Data-Driven Metric Learning for Image Databases Represented as Riemannian Manifolds Embedded into Image Spaces

Paulo A. Braz, Gastão F. Junior, Gilson A. Giraldi

Laboratório Nacional de Computação Científica (LNCC)
25651-075 – Petrópolis – RJ – Brazil

Departamento de Matemática
Universidade Federal de Sergipe (UFS) – 49107-230 – São Cristovão - SE - Brazil

{pbraz@posgrad.lncc.br, gastao@mat.ufs.br, gilson@lncc.br}

Abstract. *We present a data-driven pipeline for learning a smooth Riemannian metric over reduced image representations and computing geodesic trajectories. The method estimates local tangent structure with medoid-based clustering and cluster-wise PCA, aligns local coordinate systems, and blends metric tensors using radial basis function weights. Geodesics are solved as boundary value problem initialized from discrete manifold paths, with checkpointed continuation for difficult transitions. Results show that BVP refines graph-based paths, while linear interpolation, although short, crosses regions less supported by the empirical data distribution. The proposed framework provides a practical tool for probing intrinsic geometric structure in learned image representations.*

1. Introduction

Metric learning aims to construct distance functions that more accurately reflect the structure of data than standard Euclidean geometry. In many machine learning settings, Euclidean distance is overly restrictive, particularly when data is distributed along nonlinear manifolds embedded in high-dimensional spaces. In such cases, learning a metric adapted to the data distribution is essential for capturing local structure, neighborhood relationships, and intrinsic paths.

Riemannian metric learning extends metric learning by equipping the data space with a smoothly varying metric tensor. In generative modeling, prior work has defined geometry through generator-induced pullback metrics [Arvanitidis et al. 2018] and [Arvanitidis et al. 2021] and decoder-induced information-geometric pullbacks [Arvanitidis et al. 2022]. Other approaches use learned transformations of the data space [Diepeveen 2024] or trajectory-based methods such as Metric Flow Matching [Kapuńniak et al. 2024]. In contrast, our work reconstructs a spatially varying Riemannian metric directly from sampled reduced representations and uses it for continuous geodesic computation.

Despite these advances, there remains a gap between data-driven metric learning and explicit geometric computation. In particular, existing approaches either rely on model-dependent constructions, learn implicit geometries, or do not provide explicit mechanisms for computing geodesics under a learned metric [Harlim et al. 2023] and [Zhou et al. 2020].

In this work, we propose a data-driven framework for reconstructing a spatially varying Riemannian metric from sampled data and computing geodesics under this metric. By combining metric learning, manifold learning, and differential geometry, the method provides an explicit model-agnostic representation of data geometry, avoiding reliance on generator Jacobians or other model-dependent internal mappings.

We begin by partitioning the data into locally coherent regions, following the common assumption in manifold learning that data can be approximated by locally linear structures [Tenenbaum et al. 2000]. Within each region, we estimate local metric tensors using Principal Component Analysis (PCA), which provides a first-order approximation of the tangent space and captures local anisotropy in the data distribution. This construction is consistent with geometric interpretations of PCA as a local metric estimator [Gruffaz and Sassen 2025].

Since local PCA bases are defined independently, we align them through orthogonal Procrustes method, a standard technique for matching local coordinate systems [Goodall 2018]. To obtain a globally smooth metric, the aligned local metrics are combined using Radial Basis Function (RBF) weighting, a widely used approach for constructing smooth interpolants from local models [Buhmann 2000].

Given the learned metric, geodesic trajectories between data samples are computed by solving the geodesic equation as a boundary value problem using a collocation-based numerical solver. This formulation allows direct computation of intrinsic paths under the reconstructed metric, in contrast to approaches that induce geometry from generative mappings [Arvanitidis et al. 2018], pull back information-geometric metrics [Arvanitidis et al. 2022], or learn manifold representations for geodesic computation [Yu et al. 2026].

To validate the consistency of the learned geometry, we additionally construct a k-nearest neighbor graph with Riemannian edge weights and compare continuous geodesics with discrete shortest paths obtained via Dijkstra's algorithm.

Unlike classical metric learning, our approach learns a spatially varying Riemannian metric for intrinsic distance and geodesic computation. Building on prior work in local Riemannian manifold learning [Miranda Jr et al. 2015], we integrate medoid regions, locality-weighted estimation, RBF blending, and continuous BVP geodesics over reduced image representations.

2. Related Work

The manifold hypothesis suggests that high-dimensional data often concentrate near lower-dimensional structures embedded in ambient space. In this regard, Riemannian geometry has also been used to define intrinsic distances in representation spaces. Pullback metrics induced by generator Jacobians provide a principled way to measure distances and compute geodesics [Arvanitidis et al. 2021]. Other recent approaches construct geometric structures through deformation techniques or implicit manifolds [Yang et al. 2023] and [Hartwig et al. 2026]. Related methods also investigate likely geometries in generative models and learned transformations of the data space [Rygaard et al. 2026] and [Diepeveen 2024].

