHATE or diffusion maps have proven useful in learning manifold structure from high-dimensional data. However, they have been difficult to extend to generate or sample new points (Huguet et al., 2024). To address this, some prior works have tried to regularize an encoder to match the embeddings or distances obtained from dimensionality reduction methods (Duque et al., 2020, 2022; Liu et al., 2024b; Huang et al., 2022; Fasina et al., 2023), or by minizing the Gromov-Monge cost (Lee et al., 2024). Despite embedding or distance preservation, these methods have not focused on generative modeling of points, can struggle in gaps for trajectory inference (Lee et al., 2024), or sometimes do not decode the data at all and simply provide embeddings (Fasina et al., 2023). As a result, it is difficult to use existing embeddings to generate or sample new points on and along these manifolds faithfully.

Table 4: Single-cell trajectory inference results on Cite and Multi datasets with 50 and 100 PCA dimensions. Leave-one-out is performed and 1-Wasserstein distances between prediction and ground truth are reported.

Data Dimension	50		100	
Alg.↓ Dataset→	Cite	Multi	Cite	Multi
DSBM (Shi et al., 2024)	53.81±7.74	66.43±14.39	58.99±7.62	70.75±14.03
I-CFM (Tong et al., 2023)	41.83±3.28	49.78±4.43	48.28±3.28	57.26±3.86
OT-CFM (Tong et al., 2023)	38.76±0.40	47.58±6.62	45.39±0.42	54.81±5.86
[SF]^{2}M-Exact (Tong et al., 2024c)	40.01±0.78	45.34±2.83	46.53±0.43	52.89±1.99
[SF]^{2}M-Geo (Tong et al., 2024c)	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	36.13±0.02	41.08±0.04	45.23±0.01
OT-MFM (Kapusniak et al., 2024)	36.39±1.87	45.16±4.96	41.78±1.02	50.91±4.623
GAGA (Ours)	**23.29**±0.83	**19.68**±1.93	**26.72**±0.99	**27.04**±2.95
Improvement over SOTA	↓ 31.8%	↓ 45.5%	↓ 34.6%	↓ 40.2%

Interpolating between points For interpolating between data points, traditional approaches often rely on linear interpolation or latent space traversal that does not align with complex data trajectories (Michelis and Becker, 2021; Mi et al., 2021). Some recent methods use a neural network to learn the gradient field, where optimal trajectories can be computed by following the gradient (Huguet et al., 2022; Liu et al., 2024c). However, these methods suffer from error accumulation, which may lead to large deviations when the trajectory is sufficiently long.

Population transport Transporting populations across experimental conditions, time points, or biological states is usually approached by flow matching (Lipman et al., 2022; Tong et al., 2023) and bridge matching (Shi et al., 2023; Thornton et al., 2022). Diffusion Schrödinger Bridge Matching (Shi et al., 2023) and Minibatch Optimal Transport Flow Matching (Tong et al., 2024a) operate on Euclidean space without considering the underlying manifold, and thus cannot generate trajectories along the manifold. Simulation-Free Schrödinger Bridges (Tong et al., 2024b) requires closed-form conditional path distributions, and therefore, it's not applicable to general manifold without analytic solutions.

Some of the recent work attempts to access the manifold and leverage the non-Euclidean metric. For example, Riemannian Diffusion Schrödinger Bridge (Thornton et al., 2022) addresses the Schrödinger Bridge problem in non-Euclidean space but requires the metric to perform manifold projection. Flow Matching on General Geometries (Chen and Lipman, 2024) requires closed-form solutions for computing geodesics on simple geometries and uses premetric instead of metric on general geometries. Solving Wasserstein Lagrangian Flows (Neklyudov et al., 2024) does not learn the metric of the manifold but instead uses prefixed Wasserstein-2 and Wasserstein Fisher-Rao metrics on the statistical manifold of the probability measures, thus unable to transport populations faithfully along the data manifold. The work most comparable to GAGA's metric learning framework is Metric Flow Matching (Kapusniak et al., 2024), which learns the manifold metric but only considers a specific family of diagonal metrics: metric LAND and RBF, tunable by a few hyperparameters.

6 DISCUSSION

We have explored encoding whether a given point lies on or off the manifold within our warped pullback metric. In the future, we aim to investigate additional priors and biases that could be incorporated into the metric to further guide population transport. For instance, encoding data sparsity to enable the generation of rare events or incorporating desired chemical properties for molecule generation.

Another promising direction is utilizing our learned metric to compute other geometric quantities and operators, such as curvatures, and log and exponential maps for manifold projections.

7 CONCLUSION

In this paper, we propose a geometry-aware generative autoencoder (GAGA) that preserves geometry in latent embeddings and can generate new points uniformly on the data manifold, interpolate along the geodesics, and transport populations across the manifold. We circumvent the limitations of existing generative methods, which mainly match the modes of distributions, by training generalizable geometry-aware neural network embeddings, leveraging points both on and off the data manifold, and learning a novel warped Riemannian metric on data space that allows us to generate points from the data geometry.

Acknowledgments

This research was partially funded and supported by ESP Mérite [G.H.], CIFAR AI Chair [G.W.], NSERC Discovery grant 03267 [G.W.], NIH grants (1F30AI157270-01, R01HD100035, R01GM130847, R01GM135929) [G.W.,S.K.], NSF Career grant 2047856 [S.K.], NSF grant 2327211 [S.K., G.W., M.P., I.A.], NSF/NIH grant 1R01GM135929-01 [M.H., S.K.], the Chan-Zuckerberg Initiative grants CZF2019-182702 and CZF2019-002440 [S.K.], the Sloan Fellowship FG-2021-15883 [S.K.], and the Novo Nordisk grant GR112933 [S.K.]. The tent provided here is solely the responsibility of the authors and does not necessarily represent the official views of the funding agencies. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Chen Liu helped with refining abstract, introduction, and the schematic figure prior to submission and should be considered an author.

References

Aggarwal, C. C., Hinneburg, A., and Keim, D. A. (2001). On the surprising behavior of distance metrics in high dimensional space. In *Database theory—ICDT 2001: 8th international conference London, UK, January 4–6, 2001 proceedings 8*, pages 420–434. Springer.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR.

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

Bhaskar, D., Zhang, Y., Xu, C., Sun, X., Fasina, O., Wolf, G., Nickel, M., Perlmutter, M., and Krishnaswamy, S. (2023). Learning graph geometry and topology using dynamical systems based message-passing. *arXiv preprint arXiv:2309.09924*.

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

Chen, R. T. and Lipman, Y. (2023). Flow matching on general geometries. *arXiv preprint arXiv:2302.03660*.

Chen, R. T. Q. and Lipman, Y. (2024). Flow matching on general geometries.

Coifman, R. R. and Lafon, S. (2006). Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30.

Do Carmo, M. P. and Flaherty, F. (1992). *Riemannian geometry*, volume 2. Springer.

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

Duque, A. F., Morin, S., Wolf, G., and Moon, K. R. (2022). Geometry regularized autoencoders. *IEEE transactions on pattern analysis and machine intelligence*, 45(6):7381–7394.

Fasina, O., Huguet, G., Tong, A., Zhang, Y., Wolf, G., Nickel, M., Adelstein, I., and Krishnaswamy, S. (2023). Neural fim for learning fisher information metrics from point cloud data. In *International Conference on Machine Learning*, pages 9814–9826. PMLR.

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

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11):139–144.

Heimberg, G., Bhatnagar, R., El-Samad, H., and Thomson, M. (2016). Low dimensionality in gene expression data enables the accurate extraction of transcriptional programs from shallow sequencing. *Cell systems*, 2(4):239–250.

Huang, J., Busch, E., Wallenstein, T., Gerasimiuk, M., Benz, A., Lajoie, G., Wolf, G., Turk-Browne, N., and Krishnaswamy, S. (2022). Learning shared neural manifolds from multi-subject fmri data. In *2022 IEEE 32nd International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 01–06. IEEE.

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

Huguet, G., Tong, A., De Brouwer, E., Zhang, Y., Wolf, G., Adelstein, I., and Krishnaswamy, S. (2024). A heat diffusion perspective on geodesic preserving dimensionality reduction. *Advances in Neural Information Processing Systems*, 36.

Jindal, A., Gupta, P., Jayadeva, and Sengupta, D. (2018). Discovery of rare cells from voluminous single cell expression data. *Nature communications*, 9(1):4719.

Kapusniak, K., Potaptchik, P., Reu, T., Zhang, L., Tong, A., Bronstein, M., Bose, A. J., and Di Giovanni, F. (2024). Metric flow matching for smooth interpolations on the data manifold. *arXiv preprint arXiv:2405.14780*.

Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. *Progress in artificial intelligence*, 5(4):221–232.

Lee, W., Yang, Y., Zou, D., and Lerman, G. (2024). Monotone generative modeling via a gromov-monge embedding.

Liao, D., Liu, C., Christensen, B. W., Tong, A., Huguet, G., Wolf, G., Nickel, M., Adelstein, I., and Krishnaswamy, S. (2024). Assessing neural network representations during training using noise-resilient diffusion spectral entropy. In *2024 58th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6. IEEE.

Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M. (2022). Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*.

Liu, C., Amodio, M., Shen, L. L., Gao, F., Avesta, A., Aneja, S., Wang, J. C., Del Priore, L. V., and Krishnaswamy, S. (2024a). Cuts: A deep learning and topological framework for multigranular unsupervised medical image segmentation. In *proceedings of Medical Image Computing and Computer Assisted Intervention – MICCAI 2024*, volume LNCS 15008. Springer Nature Switzerland.

Liu, C., Liao, D., Parada-Mayorga, A., Ribeiro, A., DiStasio, M., and Krishnaswamy, S. (2024b). DiffKillR: Killing and Recreating Diffeomorphisms for Cell Annotation in Dense Microscopy Images. *arXiv preprint arXiv:2410.03058*.

Liu, C., Xu, K., Shen, L. L., Huguet, G., Wang, Z., Tong, A., Bzdok, D., Stewart, J., Wang, J. C., Del Priore, L. V., et al. (2024c). Imageflownet: Forecasting multiscale image-level trajectories of disease progression with irregularly-sampled longitudinal medical images. *arXiv preprint arXiv:2406.14794*.

Martínez-Minaya, J., Cameletti, M., Conesa, D., and Pennino, M. G. (2018). Species distribution modeling: a statistical review with focus in spatio-temporal issues. *Stochastic environmental research and risk assessment*, 32:3227–3244.

Mi, L., He, T., Park, C. F., Wang, H., Wang, Y., and Shavit, N. (2021). Revisiting latent-space interpolation via a quantitative evaluation framework. *arXiv preprint arXiv:2110.06421*.

Michelis, M. Y. and Becker, Q. (2021). On linear interpolation in the latent space of deep generative models. *arXiv preprint arXiv:2105.03663*.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. (2018). Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*.

Moitra, A. and Risteski, A. (2020). Fast convergence for langevin diffusion with manifold structure. *arXiv preprint arXiv:2002.05576*.

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

Moon, K. R., Van Dijk, D., Wang, Z., Gigante, S., Burkhardt, D. B., Chen, W. S., Yim, K., Elzen, A. v. d., Hirn, M. J., Coifman, R. R., et al. (2019). Visualizing structure and transitions in high-dimensional biological data. *Nature biotechnology*, 37(12):1482–1492.

Neklyudov, K., Brekelmans, R., Tong, A., Atanackovic, L., Makhzani, A., et al. (2024). A computational framework for solving wasserstein lagrangian flows. In *Forty-first International Conference on Machine Learning*.

Shi, Y., Bortoli, V. D., Campbell, A., and Doucet, A. (2023). Diffusion schrödinger bridge matching.

Shi, Y., De Bortoli, V., Campbell, A., and Doucet, A. (2024). Diffusion schrödinger bridge matching. *Advances in Neural Information Processing Systems*, 36.

Sun, X., Xu, C., Rocha, J. F., Liu, C., Hollander-Bodie, B., Goldman, L., DiStasio, M., Perlmutter, M., and Krishnaswamy, S. (2024). Hyperedge representations with hypergraph wavelets: Applications to spatial transcriptomics. *arXiv preprint arXiv:2409.09469*.

Thornton, J., Hutchinson, M., Mathieu, E., Bortoli, V. D., Teh, Y. W., and Doucet, A. (2022). Riemannian diffusion schrödinger bridge.

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

Tong, A., Huang, J., Wolf, G., Van Dijk, D., and Krishnaswamy, S. (2020). Trajectorynet: A dynamic optimal transport network for modeling cellular dynamics. In *International conference on machine learning*, pages 9526–9536. PMLR.

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

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

Tong, A., Wolf, G., and Krishnaswamy, S. (2022). Fixing bias in reconstruction-based anomaly detection with lipschitz discriminators. *Journal of Signal Processing Systems*, 94(2):229–243.

Tong, A. Y., Malkin, N., Fatras, K., Atanackovic, L., Zhang, Y., Huguet, G., Wolf, G., and Bengio, Y. (2024c). Simulation-free schrödinger bridges via score and flow matching. In *International Conference on Artificial Intelligence and Statistics*, pages 1279–1287. PMLR.

Van de Sande, B., Lee, J. S., Mutasa-Gottgens, E., Naughton, B., Bacon, W., Manning, J., Wang, Y., Pollard, J., Mendez, M., Hill, J., et al. (2023). Applications of single-cell rna sequencing in drug discovery and development. *Nature Reviews Drug Discovery*, 22(6):496–520.

Van Dijk, D., Sharma, R., Nainys, J., Yim, K., Kathail, P., Carr, A. J., Burdziak, C., Moon, K. R., Chaffer, C. L., Pattabiraman, D., et al. (2018). Recovering gene interactions from single-cell data using data diffusion. *Cell*, 174(3):716–729.

Wang, Y., Sun, X., and Zhao, H. (2022). Benchmarking automated cell type annotation tools for single-cell atac-seq data. *Frontiers in Genetics*, 13:1063233.

Zappia, L., Phipson, B., and Oshlack, A. (2017). Splatter: simulation of single-cell rna sequencing data. *Genome biology*, 18(1):174.

Zhao, T., Chiang, Z. D., Morriss, J. W., LaFave, L. M., Murray, E. M., Del Priore, I., Meli, K., Lareau, C. A., Nadaf, N. M., Li, J., et al. (2022). Spatial genomics enables multi-modal study of clonal heterogeneity in tissues. *Nature*, 601(7891):85–91.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes] We included them in the Methods, as well as in proofs of propositions in the Appendices.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Not Applicable]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [No] However, we will open source the code upon acceptance.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes] They are included in the proposition statements.
   (b) Complete proofs of all theoretical results. [Yes] They are included either in the main text or in the referenced Appendices.
   (c) Clear explanations of any assumptions. [Yes] Things are fairly well explained and we can further elucidate if requested.

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes] We will open source the code upon acceptance. But before that, things are reasonably reproducible using the details in the submission.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes] We mentioned all experiments are repeated with 5 random seeds, and all numbers following the plus/minus sign are standard deviations.
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes] All experiments are done on an internal server and runnable with one single GPU.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Appendix

Table of Contents

A Manifold Learning and Diffusion Geometry

The *Manifold Hypothesis* states that data are often sampled *on* or *near* an intrinsically low-dimensional manifold within high-dimensional Euclidean space. Manifold learning techniques aim to uncover and recreate this manifold in a lower-dimensional space.

Many manifold learning approaches use data *diffusion geometry*, which extracts geometric features from an approximation of heat flow on the data. Diffusion geometry models a high-dimensional point cloud as a graph by applying a kernel $\mathcal{K}$ (e.g., the Gaussian kernel $e^{-\frac{||z_1 - z_2||^2}{\sigma}}$) to the pairwise Euclidean distances between data points.

The kernel $\mathcal{K}$ is normalized to obtain a row-stochastic matrix P, where $P(z_1, z_2) = \frac{\mathcal{K}(z_1,z_2)}{||\mathcal{K}(z_1,\cdot)||_1}$. This matrix P encodes transition probabilities between points. Powering P^t represents a t-step random walk. Long-range or spurious connections are given less weight through this iterated walk than robust on-manifold paths, allowing the resulting point-wise *diffusion probabilities* to recover manifold geometry even in the presence of sparsity and noise. Methods like Diffusion Maps, PHATE, and HeatGeo use diffusion probabilities to define a *statistical distance* between data points (Coifman and Lafon, 2006; Moon et al., 2019; Huguet et al., 2024).

B Riemannian Manifolds & Metrics

The *Manifold Hypothesis* motivates our use of Riemannian geometry. Formally, an n-dimensional manifold $\mathcal{N}$ is a topological space that is locally homeomorphic to $\mathbb{R}^n$. Intuitively, while the global structure of $\mathcal{N}$ can be complex, every small region is similar to the Euclidean space.

