A Heat Diffusion Perspective on Geodesic Preserving Dimensionality Reduction

Guillaume Huguet [*1] **Alexander Tong** [*1] **Edward De Brouwer** [*2] **Yanlei Zhang** [1] **Guy Wolf** [1] **Ian Adelstein** [2 3] **Smita Krishnaswamy** [2 3]

Abstract

Diffusion-based manifold learning methods have proven useful in representation learning and dimensionality reduction of modern high dimensional, high throughput, noisy datasets. Such datasets are especially present in fields like biology and physics. While it is thought that these methods preserve underlying manifold structure of data by learning a proxy for geodesic distances, no specific theoretical links have been established. Here, we establish such a link via results in Riemannian geometry explicitly connecting heat diffusion to manifold distances. In this process, we also formulate a more general heat kernel based manifold embedding method that we call *heat geodesic embeddings*. This novel perspective makes clearer the choices available in manifold learning and denoising. Results show that our method outperforms existing state of the art in preserving ground truth manifold distances, and preserving cluster structure in toy datasets. We also showcase our method on single cell RNA-sequencing datasets with both continuum and cluster structure, where our method enables interpolation of withheld timepoints of data.

1. Introduction

The advent of high throughput and high dimensional data in various fields of science have made dimensionality reduction and visualization techniques an indispensable part of exploratory analysis. Diffusion-based manifold learning methods, based on the data diffusion operator, first defined in (Coifman & Lafon, 2006), have proven especially useful due to their ability to handle noise and density variations while preserving structure. As a result, diffusion-based dimensionality reduction methods, such as PHATE (Moon et al., 2019), T-PHATE (Busch et al., 2023), and diffusion maps (Coifman & Lafon, 2006), have emerged as methods for analyzing high throughput noisy data in various situations. While these methods are surmised to learn manifold geodesic distances, no specific theoretical links have been established. Here, we establish such a link by using Varadhan's formula (Varadhan, 1967) and a parabolic Harnack inequality (Saloff-Coste, 2010), which relate manifold distances to heat diffusion directly. This lens gives new insight into existing dimensionality reduction methods, including when they preserve geodesics, and suggests a new method for dimensionality reduction to explicitly preserve geodesics, which we call *heat geodesic embeddings*[1].

Our contributions are as follows: (1) We define the *heat-geodesic* dissimilarity based on Varadhan's formula. (2) Based on this dissimilarity, we present a versatile geodesic-preserving method for dimensionality reduction which we call *heat geodesic embedding.* (3) We establish a relationship between diffusion-based distances and the heat-geodesic dissimilarity. (4) We establish connections between our method and popular dimensionality reduction techniques such as PHATE and t-SNE, shedding light on their geodesic preservation and denoising properties based on modifications of the computed dissimilarity and distance preservation losses. (5) We empirically demonstrate the advantages of Heat Geodesic Embedding in preserving manifold geodesic distances in several experiments showcasing more faithful manifold distances in the embedding space, as well as our ability to interpolate data within the manifold.

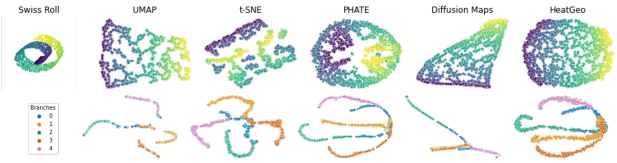

Figure 1. Embeddings of the Swiss roll (top) and Tree (bottom) datasets for different manifold learning methods. Our HeatGeo method correctly unrolls the Swiss roll while t-SNE and UMAP create undesirable artificial clusters.

[*]Equal contribution [1]Université de Montréal; Mila - Quebec AI Institute [2]Yale University [3]Co-senior authors. Correspondence to: Guillaume Huguet <guillaume.huguet@mila.quebec>.

Presented at the 2nd Annual Workshop on Topology, Algebra, and Geometry in Machine Learning (TAG-ML) at the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. 2023. *Copyright 2023 by the author(s).*

[1]https://github.com/KrishnaswamyLab/HeatGeo

2. Preliminaries

First, we introduce fundamental notions that form the basis of our manifold learning methods: Varadhan's formula (Varadhan, 1967) on a manifold, diffusion processes on graphs, efficient heat kernel approximations, and multidimensional scaling (Kruskal, 1964).

Varadhan's formula Varadhan's formula is a powerful tool in differential geometry that establishes a connection between the heat kernel and the shortest path (geodesic) distance on a Riemannian manifold. Its versatility has led to widespread applications in machine learning (Crane et al., 2013; Solomon et al., 2015; Sun et al., 2009). Let (M, g) be a closed Riemannian manifold, and Δ the Laplace-Beltrami operator on M. The heat kernel $h_t(x, y)$ on M is the minimal positive fundamental solution of the heat equation $\frac{\partial u}{\partial t} = \Delta u$ with initial condition $h_0(x, y) = \delta_x(y)$. In Euclidean space the heat kernel is $h_t(x, y) = (4\pi t)^{-n/2}\, e^{-d(x,y)^2/4t}$ so that $-4t \log h_t(x, y) = 2nt \log(4\pi t) + d^2(x, y)$ and we observe the following limiting behavior:

$$\lim_{t\to 0} -4t \log h_t(x, y) = d^2(x, y). \tag{1}$$

Varadhan (Varadhan, 1967) proved that eq. 1 (now Varadhan's formula) holds more generally on complete Riemannian manifolds M, where $d(x, y)$ is the geodesic distance on M, and the convergence is uniform over compact subsets of M.

