ENERGY GUIDED GEOMETRIC FLOW MATCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

A useful inductive bias for temporal data is that trajectories should stay close to the data manifold. Traditional flow matching relies on straight conditional paths, and flow matching methods which learn geodesics rely on RBF kernels or nearest neighbor graphs that suffer from the curse of dimensionality. We propose to use score matching and annealed energy distillation to learn a metric tensor that faithfully captures the underlying data geometry and informs more accurate flows. We demonstrate the efficacy of this strategy on synthetic manifolds with analytic geodesics, and interpolation of cell trajectories from single-cell RNA sequencing data.

1 INTRODUCTION

Generative models are the workhorse of modern AI, enabling us to sample from complex data distributions. Methods based on diffusion have recently excelled at mapping from Gaussian noise distributions to data manifolds, and flow models enable mapping from any noise distribution to the manifold. Typically, these methods are interested in the trajectory that an individual sample takes, from noise to data, only because it enables straightforward simulation with an ODE or SDE solver (Yang et al., 2024). However, for temporal data one may be interested in these trajectories themselves, for example in the context of single-cell trajectory inference.

Recently, more focus has been given to parameterize the trajectories along a manifold, e.g. with fitting expressive Neural ODEs. Early methods focused on the simulation-based setup that required differentiation through an ODE Solver (Tong et al., 2020). However, the improved efficiency of training simulation-free methods such as flow matching (Lipman et al., 2022; Tong et al., 2023) has granted simpler training to these methods.

We are especially interested in the application of inferring cell trajectories in single-cell genomics. Currently, many datasets are collected at a single timepoint, and temporal models must resort to "pseudotime" methods to artificially create multiple timepoints, but a growing number of datasets have subsets of cells measured at different times. In such settings, accurate recovery of manifold geometry is critical for characterizing cellular development and disease processes.

Our contributions are three-fold: (i) a novel combination of score-matching and annealed energy distillation to parameterize a metric tensor that characterizes the underlying data manifold; (ii) a variant of stratified sampling to robustly infer geometry from density, mitigating failures on disconnected components; and (iii) practical application of these metrics to synthetic manifolds with known geodesics and single-cell RNA trajectory inference.

2 SETUP

2.1 DENOISING SCORE MATCHING

The score matching objective (Song et al., 2020) is the backbone of modern diffusion models, providing an efficient way to learn a family of score functions corresponding to the data distribution under different levels of noise. For a full diffusion model parameterizing density over time, the model takes as input a time variable, but in our setting we consider a monotonic noise schedule of

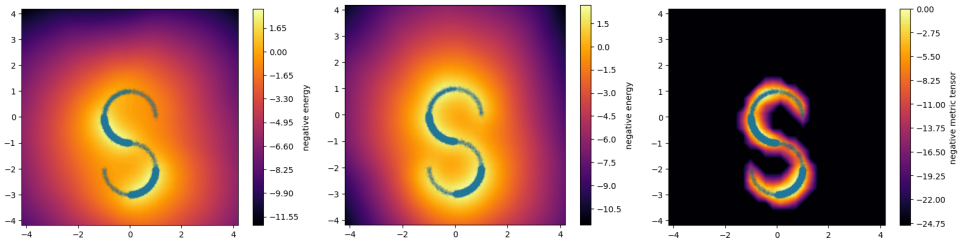

Figure 1: A visualization of our method. We fit an initial score and energy (left), then use annealing and self-normalized importance sampling to fit an updated energy that is less biased by unequal density in the data (middle), and we clip the energy to calculate a balanced metric tensor that captures the manifold (right).

$\sigma_{min} < \cdots < \sigma_{max}$ and simply condition on noise. In this case, the score matching loss is given by

$$\mathcal{L}_{score}(\theta) = \mathbb{E}_{x\sim p_{\text{data}},\epsilon\sim\mathcal{N}(0,I)}\left[\sum_i \left\| s_\theta(x+\epsilon\sigma_i,\sigma_i) + \frac{\epsilon}{\sigma_i}\right\|^2\right]$$

which yields $s_\theta(x,\sigma_i) \approx \nabla \log p_{\sigma_i}(x)$ where $p_{\sigma_i}(x)$ is the density of p_{data} convolved with Gaussian density of standard deviation σ_i.

There are many equivalent formulations of this loss that are more practical for optimization, where instead of learning s_θ directly one can instead learn to predict the noise via the loss function

$$\hat{\mathcal{L}}_{score}(\theta) = \mathbb{E}_{x\sim p_{data},\epsilon\sim\mathcal{N}(0,I)}\left[\sum_i \|\epsilon_\theta(x+\sigma_i\epsilon,\sigma_i)-\epsilon\|\right]^2$$

and approximate the score by $\nabla \log p_{\sigma_i}(x) \approx -\frac{\epsilon_\theta(x)}{\sigma_i}$. We will rely on this loss in all of our subsequent experiments.