A Riemannian manifold $(\mathcal{N}, g)$ is endowed with a Riemannian metric g, which defines an inner product on the tangent space at each point. At each $x \in \mathcal{N}$, the metric g_x assigns an inner product to tangent vectors $X, Y \in T_x\mathcal{N}$ via

$$g_x(X, Y) = X^T g(x) Y,$$

where (with a slight abuse of notation) $g(x)$ denotes an $n \times n$ matrix representing the inner product on $T_x\mathcal{N}$. This metric allows us to measure angles and lengths. In particular, the length of a tangent vector X is given by

$$\|X\| = \sqrt{g_x(X, X)},$$

and the length of a smooth curve $c : [0, T] \to \mathcal{N}$ is defined as

$$L(c) = \int_0^T \sqrt{g_{c(t)}\big(\dot{c}(t), \dot{c}(t)\big)}\, dt.$$

This expression computes the distance traveled along the curve, much like measuring a winding road on a flat map.

If the manifold is parametrized by a function $f(z)$ with $z \in \mathcal{D}$, its volume (or area, in the two-dimensional case) is calculated by

$$\int_{\mathcal{D}} \sqrt{\det g(x)}\, dx.$$

Here, $\sqrt{\det g(x)}$, called the volume element, quantifies how much local space is present at the point x.

B.1 The Pullback Metric

A key element of our approach is the *Riemannian pullback metric*. Suppose we have a map between manifolds, $f : \mathcal{M} \to (\mathcal{N}, g)$. At each point $x \in \mathcal{M}$, the differential

$$df_x : T_x\mathcal{M} \to T_{f(x)}\mathcal{N}$$

provides a linear approximation of f. Using this differential, the pullback metric f^*g on $\mathcal{M}$ is defined by

$$f^*g(X, Y) = g(df_x X, df_x Y),$$

for any tangent vectors $X, Y \in T_x\mathcal{M}$.

Intuitively, the pullback metric equips $\mathcal{M}$ with the geometry of $(\mathcal{N}, g)$ as determined by f. It allows us to measure lengths, angles, and distances on $\mathcal{M}$ in a manner that reflects the geometry of the target space. This construction is fundamental to our method, as it bridges the geometry of $\mathcal{M}$ with the geometry provided by f.

For further details, we refer the reader to Do Carmo and Flaherty (1992, Chapters 0 and 1).

C Obtaining the Function $s(x)$

Recall that $s(x)$ provides an auxiliary dimension that complements the encoder f_θ, where the value represents the deviation from the manifold. $s(x) \approx 0$ if x is on the manifold, and $s(x)$ increases as x moves away from the manifold.

C.1 Approach 1: Discriminator

There are various ways to assign the value in the auxiliary dimension. In our implementation, we employ a discriminative network (Goodfellow et al., 2020) to predict whether a point is on or off the manifold.

To train the GAN-style discriminator, we first generate negative samples away from the data manifold in the data space by adding high-dimensional Gaussian noise to the data (Eqn. (Appx. 1)), where c is a constant chosen such that the space away from the manifold is in the support of the distribution of $\check{x}$.

$$\check{x}_i = x_i + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, cI) \tag{Appx. 1}$$

Then, we define a discriminator w_ψ that maps from the data space to a score, optimized by the loss function in Eqn. (Appx. 2) inspired by Wasserstein Generative Adversarial Networks (Arjovsky et al., 2017).

$$\mathcal{L}_w(\psi) = \mathbb{E}_{\check{x}}\left[w_\psi(\check{x})\right] - \mathbb{E}_x\left[w_\psi(x)\right] + \mathrm{Var}_x(w_\psi(x)) \tag{Appx. 2}$$

w_ψ is a Lipschitz function due to weight clipping and spectral normalization (Arjovsky et al., 2017; Miyato et al., 2018). The variance term is added to encourage the discriminator to have uniform predictions. Finally, we define the GAGA embedding with auxiliary dimension in Eqn. (5).

We have the following lemma showing that the condition "$s(x) \approx 0$ if x is on the manifold, and $s(x)$ increases as x moves away from the manifold" is achieved:

Lemma C.1. *Suppose w_ψ is L-Lipschitz, and $\max_{i,j} ||x_i - \check{x}_j|| \leq M$. for any $\epsilon > 0$, if $\mathcal{L}_w(\psi) \leq -LM + \epsilon$, we have $\mathbb{E}_x[s(x)^2] \leq \epsilon$.*

C.2 Approach 2: Gaussian Process

Alternative to the discriminator, we can also obtain $s(x)$ using the variance of a Gaussian process. We take advantage of the observation that the uncertainty (covariance) of a Gaussian process increases as the evaluation point moves away from the seen training point. We use an radial basis function kernel

$$K(x, x') = \exp\left(-\frac{||x - x'||^2}{2\sigma^2}\right) \tag{Appx. 3}$$

in the model, and define $s(x)$ to be the posterior variance

$$s(x) := K(x, x) - K(x, X)[K(X, X) + \sigma_n^2 I]^{-1} K(X, x), \tag{Appx. 4}$$

where $X = \{x_1, \ldots, x_N\}$ is the data; $K(x, X) := (K(x, x_1), K(x, x_2), \ldots, K(x, x_N))$; $K(X, x) := K(x, X)^T$; and $K(X, X) := (K(x_i, x_j))_{i=1,\ldots,N}^{j=1,\ldots,N}$.

D Curve Parameterization for Generating Along Geodesics

We parameterize the curves using an interpolation between starting and ending points, with a linear term and a non-linear term parameterized by an MLP γ_η.

$$c_\eta(x_0, x_1, t) = tx_1 + (1-t)x_0 + (1-(2t-1)^2)\gamma_\eta(x_0, x_1, t), \tag{Appx. 5}$$

E Proofs of Lemmas and Propositions

E.1 proposition 3.1

For Riemannian manifolds $(\mathcal{M}, g_\mathcal{M}), (\mathcal{N}, g_\mathcal{N})$ and diffeomorphism $f : \mathcal{M} \to \mathcal{N}$, if f is a local isometry, i.e., there exists $\epsilon > 0$, such that for any $x_0, x_1 \in \mathcal{M}, d_\mathcal{M}(x_0, x_1) < \epsilon \implies d_\mathcal{M}(x_0, x_1) = d_\mathcal{N}(f(x_0), f(x_1))$, then we have $g_\mathcal{M} = f^* g_\mathcal{N}$.

Proof. We first prove that the two metrics agree on vector norms. That is, for any $u \in T_x\mathcal{M}, g_\mathcal{N}(dfu, dfu) = g_\mathcal{M}(u,u)$.:

$\forall z \in \mathcal{N}, \forall$ smooth curve $\gamma(t) \subset \mathcal{N}$, and let $\xi(t) = f^{-1}(\gamma(t))$. Then there exists $\delta > 0$ such that $\forall 0 < t < \delta$

$$\int_0^t \sqrt{g_\mathcal{M}(\dot{\xi}(\tau), \dot{\xi}(\tau))} d\tau < \epsilon \tag{Appx. 6}$$

We have

$$\int_0^t \sqrt{g_\mathcal{M}(\dot{\xi}(\tau), \dot{\xi}(\tau))} d\tau = \int_0^{\gamma^{-1}\circ\xi(t)} \sqrt{g_\mathcal{N}(\dot{\gamma}(\tau), \dot{\gamma}(\tau))} d\tau \tag{Appx. 7}$$

Take $t \to 0$, we have $g_\mathcal{N}(dfu, dfu) = g_\mathcal{M}(u,u)$ where $u = \dot{\xi}(0)$.

Next we use the identity

$$\langle u, v \rangle = \frac{1}{4}\left(\langle u+v, u+v \rangle - \langle u-v, u-v \rangle\right) \tag{Appx. 8}$$

for any 2-form $\langle \cdot, \cdot \rangle$, and apply to $g_\mathcal{M}, g_\mathcal{N}$, we have

$$g_\mathcal{N}(dfu, dfv) = g_\mathcal{M}(u,v) \forall u, v \in T_x\mathcal{M}. \tag{Appx. 9}$$

□

E.2 lemma C.1

Suppose w_ψ is L-Lipshitz, and $\max_{i,j} ||x_i - \check{x}_j|| \leq M$. $\forall \epsilon > 0$, if $\mathcal{L}_w(\psi) \leq -LM + \epsilon$, we have $\mathbb{E}_x[s(x)^2] \leq \epsilon$.

Proof. Denote p_{on} the data distribution and p_{off} the distribution of off-manifold points defined eqn. (Appx. 1).

$\forall x \sim p_{\text{on}}, \check{x} \sim p_{\text{off}}$, since w_ψ is L-Lipshitz, $|w_\psi(\check{x}) - w_\psi(x)| \leq L||\check{x} - x|| < LM$.