Graph construction and diffusion Our construction starts by creating a graph from a point cloud dataset $\boldsymbol{X}$. We use a kernel function $\kappa : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+$, such that the (weighted) adjacency matrix is $\boldsymbol{W}_{ij} := \kappa(x_i, x_j)$ for all $x_i, x_j \in \boldsymbol{X}$. The graph Laplacian $\boldsymbol{L} := \boldsymbol{Q} - \boldsymbol{W}$ (where $\boldsymbol{Q}_{ii} = \sum_j \boldsymbol{W}_{ij}$) is an operator acting on signals on $\mathcal{G}$ such that it mimics the negative of the Laplace operator. The Laplacian is symmetric positive semi-definite, and has an eigen-decomposition $\boldsymbol{L} = \Psi\Lambda\Psi^T$. Throughout the presentation, we assume that $\boldsymbol{Q}_{ii} > 0$ for all $i \in [n]$. The Laplacian allows us to define the heat equation on $\mathcal{G}$, with respect to an initial signal $\boldsymbol{f}_0 \in \mathbb{R}^n$ on $\mathcal{G}$:

$$\frac{\partial}{\partial t}\boldsymbol{f}(t) + \boldsymbol{L}\boldsymbol{f}(t) = \boldsymbol{0},\ s.t. \quad \boldsymbol{f}(0) = \boldsymbol{f}_0 \quad t \in \mathbb{R}^+. \tag{2}$$

The solution of the above differential equation is obtained with the matrix exponential $\boldsymbol{f}(t) = e^{-t\boldsymbol{L}}\boldsymbol{f}_0$, and we define the heat kernel on the graph as $\boldsymbol{H}_t := e^{-t\boldsymbol{L}}$. The matrix $\boldsymbol{H}_t$ is a diffusion matrix that characterizes how a signal propagate through the graph according to the heat equations.

Multidimensional scaling Given a dissimilarity function d between data points, multidimensional scaling (MDS) (Kruskal, 1964) finds an embedding ϕ such that the difference between the given dissimilarity and the Euclidean distance in the embedded space is minimal across all data points. Formally, for a given function $d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+$, MDS minimizes the following objective:

$$L(\boldsymbol{X}) = \left(\sum_{ij} \left(d(x_i, x_j) - \|\phi(x_i) - \phi(x_j)\|_2 \right)^2 \right)^{1/2}. \tag{3}$$

3. Heat-Geodesic Embedding

In this section, we present our Heat Geodesic Embedding which is summarized in Alg. 1. We consider the discrete case, where we have a set of n points $\{x_i\}_{i=1}^n =: \boldsymbol{X}$ in a high dimensional Euclidean space $x_i \in \mathbb{R}^d$. From this point cloud, we want to define a map $\phi : \mathbb{R}^d \to \mathbb{R}^k$ that embeds the observation in a lower dimensional space. An important property of our embedding is that we preserve manifold geodesic distances in a low dimensional space.

Heat-geodesic Dissimilarity Inspired by Varadhan's formula and the Harnack inequalities, we defined a heat-geodesic dissimilarity based on heat diffusion on graphs. From observations (datapoints) in $\mathbb{R}^n$, we define an undirected graph $\mathcal{G}$, and compute its heat kernel $\boldsymbol{H}_t = e^{-t\boldsymbol{L}}$, where $\boldsymbol{L}$ is the combinatorial or symmetrically normalized graph Laplacian (the heat kernel is thus symmetric).

Definition 3.1. For a diffusion time $t > 0$ and tunable parameter $\sigma > 0$, we define the **heat-geodesic dissimilarity** between $x_i, x_j \in \boldsymbol{X}$ as

$$d_t(x_i, x_j) := [-4t \log(\boldsymbol{H}_t)_{ij} - \sigma 4t \log(\boldsymbol{V}_t)_{ij}]^{1/2}$$

where $\boldsymbol{H}_t$ is the heat kernel on the graph $\mathcal{G}$, and $(\boldsymbol{V}_t)_{ij} := 2[(\boldsymbol{H}_t)_{ii} + (\boldsymbol{H}_t)_{jj}]^{-1}$.

Here the log is applied elementwise, and the term $-4t \log(\boldsymbol{H}_t)_{ij}$ corresponds to the geodesic approximation when $t \to 0$ as in Varadhan's formula. In practice one uses a fixed diffusion time $t > 0$ (we present in appendix how to automatically select t), so we add a symmetric volume correction term as in the Harnack inequality, ensuring that $d_t(x_i, x_j)$ is symmetric. From Sec. 2, we have $h_t(x, x) \simeq V(x, \sqrt{t})^{-1}$, and we use the diagonal of $\boldsymbol{H}_t$ to approximate the inverse of the volume. With this volume correction term and $\sigma = 1$, the dissimilarity is such that $d_t(x_i, x_i) = 0$ for all $t > 0$. When $\sigma = 0$ or the manifold has uniform volume growth (as in the constant curvature setting) we show that the heat-geodesic dissimilarity is order preserving:

Proposition 3.2. *When $\sigma = 0$ or the manifold has uniform volume growth we need only consider the $-4t \log(\boldsymbol{H}_t)_{ij}$ terms. The assumption of pointwise monotonicity of the heat kernel entails that $|x - y| > |x - z|$ implies $\boldsymbol{H}_t(x, y) < \boldsymbol{H}_t(x, z)$. We are able to conclude that $-4t \log \boldsymbol{H}_t(x, y) > -4t \log \boldsymbol{H}_t(x, z)$ and thus $d_t(x, y) > d_t(x, z)$.*