2.2 Conditional Flow Matching

Given two measures, conditional flow matching (Tong et al., 2023; Lipman et al., 2022) learns a vector field to map one distribution p_0 to another distribution p_1. A general form of this loss is given in Albergo et al. (2023) where one considers a coupling of π of the two measures and a parameterization of paths $x_t := x_t(x_0,x_1)$ that defines a conditional density $p_t(\cdot|x_0,x_1)$. Then optimizing the loss

$$L_{flow}(\vartheta) = \mathbb{E}_{t\sim U([0,1]),(x_0,x_1)\sim\pi}\left\| v_\vartheta(x_t) - \frac{d}{dt}x_t\right\|^2$$

guarantees that v_ϑ solves the continuity equation with the marginal density p_t. In this way, one can numerically integrate v_ϑ to sample along trajectories from p_0 to p_1. When the goal is sampling, a common choice is $x_t = (1-t)x_0 + tx_1 + \sigma_{flow}\epsilon$ where $\epsilon \sim \mathcal{N}(0,I)$ such that $p_t(x_0,x_1)$ is a Gaussian centered along the straight line between x_0 and x_1.

2.3 Stochastic Interpolants along Geodesics

When a prior is available, flow matching trajectories may be intentionally biased towards a known or inferred manifold. Namely, given a metric tensor $\mathbb{G}(x)$, one may parameterize conditional paths between x_0 and x_1 of the form,

$$x_t := x_t(x_0,x_1) = (1-t)x_0 + tx_1 + t(1-t)\psi(x_0,x_1,t) + \sigma_{flow}\epsilon,$$

which are guaranteed to be geodesics after minimizing the loss,

$$\mathcal{L}_{geodesic}(\psi) = \mathbb{E}_{t\sim[0,1],(x_0,x_1)\sim\pi}\left\|\dot{x}_t\right\|^2_{\mathbb{G}(x_t)}.$$

This characterization was first introduced in Kapusniak et al. (2024). Given satisfactory geodesics, one can make this choice of conditional paths and use the same conditional flow matching loss as usual.

3 Related Work

3.1 Flow Matching

Flow matching has rapidly evolved since its original formulations (Lipman et al., 2022; Albergo et al., 2023; Liu et al., 2022). In particular, research has focused on incorporating prior knowledge into the the definition of the conditional flows, based on optimal transport (Tong et al., 2023; Pooladian et al., 2023), minimal curvature (Rohbeck et al., 2025), and user-defined potential functions (Neklyudov et al., 2023), among others. While these methods enrich interpolation flexibility, they rely on Euclidean path parameterization or external priors, rather than directly learning a data-driven metric tensor.

3.2 Energy distillation from Score Matching

Although the primary application of score matching continues to be diffusion models (Song et al., 2020), several methods have focused on using score matching as a tool to distill an accurate energy of the data. Thornton et al. (2025) suggest a specific parameterization of the energy used for sequential Monte Carlo sampling, and Akhound-Sadegh et al. (2025) infer a score and energy simultaneously for annealed sampling. These methods, however, fail to connect energy to manifold geometry or flow interpolation.

3.3 Learning Geodesics along Data-driven Manifolds

Flow matching was initially extended to trajectories on manifolds by Chen & Lipman (2023) with a focus on settings where geodesics could be exactly or approximately computed. This was extended further by Kapusniak et al. (2024) allowing for learning of geodesics by minimizing energy with respect to some inferred metric tensor.

Other methods consider interpolation in a latent space and map back to ambient space to learn a curved path (Palma et al., 2025). de Kruiff et al. (2024) also considered learned metric tensors in latent space through a pullback, although they require simulation through a Neural ODE solver to train.

The methods most related to the current work also consider how to infer a non-trivial metric tensor from the data manifold to derive plausible geodesics and train simulation-free flows. Kapusniak et al. (2024) learn a metric tensor based on the linear combination of RBF kernels proposed in Arvanitidis et al. (2020). Similarly, Sun et al. (2024) learn a pullback metric using the pairwise distances inferred from the embedding method PHATE (Moon et al., 2019).

Additionally, so-called Fermat distances (Groisman et al., 2022) that define a metric tensor through a density have been studied before in more limited contexts, primarily for 2d inference of single geodesics in Sorrenson et al. (2024) and image interpolation in Yu et al. (2025). In contrast, we learn an adaptive energy metric tensor with annealing in our setting, and apply it to the task of learning general geodesics.

