(Deep) Generative Geodesics

Beomsu Kim [*1] **Michael Puthawala** [*2] **Jong Chul Ye** [1] **Emanuele Sansone** [3]

Abstract

In this work, we propose to study the global geometrical properties of generative models. We introduce a new Riemannian metric to assess the similarity between any two data points. Importantly, our metric is agnostic to the parametrization of the generative model and requires only the evaluation of its data likelihood. Moreover, the metric leads to the conceptual definition of generative distances and generative geodesics, whose computation can be done efficiently in the data space. Their approximations are proven to converge to their true values under mild conditions. We showcase three proof-of-concept applications of this global metric, including clustering, data visualization, and data interpolation, thus providing new tools to support the geometrical understanding of generative models.

1. Introduction

What does it mean for two data points to be similar, what is the correct notion of similarity, and how can we measure similarity with generative models? Deep generative models work by effectively learning a generative map from a 'simple' latent distribution, typically a Gaussian distribution in low-dimensional Euclidean space, to an empirical distribution in data space, called the data distribution.

Several works have locally studied the geometry of the data distribution by using the notion of the pullback metric, which requires access to the latent representation of a deep generative model and the Jacobian of the generative map. This enables (i) the control and manipulation of the generation process in the latent space (Arvanitidis et al., 2018; Chen et al., 2018; Shao et al., 2018; Arvanitidis et al., 2021; 2022; Lee et al., 2022b;a; Lee & Park, 2023; Issenhuth et al., 2023; Ramesh et al., 2019; Zhu et al., 2021; Choi et al., 2022; Park et al., 2023; Song et al., 2023), and (ii) the investigation of the behavior of deep generative models, for instance through latent interpolation (Berthelot et al., 2019; Zhu et al., 2020; Shao et al., 2018; Michelis & Becker, 2021; Arvanitidis et al., 2018; Chen et al., 2018; Laine, 2018; Struski et al., 2023).

In this work, we go beyond the study of the local geometry of the data distribution and instead propose a global metric, which is agnostic to the internal parametrization of the deep generative model and only requires the evaluation of its data likelihood. This allows the introduction of two new concepts: *generative distances* and *generative geodesics*. Generative distances enable to assess the similarity between any two points in the data space according to the generative model, whereas generative geodesics identify the corresponding connecting path. Notably, we show that generative distances and geodesics can be efficiently approximated by first discretizing the ambient space using a graph constructed from training and/or synthetic data, by second attaching weights to this graph based on the proposed metric and then using shortest-path algorithms over the constructed weighted graph to overcome the curse of dimensionality. Moreover, we prove that this approximation converges to the true value under mild conditions on the data distribution.

In summary, our key contributions are: (1) We conceptualize a new Riemannian metric that we call *generative distances* and *generative geodesics* along with approximation and convergence results, §2.1. (2) We combine recent advancements in geodesic computation and classical graph theory to efficiently compute our metric, §2.2. (3) We highlight the global nature of the new proposed metric by showcasing three proof-of-concept applications of our theory, including clustering, data visualization and data interpolation, §3.

1.1. Background on Geodesics and Metrics

Geodesics generalize the notion of shortest-path distance to spaces without straight-line distances or non-Euclidean distance. Suppose that you have to walk from one side of a town to the other, as in Fig. 1, and are considering the path that you'd like to take. The shortest path depends on your

[*] Equal contribution [1] Kim Jaechul Graduate School of AI, KAIST [2] South Dakota State University [3] KU Leuven. Correspondence to: Michael Puthawala <michael.puthawala@sdstate.edu>, Beomsu Kim <beomsu.kim@kaist.ac.kr>, Jong Chul Ye <jong.ye@kaist.ac.kr>, Emanuele Sansone <emanuele.sansone@kuleuven.be>.

Extended Abstract of the Geometry-grounded Representation Learning and Generative Modeling at 41st International Conference on Machine Learning, Vienna, Austria. PMLR Vol Number, 2024.

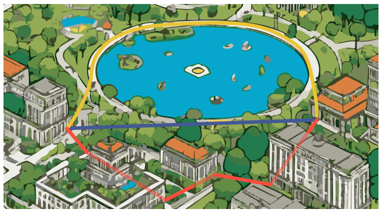


Figure 1. A visualization of how different paths can be optimal under different preferences. The blue path has the least walking, the yellow path is the most scenic, and the red path has the least sun exposure. Image generated with Stable Diffusion model.

preferences, costs, budget, namely on the underlying model you consider (in our case a generative one).

We now formalize this notion, and recall notations from differential geometry. For a detailed definition of the terms used, see e.g. Chap. 13 (Lee, 2013) and Chaps. 5, 6 (Lee, 2006). Let $\mathcal{M}$ be a smooth manifold with tangent space $T_p\mathcal{M}$ for point $p \in \mathcal{M}$. A metric is a function $g_p \colon T_p\mathcal{M} \times T_p\mathcal{M} \to \mathbb{R}$ that is symmetric and positive definite. Given this, we define the g norm and length of a smooth curve $\gamma \colon [0, 1] \to \mathcal{M}$ by

$$\|u\|_g := \sqrt{g_p(u, u)}, \quad L_g(\gamma) := \int_0^1 \|\gamma'(t)\|_g \, dt. \tag{1}$$

A non-obvious but important fact, is that the notion of length given in Eqn. 1 is parameterization independent. Put differently, $L_g(\gamma)$ measures the length of the path traced out by γ, not a property of how γ is parameterized. For any two $p, q \in \mathcal{M}$ on the same connected component of a manifold, we may define the *Riemannian distance from p to q induced by g* by

$$d_g(p, q) := \inf_{\gamma \in \Omega(p,q)} L_g(\gamma) \tag{2}$$

where $\Omega(p, q)$ is the set of all paths that connect p to q. A length minimizing path, that is an $\arg\min$ of the r.h.s. of Eqn. 2, is a *geodesic*.