Taking expectaion, we have $\mathbb{E}_{\check{x}}[w_\psi(\check{x})] - E_x[w_\psi(x)] \geq -LM$.

Thus, $\mathcal{L}_w(\psi) \leq -LM + \epsilon \implies \mathbb{E}[s(x)^2] = \text{Var}_x(w_\psi(x)) = \mathcal{L}_w(\psi) - (\mathbb{E}_{\check{x}}[w_\psi(\check{x})] - E_x[w_\psi(x)]) \leq \epsilon$. □

E.3 lemma 3.2

If there exists $\alpha \in \mathbb{R}$ such that for any $x, \check{x}, \alpha||x - \check{x}|| \leq |s(x) - s(\check{x})|$. Then for any $x, \check{x}, ||f^+(x) - f^+(\check{x})|| \geq \alpha\beta||x - \check{x}||$. Furthermore, denoting $\mathcal{D}_\mathcal{M}(y) := \inf_{x\in\mathcal{M}} ||x-y||$ and $\mathcal{D}_{f^+(\mathcal{M})}(y) := \inf_{x\in\mathcal{M}} ||f^+(x) - f^+(y)||$, then for any $\check{x}$, we have $\mathcal{D}_{f^+(\mathcal{M})}(\check{x}) \geq \alpha\beta\mathcal{D}_\mathcal{M}(\check{x})$.

Proof. Because $r(x) = \begin{pmatrix} f_\theta(x) \\ s(x) \end{pmatrix}$, where $s(x) = \beta(\bar{w} - w_\psi(x))$, we directly compute:

$$\begin{align} ||r(x) - r(\check{x})||^2 =& ||f_\theta(x) - f_\theta(\check{x})||^2 + |s(x) - s(\check{x})|^2 \tag{Appx. 10} \\ \geq& |s(x) - s(\check{x})|^2 \tag{Appx. 11} \\ \geq& \beta^2 |w_\psi(x) - w_\psi(\check{x})|^2 \tag{Appx. 12} \\ \geq& \beta^2 \alpha^2 ||x - \check{x}||^2, \tag{Appx. 13} \end{align}$$

we have $||r(x) - r(\check{x})|| \geq \beta\alpha ||x - \check{x}||$.

Taking infimum over $x \in \mathcal{M}$, we have $\mathcal{D}_{r(\mathcal{M})}(\check{x}) \geq \beta\alpha\mathcal{D}_\mathcal{M}(\check{x})$ □

E.4 proposition 3.3

Suppose $f_{\text{target}}(x) = \lambda s(x) - \log(f_{vol})(x)$ is α-strongly convex for some constant $\alpha > 0$, i.e. $\nabla^2 f(x) \succeq \alpha I$, then the distribution of X in Eqn. (6) converges exponentially fast in Wasserstein distance to a distribution supported on the data manifold, whose restriction on the manifold is proportional to the volume distribution function.

Proof. The proof follows from equation (1.4.9) in this textbook `https://chewisinho.github.io/main.pdf`: Suppose f_{target} is α-strongly convex, for any $X_t \sim \mu_t, Y_t \sim \nu_t$ following the Langevin dynamics, initialized at $X_0 \sim \mu_0, Y_0 \sim \nu_0$, we have

$$W_2^2(\mu_t, \nu_t) \leq e^{-2\alpha t} W_2^2(\mu_0, \nu_0). \tag{Appx. 14}$$

Now we check that $p(x) = \frac{1}{Z} e^{-f_{\text{target}}(x)}$, where $Z = \int e^{-f_{\text{target}}(x)} dx$, corresponds to a stochastic process governed by this SDE by showing that it satisfies the Fokker-Planck equation.

$$\begin{aligned} \text{LHS: } \frac{\partial p(x)}{\partial t} &= 0 \\ \text{RHS: } \nabla \cdot (p(x) \nabla f_{\text{target}}(x)) + \Delta p(x) \\ = \frac{1}{Z}(\nabla \cdot (e^{-f_{\text{target}}(x)} \nabla f_{\text{target}}(x)) + \Delta e^{-f_{\text{target}}(x)}) \\ = \frac{1}{Z}(-\nabla \cdot \nabla e^{-f_{\text{target}}(x)} + \Delta e^{-f_{\text{target}}(x)}) \\ &= 0 \end{aligned}$$

Therefore, for any initialization $X_0 \sim \mu_0$, let $Y_0 \sim p$, we have

$$W_2^2(\mu_t, p) \leq e^{-2\alpha t} W_2^2(\mu_0, p). \tag{Appx. 15}$$

where $p(x) = e^{-\lambda s(x)} f_{vol}(x)$. Since $s(x) \approx 0$ if x in on the manifold, and is large when x is away from the manifold, we have $p(x) \approx f_{vol}(x)$ if x is on the manifold, and $p(x) \approx 0$ if x is away from the manifold. □

E.5 lemma 3.4

Assume that the ω-thickening of the manifold $\mathcal{M} \subset \mathbb{R}^n$, defined as $\mathcal{M}^\omega := \{x \in \mathbb{R}^n : \inf_{m \in \mathcal{M}} d(x, m) < \omega\}$, (i.e., the set of points whose distance from $\mathcal{M}$ is less than ω) maps into a subset of the ϵ-thickening of $f(\mathcal{M})$. Here, the ϵ-thickening is defined analogously, with ϵ chosen such that for every $x \in f(\mathcal{M})$, the ball $B_\epsilon(x) := \{y \in \mathbb{R}^n : \|y - x\| < \epsilon\}$ intersects $f(\mathcal{M})$ in exactly one connected component.

Then, for any smooth curve $c : [0, 1] \to \mathbb{R}^n$ connecting x_0 and x_1 (i.e., $c(0) = x_0$ and $c(1) = x_1$), there exists a smooth curve $c' : [0, 1] \to \mathcal{M}$ lying entirely **on the manifold** (with $c'(0) = x_0$ and $c'(1) = x_1$) such that $\mathcal{L}_{\text{Geo}}(c') \leq \mathcal{L}_{\text{Geo}}(c) - \alpha^2\beta^2 \frac{1}{M} \sum_{m=1}^{M} \big(\mathcal{D}_\mathcal{M}(c(t_m)) - \mathcal{D}_\mathcal{M}(c(t_{m-1}))\big)^2 + \xi$. Here, α is defined as in Lemma 3.2, $\mathcal{D}_\mathcal{M}$ denotes the distance from a point to $\mathcal{M}$ (also as in Lemma 3.2), and ξ is a fixed positive constant independent of x_t and β.

Proof. Consider a smooth $c : [0, 1] \to \mathbb{R}^n$ with $c(0) = x_1, c(0) = x_1$ which lies within the ω-thickening of $\mathcal{M}$. We construct an open cover of its image $f(c)$ as the collection of open balls $\{B_\epsilon(c(t)) : t \in [0, 1]\}$. By compactness, this admits a finite subcover at some collection of times $\{t_1 \dots t_N\}$. For each t_i, we can choose point $c'[t_i]$ from $B_\epsilon(c(t_i)) \cap f(\mathcal{M})$. By the continuity of $f \circ c$, these are all part of the same connected component of $f(\mathcal{M})$, hence there exists a curve $c' : [0, 1] \to \mathbb{R}^n$ with the same endpoints as c, whose image contains $\{c'[t_i]\}$. Furthermore, by the smoothness of f and c, there exists a uniform $K > 0$ independent of c, c' such that $| \int \dot{c}(t)^T J_f^T J_f \dot{c}(t) - \dot{c}'(t)^T J_f^T J_f \dot{c}'(t) dt| < K\epsilon$. Following lemma C.1, because $c' \in \mathcal{M}$, we also have $| \int \dot{c}'(t)^T J_s^T J_s \dot{c}'(t)| < \epsilon'$ for some uniform $\epsilon' > 0$ independent on c, c'.