4 Methods

We propose a multi-stage training procedure that yields a geometrically faithful flow-matching trajectory aligned with the data manifold. First, given samples $x \sim p_{\text{data}}$, we estimate a density $p_\theta(x)$ by jointly learning the score $s_\theta(x) = \nabla_x \log p_\theta(x)$ and the energy $E_\zeta(x)$ via score- and energy-matching. This estimation is iteratively refined to better align p_θ with p_{data}. Next, we construct a metric tensor $\mathbb{G}$ from the learned energy, and jointly learn geodesics and the associated geodesic distance d under $\mathbb{G}$. Finally, we compute an optimal transport coupling under the cost defined by d and train a flow-matching vector field to realize the resulting displacement interpolation, yielding a flow that respects the learned manifold geometry.

4.1 SCORE AND ENERGY MATCHING

We first aim to learn the shape of the manifold through score and energy matching in the form of the data generating density. Formally, denote the score matching network as $s_\theta(x, \sigma)$ where θ is the set of learnable parameters and σ is the noise scale. Denote noised data at noise scale σ_i as $y = x + \epsilon\sigma_i$. We perform denoising score matching by minimizing,

$$\mathcal{L}(\theta) = \mathbb{E}_{x\sim p_{\text{data}}, \epsilon\sim\mathcal{N}(0,I)} \left[\sum_i \left\| s_\theta(y, \sigma_i) + \frac{\epsilon}{\sigma_i \cdot \beta} \right\|^2 \right],$$

where the score net $s_\theta(\cdot, \sigma)$ takes noised data from various noise levels and conditions on an embedding of the noise scales. β is the a temperature scalar. The log density of the learned generating distribution is then given by $s_\theta(\cdot, \sigma) \approx \nabla_x \log p_\theta(x)$. We can then recover energy as,

$$E(x) \approx -\log p_\theta(x),$$

which can be learned by optimizing

$$\mathcal{L}(\zeta) = \mathbb{E}_{x\sim p_{\text{data}}, \epsilon\sim\mathcal{N}(0,I)} \left[\sum_i \left\| \nabla E_\zeta(y, \sigma_i) + s_\theta(y, \sigma_i) \right\|^2 \right]$$

where ζ is the set of parameters from the energy network $E_\zeta(., \sigma)$ which conditions on the noise embedding and takes in noised data.

4.2 ITERATIVE DENSITY REFINEMENT

The primary issue with simply using the score is imbalance in data density. To better align the learned density p_θ to the underlying geometry, we repeat score and energy matching multiple times while refining the learned density with the following operations to achieve stable and annealed density estimation.

Self normalized importance sampling (SNIS). One important component of these refinement steps is self normalized importance sampling. At refinement step k, the reweighting is,

$$\hat{p}^k_{\text{data}}(x) \propto \frac{p_{\text{data}}(x)}{\exp(-\beta_w \text{clip}(E^k_\zeta(x, \sigma_{min})))},$$

where β_w is a hyperparameter and the clipping is based on energy quantiles. We use this reweighting over our data, which is closer to uniform, in subsequent steps.

Density annealing. Aside from reweighting, we also give the option to anneal the density based on the previous step density and temperature. Let $y = x + \epsilon\sigma_i$, at refinement step k, the density annealing loss is given by,

$$\mathcal{L}^k(\theta) = \mathbb{E}_{x\sim\hat{p}^k_{\text{data}}, \epsilon\sim\mathcal{N}(0,I)} \left[\sum_i \alpha \left\| s^k_\theta(y, \sigma_i) + \frac{\epsilon}{\sigma_i \cdot \beta_k} \right\|^2 + (1-\alpha) \left| s^k_\theta(y, \sigma_i) - \frac{\beta_k}{\beta_{k-1}} s^{k-1}_\theta(y, \sigma_i) \right| \right]$$

when $\alpha = 1$ this is equivalent to optimizing denoising score matching with current temperature β_k at step k.

The iterative density refinement is possible due to the scalability and efficiency of score/energy matching models. Our final energy estimation is based on the refined density after K steps.

Stratified Sampling. One shortcoming of score matching is learning the correct normalization of isolated components (Wenliang & Kanagawa, 2020). Intuitively, this presents a challenge for using the score to induce a metric tensor as even a very accurate score may assign unequal densities to identical but separated components. We consider a novel approach for addressing this issue by introducing clustering and stratified sampling (Owen, 2013).

Let p^* denote the uniform distribution on the underlying data manifold. Since our ultimate goal is to learn a density that is close to p^*, and typical notions of distances between measures like

integral probability metrics are based on integrating against test functions, we equivalently want a low-variance estimator for $\mathbb{E}_{x\sim p^*}[g(x)]$ for any test function g.