2. Our Contribution

2.1. Generative Geodesics

Let $\Omega \subset \mathbb{R}^n$, $p_\Psi \colon \Omega \to \mathbb{R}^+$ be pointwise non-negative density function on Ω, and $\lambda \geq 0$ and $p_0 > 0$ be given. We may define the *generative metric* $g_{x,\Psi,\lambda}$ on Ω at point x by

$$g_{x,\Psi,\lambda}(u, v) = \left(\frac{p_0 + \lambda}{p_\Psi(x) + \lambda} \right)^2 u \cdot v \tag{3}$$

where $\cdot$ denotes the Euclidean dot product. Define the Riemannian metric on Ω, denoted $g_{\Psi,\lambda}$, as the flat metric defined point-wise on $g_{x,\Psi,\lambda}$. From this define the *generative geodesic* $d_{\Psi,\lambda}$ on Ω using Eqn.s 1 and 2.

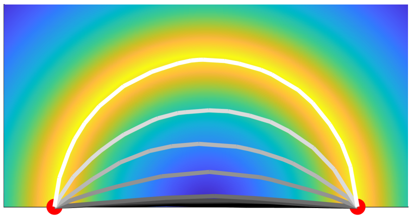


Figure 2. A showcase of how decreasing λ changes the shortest path between the two red points. The black curve has $\lambda = 5$, the white curve $\lambda = 0.05$, and the other curves interpolate between the two. The background $p_\Psi(x)$ probability distribution is shown with a yellow to blue gradient with yellow indicating high probability and blue low probability.

Lemma 2.1 ($d_{\Psi,\lambda}$ Metric)**.** *Let $\lambda, p_0 > 0$, and p_Ψ be a smooth probability density on $\mathbb{R}^n$. Then $d_{\Psi,\lambda}$ is a Riemannian metric on $\mathbb{R}^n$. Further, for $x, y \in \mathbb{R}^n$,*

1. *as $\lambda \to \infty$, $d_{\Psi,\lambda}(x, y)$ converges to $d_2(x, y)$. That is $d_{\Psi,\lambda}$ converges to the Euclidean distance.*

2. *Let $p_\Psi(x) > 0$ for all x, and define $\tilde{d}$ to be the distance induced by the metric $\frac{p_0}{p_\Psi(x)} u \cdot v$. As $\lambda \to 0$, $d_{\Psi,\lambda} \to \tilde{d}$.*

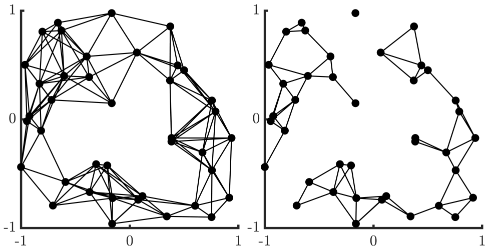
Moreover, if Ω is bounded, then this convergence is uniform (i.e. doesn't depend on x, y).

The lemma allows us to understand the role and effects of changing λ. If λ is large relative to p_0 and p_Ψ, then $\frac{p_0+\lambda}{p_\Psi(x)+\lambda} \approx 1$, and $d_{\Psi,\lambda}$ reduces to the Euclidean distance. As λ decreases, $d_{\Psi,\lambda}$ tends to consider points to be close if they are connected via a path of high likelihood, and far apart otherwise. This idea is illustrated through an example in Figure 2. If p_Ψ is the likelihood associated with a generative model, then being connected via a path of high likelihood implies that the two points lie in the same connected components of the data manifold. It is for this reason that we refer to it as the generative geodesic.

Definition 2.2 (Generative Geodesic)**.** For any $\lambda > 0$ and $x, y \in \Omega$ we call $d_{\Psi,\lambda}(x, y)$ the *generative distance* from x to y, and we call a $d_{\Psi,\lambda}$ length-minimizing $\gamma^* \in \Omega(x, y)$ a *generative geodesic*.

2.2. Approximation and Convergence Guarantee

Our metric can be readily approximated using the graph-theory approach given by (Davis & Sethuraman, 2019). The idea is to first form an epsilon graph from the points, then give the graph edge weights by using a quadrature for Eqn. 1, then use a shortest path finding algorithm. We denote by $\mathcal{X}$ the set of data, either training or synthetic. Define the ϵ graph, notated $\mathcal{G}_\epsilon(\mathcal{X})$, as the graph formed by drawing edges between all pairs in $\mathcal{X}$ less than ϵ apart. For a visualization of $\mathcal{G}$ on two-dimensional domain, see Figure 3.



(a) Plot of $\mathcal{G}_{0.56}(\mathcal{X})$ (b) Plot of $\mathcal{G}_{0.4}(\mathcal{X})$

Figure 3. A figure demonstrating the formation of the graph $\mathcal{G}_\epsilon(\mathcal{X})$, for different choices of ϵ. In both figures, the black dots denote points in $\mathcal{X}$, and black lines denote the edges in E_ϵ. Notice how when $\epsilon = 0.56$, as in 3a, the $\mathcal{G}$ has one connected component, whereas when $\epsilon = 0.4$, as in 3b, $\mathcal{G}$ has several disconnected components.