Denoising Distances with Triplet Computations We note that both diffusion maps and PHATE compute a triplet distance between datapoints, i.e., rather than using the direct diffusion probability between datapoints, they use the distance between corresponding rows of a diffusion operator. For a reference dissimilarity like the heat-geodesic, it is defined as $D_{\mathrm{T}}(x_i, x_j) := \|d_t(x_i, \cdot) - d_t(x_j, \cdot)\|_2$. For linear perturbations of the form $d_t(x_i, x_j) + \epsilon$, where $\epsilon \in \mathbb{R}$, the effect of ϵ on $D_{\mathrm{T}}(x_i, x_j)$ is less severe than on $d_t(x_i, x_j)$.

Proposition 3.3. *Denote the perturbed triplet distance by* $\widetilde{D_{\mathrm{T}}}(x_i, x_j) = ||\tilde{d}_t(x_i, \cdot) - \tilde{d}_t(x_j, \cdot)||_2$ *where* $\tilde{d}_t(x_i, x_j) := d_t(x_i, x_j) + \epsilon$ *and* $\tilde{d}_t(x_i, x_k) := d_t(x_i, x_k)$ *for* $k \neq j$. *Then the triplet distance* D_{T} *is robust to perturbations , i.e., for all* $\epsilon > 0$,

$$\left(\frac{\widetilde{D_{\mathrm{T}}}(x_i, x_j)}{D_{\mathrm{T}}(x_i, x_j)}\right)^2 \leq \left(\frac{d_t(x_i, x_j) + \epsilon}{d_t(x_i, x_j)}\right)^2.$$

Heat-geodesic embedding To define a lower dimensional embedding of a point cloud $\boldsymbol{X}$, we construct a matrix from the heat-geodesic dissimilarity, and then use MDS to create the embedding. Our embedding defines a map ϕ that minimizes $\left(d_t(x_i, x_j) - \|\phi(x_i) - \phi(x_j)\|_2\right)^2$, for all $x_i, x_j \in \boldsymbol{X}$. Hence, it preserves the heat-geodesic dissimilarity as the loss decreases to zero. In Alg. 1, we present the main steps of our algorithm using the heat-geodesic dissimilarity.

Algorithm 1 Heat Geodesic Embedding

1: **Input:** $N \times d$ dataset matrix $\boldsymbol{X}$, denoising parameter $\rho \in [0, 1]$, regularization $\sigma > 0$, output dimension k.
2: **Returns:** $N \times k$ embedding matrix $\boldsymbol{E}$.
3: $\boldsymbol{H}_t \leftarrow p_K(\boldsymbol{L}, t)$ *Heat approximation*
4: $\boldsymbol{D} \leftarrow -4t \log \boldsymbol{H}_t + t\sigma \boldsymbol{V}$ $\log$ *is applied elementwise*
5: $\boldsymbol{D} \leftarrow (1 - \rho)\boldsymbol{D} + \rho D_{\mathrm{T}}$ *Triplet interpolation step*
6: Return $\boldsymbol{E} \leftarrow \mathrm{MetricMDS}(\boldsymbol{D}, \|\cdot\|_2, k)$

Relation to PHATE The potential distance in PHATE (Sec. B) is defined by comparing the transition probabilities of two t-steps random walks initialized from different vertices. The transition matrix $\boldsymbol{P}^t$ mimics the heat propagation on a graph. The heat-geodesic dissimilarity provides a new interpretation of PHATE. In the following proposition, we show how the heat-geodesic relates to the PHATE potential distance with a linear combination of t-steps random walks.

Proposition 3.4. *The PHATE potential distance with the heat kernel* $PH_{\boldsymbol{H}_t}$ *can be expressed in terms of the heat-geodesic dissimilarity with* $\sigma = 0$

$$PH_{\boldsymbol{H}_t} = (1/4t)^2 \|d_t(x_i, \cdot) - d_t(x_j, \cdot)\|_2^2,$$

and it is equivalent to a multiscale random walk distance with kernel $\sum_{k>0} m_t(k)\boldsymbol{P}^k$, *where* $m_t(k) := t^k e^{-t}/k!$.

4. Results

In this section, we show the versatility of our method, showcasing its performance in terms of clustering and preserving the structure of continuous manifolds. We use the following methods in our experiments: our *Heat Geodesic Embedding*, *diffusion maps* (Coifman & Lafon, 2006), *PHATE* (Moon et al., 2019), *shortest-path* which estimates the geodesic distance by computing the shortest path between two nodes in a graph built on the point clouds, *t-SNE* (Van der Maaten & Hinton, 2008), and *UMAP* (McInnes et al., 2018). Details about each of these methods, and results for different parameters (graph type, heat approximation, etc.) are given in Appendix E.

Table 1. Pearson and Spearman correlation between the inferred and ground truth distance matrices on the Swiss roll and Tree datasets (higher is better). Best models on average are bolded.