Trajectory-based formulations, including Metric Flow Matching and Riemannian

distance-based model identification, provide further perspectives on geometric learning and analysis [Kapuńniak et al. 2024] and [Song and Itti 2025]. However, many of these methods either require access to model internals, avoid explicit metric reconstruction, or focus on learning plausible trajectories without directly estimating a spatially varying metric system from data. Recent work has also emphasized computational stability and efficient operators for Riemannian metrics on SPD manifolds, including Cholesky-based product geometries for fast and stable SPD metric design [Chen et al. 2026]. Although this line of work focuses on SPD-valued data and SPD neural networks, it reinforces the importance of stable metric construction when Riemannian geometry is used in learning systems.

Metric learning provides another related perspective, aiming to replace Euclidean distances with task- or data-adapted notions of similarity [Kulis 2013] and [Kaya and Bilge 2019]. Classical and deep metric learning methods usually learn global, parametric, or embedding-based distances for supervised objectives, including face recognition and deep visual representations [Schroff et al. 2015] and [Zheng et al. 2021]. In contrast, our work reconstructs an explicit Riemannian metric directly from sampled data in a reduced representation space. Local anisotropy is estimated through cluster-wise PCA, local metric systems are aligned using Procrustes method, and the resulting metric systems are blended with RBF weights into a smooth global metric system. This enables continuous geodesic computation through a boundary value formulation, independent of generator Jacobians or model-specific internal derivatives.

Positioning with respect to recent approaches. Recent work has explored Riemannian geometry in representation and generative modeling through learned latent-space metrics [Arvanitidis et al. 2021], global geometric transformations of the data space [Diepeveen 2024], and trajectory-based methods such as Metric Flow Matching [Kapuńniak et al. 2024]. Related work on SPD manifolds has also emphasized stable and computationally efficient Riemannian metric design [Chen et al. 2026], while Riemannian distances have been used in application-oriented settings such as model identification [Song and Itti 2025]. Our method is complementary to these directions: it learns a spatially varying metric from sampled reduced representations and explicitly couples metric reconstruction with continuous geodesic computation.

3. Riemannian Geometry Preliminaries

We briefly recall the geometric concepts used throughout this work. We assume that the data lies on a d-dimensional differentiable manifold $\mathcal{M} \subset R^D$. For each point $\mathbf{p} \in \mathcal{M}$, the set of all tangent vectors to the d-dimensional manifold $\mathcal{M}^d$ at $\mathbf{p}$ forms a vector space, denoted by $T_{\mathbf{p}}\mathcal{M}$, called the *tangent space* of $\mathcal{M}$ at $\mathbf{p}$.

A Riemannian metric endows each tangent space with an inner product that varies smoothly across the manifold, allowing the definition of local distances, angles, and geometric quantities such as geodesics.

[Riemannian Metric] A metric on a differentiable manifold $\mathcal{M}^d$ is a correspondence that associates to each point $\mathbf{p} \in \mathcal{M}^d$ an inner product $\langle \cdot, \cdot \rangle_{\mathbf{p}}$ on the tangent space $T_{\mathbf{p}}\mathcal{M}^d$, varying smoothly with $\mathbf{p}$ [do Carmo 2008].

In local coordinates, the metric is represented by the coefficients, for each $\mathbf{p} \in \mathcal{M}$

$$g_{ij} = \left\langle \frac{\partial}{\partial x_i}, \frac{\partial}{\partial x_j} \right\rangle, \tag{1}$$

where $\left\{\frac{\partial}{\partial x_i}\right\}_{i=1}^{d}$ denotes the natural basis on $T_{\mathbf{p}}\mathcal{M}^d$ induced by the coordinates chart. These coefficients define a symmetric positive-definite matrix $g = (g_{ij})$. The pair $(\mathcal{M}^d, g)$ is called a Riemannian manifold.

[Christoffel Symbols] Let $(\mathcal{M}^d, g)$ be a Riemannian manifold. The Christoffel symbols of Levi-Civita connection are defined in local coordinates by

$$\Gamma_{ij}^{k} = \frac{1}{2} g^{km} \left(\frac{\partial g_{jm}}{\partial x_i} + \frac{\partial g_{im}}{\partial x_j} - \frac{\partial g_{ij}}{\partial x_m} \right), \tag{2}$$

where g^{km} denotes the inverse of the metric tensor g_{km} [do Carmo 2008].

These coefficients encode how directions evolve along the manifold via the connection and directly define the geodesic equation used to compute intrinsic paths under the learned metric.