Thus, we consider a form of Rao-Blackwellization by first clustering the underlying data into clusters $C_1, \dots C_J$, doing score matching and annealing on each cluster component independently, before finally reweighting to learn a score over the entire dataset. Formally, this loss is

$$\mathcal{L}^k(\theta) = \mathbb{E}_{j\sim[J], x\sim\hat{p}^k_{\text{data}}(\cdot|C_j), \epsilon\sim\mathcal{N}(0,I)}\left[\sum_i s^k(x+\epsilon\sigma_i, \sigma_i, j) + \frac{\epsilon}{\sigma_i}\right]$$

When fully trained, this loss yields a score function that can be conditioned on any individual cluster, with a respective energy that can be distilled per cluster as well. The utility of this clustering is that, because the score can be better approximated per local cluster, the SNIS estimator for a uniform distribution on that cluster is more accurate, and therefore the combined reweighting across all clusters will be closer to uniform. Explicitly, we use the normalization

$$\hat{p}^k_{\text{data}}(x, j) \propto \frac{p_{\text{data}}(x)}{\exp(-\beta_w \text{clip}(E^k_\zeta(x, \sigma_{min}, j)))},$$
$$\hat{p}^k_{\text{data}}(x) = \sum_{j=1}^{J} \frac{\hat{p}^k_{\text{data}}(x, j)}{|C_j|}.$$

We use Leiden clustering (Traag et al., 2019) to infer our clusters, as is common in single-cell literature.

4.3 Metric Tensor and Learning Geodesic

Having estimated the energy E^K, we next construct a metric tensor based on the learned energy. More specifically, the metric tensor is,

$$\mathbb{G}(x) = \gamma + \text{clip}(\lambda \exp(E^K(x)))$$

where $E^K(x)$ is the learned energy after K refinement steps, and we clip according to quantiles of the energy taken as hyperparameters, as specified in the Appendix. Similar to Kapusniak et al. (2024), we penalize trajectories passing through off-manifold regions with low data density by writing trajectory as,

$$x_t|x_0, x_1 = (1-t)x_0 + tx_1 + t(1-t)\psi(x_0, x_1, t) + \sigma_{flow}\epsilon$$

where ψ is a neural network that aims to approximate the geodesic given the metric tensor. We learn ψ by energy minimization, i.e. minimizing

$$\mathcal{L}(\psi) = \mathbb{E}_{t\sim\mathcal{U}[0,1],(x_0,x_1)\sim\pi}\left[\|\dot{x_t}\|^2_{\mathbb{G}(x_t)}\right]$$

where $\dot{x_t}$ is the time derivative of x_t.

4.4 Distance Learning and Flow Matching

We use metric learning to obtain a distance that is consistent with the metric tensor defined. More specifically, we learn an isometric embedding $f : (\mathcal{M}, \mathbb{G}) \mapsto (\mathbb{R}^d, <.,.>_{\mathbb{R}^d})$ that maps defined geodesic into a straight-line, constant-speed segment in Euclidean space. To achieve this, we optimize

$$\mathcal{L}(f) = \mathbb{E}_{t\sim\mathcal{U}[0,1],(x_0,x_1)\sim\pi}\left[\|\partial_t f(x_t)\|^2 - \|\dot{x_t}\|^2_{\mathbb{G}(x_t)}\right]$$

where π is the optimal coupling with the learned distance $d(x_0, x_1) = \|f(x_0) - f(x_1)\|$. Finally, flow matching is used to parameterize paths that satisfy the continuity equation. Denoting the flow network as $v(t, x_t)$, we optimize

$$\mathcal{L}(v) = \mathbb{E}_{t\sim\mathcal{U}[0,1],(x_0,x_1)\sim\pi,\epsilon}\left[\|v(t, x_t) - \dot{x_t}\|^2\right]$$

4.5 THEORETICAL ANALYSIS

In this section we show that with our stratified sampling approach, the model learns a density that is a mixture among the clusters. Let $\{C_j\}_{j=1}^J$ be a disjoint partition of a data manifold $\mathcal{M}$. Our aim is to estimate $\mu = \mathbb{E}_{p*}[g(x)]$ for some testing function $g(x)$. Assume cluster C_j has size n_j, the within-in cluster proposal distribution is given by

$$q_j(i) := \frac{r_i}{\sum_{k \in C_j} r_k}, \qquad i \in C_j$$

where $r_i = f(E(x_i)) > 0$ is a positive function of learned energy. Set $R_j = \sum_{k \in C_j} r_k$, if the target distribution is uniform on cluster rescaled by number of clusters, i.e.

$$p_j^* = \frac{1}{n_j} \cdot \frac{1}{J}, \quad \mu_j = \frac{1}{n_i} \sum_{i \in C_j} g(x_i)$$