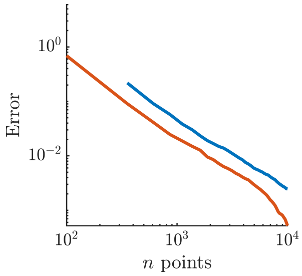


Figure 4. Error analysis of the true geodesic length vs the length of the approximated geodesic as described in Theorem 2.5, in a log plot. The x axis indicates the number of points and the y axis indicates the absolute error in the geodesic approximation. The blue line uses points sampled uniformly from the ambient space, and the orange line uses points sampled from p_Ψ.

Definition 2.3 (ϵ Graph)**.** Let $\epsilon > 0$ and $\mathcal{X} := \left\{x^{(i)}\right\}_{i\in I}$ be given. The ϵ *graph*, denoted by $\mathcal{G}_\epsilon(\mathcal{X})$, is defined as the graph with vertices $V := \mathcal{X}$ and edges

$$E_\epsilon := \{(x, x')\colon \text{where } x, x' \in \mathcal{X} \text{ are s.t. } \|x - x'\| < \epsilon\}\,.$$

Definition 2.4 (Weighted ϵ graph)**.** Let $\mathcal{G}_\epsilon(\mathcal{X})$, Ψ, p_0 and λ be given. We call the weighted ϵ graph, denoted by $\hat{\mathcal{G}}_\epsilon(\mathcal{X})$, the graph $\mathcal{G}_\epsilon(\mathcal{X})$ with edge weights given by $L_{g_{\Psi,\lambda}}(\ell_{x,y})$ where $(x, y) \in E_\epsilon$ $\ell_{x,y}$ is a straight line from x to y.

Similarly, we define the K*-approximate weighted* ϵ *graph*, denoted by $\hat{\mathcal{G}}_{\epsilon,K}(\mathcal{X})$, as the graph $\mathcal{G}_\epsilon(\mathcal{X})$ with edge weights given by applying an integral quadrature[1] to $L_{g_{\Psi,\lambda}}(\ell_{x,y})$.

Let $\hat{\mathcal{G}}$ be an edge weighted graph with vertices $\mathcal{X}$. Given two vertices $x, y \in \hat{\mathcal{G}}$ define the *linear interpolating cost*, denoted $\hat{L}(x, y; \hat{\mathcal{G}})$, as the cost of the minimal path connecting x to y in $\hat{\mathcal{G}}$. If x and y are not connected in $\hat{\mathcal{G}}$, then define $\hat{L}_{n,K}(x, y; \hat{\mathcal{G}}) = \infty$. The idea is that we may approximate $d_{\Psi,\lambda}(x, y)$ by computing $\hat{L}_{n,K}(x, y; \hat{\mathcal{G}})$. The following theorem is the main theoretical contribution of this paper and says that this approximation converges as the number of points in $\mathcal{X}$ increases and ϵ decreases[2]. This is true not only in value, but also in minimizing path, as the following theorem shows. Its proof is in Appendix A.1.

Theorem 2.5 (Convergence of Linear Interpolation Costs to Riemannian Distance)**.** *Let $\mathcal{X}_n$ be n realizations of a uniform random variable* X *over Ω. Then for any $x, y \in \Omega$ sequence $(K_n)_{n=1,\ldots}$ so that $\lim_{n\to\infty} K_n = \infty$, and $\epsilon_n \to 0$ sufficiently quickly. Then the following results hold with probability 1 (over* X*).*

1. ***Graph Becomes Connected*** *There is an N so that for all $n \geq N$, there is a path from x to y in $\hat{\mathcal{G}}_{\epsilon_n,K_n}(\mathcal{X}_n \cup \{x, y\})$.*
2. ***Convergence in Length*** $\lim_{n\to\infty} \hat{L}_{n,K_n}(x, y; \hat{\mathcal{G}}_{\epsilon_n,K_n}(\mathcal{X}_n \cup \{x,y\})) = d_{\Psi,\lambda}(x,y)$.
3. ***Convergence in Geodesic*** *For any sequence of optimal paths $l_{\mathbf{v}^{(n)}} \in \arg\min \hat{L}_{n,K_n}(x, y)$, any subsequence of $(l_{\mathbf{v}^{(n)}})_{n=1,\ldots}$ has a further subsequence of linear paths that converges uniformly to a limit path $\gamma \in \arg\min L$ and of the discrete paths in the Hausdorff sense to the true generative geodesic, S_γ. Further, if the path is unique (up to reparameterization), then $(l_{\mathbf{v}^{(n)}})_{n=1,\ldots}$ converges in Hausdorff distance to S_γ.*

A visualization of the convergence described by Theorem 2.5 is shown in Figure 4. The statement in Theorem 2.5 requires the data to be sampled from a uniform distribution, but we still numerically observe convergence even if the samples are not uniform. In fact, we observe faster convergence when the data is sampled from p_Ψ. In Appendix B, we provide details about the algorithm to compute generative distances and corresponding geodesics.

3. Applications/Experiments

Detailed experiment settings can be found in Appendix C.