[Geodesics] Let $\gamma : I \subset R \to \mathcal{M}^d$ be a differentiable curve. In local coordinates it can be written as $\gamma(t) = (x_1(t), \ldots, x_d(t))$. The derivative of the velocity vector along the curve is given by

$$\frac{D}{dt}\left(\frac{d\gamma}{dt}\right) = \sum_{k=1}^{d} \left(\frac{d^2 x_k}{dt^2} + \sum_{i,j=1}^{d} \Gamma_{ij}^{k} \frac{dx_i}{dt} \frac{dx_j}{dt} \right) \frac{\partial}{\partial x_k}. \tag{3}$$

The curve γ is called a geodesic if its covariant acceleration vanishes identically, i. e., $Ddt\,(d\gamma dt) = 0$. In local coordinates, this geometric zero-acceleration condition yields the geodesic equation

$$\frac{d^2 x_k}{dt^2} + \sum_{i,j=1}^{d} \Gamma_{ij}^{k} \frac{dx_i}{dt} \frac{dx_j}{dt} = 0, \quad k = 1, \ldots, d. \tag{4}$$

Geodesics represent curves of constant speed that locally minimize the length induced by the Riemannian metric.

4. Methodology

We propose a data-driven pipeline for learning a smooth Riemannian metric over a reduced image representation space and computing geodesic trajectories between representative samples. The proposed pipeline starts from a sample set $\mathcal{S} = \{\mathbf{y}_i\}_{i=1}^{N}$ embedded in the reduced representation space and is not intrinsically tied to a particular generative model.

In the MNIST experiments, $\mathcal{S}$ is built from a curated empirical sample of digit images, including generated samples filtered by image-similarity criteria, SSIM (Structural Similarity Index Measure) [Wang et al. 2004], and embedded into the representation space. This generative component is used only to enrich the empirical sampling of the

digit manifold; it is not part of the subsequent metric estimation or geodesic computation procedure.

Given $\mathcal{S}$, we partition the representation space into K local regions. In our experiments, we use K-Medoids because its representatives are actual sampled points, avoiding synthetic centroids that may lie in poorly supported regions of the representation space. Each cluster is treated as a local approximation region of the sampled manifold, and its medoid is used as a local reference point and as the center of the corresponding radial basis weight.

4.1. Local-to-Global Metric Construction

Our construction follows the LRML (Local Riemannian Manifold Learning) principle of representing the manifold through local coordinate systems and composing these local descriptions into a common representation [Miranda Jr et al. 2015]. In our setting, each cluster C_k defines a local neighborhood, with medoid $\mathbf{m}_k$ as reference point. A local PCA frame $V_k = [\mathbf{v}_{k,1}, \ldots, \mathbf{v}_{k,d}]$ is estimated from the centered data matrix X_k by solving

$$X_k^\top X_k \mathbf{v}_{k,i} = \lambda_{k,i} \mathbf{v}_{k,i}. \tag{5}$$

Analogously to local normal coordinates, points near $\mathbf{m}_k$ are represented in the local chart by

$$\mathbf{u}_k(\mathbf{y}) = V_k^\top (\mathbf{y} - \mathbf{m}_k). \tag{6}$$

In each local metric system, the metric is represented by a symmetric matrix $A_k \in R^{d \times d}$. Following the quadratic-form formulation used in LRML, local graph-based distances are approximated by

$$\ell_{ij}^2 \approx \left(\mathbf{u}_k(\mathbf{y}_i) - \mathbf{u}_k(\mathbf{y}_j)\right)^\top A_k \left(\mathbf{u}_k(\mathbf{y}_i) - \mathbf{u}_k(\mathbf{y}_j)\right). \tag{7}$$

Here, ℓ_{ij} is computed as the shortest-path distance between $\mathbf{y}_i$ and $\mathbf{y}_j$ on the local k-NN graph restricted to the neighborhood C_k. Although each evaluation is scalar, the learned object is the full local metric matrix A_k, which has $d(d+1)/2$ independent coefficients. These coefficients are estimated by stacking local pairwise constraints into a weighted least-squares problem,

$$\hat{\mathbf{a}}_k = \arg\min_{\mathbf{a}} \left\| W_k^{1/2} (\Psi_k \mathbf{a} - \mathbf{b}_k) \right\|_2^2 + \rho \|\mathbf{a}\|_2^2, \tag{8}$$

where $\mathbf{a}$ contains the independent coefficients of A_k, Ψ_k encodes the quadratic displacement terms, and $\mathbf{b}_k$ contains the squared local distances. The matrix W_k is diagonal and implements locality weighting: shorter shortest-path constraints inside C_k receive larger weights, since the local quadratic approximation is more appropriate for small displacements. The solution is unpacked into a symmetric matrix which defines a symmetric positive definite local quadratic form.