The importance weight for $i \in C_j$ is

$$W_j(i) \;=\; \frac{p_j^*(i)}{q_j(i)} \;=\; \frac{R_j}{J\, r_i}$$

Given i.i.d. samples $I_1, \ldots, I_{m_j} \sim q_j$, the importance sampling estimator within cluster is:

$$\widehat{\mu}_j^{\text{SNIS}} \;=\; \frac{\sum_{t=1}^{m_j} W_j(I_t)\, g(x_{I_t})}{\sum_{t=1}^{m_j} W_j(I_t)}.$$

Across all clusters, our target and estimator becomes

$$p^* = \frac{1}{J} \sum_{j=1}^{J} \text{Unif}(C_j), \quad \widehat{\mu}^{\text{SNIS}} = \frac{1}{J} \sum_{j=1}^{J} \widehat{\mu}_j^{\text{SNIS}}$$

With this setup we can introduce the following theorem to show estimation convergence:

Theorem 4.1. *Assume for each cluster C_j, $r_i > 0$ for all $i \in C_j$ and the importance weights have finite second moment under per-cluster measure q_j, as number of samples m_j in C_j goes to infinity we have*

$$\widehat{\mu}^{SNIS_j} \xrightarrow{p} \mu_j, \quad \widehat{\mu}^{SNIS} \xrightarrow{p} \mu$$

Additionally, if $\mathcal{L}(x;\theta)$ is the denoising score matching loss, the objective converges to $\mathbb{E}_{p^\star}[\mathcal{L}(x;\theta)]$

Proof. Since $W_j(i) \;=\; \frac{p_j^*(i)}{q_j(i)}$ and $I_j \overset{\text{i.i.d.}}{\sim} q_j$, then

$$\begin{aligned}\mathbb{E}_{q_j}[W_j(i) g(x_i)] &= \sum_{i \in C_j} q_j(i) \frac{p_j^*(i)}{q_j(i)} g(x_i) \\ &= \sum_{i \in C_j} p_j^*(i) g(x_i) = \frac{1}{J} \mu_j\end{aligned}$$

We also have

$$\mathbb{E}_{q_j}[W_j] = \sum_{i \in C_j} q_j(i) \frac{p_j^*(i)}{q_j(i)} = \frac{1}{J}$$

Hence under the law of large numbers, we have

$$\widehat{\mu}^{\text{SNIS}_j} = \frac{\sum_{t=1}^{m_j} W_j(I_t)\, g(x_{I_t})}{\sum_{t=1}^{m_j} W_j(I_t)} \xrightarrow{p} \frac{\mathbb{E}_{q_j}[W_j(I_t) g(x_i)]}{\mathbb{E}_{q_j}[W_j]} = \frac{(1/J)\mu_j}{(1/J)} = \mu_j$$

Since global estimator is given by averaging over clusters, we have

$$\widehat{\mu}^{\text{SNIS}} = \frac{1}{J} \sum_j \widehat{\mu}_j^{\text{SNIS}} \xrightarrow{p} \frac{1}{J} \sum_j \widehat{\mu}_j = \mu$$

Additionally, if we set $g(x_i) = \mathcal{L}(x_i; \theta)$ as the testing function, the SNIS estimator becomes a risk estimator

$$\widehat{\mathcal{R}}_j^{\text{SNIS}} = \frac{\sum_{t=1}^{m_j} W_j(I_t)\mathcal{L}(x_{I_t}; \theta)}{\sum_{t=1}^{m_j} W_j(I_t)}$$

Hence

$$\widehat{\mathcal{R}}_j^{\text{SNIS}} \xrightarrow{p} \frac{\mathbb{E}_{q_j}[W_j(I_t)\mathcal{L}(x_{I_t}; \theta)]}{\mathbb{E}_{q_j}[W_j]} = \frac{\sum_{i \in C_j} p_j^*(i)\mathcal{L}(x; \theta)}{\sum_{i \in C_j} p_j^*(i)} = \mathbb{E}_{p_j^*}[\mathcal{L}(x; \theta)]$$

Similar mapping from cluster specific estimator to global estimator holds, hence

$$\widehat{\mathcal{R}}^{\text{SNIS}} \xrightarrow{p} \mathbb{E}_{p^*}[\mathcal{L}(x; \theta)]$$

□

5 RESULTS

5.1 BASELINES

We consider comparison against two primary methods, namely CFM where the metric tensor is constrained to be the identity (and therefore geodesics and a distance embedding don't need to be learned), and MFM where the metric tensor is given by a conformal metric defined by a linear combination of RBF kernels as in Arvanitidis et al. (2020). We consider GAGA (Sun et al., 2024) as another worthwhile baseline, but unfortunately we were unable to reproduce and run their publicly available code due to issues with CUDA memory and heldout test data used in the definition of their graph-based metric tensor.