We can decompose the pullback metric as

$$J_r^T J_r = J_f^T J_f + J_s^T J_s. \tag{Appx. 16}$$

and compute the difference

$$\mathcal{L}_{\text{Geo}}(c) - \mathcal{L}_{\text{Geo}}(c') = \frac{1}{M} \sum_{m=1}^{M} (\dot{c}(t)^T J_f^T J_f \dot{c}(t) + \dot{c}(t)^T J_s^T J_s \dot{c}(t) - (\dot{c}'(t)^T J_f^T J_f \dot{c}'(t) + \dot{c}'(t)^T J_s^T J_s \dot{c}'(t))) \tag{Appx. 17}$$

$$= \frac{1}{M} \sum_{m=1}^{M} (\dot{c}(t)^T J_f^T J_f \dot{c}(t) - \dot{c}'(t)^T J_f^T J_f \dot{c}'(t) + \dot{c}(t)^T J_s^T J_s \dot{c}(t) + \dot{c}'(t)^T J_s^T J_s \dot{c}'(t)) \tag{Appx. 18}$$

$$\geq - K\epsilon - \epsilon' + \frac{1}{M} \sum_{m=1}^{M} \dot{c}(t)^T J_s^T J_s \dot{c}(t). \tag{Appx. 19}$$

$$\geq - K\epsilon - \epsilon' - \epsilon'' + \frac{1}{M} \sum_{m=1}^{M} \dot{(} s(c(t_m)) - s(c(t_{m-1}))^2 \tag{Appx. 20}$$

$$\geq - K\epsilon - \epsilon' - \epsilon'' + \frac{1}{M} \sum_{m=1}^{M} \dot{(} s(c(t_m)) - s(c(t_{m-1}))^2. \tag{Appx. 21}$$

$$\geq - K\epsilon - \epsilon' - \epsilon'' + \frac{1}{M} \alpha\beta \sum_{m=1}^{M} \dot{(} D_{\mathcal{M}}(c(t_m)) - D_{\mathcal{M}}(c(t_{m-1}))^2, \tag{Appx. 22}$$

$$\tag{Appx. 23}$$

where ϵ', ϵ'' are positive constants independent on x_t, β. □

E.6 proposition 3.5

When $\mathcal{L}_{\text{Geo}}$ is minimized, $\max\limits_{m=1,\dots,M} \mathcal{D}_{\mathcal{M}}(c(t_m)) \leq \frac{\sqrt{\xi}}{\alpha\beta}$, i.e., for sufficiently large β, $c(t)$ is close to the manifold with a maximum distance of $\frac{\sqrt{\xi}}{\alpha\beta}$. Furthermore, let $c'(t)$ be a geodesic between x_0 and x_1 under the metric $g_{\mathcal{M}}$, we have $\frac{1}{M} \sum_{m=1}^{M} g_{\mathcal{M}}(\dot{c}, \dot{c})(x_0, x_1, t_m) \leq \frac{1}{M} \sum_{m=1}^{M} g_{\mathcal{M}}(\dot{c}', \dot{c}')(x_0, x_1, t_m) + \xi' \frac{\sqrt{\xi}}{\alpha\beta}$ for some positive constant ξ'. That is, c approximately minimizes the energy (and hence curve length) under $g_{\mathcal{M}}$.

Proof. Suppose c minimizes $\mathcal{L}_{\text{Geo}}$. Then by lemma 3.4, there exists c' such that

$$\mathcal{L}_{\text{Geo}}(c') \leq \mathcal{L}_{\text{Geo}}(c) - \alpha^2 \beta^2 \frac{1}{M} \sum_{m=1}^{M} (D_{\mathcal{M}}(c(t_m)) - D_{\mathcal{M}}(c(t_{m-1})))^2 + \xi. \tag{Appx. 24}$$

On the other hand, because c is a minimizer, we have

$$\mathcal{L}_{\text{Geo}}(c) \leq \mathcal{L}_{\text{Geo}}(c'). \tag{Appx. 25}$$

Combining them, we have

$$\alpha^2 \beta^2 \frac{1}{M} \sum_{m=1}^{M} (D_{\mathcal{M}}(c(t_m)) - D_{\mathcal{M}}(c(t_{m-1})))^2 \leq \mathcal{L}_{\text{Geo}}(c) - \mathcal{L}_{\text{Geo}}(c') + \xi \leq \xi. \tag{Appx. 26}$$

Rearrange $t_0, \ldots, t_M$ with a permutation σ such that $D_{\mathcal{M}}(t_{\sigma(0)}) \leq \cdots \leq D_{\mathcal{M}}(t_{\sigma(M)})$, and because $D_{\mathcal{M}}(t_0) = 0$ (the minimum), WLOG, let $t_{\sigma(0)} = 0$. We have

$$\alpha^2 \beta^2 \frac{1}{M} \sum_{m=1}^{M} (D_{\mathcal{M}}(c(t_{\sigma(m)})) - D_{\mathcal{M}}(c(t_{\sigma(m-1)})))^2 \leq \xi \tag{Appx. 27}$$

$$\implies \alpha^2 \beta^2 (\frac{1}{M} \sum_{m=1}^{M} (D_{\mathcal{M}}(c(t_{\sigma(m)})) - D_{\mathcal{M}}(c(t_{\sigma(m-1)})))^2 \leq \xi \text{ (by Jensen's inequality)} \tag{Appx. 28}$$

$$\implies \alpha^2 \beta^2 (D_{\mathcal{M}}(c(t_{\sigma(M)}) - D_{\mathcal{M}}(c(t_{\sigma(0)}))^2 \leq \xi \tag{Appx. 29}$$

$$\implies \max_{m=1,\ldots,M} D_{\mathcal{M}}(c(t_m)) = D_{\mathcal{M}}(c(t_{\sigma(M)})) \leq \frac{\sqrt{\xi}}{\alpha\beta}. \tag{Appx. 30}$$

The proof for the second part follows from the Lipshitz property of $s(x)$ and the smoothness of f in lemma 3.4. □

E.7 Proposition 3.6

At the convergence of Algorithm 2,

$$x(t) = x_0 + \int_0^t v_\nu(x_0, \tau) d\tau \tag{Appx. 31}$$

are geodesics between points in $\mathcal{X}$ and points in $\mathcal{Y}$ following the optimal transport plan that minimizes the geodesic lengths.

Proof. We first prove that when Eqn. (7) and Eqn. (8) are minimized, Eqn. (Appx. 31) yields geodesics from $x_0 \in \mathcal{X}$ to $x_1 \in \mathcal{Y}$. This is because by Lemma 3.4, the curves c_η are geodesics. When Eqn. (8) is minimized, v_ν approximates the gradient of c_η, and its integration starts at the same point x_0 approximates c_η.

The rest follows from the the proof of Algorithm 3 in (Tong et al., 2023).

□

F Additional Convergence Proposition for Volume-Guided Generation

Proposition F.1. *Suppose when X_t is initialized near the manifold $\mathcal{M}$, it stays in the neighborhood $\mathcal{D} := \{X \in \mathcal{M} : \mathcal{D}_{\mathcal{M}}(X) \leq s\}$ near $\mathcal{M}$ with high probability up to time T, and that $\exp(-f_{\text{target}}(x))$ satisfies Poincaré's inequality along and across the level sets of f_{target} near the manifold, then the distribution in Eqn. (6) converges exponentially fast in total variation distance to a distribution supported on the data manifold, whose restriction on the manifold is proportional to the volume distribution function.*

Proof. This is a direct application of Moitra and Risteski (2020, Theorem 4). □

These assumptions are attainable in our setup, given the fact that the manifold is bounded because the $\lambda s(x)$ term prevents the points from going far away from the manifold. In addition, if we restrict the domain to a ball containing the manifold, the function $e^{-f_{\text{target}}(x)}$ is differentiable and hence satisfies Poincare's inequality on this compact domain.

G Experiment Details

G.1 Geometry-aware autoencoder

G.1.1 Datasets: Splatter

We evaluate our geometry-aware autoencoder on simulated scRNA-seq datasets Splatter(Zappia et al., 2017). Splatter uses parametric models to simulate cell populations with multiple cell types, structures, and differentiation patterns. Specifically, we evaluate on single-cell data of group and path structures with biological coefficient of variation (bcv) parameters $\{0, 0.18, 0.25, 0.5\}$. A higher bcv corresponds to a lower signal-to-noise ratio. The cellular state space is a simulation parameter indicating whether the cells are arranged in clusters or trajectories in the data space. In Splatter, it is specified by the `method` parameter, where clusters correspond to `groups` and trajectories correspond to `paths`.

G.1.2 Evaluation Criteria

For the encoder, we leverage DEMaP (Moon et al., 2019) to measure the correlation between Euclidean distances in latent space and ground truth geodesic distances in original data space.

$$\text{DEMaP}(f) = \frac{2}{N(N-1)} \sum_{i<j} \text{Corr}(||f(x_i) - f(x_j)||_2, d_{ij}), \tag{Appx. 32}$$

where f is the encoder to be evaluated, Corr is Pearson correlation, x_i, x_j are points from test data, and d_{ij} is the ground truth geodesic distance between x_i, x_j, computed from shortest path distance under noiseless setting.