The matrix A_k defines the metric in the k-th local metric system. For any local displacement $\delta u^{(k)}$, its squared local metric norm is

$$\|\delta u^{(k)}\|_{A_k}^2 = (\delta u^{(k)})^\top A_k \delta u^{(k)}. \tag{9}$$

Thus, A_k is a full $d \times d$ local metric matrix, while $(\delta u^{(k)})^\top A_k \delta u^{(k)}$ is only the scalar evaluation of this metric on a particular displacement.

Because local metric systems are estimated independently, their orientations may differ across clusters. Following the composition motivation of LRML, we align local metrics before global blending using orthogonal Procrustes technique [Goodall 2018]. The orthogonal matrix $R_k \in O(d)$ is the alignment matrix, the aligned metric is

$$\widetilde{A}_k = R_k^\top A_k R_k. \tag{10}$$

Finally, the aligned local metrics are blended into a smooth global metric field using normalized radial basis weights:

$$G(\mathbf{y}) = \sum_{k=1}^{K} w_k(\mathbf{y}) \widetilde{A}_k, \qquad w_k(\mathbf{y}) = \frac{\exp\left(-\frac{\|\mathbf{y}-\mathbf{m}_k\|^2}{2\sigma_k^2}\right)}{\sum_{j=1}^{K} \exp\left(-\frac{\|\mathbf{y}-\mathbf{m}_j\|^2}{2\sigma_j^2}\right)}. \tag{11}$$

The weights $w_k(\mathbf{y})$ are scalar functions, but they weight full $d \times d$ metric matrices. We shall highlight that G(y) is symmetric positive-definite because it is a convex linear combination of symmetric positive-definite matrices. So, we can affirm that G(y) is a Riemannian metric over the reduced representation space.

4.2. Continuous Geodesic Computation

Given two data points $y_a, y_b \in R^d$, we compute a curve

$$\gamma : [0, 1] \to R^d \tag{12}$$

that connects them under the learned metric. The curve is obtained by solving the geodesic equation

$$\frac{d^2\gamma^i}{dt^2} + \Gamma^i_{jk}(\gamma)\frac{d\gamma^j}{dt}\frac{d\gamma^k}{dt} = 0, \tag{13}$$

with boundary constraints

$$\gamma(0) = y_a, \qquad \gamma(1) = y_b. \tag{14}$$

We address the geodesic equation as a two-point boundary value problem (BVP). The BVP specifies the differential equation and endpoint constraints, whereas the numerical collocation solver necessitates an initial approximation of the unknown curve on its mesh. We derive this initial mesh curve from a Dijkstra path calculated on the Riemannian k-NN graph, rather than employing a Euclidean interpolation between endpoints. The path serves solely as a numerical initialization. The ultimate trajectory is continuous within the reduced representation space and is not obligated to adhere to the graph.

For medium and hard pairs, direct BVP solving may become unstable when the trajectory crosses several local metric regions. In these cases, we use checkpointed continuation: if the direct solve fails, a stable intermediate point is selected, and a new BVP is solved from that point to the original target. The final trajectory is obtained by concatenating the stable prefix and the new segment, with at most three continuation steps.

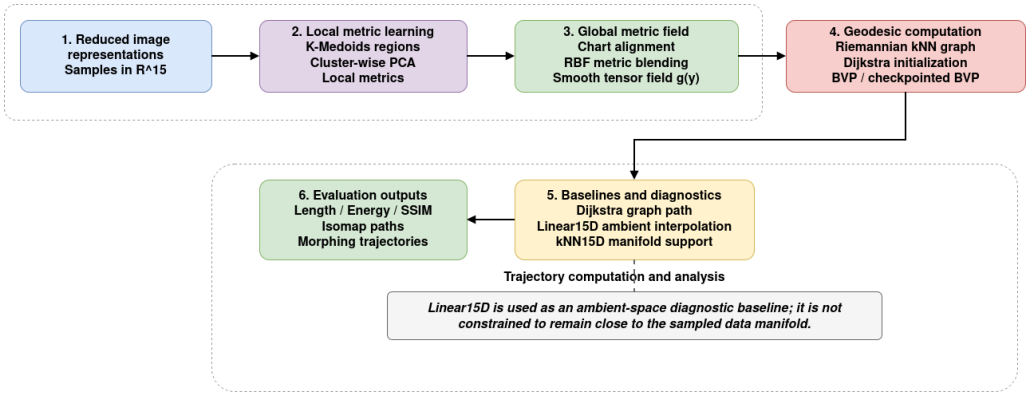


Figure 1. Full Technique Pipeline

4.3. Trajectory Evaluation

For each trajectory γ, we report its Riemannian length:

$$L(\gamma) = \int_0^1 \|\dot{\gamma}(t)\|_{g(\gamma(t))} dt = \int_0^1 \sqrt{\dot{\gamma}(t)^\top g(\gamma(t))\dot{\gamma}(t)}\, dt. \tag{15}$$