5.2 SYNTHETIC DATA

We first consider settings where we can obtain analytic, closed form characterizations of the geodesics to measure against the learned geodesics. In those cases, we measure using an Average Geodesic Error calculated as

$$\mathcal{L}_{AVE}(\gamma, \gamma^*) = E_{t \sim U(0,1), (x_0, x_1) \sim \mu \times \mu} \left\| \gamma(x_0, x_1, t) - \gamma^*(x_0, x_1, t) \right\|^2$$

where γ is the learned geodesic, γ^* is the analytic geodesic, and μ is some base measure on the underlying manifold, for example the uniform Haar measure.

We consider data sampled uniformly on the surface of a sphere in varying dimensions. Note that while this task is handled by RFM (Chen & Lipman, 2023) if the manifold is known a priori, learning it directly from the data requires parameterizing the geodesics, and therefore provides a setting to check how accurately our learned geometry matches the ground truth. We omit comparison against CFM, as the optimal solution is simply straight geodesics that ignore the geometry of the sphere entirely.

Table 1: Comparison of average geodesic error along sphere geodesics in varying dimensions.

Method	Dimension	AVE
MFM	10	$.21 \pm .07$
EGGFM	10	$\mathbf{.12 \pm .07}$
MFM	20	$.20 \pm .05$
EGGFM	20	$\mathbf{.19 \pm .05}$

We observe that EGGFM outperforms MFM in this synthetic setting where we can measure correct geodesics exactly. Visually, we can observe 2d slices of the energy of EGGFM (which is equivalent to the metric tensor up to rescaling) and the metric tensor of MFM in Figure 2. Empirically the energy-induced metric is accurate in projected space, while the MFM metric is parameterized through RBF kernels around individual points. Although we found the best performance for MFM

using a total of 2000 points to parameterize the metric, the curse of dimensionality makes it difficult to cover even a moderately high dimensional manifold this way, and hence most of the points have no intersection with the 2d plane.

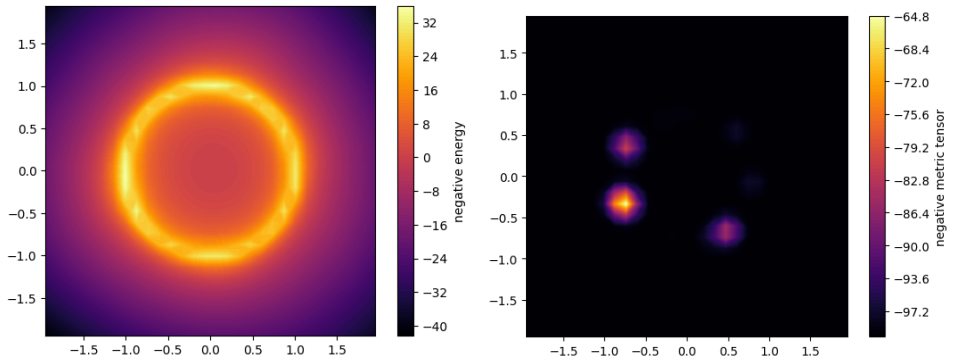

Figure 2: The negative energy of EGGFM (left) and negative metric tensor of MFM (right) on the 10 dimensional sphere, projected onto the first two dimensions.

5.3 Interpolation of single-cell RNA data

We then consider the case of interpolating temporal single-cell RNA data. We hold out single time points for testing and measure model performance with one Wasserstein distance.

5.3.1 Embryonic body dataset

We test our model on the Embryonic Body (EB) dataset introduced by Moon et al. (2019) and processed by Tong et al. (2020). The dataset consists of 5 time points over 30 days, which we denote with index 1-5. In our experiments, we hold out time points 2, 3, and 4 and train separate models on the full-time scale. We compare our model with conditional flow matching(CFM) and metric flow matching (MFM)(Kapusniak et al. (2024)). We compute distances in a PCA representation, a standard practice in single-cell RNA-seq analysis that reduces noise while extracting dominant biological variation.

Table 2: Wasserstein 1 distance averaged across left-out marginals(↓ better) for 5-dim PCA representation of the EB dataset. Results are averaged across 3 independent runs.

Method	W1 distance(↓)
CFM	0.711 ± 0.018
MFM	0.727 ± 0.042
EGGFM	**0.674 ± 0.039**

In Table 2, we observe significant performance improvement for EGGFM, which demonstrates the benefit of constraining learned trajectories to flow through the actual data manifold. In contrast, the weaker performance of MFM highlight the fact that even in relatively low-dimensional manifolds (5d), local RBF-kernel based estimation of the geodesic is insufficient to capture data geometry.