	Swiss roll		Tree	
Method	Pearson	Spearman	Pearson	Spearman
Diffusion Map	0.476 ± 0.226	0.478 ± 0.138	0.656 ± 0.054	0.653 ± 0.057
PHATE	0.457 ± 0.01	0.404 ± 0.024	0.766 ± 0.023	0.743 ± 0.028
Shortest Path	0.497 ± 0.144	0.558 ± 0.134	0.780 ± 0.009	0.757 ± 0.019
HeatGeo (ours)	$\mathbf{0.702 \pm 0.086}$	$\mathbf{0.700 \pm 0.073}$	$\mathbf{0.822 \pm 0.008}$	$\mathbf{0.807 \pm 0.016}$

Distance matrix comparison We start by evaluating the ability of the different methods to recover the ground truth distance matrix of a point cloud. For this task, we use point clouds from the Swiss roll and Tree datasets, for which the ground truth geodesic distance is known. In Fig. 1, we present embeddings of both datasets. Our method recovers the underlying geometry, while other methods create artificial clusters or have too much denoising. We compare the methods according to the Pearson and Spearman correlations of the estimated distance matrices with respect to ground truth. Results are displayed in Tab. 1. We observe that Heat Geodesic Embedding typically outperforms previous methods in terms of the correlation with the ground truth distance matrix.

Temporal data representation. We aim at representing data points from population observed at consecutive points in time. We use single cell gene expression datasets collected across different time points, including the Embryoid Body (EB), IPSC (Moon et al., 2019), and two from the 2022 NeurIPS multimodal single-cell integration challenge (Cite & Multi). We first embed the entire dataset and obfuscate all samples from a particular time point (*e.g.,* $t = 2$). We then estimate the distribution of the missing time point by using displacement interpolation (Villani & Villani, 2009) between the adjacent time points (*e.g.,* $t = 1$ and $t = 3$). We report the Earth Mover Distance (EMD) between the predicted distribution and true distribution. A low EMD suggests that the obfuscated embeddings are naturally located between the previous and later time points, and that the generated embedding captures the temporal evolution

of the data adequately. Results are presented in Tab. 2. Heat Geodesic Embedding outperforms other methods on the EB, Multi, and IPSC datasets and is competitive with other approaches on Cite. We show a graphical depiction of the different embeddings for the embryoid (EB) dataset in Fig. 2.

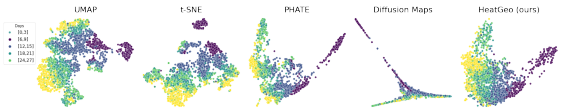

Figure 2. Embeddings of 2000 differentiating cells from embryoid body (Moon et al., 2019) over 28 days. UMAP and t-SNE do not capture the continuous manifold representing the cells' evolution.

Table 2. EMD between a linear interpolation of two consecutive time points $t-1$, $t+1$, and the time points t. Best models on average are bolded (lower is better).

Method	Cite	EB	Multi	IPSC
UMAP	**0.791 ± 0.045**	0.942 ± 0.053	1.418 ± 0.042	0.866 ± 0.058
t-SNE	0.905 ± 0.034	0.964 ± 0.032	1.208 ± 0.087	1.006 ± 0.026
PHATE	1.032 ± 0.037	1.088 ± 0.012	1.254 ± 0.042	0.955 ± 0.033
Diffusion Maps	0.989 ± 0.080	0.965 ± 0.077	1.227 ± 0.086	0.821 ± 0.039
HeatGeo (ours)	0.890 ± 0.046	**0.733 ± 0.036**	**0.958 ± 0.044**	**0.365 ± 0.056**

5. Conclusion

The ability to visualize complex high-dimensional data in an interpretable and rigorous way is a crucial tool of scientific discovery. In this work, we took a step in that direction by proposing a general framework for understanding diffusion-based dimensionality reduction methods through the lens of Riemannian geometry. This allowed us to define a novel embedding based on the heat geodesic dissimilarity—a more direct measure of manifold distance. Theoretically, we showed that our methods brings greater versatility than previous approaches and can help gaining insight into popular manifold learning methods such as diffusion maps, PHATE, and SNE. Experimentally, we demonstrated that it also results in better geodesic distance preservation and excels both at clustering and preserving the structure of a continuous manifold. This contrasts with previous methods that are typically only effective at a single of these tasks.

References

Busch, E. L., Huang, J., Benz, A., Wallenstein, T., Lajoie, G., Wolf, G., Krishnaswamy, S., and Turk-Browne, N. B. Multi-view manifold learning of human brain-state trajectories. *Nature Computational Science*, pp. 1–14, 2023.

Coifman, R. R. and Lafon, S. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006.

Crane, K., Weischedel, C., and Wardetzky, M. Geodesics in heat: A new approach to computing distance based on heat flow. *ACM Transactions on Graphics (TOG)*, 32(5): 1–11, 2013.

Hinton, G. E. and Roweis, S. Stochastic neighbor embedding. *Advances in neural information processing systems*, 15, 2002.

Huang, S.-G., Lyu, I., Qiu, A., and Chung, M. K. Fast polynomial approximation of heat kernel convolution on manifolds and its application to brain sulcal and gyral graph pattern analysis. *IEEE transactions on medical imaging*, 39(6):2201–2212, 2020.

Kruskal, J. B. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

Linderman, G. C. and Steinerberger, S. Clustering with t-SNE, provably. *arXiv:1706.02582 [cs, stat]*, 2017.