The *energy* is obtained by integrating its squared value. Length measures the geometric extent of the trajectory, whereas energy is more sensitive to high-cost or irregular segments. For image-based evaluation, trajectory points in the reduced space are mapped back to the image domain using the corresponding local inverse PCA transformation implemented in scikit-learn [Pedregosa et al. 2011]. We also report frame-wise SSIM to assess the similarity and quality of consecutive reconstructed images, and the average Euclidean distance from trajectory points to their nearest real samples in the reduced space, used as a diagnostic of empirical manifold support.

5. Experiments and Results

5.1. Experimental Setup

The full pipeline is presented in Figure 1. We evaluate the proposed framework on MNIST. A pretrained generative model is used as a data augmentation strategy to expand the sample set before metric estimation, providing a denser approximation of the data manifold. All geometric computations are performed in a reduced representation space R^{15}, used for local metric estimation, global metric blending, and geodesic computation.

The dimension $d = 15$ was selected from a preliminary spectral analysis of the standardized MNIST space after outlier filtering. This analysis follows the standard PCA principle that the covariance eigenvalue spectrum identifies the dominant directions of data variance [Jolliffe and Cadima 2016]. In this diagnostic, we retained the components whose eigenvalues remained above an optimized spectral threshold; the procedure selected 15 components, which were then used for metric estimation and BVP geodesic computation.

The sampled points are partitioned into K clusters using K-Medoids ($K = 30$). Each cluster is interpreted as a local metric of the sampled manifold approximation, where PCA provides dominant local directions and the corresponding local metric captures variation of the data distribution.

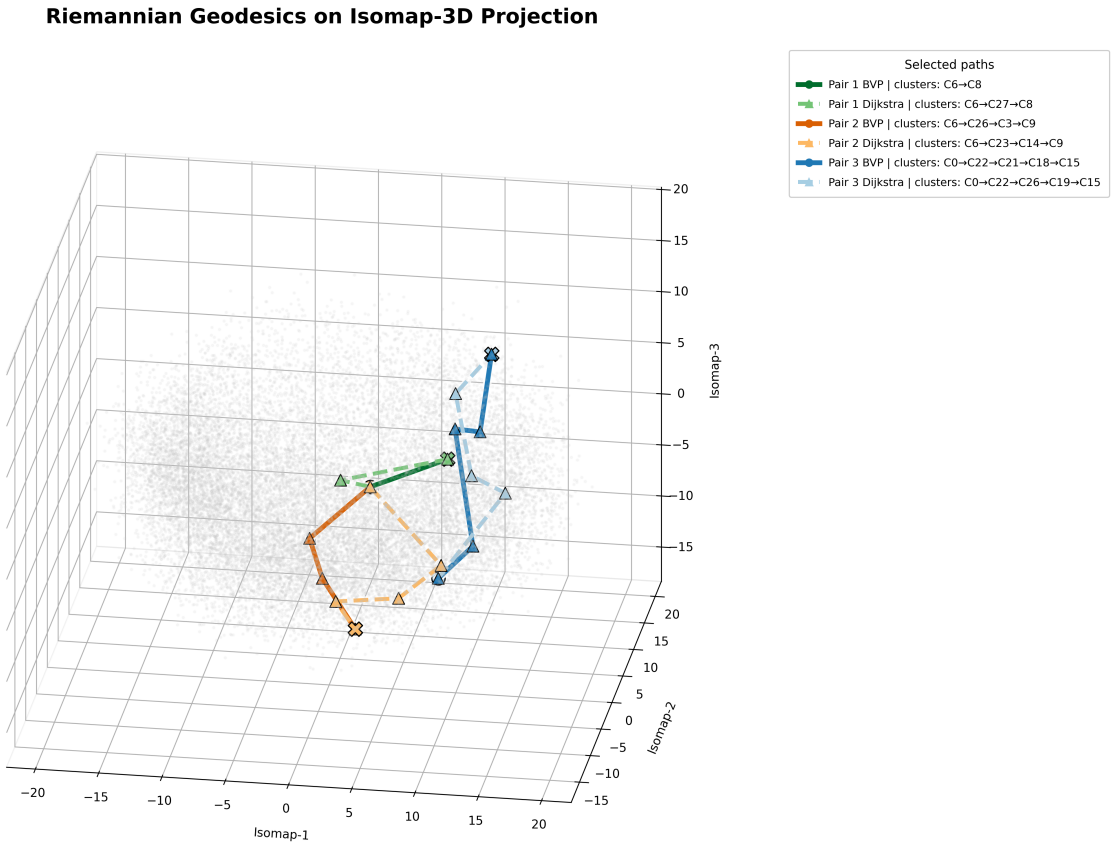


Figure 2. Representative trajectories projected with Isomap for visualization. Geodesics are computed in 15D.