For decoder evaluation, we propose a novel criteria, DRS (Denoised Reconstruction Score), to account for the noisy and sparse nature of single-cell data. DRS computes the correlation between reconstructed genes and denoised genes through denoising and imputation method MAGIC(Van Dijk et al., 2018).

$$\text{DRS}(f, h) = \frac{1}{N_{\text{gene}}} \sum_{i=1}^{N_{\text{gene}}} \text{Corr}(y_i, y_i^{\text{MAGIC}}), \tag{Appx. 33}$$

where f, h are the encoder and decoder pair, $y_i = \text{PCA}^{-1}(h(f(x_i)))$, $y_i^{\text{MAGIC}} = \text{PCA}^{-1}(\text{MAGIC}(x_i))$. PCA^{-1} here is the inverse PCA operator since the original data are first PCA transformed and then fed into the autoencder. Therefore we use inverse PCA to map the reconstructed points back to the gene space for evaluation.

G.2 Volume-guided Generation on Manifold

G.2.1 Generate imbalanced data on toy manifolds

We generate imbalanced data on hemishpere, saddle, and paraboloid. Table Appx. 2 shows their parametrizations and volume elements.

In order to generate imbalanced data on the manifold, we generate $3,000$ points following a bivariate Gaussian distribution $\mathcal{N}\left(\begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}2 & 0\\0 & 2\end{pmatrix}\right)$, with range restricted to $[-2,2] \times [-2,2]$. These points are used as parameters (u, v), which we use to compute (x, y, z) with the parametrizations in Table Appx. 2. These points $(x, y, z) \in \mathbb{R}^3$ are used as training points for GAGA.

G.2.2 Details on evaluation metric for volume guided generation

We evaluate the generated points by comparing its density estimation with the ground truth volume element in the parameter space. We first convert the generated points in $\mathbb{R}^3$ back to the parameter space using $u = x, v = y$. Then, we use apply kernel density estimation to the parameters $(u, v) \in \mathbb{R}^2$. We use a Gaussian kernel and use Scott's rule to determine the bandwidth. To avoid the error from boundary effects of kernel density estimation, as well as the numerical instability of the volume element computation of the hemisphere near the boundary, we mask out the points near the boundary by only computing kernel density estimation and volume element on $\{(u, v) : u^2 + v^2 < 0.8\}$ for hemisphere, and $\{(u, v) : |u|, |v| < 1.6\}$ for saddle and paraboloid.

Table Appx. 1: Average DEMaP and DRS on simulated single-cell datasets over different noise settings.

	Objective	State Space	DEMaP (↑)	DRS (↑)
Autoencoder	$\mathcal{L}_{\text{Recon}}$	Clusters	0.347±0.117	0.642±0.129
GAGA	$\mathcal{L}_{\text{Recon}}, \mathcal{L}_{\text{Dist}}$	Clusters	**0.645**±0.195	**0.667**±0.165
Autoencoder	$\mathcal{L}_{\text{Recon}}$	Trajectories	0.433±0.135	**0.587**±0.148
GAGA	$\mathcal{L}_{\text{Recon}}, \mathcal{L}_{\text{Dist}}$	Trajectories	**0.600**±0.191	0.559±0.143

Manifold	Parametrization (u, v)	Volume Element $f_{vol}(u, v)$
Hemisphere	$\begin{cases} x = u \\ y = v \\ z = \sqrt{1 - u^2 - v^2} \end{cases}$	$\frac{1}{\sqrt{1-u^2-v^2}}$
Saddle	$\begin{cases} x = u \\ y = v \\ z = u^2 - v^2 \end{cases}$	$\sqrt{1 + 4u^2 + 4v^2}$
Paraboloid	$\begin{cases} x = u \\ y = v \\ z = u^2 + v^2 \end{cases}$	$\sqrt{1 + 4u^2 + 4v^2}$

Table Appx. 2: Parameterizations and volume elements of toy manifolds. We have access to the analytical forms of the volume elements computed from the parameterizations. We use them as the ground truth in evaluation.

G.3 Generating along geodesics

G.3.1 Datasets: Simulated manifolds

We generate four toy manifolds: ellipsoid, torus, saddle, and hemisphere in $\mathbb{R}^3$. We add Gaussian noise of different scales to the original toy manifolds and rotate the data to higher dimensions using a random rotation matrix. We simulate datasets under $\{0, 0.1, 0.3, 0.5\}$ noise scales and $\{3, 5, 10, 15\}$ dimensions. For each dataset, we randomly select 20 pairs of starting and ending points on the manifold.

We benchmark all methods on the noisy, high-dimensional data, and compute the pairwise geodesics.

G.3.2 Evaluation Criteria

Quantitatively, we evaluate these methods on the MSE criteria: the mean squared error between the predicted geodesic length and ground truth length.

$$\text{Length MSE} = \frac{1}{k} \sum_{i=1}^{k} (\hat{l}_i - l_i)^2, \tag{Appx. 34}$$

where k is the total number of geodesics, $l_i, \hat{l}_i$ are the lengths of the i-th ground truth and predicted geodesics. We obtain the ground truth geodesics analytically if the solution is available or using Dijkstra's algorithm on noiseless data otherwise.

G.4 Geodesics-guided flow matching

G.4.1 Datasets: Randomly sampled populations on toy manifolds

To showcase GAGA's ability on transporting distributions on manifolds, we generate four toy manifolds: ellipsoid, torus, saddle, and hemisphere in $\mathbb{R}^3$. To simulate starting and ending distributions, we first randomly sample two points on the manifold as the starting and ending center and then sample N points near these selected centers. We compute and visualize the flow paths between the two distributions.

G.5 Hyperparameters

We chose our hyperparameters using grid search on the validation sets. The hyperparameters we used for our experiments are the following:

For autoencoder training, both the encoder and decoder are multi-layer MLPs with hidden dimensions [256, 128, 64], [64, 128, 256] respectively. Each intermediate fully connected layer is followed by a spectral normalization layer, a batch normalization layer, a ReLU layer, and a dropout layer with 0.2 dropout probability.

For training the discriminator $s(x)$, we use a multi-layer MLP with hidden dimensions [256, 128, 64], and each intermediate fully connected layer is followed by a spectral normalization layer, a batch normalization layer, and a ReLU layer.

For the geodesic-guided flow matching model, we use a multi-layer MLP with hidden dimensions [192, 192, 192] for curve parameterization and a multi-layer MLP with hidden dimensions [64, 64, 64] for the flow matching model.

All models were trained with AdamW optimizer with learning rate 1e-3 and 1e-4 weight decay. The autoencoder was trained with 200 maximum epochs, the discriminator with 100 maximum epochs, and geodesic-guided flow matching with 100 maximum epochs. We used early stopping for all models, and the patience used is 50.

We used the same set of loss weights in all experiments reported: $\lambda_1 = 77.4$, $\lambda_2 = 0.32$, $\zeta = 0.5$ for autoencoder loss (Eqn. (4)). $\beta = 10$ for the extended embedding (Eqn. (5)). For volume-guided generation, we used $\lambda = 10$ (Eqn. (6)). For geodesic-guided flow matching, we used $\lambda_3 = 1$ and $\lambda_4 = 1$ (Eqn. (9)).

For applying GAGA on new datasets, we recommend starting with a relatively larger λ_1 and a smaller λ_2 for training the autoencoder. We found that $\lambda_1 = 77.4$ and $\lambda_2 = 0.32$ generally work well for single cell datasets. The much smaller λ_2 encourages the neural network to focus more on learning a good latent space instead of reconstructing the original signal since learning a latent space that preserves manifold distances is much more challenging than reconstruction. In addition, biological data are often very noisy, so better reconstruction does not necessarily aid in learning better representations. The decay parameter ζ encourages the latent space to focus more on matching local distances. We recommend starting with a relatively large β for the extended embedding and a large λ for volume-guided generation since it would place a significant penalty when generated points stray off from the manifold. In practice, we found $\beta = 8$ and $\beta = 10$ both work well in our experiments. For the geodesic-guided population transport, we recommend starting with equal λ_3 and λ_4 since we want to learn both the flow and the geodesic transportation path.

H Additional Experiment Results

H.1 Geometry-aware autoencoder under increasingly noisy data

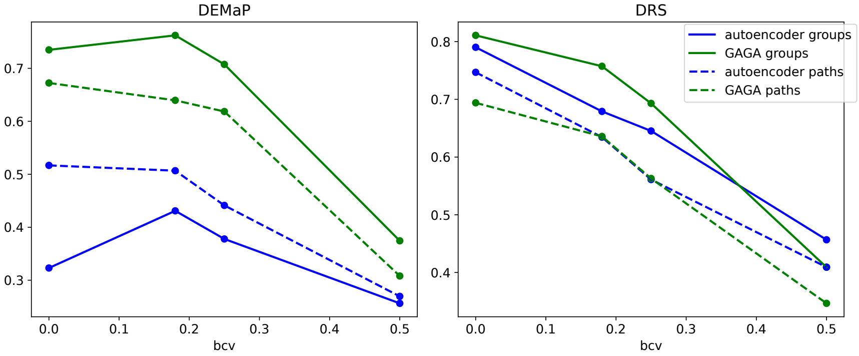

Figure Appx. 1: Comparison for GAGA and standard autoencoder on increasingly noisy single-cell datasets.