McInnes, L., Healy, J., and Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Moon, K. R., van Dijk, D., Wang, Z., Gigante, S., Burkhardt, D. B., Chen, W. S., Yim, K., Elzen, A. v. d., Hirn, M. J., Coifman, R. R., Ivanova, N. B., Wolf, G., and Krishnaswamy, S. Visualizing structure and transitions in high-dimensional biological data. *Nat Biotechnol*, 37 (12):1482–1492, 2019.

Saloff-Coste, L. The heat kernel and its estimates. *Probabilistic approach to geometry*, 57:405–436, 2010.

Satopaa, V., Albrecht, J., Irwin, D., and Raghavan, B. Finding a" kneedle" in a haystack: Detecting knee points in system behavior. In *2011 31st international conference on distributed computing systems workshops*, pp. 166–171. IEEE, 2011.

Solomon, J., De Goes, F., Peyré, G., Cuturi, M., Butscher, A., Nguyen, A., Du, T., and Guibas, L. Convolutional wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Transactions on Graphics (ToG)*, 34(4):1–11, 2015.

Sun, J., Ovsjanikov, M., and Guibas, L. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer graphics forum*, volume 28, pp. 1383–1392. Wiley Online Library, 2009.

Van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Varadhan, S. R. S. On the behavior of the fundamental solution of the heat equation with variable coefficients. *Communications on Pure and Applied Mathematics*, 20 (2):431–455, 1967.

Villani, C. and Villani, C. Displacement interpolation. *Optimal Transport: Old and New*, pp. 113–162, 2009.

A. Theory and algorithm details

A.1. Kernel preserving embeddings

In this section, we attempt to create a generalized framework for dimensionality reduction methods. These methods often have been viewed as disparate or competing but here we show that many of them are related to one another given the right template for methodology comparison. In order to do this, we introduce a general definition suited for distance-preserving dimensionality reduction methods. With this definition, we can cast many dimensionality reduction methods within the same framework, and easily compare them. We recall that the observations in the ambient space are denoted x, and those in the embedded space are denoted y. The definition relies on kernel functions H_t^x , H_t^y defined respectively on the ambient and embedded spaces and on transformations T^x, T^y applied to the kernels. We recall that a divergence $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+$ is such that $f(a, b) = 0$ if and only if $a = b$ and $f(a, a + \delta)$ is a positive semi-definite quadratic form for infinitesimal δ.

Definition A.1. We define a **kernel features preserving embedding** as an embedding which minimizes a loss L between a transformation T^x of the ambient space kernel H_t^x and its embedded space counterpart

$$L := f(T^x(H_t^x), T^y(H_{t'}^y)), \tag{4}$$

where f is any C^2 divergence on $\mathbb{R}_{\geq 0}$.

Example A.2. We formulate MDS as a kernel feature-preserving embedding. Suppose we want to preserve the Euclidean distance, we have $H_t^x(x_i, x_j) = \|x_i - x_j\|_2$, $H_t^y(y_i, y_j) = \|y_i - y_j\|_2$, $f(a, b) = \|a - b\|_2$, and $T^x = T^y = I$.

In the following, we present popular dimensionality reduction methods that are kernel features preserving embeddings. With this definition, we can distinguish between methods that a preserve a kernel via affinities or distances. For the methods considered in this work, H_t^x is an affinity kernel, but its construction varies from one method to another. In PHATE and Diffusion maps, H_t^x is a random walk $\boldsymbol{P}$, while in Heat Geodesic Embedding we use the heat kernel $\boldsymbol{H}_t$. t-SNE defines H_t^x as a symmetrized random walk matrix from a Gaussian kernel, while UMAP uses an unnormalized version. Methods such as PHATE and diffusion maps define a new distance matrix from a kernel in the ambient space and preserve these distances in the embedded space. Other methods like t-SNE and UMAP define similarities from a kernel and aim to preserve these similarities in the ambient space and embedded space via an entropy-based loss. We note the Kullback–Leibler divergence $D_{\mathrm{KL}}(a, b) = \sum_{ij} a_{ij} \log[a_{ij}/b_{ij}]$.

Proposition A.3. *The embeddings methods HeatGeo, PHATE, Diffusion Maps, SNE, t-SNE, and UMAP are kernel feature-preserving embeddings.*

Proof. We assume that the affinity kernel in the ambient space H_t^x, is given, to complete the proof we need to define f, H_t^y, T^x, T^y for all methods.

We start with the distance preserving embeddings; HeatGeo, PHATE, and Diffusion Maps. For these methods, the kernel in the embed space is simply $H_t^y(y_i, y_j) = \|y_i - y_j\|_2$, without transformation, i.e. $T^y = I$. Since they preserve a distance, the loss is $f(T^x(H_t^x), T^y(H_{t'}^y)) = \|H_t^x - H_{t'}^y\|_2$.

In the Heat Geodesic Embedding we apply a transformation on $H_t^x = \boldsymbol{H}_t$ to define a dissimilarity, hence $T^x(H_t^x) = -t \log H_t^x$ (for $\sigma = 0$), where $\log$ is applied elementwise.

In PHATE, the potential distance is equivalent to $(T^x(H_t^x))_{ij} = \| - \log(H_t^x)_i + \log(H_t^x)_j\|_2$. In Diffusion Maps, the diffusion distance is $(T^x(H_t^x))_{ij} = \|(H_t^x)_i - (H_t^x)_j\|_2$.