5.3.2 Hematopoietic stem cell dataset

We additionally apply our method to a low-dimensional projection of the CITE sequencing dataset (measuring RNA and protein features) of hematopoietic stem cells (Burkhardt et al., 2022). This dataset includes four timepoints on days 2, 3, 4, and 7, and therefore we can only evaluate on two heldout timepoints that have a distribution before and after. Our metrics in Table 3 show our method continues to show favorable results.

Table 3: Wasserstein 1 distance averaged across left-out marginals(↓ better) for 5-dim PCA representation of the CITE dataset. Results are averaged across 3 independent runs.

Method	W1 distance(↓)
CFM	0.538 ± 0.012
MFM	0.571 ± 0.021
EGGFM	**0.531 ± 0.010**

6 Conclusion

This work takes a step toward making flow-based generative models geometry-aware, by showing how score matching and energy distillation can be harnessed to recover an adaptive metric tensor that faithfully encodes manifold structure. We have demonstrated the utility of calculating a metric tensor through annealed score and energy weighting, applied to synthetic and genomic datasets. Currently, the main limitation of the work is in application to low-dimensional data, and future work aims to generalize other strategies for using generative methods to infer geometry that are more robust at scale. Additionally, the application to genomic data can be further buoyed by downstream analysis of the learned trajectories, in order to understand how the prior knowledge of the data manifold corroborates biological priors about how gene expression changes during differentiation or other temporal processes.

References

Tara Akhound-Sadegh, Jungyoon Lee, Avishek Joey Bose, Valentin De Bortoli, Arnaud Doucet, Michael M Bronstein, Dominique Beaini, Siamak Ravanbakhsh, Kirill Neklyudov, and Alexander Tong. Progressive inference-time annealing of diffusion models for sampling from boltzmann densities. *arXiv preprint arXiv:2506.16471*, 2025.

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

Georgios Arvanitidis, Søren Hauberg, and Bernhard Schölkopf. Geometrically enriched latent spaces. *arXiv preprint arXiv:2008.00565*, 2020.

Daniel Burkhardt, Malte Luecken, Andrew Benz, Peter Holderrieth, Jonathan Bloom, Christopher Lance, Ashley Chow, and Ryan Holbrook. Open problems - multimodal single-cell integration. `https://kaggle.com/competitions/open-problems-multimodal`, 2022. Kaggle.

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

Friso de Kruiff, Erik Bekkers, Ozan Öktem, Carola-Bibiane Schönlieb, and Willem Diepeveen. Pullback flow matching on data manifolds. *arXiv preprint arXiv:2410.04543*, 2024.

Pablo Groisman, Matthieu Jonckheere, and Facundo Sapienza. Nonhomogeneous euclidean first-passage percolation and distance learning. *Bernoulli*, 28(1):255–276, 2022.

Kacper Kapusniak, Peter Potaptchik, Teodora Reu, Leo Zhang, Alexander Tong, Michael Bronstein, Joey Bose, and Francesco Di Giovanni. Metric flow matching for smooth interpolations on the data manifold. *Advances in Neural Information Processing Systems*, 37:135011–135042, 2024.

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

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

Kevin R Moon, David Van Dijk, Zheng Wang, Scott Gigante, Daniel B Burkhardt, William S Chen, Kristina Yim, Antonia van den Elzen, Matthew J Hirn, Ronald R Coifman, et al. Visualizing structure and transitions in high-dimensional biological data. *Nature biotechnology*, 37(12):1482–1492, 2019.

Kirill Neklyudov, Rob Brekelmans, Alexander Tong, Lazar Atanackovic, Qiang Liu, and Alireza Makhzani. A computational framework for solving wasserstein lagrangian flows. *arXiv preprint arXiv:2310.10649*, 2023.

Art B Owen. Monte carlo theory, methods and examples, 2013.

Alessandro Palma, Sergei Rybakov, Leon Hetzel, Stephan Günnemann, and Fabian J Theis. Enforcing latent euclidean geometry in single-cell vaes for manifold interpolation. *arXiv preprint arXiv:2507.11789*, 2025.

Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky TQ Chen. Multisample flow matching: Straightening flows with minibatch couplings. In *International Conference on Machine Learning*, pp. 28100–28127. PMLR, 2023.

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

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Peter Sorrenson, Daniel Behrend-Uriarte, Christoph Schnörr, and Ullrich Köthe. Learning distances from data with normalizing flows and score matching. *arXiv preprint arXiv:2407.09297*, 2024.

Xingzhi Sun, Danqi Liao, Kincaid MacDonald, Yanlei Zhang, Chen Liu, Guillaume Huguet, Guy Wolf, Ian Adelstein, Tim GJ Rudner, and Smita Krishnaswamy. Geometry-aware generative autoencoders for warped riemannian metric learning and generative modeling on data manifolds. *arXiv preprint arXiv:2410.12779*, 2024.