In Figure Appx. 1, we observe that GAGA consistently outperforms standard autoencoder on DEMaP under increasingly noisy sinle-cell data simulated with increasing bcv parameter. Moreover, we can see that GAGA generally rivals the standard autoencoder on DRS, indicating our distance-matching loss does not detract from data reconstruction.

H.2 Visualizing GAGA's latent embeddings

Qualitatively, we visualize the latent embeddings of GAGA on real-world scRNA-seq dataset EB, embryoid body data generated over 27 day time course (Moon et al., 2019). We show that GAGA is able to capture geometric structures in the data, which are essential for biological insights and interpretations. In addition to PHATE, we trained GAGA with two other geodesic distances obtained under different settings of HeatGeo (Huguet et al., 2024). We can see from Figure Appx. 2 that GAGA captures both local and global geometric structures such as clusters, branches, and paths. Moreover, Figure Appx. 2 shows that GAGA can match closely with the embedding method that it's based on, preserving the latent space of the original dimension reduction method and, at the same time, capable of generalizing to unseen points.

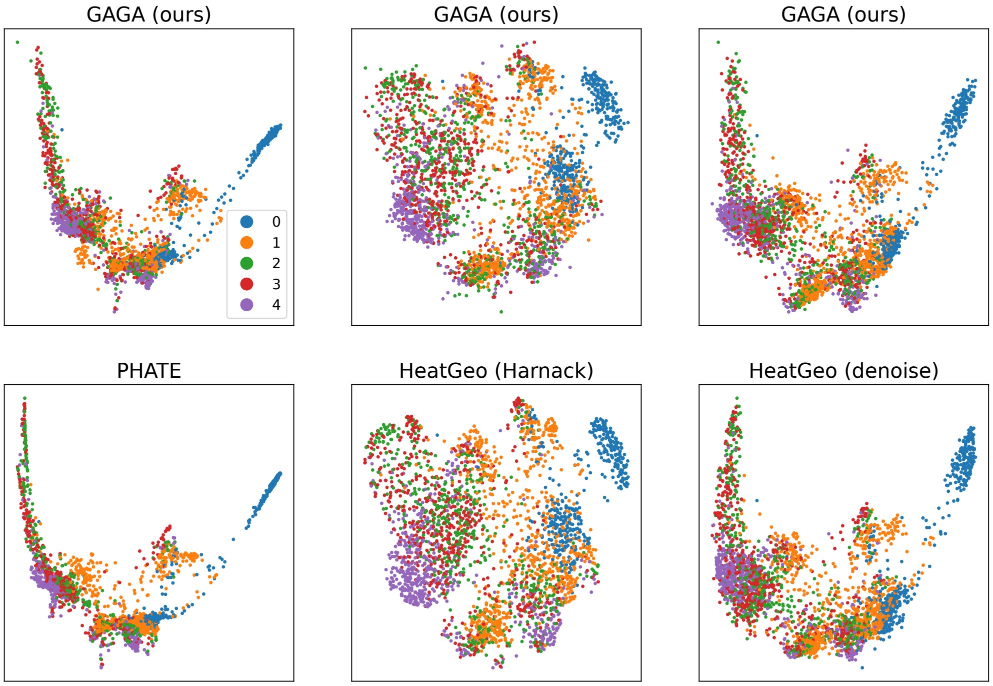

Figure Appx. 2: Visualization of the embedding shows GAGA preserves local and global structures.

H.3 Volume-guided Generation on Manifold

In Figure Appx. 3 (B,C,D) we show that the densities of the points generated by GAGA are closer to the ground truth volume elements compared to the original data points, indicating that GAGA largely reduces data imbalance. In addition, Figure Appx. 3 (A) shows that the generated points stay on the data manifold and cover the sparse regions well in the original data.

H.4 Comparing Volume-Guided Generation with On-Manifold Generation

To assess the faithfulness of generated points to the data geometry, we compared our method with Riemannian Flow matching (RFM) (Chen and Lipman, 2023). Notably, found that RFM only supports a number of specific manifolds in their implementation and that among the manifolds we conducted our experiment on, it only supports the hemisphere (supported through the sphere implementation). We used the hyperparameters for the sphere manifold (the volcano experiment) in their codebase. We present the comparison result in the following table:

Data	R	R^2
Original	-0.26	0.07
RFM Generated	-0.15	0.02
GAGA (Ours) Generated	0.85	0.71

Table Appx. 3: Correlation R and R^2 between the density and the volume element, where larger R and R^2 indicate more faithful generation along the data geometry.

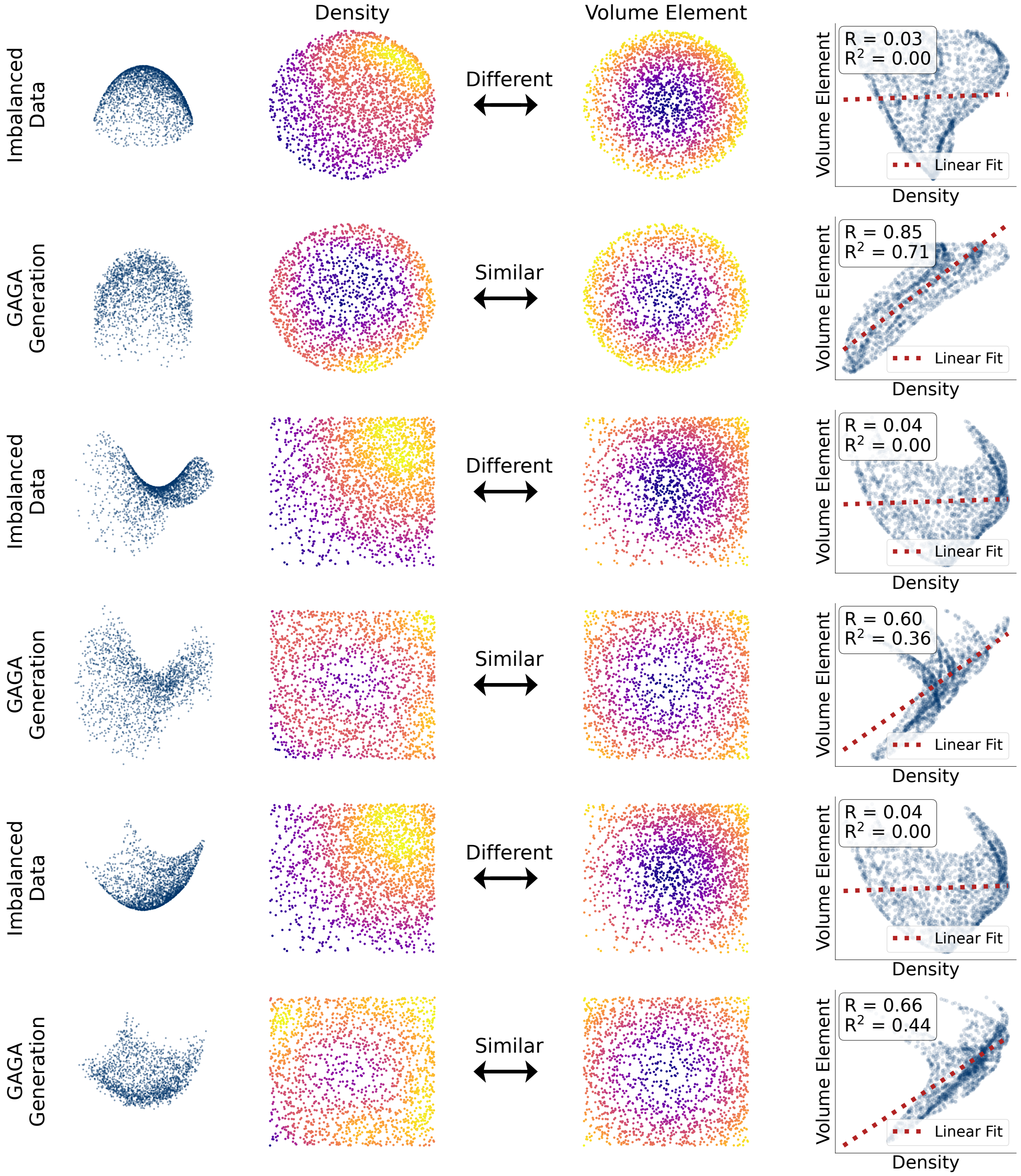

Figure Appx. 3: Geometry-aware generation with GAGA on hemisphere saddle, and paraboloid. **(A)** Generated points remain on the manifold, and are more evenly distributed compared to raw data. **(B)** Kernel density estimation. **(C)** Ground truth volume elements computed analytically. **(D)** In raw data, density does not correlate to volume element, indicating data imbalance. GAGA generation corrects the imbalance indicated by higher correlation between volume element and density.