SNE, t-SNE, and UMAP preserve affinities from a kernel. For these three methods, the loss is a divergence between distributions, namely $f = D_{\mathrm{KL}}$. They vary by defining different affinity kernel and transformation in the embedded space. SNE uses the unnormalized kernel $H_t^y(y_i, y_j) = \exp(-(1/t)\|y_i - y_j\|_2^2)$, with $T^x = T^y = I$. Whereas, t-SNE uses $(H_1^y)_{ij} = (1 + \|y_i - y_j\|^2)^{-1}$, and $T^x = T^y = I$. UMAP define a pointwise transformation in the embedded space with $(H_1^y)_{ij} = (1 + \|y_i - y_j\|^2)^{-1}$, $(T^y(H_t^y))_{ij} = (H_1^y)_{ij}/(1 - (H_1^y)_{ij})$, and $T^x = I$.

We summarize the choice of kernels and functions in Tab. 3 □

A.2. Proofs

Proposition A.4. *When $\sigma = 0$ or the manifold has uniform volume growth we need only consider the $-4t \log(\boldsymbol{H}_t)_{ij}$ terms. The assumption of pointwise monotonicity of the heat kernel entails that $|x - y| > |x - z|$ implies $\boldsymbol{H}_t(x, y) < \boldsymbol{H}_t(x, z)$.*

Table 3. Overview of kernel preserving methods.

Method	$H_t^y(y_i, y_j)$	$T^x(H_t^x)$	$T^y(H_t^y)$	f
PHATE	$\|y_i - y_j\|_2$	$\| -\log(H_t^x)_i + \log(H_t^x)_j\|_2$	H_t^y	$\|\cdot\|_2$
Heat Geodesic	$\|y_i - y_j\|_2$	$-t\log(H_t^x)_{ij}$	H_t^y	$\|\cdot\|_2$
Diffusion Maps	$\|y_i - y_j\|_2$	$\|(H_t^x)_i - (H_t^x)_j\|_2$	H_t^y	$\|\cdot\|_2$
SNE	$\exp(-(\frac{1}{t})\|y_i - y_j\|_2^2)$	H_t^x	H_t^y	D_{KL}
t-SNE	$(1 + \|y_i - y_j\|^2)^{-1}$	H_t^x	H_t^y	D_{KL}
UMAP	$(1 + \|y_i - y_j\|^2)^{-1}$	H_t^x	$\frac{(H_1^y)_{ij}}{(1-(H_1^y)_{ij})}$	D_{KL}

We are able to conclude that $-4t\log \boldsymbol{H}_t(x,y) > -4t\log \boldsymbol{H}_t(x,z)$ and thus $d_t(x,y) > d_t(x,z)$.

Proof of Proposition 3.2. When $\sigma = 0$ or the manifold has uniform volume growth we need only consider the $-4t\log(\boldsymbol{H}_t)_{ij}$ terms. The assumption of pointwise monotonicity of the heat kernel entails that $|x-y| > |x-z|$ implies $\boldsymbol{H}_t(x,y) < \boldsymbol{H}_t(x,z)$. We are able to conclude that $-4t\log \boldsymbol{H}_t(x,y) > -4t\log \boldsymbol{H}_t(x,z)$ and thus $d_t(x,y) > d_t(x,z)$. □

Proposition A.5. *Denote the perturbed triplet distance by $\widetilde{D_{\mathrm{T}}}(x_i,x_j) = ||\tilde{d}_t(x_i,\cdot) - \tilde{d}_t(x_j,\cdot)||_2$ where $\tilde{d}_t(x_i,x_j) := d_t(x_i,x_j) + \epsilon$ and $\tilde{d}_t(x_i,x_k) := d_t(x_i,x_k)$ for $k \neq j$. Then the triplet distance D_{T} is robust to perturbations , i.e., for all $\epsilon > 0$,*

$$\left(\frac{\widetilde{D_{\mathrm{T}}}(x_i,x_j)}{D_{\mathrm{T}}(x_i,x_j)}\right)^2 \leq \left(\frac{d_t(x_i,x_j)+\epsilon}{d_t(x_i,x_j)}\right)^2.$$

Proof of Proposition 3.3. The effect of the noise on the square distance is $(d_t(x_i,x_j)+\epsilon)^2/d(x_i,x_j)^2 = 1 + (2\epsilon d_t(x_i,x_j) + \epsilon^2)/d(x_i,x_j)^2$. Denoting the perturbed triplet distance by $\widetilde{D_{\mathrm{T}}}$, we have

$$\frac{\widetilde{D_{\mathrm{T}}}(x_i,x_j)^2}{D_{\mathrm{T}}(x_i,x_j)^2} = \frac{\sum_{k\neq i,j}\big(d_t(x_i,x_k) - d_t(x_j,x_k)\big)^2 + 2(d_t(x_i,x_j)+\epsilon)^2}{D_{\mathrm{T}}(x_i,x_j)^2} = 1 + \frac{4\epsilon d(x_i,x_j) + 2\epsilon^2}{D_{\mathrm{T}}(x_i,x_j)^2},$$

and we have

$$\frac{4\epsilon d(x_i,x_j) + 2\epsilon^2}{D_T(x_i,x_j)^2} \leq \frac{2\epsilon d_t(x_i,x_j) + \epsilon^2}{d_t(x_i,x_j)^2}$$

For $\epsilon > 0$, this gives

$$\epsilon \geq \frac{4d_t(x_i,x_j)^3 - 2d_t(x_i,x_j)D_T(x_i,x_j)^2}{D_t(x_i,x_j)^2 - 2d_t(x_i,x_j)^2} = -2d_t(x_i,x_j).$$

For $\epsilon < 0$, we have

$$\epsilon \leq \frac{4d_t(x_i,x_j)^3 - 2d_t(x_i,x_j)D_T(x_i,x_j)^2}{D_t(x_i,x_j)^2 - 2d_t(x_i,x_j)^2} = -2d_t(x_i,x_j).$$

Thus $\epsilon \in (-\infty, -2d_t(x_i,x_j)) \cup (0,\infty)$. As we require the perturbation factor $\epsilon << d_t(x_i,x_j)$, hence we choose $\epsilon \in (0,\infty)$.