Continuous geodesics are computed by solving the geodesic boundary value problem between selected pairs of points using a collocation-based BVP solver. The resulting trajectories are compared against discrete shortest paths computed on a k-nearest-neighbor graph whose edge weights are defined using the same learned Riemannian metric.

5.2. Results

We evaluate 25 medoid-to-medoid trajectories in the reduced 15-dimensional (R^{15}) representation space: 5 easy, 10 medium, and 10 hard pairs, defined by quartiles of pairwise medoid distances. For each pair, we compare the proposed continuous BVP geodesic, a Riemannian-weighted Dijkstra path on a k-nearest-neighbor graph, and Linear15D, a straight-line interpolation in the reduced ambient space. We report Riemannian length, energy, and frame-wise SSIM between consecutive reconstructed images. The proposed full pipeline workflow is shown in Figure 1.

Figure 2 shows representative trajectories for the three difficulty buckets: easy, medium, and hard, projected with Isomap for visualization. For each pair, BVP denotes the proposed continuous geodesic trajectory computed in the reduced 15D representation space under the learned Riemannian metric, while Dijkstra denotes the corresponding discrete shortest path on the Riemannian k-NN graph. The 3D projection provides a

qualitative view of the trajectory layout and should not be interpreted as preserving the full geometry of the original 15D paths. The visualization illustrates how the continuous solver refines graph-based paths across local, intermediate, and long-range medoid-to-medoid transitions.

Table 1 shows that BVP achieves the lowest length and energy in the easy bucket and remains lower-cost than Dijkstra in medium and hard trajectories, supporting its role as a continuous refinement of graph-based Riemannian paths. Linear15D becomes shorter in longer-range transitions, but Table 2 shows that it has weaker empirical support: its trajectory points are farther from their 10 nearest real samples in medium and hard buckets. Thus, Linear15D should be interpreted as an ambient off-manifold shortcut, not as a better manifold-consistent trajectory.

Table 1. Quantitative comparison over 25 trajectories. Lower length and energy indicate lower path cost, but must be interpreted jointly with empirical support diagnostics of Table 2.

Bucket	Method	N	Length	Energy	SSIM
Easy	BVP	5	$\mathbf{0.531 \pm 0.122}$	$\mathbf{0.298 \pm 0.146}$	**0.988**
	Dijkstra	5	0.884 ± 0.156	0.807 ± 0.298	0.986
	Linear15D	5	0.554 ± 0.135	0.322 ± 0.167	0.988
Medium	BVP	10	1.338 ± 0.071	1.828 ± 0.201	0.965
	Dijkstra	10	1.408 ± 0.148	2.032 ± 0.421	0.965
	Linear15D	10	$\mathbf{0.835 \pm 0.077}$	$\mathbf{0.713 \pm 0.127}$	**0.971**
Hard	BVP	10	1.531 ± 0.124	2.384 ± 0.393	0.960
	Dijkstra	10	1.649 ± 0.112	2.751 ± 0.388	0.957
	Linear15D	10	$\mathbf{0.980 \pm 0.103}$	$\mathbf{0.979 \pm 0.226}$	**0.968**

Table 2. Average distance to the 10 nearest real samples in the reduced 15D space. Lower values indicate stronger empirical support.

Bucket	BVP	Dijkstra	Linear15D
Easy	5.341	**5.026**	5.288
Medium	**4.527**	4.608	5.171
Hard	**4.423**	4.569	5.274