James Thornton, Louis Béthune, Ruixiang Zhang, Arwen Bradley, Preetum Nakkiran, and Shuangfei Zhai. Composition and control with distilled energy diffusion models and sequential monte carlo. *arXiv preprint arXiv:2502.12786*, 2025.

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

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

Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):1–12, 2019.

Li K Wenliang and Heishiro Kanagawa. Blindness of score-based methods to isolated components and mixing proportions. *arXiv preprint arXiv:2008.10087*, 2020.

Ling Yang, Zixiang Zhang, Zhilong Zhang, Xingchao Liu, Minkai Xu, Wentao Zhang, Chenlin Meng, Stefano Ermon, and Bin Cui. Consistency flow matching: Defining straight flows with velocity consistency. *arXiv preprint arXiv:2407.02398*, 2024.

Qingtao Yu, Jaskirat Singh, Zhaoyuan Yang, Peter Henry Tu, Jing Zhang, Hongdong Li, Richard Hartley, and Dylan Campbell. Probability density geodesics in image diffusion latent space. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 27989–27998, 2025.

A Experimental Details

A.1 Synthetic Experiments

Table 4: Base configs for synthetic experiments. If not specified, all subsequent experiments used these hyperparameters as well

Hyperparameter	Value
Learning rate	0.0001
Hidden Dim	512
Number of Layers	4
Gradient Clipping	10
Score / Energy batch size	4196
Geodesic / Flow batch size	256
Frequencies for sinusoidal embedding of noise / cluster id	32
EMA decay	0.999
Annealing steps	2
Metric Scale (λ)	10
Number of noise scales in score matching	20
Min score matching noise(σ_{min})	0.01
Max score matching noise(σ_{max})	0.2
Metric constant (γ)	0.2
Weight beta (β_w)	0.3
Energy clip quantiles	[0.05, 0.98]
Metric clip lower quantile	0.05
Flow matching noise (σ_{flow})	0.1
Leiden n_neighbors	10
Leiden resolution	0.3

For all our architectures, we use MLPs to parameterize the distance embedding, geodesic and flow networks, and MLPs with residual connections for the score and energy networks. For energy we use the parameterization proposed in Thornton et al. (2025) and calculate $E(x) = \langle E_\theta(x), x\rangle$. We also use skip connections in the embedding network.

We consider the sphere in 10 and 20 dimensions, sampled uniformly with a total of 40000 points. For the sake of simplicity, we consider 20000 points in each of two timepoints, and train geodesics, distance and flow losses with a product measure coupling in order to learn geodesics over the entire sphere.

We compare against CFM with comparable parameters using the same architectures as well as MFM, which only requires additional parameters for number of clusters $K = 2000$ and regularization $\kappa = 0.5$, as per Kapusniak et al. (2024).

A.2 EB Experiments

For EB experiments, we use the following configuration for our model:

Table 5: Configuration for EB model

Hyperparameter	Value
Epochs(score net)	500
Epochs(energy net)	3000
Epochs(embedding net)	2000
Epochs(flow net)	2000
Number of layers	5
Annealing steps	3
Metric scale (λ)	4
Min temperature	5.0
Max temperature	10.0
Min score matching noise(σ_{min})	0.1
Max score matching noise(σ_{max})	0.2
Flow matching noise (σ_{flow})	0.05
Leiden resolution	0.0

For metric flow matching, we used 500 clusters with $\kappa = 0.1$. To recreate MFM's metric tensor, we skip the score and energy network training. We train the embedding and flow networks with identical hyperparameters. For conditional flow matching, the metric tensor is the identity matrix, hence we also skip the score and energy networks. The rest of the model is trained identically. The data is available at https://data.mendeley.com/datasets/hhny5ff7yj/1 and preprocessed according to the same code as Tong et al. (2020).

A.3 CITE Experiments

We use the CITE donor RNA-seq data publicly available at https://data.mendeley.com/datasets/hhny5ff7yj/1. The data is already preprocessed, so we simply map to 5PCs and whiten each PC. As with EB, we compare against MFM on a performant parameters with $K = 2000$ clusters and $\kappa = 1.0$.

Table 6: Configuration for CITE model

Hyperparameter	Value
Epochs(score net)	500
Epochs(energy net)	3000
Epochs(embedding net)	500
Epochs(flow net)	2000
Number of layers	4
Annealing steps	2
Metric scale (λ)	1
Min temperature	1.0
Max temperature	1.0
Min score matching noise(σ_{min})	0.02
Max score matching noise(σ_{max})	0.3
Flow matching noise (σ_{flow})	0.2
Energy clip quantiles	[0.05, 0.95]
Metric clip lower quantile	0.05
Metric constant (γ)	0.5
Weight beta (β_w)	0.2