3.1. Clustering with Geodesics

In this section, we verify the effectiveness of the geodesic on three datasets: two parallel and axis-aligned MoG (Narror MoG), two concentric circles, and two spirals. Specifically, given a set of points $\{x_i\}$ from a dataset, we use the geodesic to construct an affinity matrix A whose entries are defined as $A_{ij} := \exp(-d_{\Psi,\lambda}(x_i, x_j)/\tau$ where τ is a temperature hyper-parameter to control the level of smoothness of A. Then the (i, j)-th entry of A measures similarity between x_i and x_j, with higher value indicating larger similarity under the Riemannian metric. We then use A to run clustering

[1] We use equispaced trapezoidal rule for our numerical experiments.

[2] This point of ϵ decreasing is somewhat technical, and not elaborated upon in this manuscript. It is summarized in Eqn. 2.6 (Davis & Sethuraman, 2019) and the surrounding discussion.

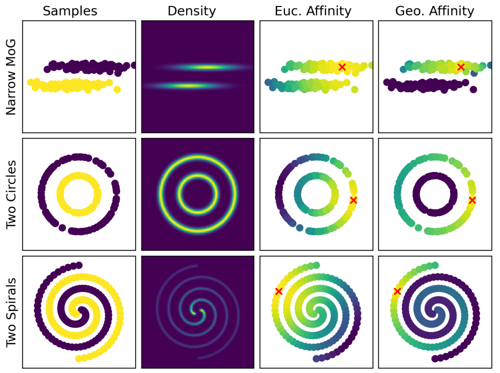


Figure 5. Euclidean and geodesic affinity visualization on various toy datasets. **First column** shows data samples, where color denotes the cluster each sample was generated from. **Second column** shows data density, where brighter color means higher density. **Third and fourth columns** show Euclidean and geodesic affinity, respectively, between the red cross and rest of the points. Brighter color means higher affinity.

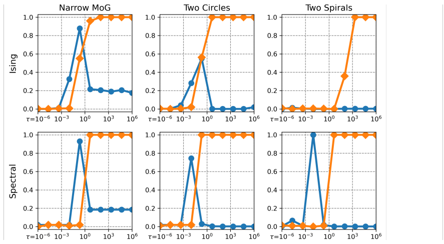


Figure 6. Clustering normalized mutual information (NMI) w.r.t. the temperature hyper-parameter τ using Euclidean and geodesic affinity matrices. Higher NMI is better. NMI = 0 means random cluster assignment and NMI = 1 means perfect clustering.

algorithms. We consider two clustering methods: Ising kernel (Kumagai et al., 2023) and spectral clustering. Figure 5 illustrates how geodesic affinity is able to provide robust clustering results. Indeed, compared to Euclidean affinity, we observe that geodesic affinity takes into account the geometric structure of the dataset.

In Figure 6, we see that clustering with geodesic affinity matrices out-perform clustering with Euclidean affinity on most of the datasets. Moreover, clustering with geodesic affinity is more robust to the choice of τ. Specifically, Euclidean affinity achieves its peak performance only for a narrow subset $\tau \in (10^{-3}, 1)$ whereas geodesic affinity performs consistently well for all sufficiently large τ.

3.2. Embedding with Geodesics

We also visualize t-SNE embeddings computed w.r.t. the Euclidean metric and our Riemannian metric. Again, in Figure 7, we can observe that points on the same connected component, i.e., cluster, are nearby in the embedding space for the geodesic. In fact, t-SNE with the Riemannian metric always produces linearly separable embeddings, so we can achieve near-perfect classification with a linear classifier trained on the embeddings.

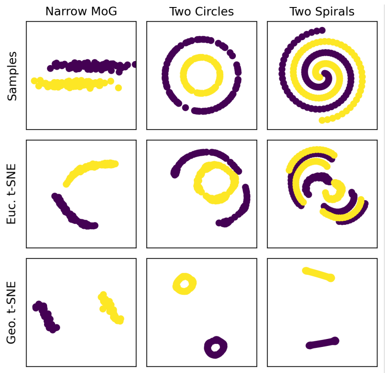


Figure 7. t-SNE with Euclidean and Riemannian metrics. Color denotes cluster label of each point.

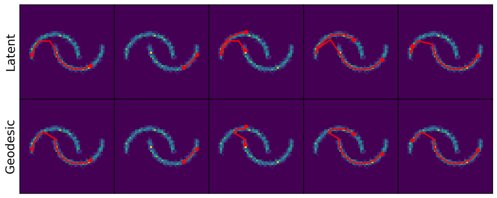


Figure 8. Interpolation on two moons. Each interpolation is illustrated by a red path.

3.3. Interpolating with Geodesics

Finally, we visualize geodesics approximated by our Riemannian metric. In Figure 8, we train a RealNVP (Dinh et al., 2017) on the two moons dataset, use the density estimated by RealNVP to calculate the geodesics. For baseline, we perform linear interpolation in the latent space of RealNVP. We observe that latent interpolation often produces convoluted paths whereas geodesics traverse the shortest path along the manifold. We also visualize geodesics between MNIST images in Figure 9, where the data density is approximated by a mixture of Gaussians with $2k$ components and variance 0.005^2. For comparison, we also visualize interpolation in the latent space of a Consistency Trajectory Model (Kim et al., 2024) which estimates diffusion probability flow ODE trajectories. We observe that the diffusion model often has sudden skips or artifacts when transitioning between digits of different identities. On the other hand, the geodesic always provides smooth transition between digits.