We computed correlation R and R^2 between the density and the volume element, where larger R and R^2 indicate more faithful generation along the data geometry. Here "Original" refers to the original unbalanced dataset on which the models are trained. "RFM Generated" is the data generated by RFM, and "GAGA (Ours) Generated" is the data generated by our method.

Please refer to Appendix G.2.2 for detailed descriptions of how we generated the original dataset and computed the evaluation metrics.

We observe that the RFM generated data exhibit weak correlations with the volume element, similar to the original unbalanced data. This occurs because the flow matching model learns the density of the training data rather than its geometry, making it unable to address sampling bias effectively. This limitation is illustrated in Figure 2.

H.5 Geodesic computation in noisy data setting

To better demonstrate the effectiveness of our method, especially in noisy data settings, we compared our method to Dijkstra's algorithm in a setting of noise=0.7, dimension=15. We computed the mean squared error of geodesic lengths over 20 pairs of starting/ending points. To get a rigorous sense of significance, we used a Wilcoxon signed-rank test to compute the p-values (the null hypothesis is that the errors of the two methods are the same).

Dataset	**GAGA (Ours)**	**Dijkstra**	**p-value**
Ellipsoid	0.22	0.79	4.22e-03
Hemisphere	2.25	5.67	3.22e-04
Saddle	2.73	6.34	1.99e-03
Torus	0.93	2.14	0.73

Table Appx. 4: Mean squared error of geodesic lengths of GAGA (Ours) vs. Dijkstra across different datasets under 0.7 noise scale and 15 dimensions.

We observe that our method significantly outperforms Dijkstra's algorithm across all manifolds except the torus.

Beyond the quantitative benchmarks, we would like to emphasize several fundamental advantages of our method over Dijkstra's algorithm: 1) point generation: GAGA generates new points along the geodesic, whereas Dijkstra's algorithm only connects existing points. 2) smoothness: GAGA learns smooth curves, while the curves produced by Dijkstra's algorithm are discrete and prone to jittering, especially in the presence of noisy data. 3) geodesic insights: The smoothness of GAGA-generated curves allows us to compute other geometric quantities, such as velocities, providing valuable insights into the underlying manifold. For instance, we can compute the curvature of the geodesic.

H.6 Visualizing geodesics on toy manifolds

Figure Appx. 4 shows the geodesics of different methods on the same set of starting and ending points on multiple toy manifolds. Each row corresponds to one manifold and each column corresponds to one method. From left to right column, the method is 1) ground truth, 2) GAGA, 3) local metric, 4) density regularization. Density refers to geodesics learned with using density regularization.

We can see that GAGA generally outperforms all the other methods except Djikstra's on the saddle datasets. Directly using the local metric performs the worst, lagging far behind all other methods. The inferior performance of the local metric again illustrates the challenges of staying on the manifold while optimizing for the shortest path.

H.7 Single-cell trajectory inference

Single-cell trajectory inference, a central task in cellular dynamics, aims to predict the continuous trajectories of cells over time. Specifically, we conducted left-one-timepoint-out experiment in which cells at one specific timepoint were excluded, and the goal is to predict the left-out cells using the cells from the remaining timepoints (Tong et al., 2020).

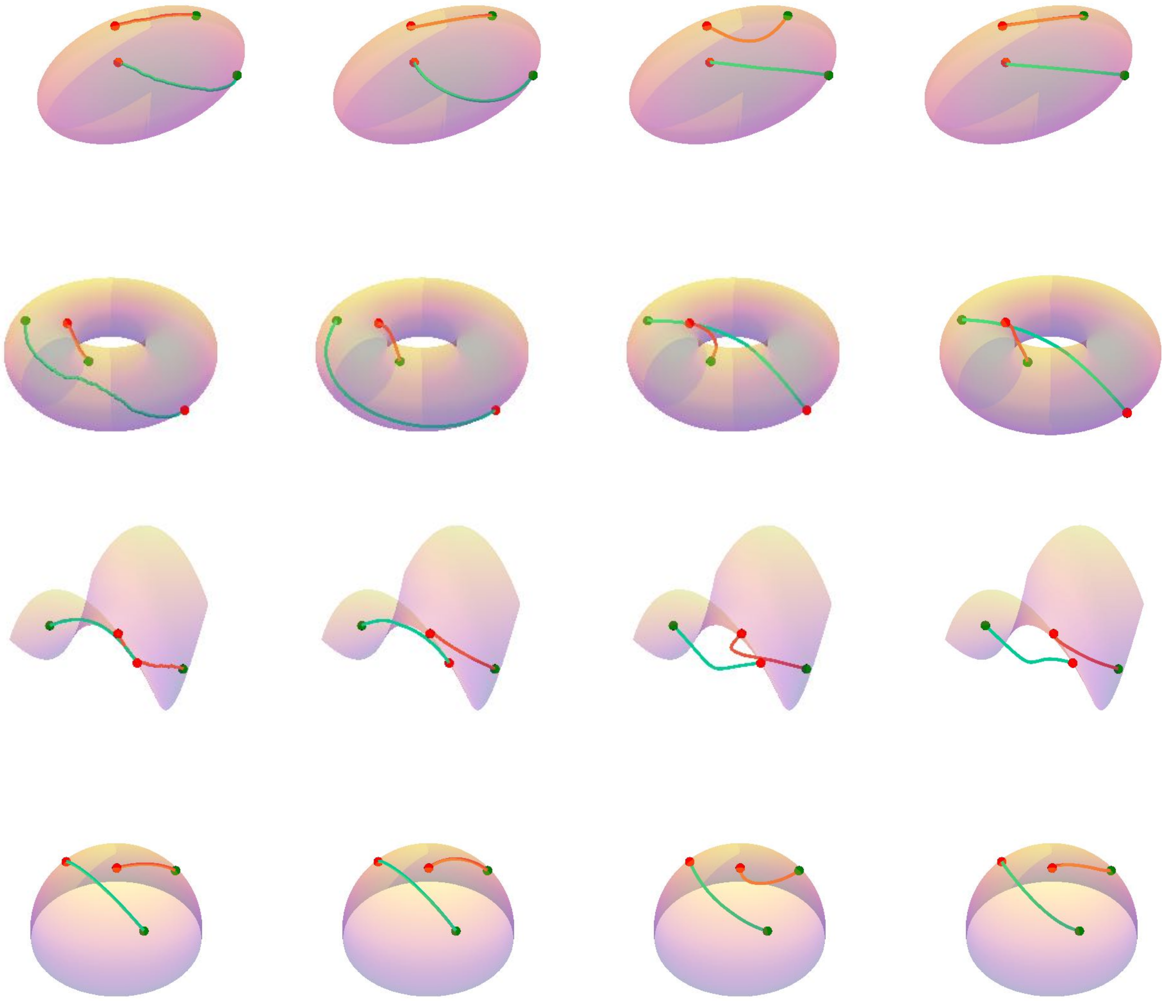

Figure Appx. 4: Comparison of geodesics. From left to right columns: 1) ground truth, 2) GAGA, 3) local metric, 4) density regularization.

We repurposed the Cite and Multi single-cell datasets from the Multimodal Single-cell Integration Challenge at NeurIPS 2022 (Burkhardt et al., 2022). Following the experiment setup in (Tong et al., 2024c), we trained and evaluated GAGA 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.

To perform left-one-timepoint-out experiment, we excluded day 3 and day 4, respectively, and used the remaining cells to infer the left-out populations. The train and test split ratio is 9:1, and the left-out timepoint was excluded from the training set. Our models were trained on the training set and evaluated on the test set. To reconstruct the left-out cells X_t at time t in the test set, GAGA generates the population level trajectories between X_{t-1} and X_{t+1} in the test set, and we use the points generated along the trajectories as the predicted cells $\hat{X}_t$. We ran experiments on 50 and 100 PCA dimensions of cells and the average Wasserstain-1 distance across the left-out timepoints was reported. The numbers listed for other methods were taken from the corresponding work.