□

Proposition A.6. *The PHATE potential distance with the heat kernel $PH_{\boldsymbol{H}_t}$ can be expressed in terms of the heat-geodesic dissimilarity with $\sigma = 0$*

$$PH_{\boldsymbol{H}_t} = (1/4t)^2\|d_t(x_i,\cdot) - d_t(x_j,\cdot)\|_2^2,$$

and it is equivalent to a multiscale random walk distance with kernel $\sum_{k>0} m_t(k)\boldsymbol{P}^k$, where $m_t(k) := t^k e^{-t}/k!$.

Proof of Proposition 3.4. For $\sigma = 0$, we have $d_t(x_i,x_j) = -4t\log(\boldsymbol{H}_t)_{ij}$, the relation between the PHATE potential and the heat-geodesic follows from the definition

$$\begin{aligned} PH_{\boldsymbol{H}_t} &= \sum_k \big(-\log \boldsymbol{H}_t(x_i,x_k) + \log \boldsymbol{H}_t(x_j,x_k)\big)^2 \\ &= (1/4t)^2\|d_t(x_i,\cdot) - d_t(x_j,\cdot)\|_2^2. \end{aligned}$$

Using the heat kernel $\boldsymbol{H}_t$ with the random walk Laplacian $\boldsymbol{L}_{rw} = \boldsymbol{Q}^{-1}\boldsymbol{L} = \boldsymbol{I}_n - \boldsymbol{Q}^{-1}\boldsymbol{W}$ corresponds to a multiscale random walk kernel. Recall that we can write $\boldsymbol{L}_{rw}$ in terms of the symmetric Laplacian $\boldsymbol{L}_{rw} = \boldsymbol{Q}^{-1/2}\boldsymbol{L}_s\boldsymbol{Q}^{1/2}$, meaning that the two matrices are similar, hence admit the same eigenvalues Λ. We also know that $\boldsymbol{L}_s$ is diagonalizable, since we can write $\boldsymbol{L}_s = \boldsymbol{Q}^{-1/2}\boldsymbol{L}\boldsymbol{Q}^{-1/2} = \boldsymbol{Q}^{-1/2}\Psi\Lambda\Psi^T\boldsymbol{Q}^{-1/2}$. In particular, we have $\boldsymbol{L}_{rw} = \boldsymbol{S}\Lambda\boldsymbol{S}^{-1}$, where $\boldsymbol{S} := \boldsymbol{Q}^{-1/2}\Psi$. The random walk matrix can be written as $\boldsymbol{P} = \boldsymbol{I}_n - \boldsymbol{R}_{rw}$, hence its eigenvalues are $(\boldsymbol{I}_n - \Lambda)$, and we can write $\boldsymbol{P}^t = \boldsymbol{S}(\boldsymbol{I}_n - \Lambda)^t\boldsymbol{S}^{-1}$. Similarly, the heat kernel with the random walk Laplacian can be written as $\boldsymbol{H}_t = \boldsymbol{S}e^{-t\Lambda}\boldsymbol{S}^{-1}$. Interestingly, we can relate the eigenvalues of $\boldsymbol{H}_t$ and $\boldsymbol{P}$ with the Poisson distribution. Note the probability mass function of a Poisson as $m_t(k) := t^k e^{-t}/k!$, for $t \geq 0$, we have

$$e^{-t(1-\mu)} = e^{-t}\sum_{k\geq 0}\frac{(t\mu)^k}{k!} = \sum_{k\geq 0} m_t(k)\mu^k. \tag{5}$$

We note that (5) is the probability generating function of a Poisson distribution with parameter t, i.e. $\mathbb{E}[\mu^X]$, where $X \sim \text{Poisson}(t)$. With this relationship, we can express $\boldsymbol{H}_t$ as a linear combination of $\boldsymbol{P}^t$ weighted by the Poisson distribution. Indeed, substituting $\lambda = 1 - \mu$ in (5) links the eigenvalues of $\boldsymbol{H}_t$ and $\boldsymbol{P}$. We write the heat kernel as a linear combination of random walks weighted by the Poisson distribution, we have

$$\boldsymbol{H}_t = \boldsymbol{S}e^{-t\Lambda}\boldsymbol{S}^{-1} = \boldsymbol{S}\sum_{k=0}^{\infty} m_t(k)(\boldsymbol{I}_n - \Lambda)^k\boldsymbol{S}^{-1} = \sum_{k=0}^{\infty} m_t(k)\boldsymbol{P}^k.$$

□

A.3. Algorithm details

For the knee-point detection we use the Kneedle algorithm (Satopaa et al., 2011). It identifies a knee-point as a point where the curvature decreases maximally between points (using finite differences). We summarize the four main steps of the algorithm for a function $f(x)$, and we refer to (Satopaa et al., 2011) for additional details.