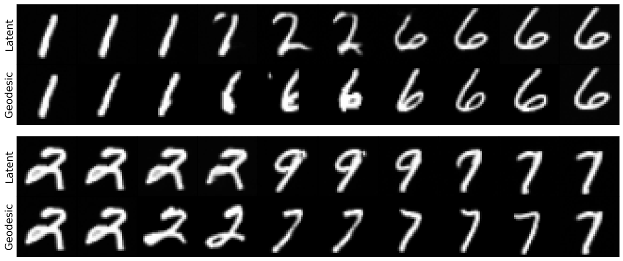


Figure 9. Interpolation on MNIST.

Statement of Contribution

B. K. actively contributed to the discussion in the meetings, implemented and ran the experiments for Section 3.1-3.3 and wrote the corresponding text. M. P. developed the theory of the approach, wrote all theorems/lemmas and proofs, the text of Sections 1.1 and 2, the code behind Figures 3, 2 and 4, and wrote the pseudo-code for the geodesic computation. J. C. Y. provided guidance and additional computation resources for the numerical experiments. E. S. setup and coordinated the team, helped to identify/conceptualize the metric, identified the applications in the experiments, reviewed the literature, wrote the Introduction section and proofread the manuscript.

Acknowledgements

This work was partially supported by the National Research Foundation of Korea under Grant RS-2024-00336454. M. P. is partially supported by CAPITAL Services of Sioux Falls, South Dakota. E. S. is supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement n° 101021347, KeepOnLearning).

References

Arvanitidis, G., Hansen, L. K., and Hauberg, S. Latent Space Oddity: on the Curvature of Deep Generative Models. In *ICLR*, 2018.

Arvanitidis, G., Hauberg, S., and Schölkopf, B. Geometrically Enriched Latent Spaces. In *AISTATS*, 2021.

Arvanitidis, G., Georgiev, B. M., and Schölkopf, B. A Prior-Based Approximate Latent Riemannian Metric. In *AISTATS*, 2022.

Berthelot, D., Raffel, C., Roy, A., and Goodfellow, I. Understanding and improving interpolation in autoencoders via an adversarial regularizer. In *ICLR*, 2019.

Chen, N., Klushyn, A., Kurle, R., Jiang, X., Bayer, J., and van der Smagt, P. Metrics for Deep Generative Models. In *AISTATS*, 2018.

Choi, J., Lee, J., Yoon, C., Park, J. H., Hwang, G., and Kang, M. Do Not Escape From the Manifold: Discovering the Local Coordinates on the Latent Space of GANs. In *ICLR*, 2022.

Crauser, A., Mehlhorn, K., Meyer, U., and Sanders, P. A Parallelization of Dijkstra's Shortest Path Algorithm. In *Mathematical Foundations of Computer Science*, 1998.

Davis, E. and Sethuraman, S. Approximating geodesics via random points. *The Annals of Applied Probability*, 29(3): 1446–1486, 2019.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density Estimation using Real NVP. *ICLR*, 2017.

Issenhuth, T., Tanielian, U., Mary, J., and Picard, D. Unveiling the Latent Space Geometry of Push-Forward Generative Models. In *ICML*, 2023.

Kim, D., Lai, C., Mitsufuji, Y., and Ermon, S. Consistency Trajectory Models: Learning Probability Flow ODE Trajectory of Diffusion. *ICLR*, 2024.

Kumagai, M., Komatsu, K., Sato, M., and Kobayashi, H. Ising-based kernel clustering. *Algorithms*, 16(4):214, 2023.

Laine, S. Feature-based metrics for exploring the latent space of generative models. In *ICLR Workshop*, 2018.

Lee, J. M. *Riemannian manifolds: an introduction to curvature*, volume 176. Springer Science & Business Media, 2006.

Lee, J. M. *Smooth manifolds*. Springer, 2013.

Lee, Y. and Park, F. C. On Explicit Curvature Regularization in Deep Generative Models. *CoRR*, 2023.

Lee, Y., Kim, S., Choi, J., and Park, F. C. A Statistical Manifold Framework for Point Cloud Data. In *ICML*, 2022a.

Lee, Y., Yoon, S., Son, M., and Park, F. C. Regularized Autoencoders for Isometric Representation Learning. In *ICLR*, 2022b.

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

Ortega-Arranz, H., Torres, Y., Llanos, D. R., and Gonzalez-Escribano, A. A New GPU-Based Approach to the Shortest Path Problem. In *International Conference on High Performance Computing & Simulation*, 2013.

Park, Y., Kwon, M., Choi, J., Jo, J., and Uh, Y. Understanding the Latent Space of Diffusion Models through the Lens of Riemannian Geometry. In *NeurIPS*, 2023.

Ramesh, A., Choi, Y., and LeCun, Y. A spectral regularizer for unsupervised disentanglement. In *ICML*, 2019.

Shao, H., Kumar, A., and Fletcher, P. T. The Riemannian Geometry of Deep Generative Models. In *CVPR Workshop*, 2018.

Song, Y., Keller, A., Sebe, N., and Welling, M. Latent Traversals in Generative Models as Potential Flows. In *ICML*, 2023.

Struski, L., Sadowski, M., Danel, T., Tabor, J., and Podolak, I. T. Feature-based interpolation and geodesics in the latent space of generative models. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

Team, R. D. *RAPIDS: Libraries for End to End GPU Data Science*, 2023. URL https://rapids.ai.