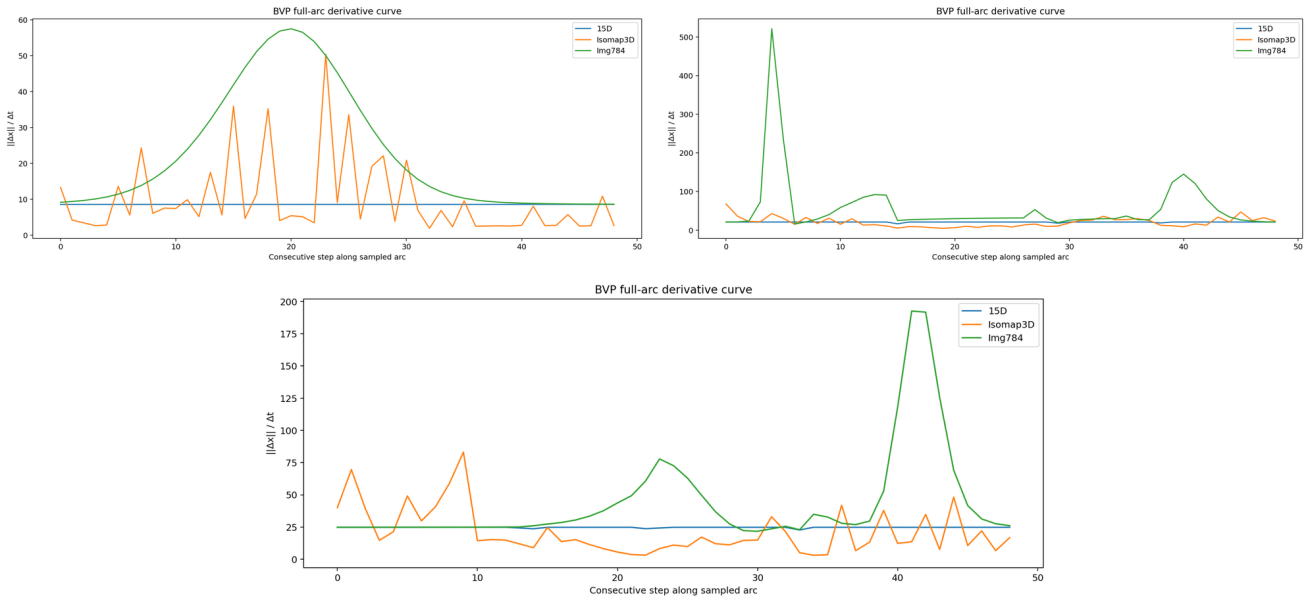


Figure 3. Derivative profiles for representative easy, medium, and hard BVP trajectories. The 15D curves correspond to the reduced representation space.

Figure 3 analyzes the numerical regularity of representative BVP trajectories from

the easy, medium, and hard buckets. Each panel shows derivative magnitudes in three representations: *t*he reduced space where the BVP is solved *(15D)*; *Img784*, the reconstructed image space with dimension $D = 784$; and *Isomap3D*, the 3D visualization projection. The 15D curve is therefore the primary regularity diagnostic, while Img784 and Isomap3D reflect reconstruction and visualization effects. Across the three examples, the 15D derivative profiles show no abrupt jumps along the sampled trajectories. This suggests a smooth BVP refinement of the graph-based initialization, without irregular or oscillatory behavior.

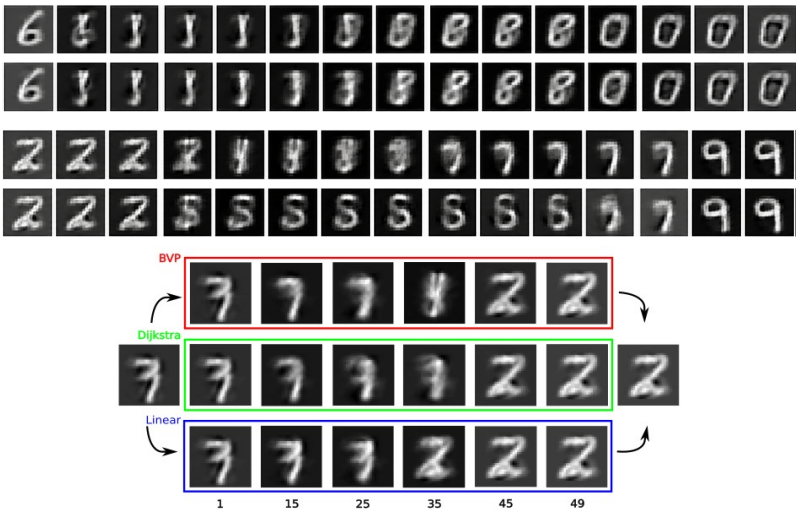


Figure 4. Representative hard morphing trajectories.Red represents our approach, green dijkstra, and blue linear interpolation.

Finally, Figure 4 shows representative hard morphing trajectories. The first two panels compare BVP and Dijkstra for two hard source-target pairs, with BVP shown first and Dijkstra second. The last panel compares BVP, Dijkstra, and Linear15D for the same pair. The Riemannian paths show more gradual digit-like transitions, while Linear15D changes more directly toward the target, supporting the nearest-neighbor evidence that visually smooth interpolation may still cross less supported regions of the reduced representation space.

6. Conclusion

We presented a data-driven pipeline that combines medoid-based clustering, local PCA metric estimation, Procrustes alignment, and RBF blending to learn a spatially varying Riemannian metric over reduced image representations. Geodesics are then computed as BVP trajectories initialized from Riemannian graph paths, with checkpointed continuation for hard cases.