Zhu, J., Feng, R., Shen, Y., Zhao, D., Zha, Z., Zhou, J., and Chen, Q. Low-Rank Subspaces in GANs. In *NeurIPS*, 2021.

Zhu, X., Xu, C., and Tao, D. Learning disentangled representations with latent variation predictability. In *ECCV*, 2020.

A. Proofs

A.1. Proof of Theorem 2.5

The proof of Theorem 2.5 has two parts. In the first part we prove that the linear interpolating cost (defined below) converges to $d_{\Psi,\lambda}$. Next, we show that the linear interpolating cost may be approximated again with $\hat{L}_{\Psi,\lambda}$ using a quadrature.

Definition A.1 (Linear Interpolating Cost)**.** Let $x, y \in \mathcal{X}$ and $V_n(x,y)$ notate the set of all paths in $\mathcal{G}_n$ from x to y of the form $V_n(x,y) \coloneqq \{(v_0, v_1, \dots, v_{m-1}, v_m) | m > 0, v_0 = x, v_m = y, v_1, \dots, v_{m-1} \in \mathcal{X}\}$. If $V_n(x,y)$ is nonempty, we say that *Linear Interpolating Cost* from x, y is

$$L_n(x,y) \coloneqq \min_{(v_0,\dots,v_m)\in V_n(x,y)} \sum_{i=0}^{m-1} L(\overline{(v_i, v_{i+1})}; \Psi, \lambda). \tag{4}$$

If $V_n(x,y)$ is empty, we say that $L_n(x,y) = +\infty$.

Before proceeding to the Proof of Theorem 2.5, we first prove the following proposition about L_n.

Proposition A.2 (Convergence of Linear Interpolation Costs to Geodesic Cost)**.** *Let $(\epsilon_n)_{i=1,\dots}$ satisfy the decay rate assumption. Let $\mathcal{X}_n$ be n realizations of a uniform random variable X over Ω. Then the following results hold with probability 1.*

1. *For any $x, y \in \mathcal{X}_n$, $V_n(x,y)$ is non-empty.*
2. $\lim_{n\to\infty} L_n(x,y) = d_{\Psi,\lambda}(x,y)$.
3. *For any sequence of optimal paths $l_{\mathbf{v}^{(n)}} \in \arg\min L_n(x,y)$ and subsequence of $(l_{\mathbf{v}^{(n)}})_{n=1,\dots}$ has a further subsequence of linear paths that converges uniformly to a limit path $\gamma \in \arg\min L$ and of the discrete paths in the Hausdorff sense to S_γ an optimal path. Further, if the path is unique (up to reparameterization), then $(l_{\mathbf{v}^{(n)}})_{n=1,\dots}$ converges in Hausdorff distance to S_γ.*

Proof. The first point follows from the paragraph at the end of Pg. 5 (Davis & Sethuraman, 2019).

For all x and $\lambda > 0$, $\frac{p_0+\lambda}{p_\Psi(x)+\lambda}$ is positive, hence Lemma 2.6 (Davis & Sethuraman, 2019) applies by taking $M(x) \coloneqq \frac{p_0+\lambda}{p_\Psi(x)+\lambda}$. Hence, (Lip), (Hilb), (TrIneq) and (Pythag) all apply for $\alpha > 1$. Therefore the full consequence of Theorem 2.8 (Davis & Sethuraman, 2019) apply.

The second point follows from equation $\lim_{n\to\infty} \min_{\gamma\in\Omega_n^l(x,t)} L_n(x,y) = \min_{\gamma\in\Omega(x,y)} L(\gamma;\Psi,\lambda)$. But, we have that $\min_{\gamma\in\Omega(x,y)} L(\gamma;\Psi,\lambda) = d_{\Psi,\lambda}(x,y)$ from the work above.

Finally, the third point follows from the final three sentences in the conclusion of Theorem 2.8 (Davis & Sethuraman, 2019). □

Next we prove two lemmas that show that L_n and $\hat{L}_{n,K}$ are close, and become closer as $n, K \to \infty$.

Lemma A.3 (Approximating Edge Weights)**.** *Let $\mathcal{G}_n$ be a λ weighted ϵ graph with edge (x,y). Then for any $\delta > 0$, there is a $K \in \mathbb{N}$ so that*

$$\left| L(\overline{(x,y)}; \Psi, \lambda) - \frac{\|y-x\|_2}{K} \sum_{i=0}^{K-1} \frac{p_0+\lambda}{p_\Psi\left(\left(1-\frac{i}{K}\right)x + \frac{i}{K}y\right)+\lambda} \right| < \delta \tag{5}$$

Proof. For $x, y \in \Omega$, the unit-speed parameterization of the straight-line path connecting x to y is given by

$$\gamma(t) \coloneqq \left(1 - \frac{t}{\|y-x\|_2}\right)x + \frac{t}{\|y-x\|_2}y, \quad \dot{\gamma}(t) = \frac{y-x}{\|y-x\|_2}.$$

Hence, we may write

$$L(\overline{(x,y)};\Psi,\lambda)=\int_0^{\|y-x\|_2} f\left(\left(1-\frac{t}{\|y-x\|_2}\right)x+\frac{t}{\|y-x\|_2}y,\frac{y-x}{\|y-x\|_2};\lambda\right)dt$$
$$=\int_0^{\|y-x\|_2}\frac{p_0+\lambda}{p_\Psi\left(\left(1-\frac{t}{\|y-x\|_2}\right)x+\frac{t}{\|y-x\|_2}y\right)+\lambda}dt. \tag{6}$$

For $\Delta t:=\frac{\|y-x\|_2}{K}$, the K-piece left Riemann sum of the r.h.s. of 6 is

$$\sum_{i=0}^{K-1} f(\gamma(\Delta t i),\dot\gamma(\Delta t i);\Psi,\lambda)\Delta t=\frac{\|y-x\|_2}{K}\sum_{i=0}^{K-1}\frac{p_0+\lambda}{p_\Psi\left(\left(1-\frac{i}{K}\right)x+\frac{i}{K}y\right)+\lambda} \tag{7}$$

The quantity in the integrand of Eqn. 6 is continuous as a function of t, and so from standard calculus results there is some K so that

$$\left|L(\overline{(x,y)};\Psi,\lambda)-\frac{\|y-x\|_2}{K}\sum_{i=0}^{K-1}\frac{p_0+\lambda}{p_\Psi\left(\left(1-\frac{i}{K}\right)x+\frac{i}{K}y\right)+\lambda}\right|<\delta. \tag{8}$$

□

Lemma A.4 (Approximating Linear Interpolating Cost)**.** *Let $\mathcal{X}$ and λ be given, then for any δ there is a $K\in\mathbb{N}$ such that for all $x,y\in\mathcal{X}$,*

$$\left|L_n(x,y)-\hat L_{n,K}(x,y)\right|<\delta. \tag{9}$$

Proof. Let M be the length (in number of edges) between all L_n minimal paths connecting x,y for any two x,y. Then, we may apply Lemma Approximating Edge Weights to get a K such that the edge weight error is no more than δ/M. Then, we have for that choice of K and any path $(v_0,\dots,v_m)$ that

$$\left|\sum_{i=0}^{m-1}L(\overline{(v_i,v_{i+1})};\Psi,\lambda)-\frac{\|y-x\|_2}{K}\sum_{i=0}^{m-1}\sum_{j=0}^{K-1}\frac{p_0+\lambda}{p_\Psi\left(\left(1-\frac{j}{K}\right)v_i+\frac{j}{K}v_{i+1}\right)+\lambda}\right|$$
$$=\left|\sum_{i=0}^{m-1}\left(L(\overline{(v_i,v_{i+1})};\Psi,\lambda)-\frac{\|y-x\|_2}{K}\sum_{j=0}^{K-1}\frac{p_0+\lambda}{p_\Psi\left(\left(1-\frac{j}{K}\right)v_i+\frac{j}{K}v_{i+1}\right)+\lambda}\right)\right|$$
$$\le\sum_{i=0}^{m-1}\left|L(\overline{(v_i,v_{i+1})};\Psi,\lambda)-\frac{\|y-x\|_2}{K}\sum_{j=0}^{K-1}\frac{p_0+\lambda}{p_\Psi\left(\left(1-\frac{j}{K}\right)v_i+\frac{j}{K}v_{i+1}\right)+\lambda}\right|$$
$$\le\sum_{i=0}^{m-1}\frac{\delta}{M}=\frac{m}{M}\delta\le\delta$$

This bound holds for all paths, hence it holds for the minimal paths too. Let $\mathbf{v}^*:=(v_0^*,\dots,v_m^*)$ be an L_n minimizing path from x,y, then

$$\left|L_n(x,y)-\hat L_{n,K}(x,y)\right|$$
$$\le\left|\sum_{i=0}^{m-1}L(\overline{(v_i^*,v_{i+1}^*)};\Psi,\lambda)-\frac{\|y-x\|_2}{K}\sum_{i=0}^{m-1}\sum_{j=0}^{K-1}\frac{p_0+\lambda}{p_\Psi\left(\left(1-\frac{j}{K}\right)v_i^*+\frac{j}{K}v_{i+1}^*\right)+\lambda}\right|$$
$$\le\delta.$$

□

Finally, we present the final proof of Theorem 2.5.

Proof. The vertices and edges of $\mathcal{G}_n$ and $\hat{\mathcal{G}}_{n,K_n}$ are the same, so the existence of a path connecting x and y in $\mathcal{X}$ follow from the same arguments as in Prop. A.2.

Point 2 follows from combining point 2 of Prop. A.2 with Lemma A.4.

For $\lambda > 0$, $\frac{p_0+\lambda}{p_\Psi(x)+\lambda}$ is smooth on compact set, and so has uniformly bounded gradients. Hence, for any δ, there is a uniform K that can be so that for all $\overline{(x, y)}$ such that $x, y \in \Omega$. Hence, as $n \to \infty$,

$$\left| L(e; \Psi, \lambda) - \frac{\|y - x\|_2}{K} \sum_{i=0}^{K-1} \frac{p_0 + \lambda}{p_\Psi\left(\left(1 - \frac{i}{K}\right) x + \frac{i}{K} y\right) + \lambda} \right| \to 0 \tag{10}$$

uniformly in e. Hence, for any sequence $(l_{\mathbf{v}^{(n)}})_{n=1,\dots}$ of paths uniformly bounded in length,

$$\lim_{n\to\infty} \left| \sum_{i=0}^{m_n-1} \left(L(\overline{(v_i, v_{i+1})}; \Psi, \lambda) - \frac{\|y - x\|_2}{K} \sum_{j=0}^{K_n-1} \frac{p_0 + \lambda}{p_\Psi\left(\left(1 - \frac{j}{K}\right) v_i + \frac{j}{K} v_{i+1}\right) + \lambda} \right) \right| = 0, \tag{11}$$

where $l_{\mathbf{v}^{(n)}} = (v_0, \dots, v_{m_n})$. Put simply, for a sequence of finite-length paths in $\mathcal{G}$, the costs of L_n and $\hat{L}_{n,K_n}$ converge to each other as $n \to \infty$.