MNIST experiments show that BVP refines the Dijkstra approximation under the learned metric, reducing Riemannian length and energy for most trajectories. Linear15D highlights the limitation of ambient interpolation: visually smooth straight-line paths may still move away from the sampled data manifold, as confirmed by the 15D nearest-neighbor diagnostic in medium and hard transitions.

The results demonstrate that data-driven Riemannian measures serve as an effective instrument for examining the fundamental geometric structure in learned image

representations. We demonstrate that graph-based pathways, continuous boundary value problem geodesics, and ambient linear interpolation include distinct facets of the representation geometry and should not be regarded as interchangeable benchmarks.

Future work will incorporate density-aware penalties to discourage off-manifold shortcuts, quantify computational cost, analyze sensitivity to representation dimension and number of local metric regions, compare with global Riemannian distance-field approach such as [Li et al. 2026] and a non-parametric one [Diepeveen and Needell 2025], and develop curvature-based diagnostics as post-hoc analyses of the learned metric field.

References

Arvanitidis, G., González-Duque, M., Pouplin, A., Kalatzis, D., and Hauberg, S. (2022). Pulling back information geometry. In *International Conference on Artificial Intelligence and Statistics*, pages 4872–4894. PMLR.

Arvanitidis, G., Hansen, L. K., and Hauberg, S. (2018). Latent space oddity: on the curvature of deep generative models. In *International Conference on Learning Representations*.

Arvanitidis, G., Hauberg, S., and Schölkopf, B. (2021). Geometrically enriched latent spaces. In *International Conference on Artificial Intelligence and Statistics*, pages 631–639. PMLR.

Buhmann, M. D. (2000). Radial basis functions. *Acta numerica*, 9:1–38.

Chen, Z., Song, Y., Wu, X., and Sebe, N. (2026). Fast and stable riemannian metrics on SPD manifolds via cholesky product geometry. In *The Fourteenth International Conference on Learning Representations*.

Diepeveen, W. (2024). Pulling back symmetric riemannian geometry for data analysis.

Diepeveen, W. and Needell, D. (2025). Manifold learning with normalizing flows: Towards regularity, expressivity and iso-riemannian geometry. *arXiv preprint arXiv:2505.08087*.

do Carmo, M. P. (2008). *Geometria riemanniana*. Instituto de Matemática Pura e Aplicada.

Goodall, C. (2018). Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society: Series B (Methodological)*, 53(2):285–321.

Gruffaz, S. and Sassen, J. (2025). Riemannian metric learning: Closer to you than you imagine. *arXiv preprint arXiv:2503.05321*.

Harlim, J., Jiang, S. W., and Peoples, J. W. (2023). Radial basis approximation of tensor fields on manifolds: from operator estimation to manifold learning. *Journal of Machine Learning Research*, 24(345):1–85.

Hartwig, F., Sassen, J., Braunsmann, J., Rumpf, M., and Wirth, B. (2026). Geodesic calculus on implicitly defined latent manifolds.

Jolliffe, I. T. and Cadima, J. (2016). Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202.

Kapuńniak, 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. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

Kaya, M. and Bilge, H. Ş. (2019). Deep metric learning: A survey. *Symmetry*, 11(9):1066.

Kulis, B. (2013). Metric learning: A survey. *Foundations and Trends® in Machine Learning*, 5(4):287–364.

Li, Y., Qiu, J., and Calinon, S. (2026). A riemannian take on distance fields and geodesic flows in robotics. *The International Journal of Robotics Research*, (02783649261420233).

Miranda Jr, G. F., Giraldi, G., Thomaz, C. E., and Millàn, D. (2015). Composition of local normal coordinates and polyhedral geometry in riemannian manifold learning. *International Journal of Natural Computing Research (IJNCR)*, 5(2):37–68.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Rygaard, F. M., Zhu, S., Jin, Y., Hauberg, S., and Fletcher, T. (2026). A likely geometry of generative models.

Schroff, F., Kalenichenko, D., and Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823.

Song, H. J. and Itti, L. (2025). Riemannian-geometric fingerprints of generative models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11425–11435.

Tenenbaum, J. B., de Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612.

Yang, H., Sun, B., Chen, L., Pavel, A., and Huang, Q. (2023). Geolatent: A geometric approach to latent space design for deformable shape generators. *ACM Transactions on Graphics (TOG)*, 42(6):1–20.

Yu, H., Hauberg, S., Hartmann, M., Klami, A., and Arvanitidis, G. (2026). Learning geometry and topology via multi-chart flows. In *The 29th International Conference on Artificial Intelligence and Statistics*.

Zheng, W., Zhang, B., Lu, J., and Zhou, J. (2021). Deep relational metric learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12065–12074.

Zhou, Y., Chen, C., and Xu, J. (2020). Learning manifold implicitly via explicit heat-kernel learning. *Advances in Neural Information Processing Systems*, 33:477–487.