Now, let $l_{\mathbf{v}^{(n)}} \in \arg\min \hat{L}_{n,K_n}(x, y)$. We have that $\frac{p_0+\lambda}{p_\Psi(x)+\lambda}$ is bounded from below, hence $l_{\mathbf{v}^{(n)}}$ are all uniformly bounded in length. Hence, Eqn. 11 applies, and so $\liminf_{n\to\infty} L_n(l_{\mathbf{v}^{(n)}}) \leq \lim_{n\to\infty} \hat{L}_{n,K_n}(l_{\mathbf{v}^{(n)}})$. The same argument holds reversing the roles of L_n and $\hat{L}_{n,K_n}$, hence any minimizing sequence of L_n is one of $\hat{L}_{n,K_n}$ and vice-versa.

Now suppose that $(l_{\mathbf{v}^{(n)}})_{n=1,\dots} \in \arg\min \hat{L}_{n,K_n}(x, y)$ is a sequence of optimal paths. It is also a sequence of optimal paths of L_n. Thus, any subsequence of $(l_{\mathbf{v}^{(n)}})$ has a further subsequence so that Theorem A.2 point 3 applies, and so has a further subsequence that converges in Hausdorff distance to S_γ. □

B. Algorithm

Algorithm 1 describes the strategy to compute $\hat{\mathcal{G}}_{n,K}$. From $\hat{\mathcal{G}}_{n,K}$, we can compute the minimal path between two end points x and y and its length. This requires us to solve a standard graph theorem problem, which can be solved with Dijkstra's algorithm, see e.g. (Crauser et al., 1998; Ortega-Arranz et al., 2013). A GPU accelerated python implementation of the single source shortest path (SSSP) algorithm can be found in the RAPIDs library (Team, 2023).

Algorithm 1 Compute $\hat{\mathcal{G}}_{n,K}$

```
Require: edge $\epsilon, X \subset \mathcal{X}, \lambda, p$
  # Compute $\mathcal{G}_n$ with vertices $V$, edges $E$
  $V, E \leftarrow X, \emptyset$
  for $(x, y) \in X \times X, x \neq y$ do
    if $d_2(x, y) \leq \epsilon$ then
      $E$.add$(x, y)$
    end if
  end for
  # Compute edge weights, $W\colon E \to \mathbb{R}$
  $W$ reset
  for edge $(x, y) \in E$ do
    $W((x, y)) \leftarrow$ Integral Quadrature for $L_g(\ell_{x,y})$
  end for
  return $\hat{\mathcal{G}}_{n,K} = (\mathcal{G}_n, W)$.
```

C. Experiment Settings

C.1. Section 2.2

In Figure 4, we used a p_Ψ given by

$$p_\Psi(x) := \frac{e^{-10\left|\frac{3}{4}-\|x\|_2\right|}}{Z}. \tag{12}$$

defined over $[0,1]^2$, where Z is a normalization constant. We used $K = 10$, $\lambda = 0.01$ $\epsilon = 0.158$. The values of n chosen ranged from $n = 100$ to $n = 10,000$. The geodesic error was measured between the points $(-1, 0)$ and $(1, 0)$. For each value of n, we computed the associated geodesic error 20 times. Figure 4 reports the average error across the 20 trials.

C.2. Section 3.1

We use $n = 200$, $K = 10$, $\lambda = 10^{-8}$, and $\epsilon = 10$ to compute $\hat{\mathcal{G}}_{n,K}$. We use spectral clustering in the `scikit-learn` Python library, and Ising clustering from `https://github.com/kumagaimasahito/Ising-based_Kernel_Clustering`. We use the default clustering hyper-parameters. For evaluation, we use normalized mutual information also in `scikit-learn`.

C.3. Section 3.2

We use $n = 200$, $K = 10$, $\lambda = 10^{-8}$, and $\epsilon = 10$ to compute $\hat{\mathcal{G}}_{n,K}$. We use t-SNE in the `scikit-learn` Python library.

C.4. Section 3.3

Two Moons. We train a RealNVP model using the code in `https://github.com/xqding/RealNVP`. We use $n = 1024$, $K = 10$, $\lambda = 10^{-8}$, and $\epsilon = 10$ to compute $\hat{\mathcal{G}}_{n,K}$. For latent interpolation, given two points, we map them to latent vectors using the RealNVP model, linearly interpolate the latent vectors, and then push the latent vectors to data space by again using the RealNVP model.

MNIST. We use $n = 2000$, $K = 10$, $\lambda = 0.005$, and adaptively choose ϵ for each vertex in $\hat{\mathcal{G}}_{n,K}$ such that it always has at least 5 neighbors. For the baseline, we train a Consistency Trajectory Model on MNIST using the code from `https://github.com/sony/ctm`, and interpolate images by linearly interpolating latent vectors and pushing the vectors through the model.