1. Smoothing with a spline to preserve the shape of the function.
2. Normalize the values, so the algorithm does not depend on the magnitude of the observations.
3. Computing the set of finite differences for x and $y := f(x)$, e.g. $y_{d_i} := f(x_i) - x_i$.
4. Evaluating local maxima of the difference curve y_{d_i}, and select the knee-point using a threshold based on the average difference between consecutive x.

B. Related Work

We review state-of-the-art embedding methods and contextualize them with respect to Heat Geodesic Embedding. Given a set of high-dimensional datapoints, the objective of embedding methods is to create a map that embeds the observations in a lower dimensional space, while preserving distances or similarities. Different methods vary by their choice of distance or dissimilarity functions, as shown below.

Diffusion maps In diffusion maps (Coifman & Lafon, 2006), an embedding in k dimensions is defined via the first k non-trivial right eigenvectors of $\boldsymbol{P}^t$ weighted by their eigenvalues. The embedding preserves the ***diffusion distance*** $DM_{\boldsymbol{P}}(x_i, x_j) := \|(\boldsymbol{\delta_i}\boldsymbol{P}^t - \boldsymbol{\delta_j}\boldsymbol{P}^t)(1/\boldsymbol{\pi})\|_2$, where $\boldsymbol{\delta}_i$ is a vector such that $(\boldsymbol{\delta}_i)_j = 1$ if $j = i$ and 0 otherwise, and $\boldsymbol{\pi}$ is the stationary distribution of $\boldsymbol{P}$. Intuitively, $DM_{\boldsymbol{P}}(x_i, x_j)$ considers all the t-steps paths between x_i and x_j. A larger diffusion time can be seen as a low frequency graph filter, i.e. keeping only information from the low frequency transitions such has the stationary distributions. For this reason, using diffusion with $t > 1$ helps denoising the relationship between observations.

PHATE This diffusion-based method preserves the ***potential distance*** (Moon et al., 2019) $PH_{\boldsymbol{P}} := \| - \log \boldsymbol{\delta}_i\boldsymbol{P}^t + \log \boldsymbol{\delta}_j\boldsymbol{P}^t\|_2$, and justifies this approach using the $\log$ transformation to prevent nearest neighbors from dominating the distances. An alternative approach is suggested using a square root transformation. Part of our contributions is to justify the $\log$ transformation from a geometric point of view. The embedding is defined using multidimensional scaling, which we present below.

SNE, t-SNE, UMAP Well-known attraction/repulsion methods such as SNE (Hinton & Roweis, 2002), t-SNE (Van der Maaten & Hinton, 2008), and UMAP (McInnes et al., 2018) define an affinity matrix with entries p_{ij} in the ambient space, and another affinity matrix with entries q_{ij} in the embedded space. To define the embedding, a loss between the two affinity matrices is minimized. Specifically, the loss function is $D_{\mathrm{KL}}(p||q) := \sum_{ij} p_{ij} \log p_{ij}/q_{ij}$ in SNE and t-SNE, whereas UMAP adds $D_{\mathrm{KL}}(1-p||1-q)$. While these methods preserves affinities, they do not preserve any types of distances in the embedding.

C. Additional details and results

C.1. Fast computation of Heat diffusion

Exact computation of the (discrete) heat kernel H_t is computationally costly, requiring a full eigendecomposition in $O(n^3)$ time. Fortunately, multiple fast approximations have been proposed, including using orthogonal polynomials or the Euler backward methods. In this work, we use Chebyshev polynomials, as they have been shown to converge faster than other polynomials on this problem (Huang et al., 2020).

Chebyshev polynomials are defined by the recursive relation $\{T_k\}_{k\in\mathbb{N}}$ with $T_0(y) = 0, T_1(y) = y$ and $T_k(y) = 2yT_{k-1}(y) - T_{k-2}(y)$ for $k \geq 2$. Assuming that the largest eigenvalue is less than two (which holds for the normalized Laplacian), we approximate the heat kernel with the truncated polynomials of order K

$$\boldsymbol{H}_t \approx p_K(\boldsymbol{L}, t) := \frac{b_{t,0}}{2} + \sum_{k=1}^{K} b_{t,k} T_k(\boldsymbol{L} - \boldsymbol{I}_n), \tag{6}$$

where the $K+1$ scalar coefficients $\{b_{t,i}\}$ depend on time and are evaluated with the Bessel function. Computing $p_K(\boldsymbol{L}, t)\boldsymbol{f}$ requires K matrix-vector product and $K+1$ Bessel function evaluation. The expensive part of the computation are the matrix-vector products, which can be efficient if the Laplacian matrix is sparse. Interestingly, we note that the evaluation of T_k do not depend on the diffusion time. Thus, to compute multiple approximations of the heat kernel $\{p_K(\boldsymbol{L}, t)\}_{t\in\mathcal{T}}$, only necessitates reweighting the truncated polynomial $\{T_k\}_{k\in[1,\ldots,K]}$ with the corresponding $|\mathcal{T}|$ sets of Bessel coefficients. The overall complexity is dominated by the truncated polynomial computation which takes $O(K(E+n))$ time where E is the number of non-zero values in $\boldsymbol{L}$.

Another possible approximation is using the Euler backward method. It requires solving K systems of linear equations $\boldsymbol{f}(t) = (\boldsymbol{I}_n + (t/K)\boldsymbol{L})^{-K}\boldsymbol{f}(0)$, which can be efficient for sparse matrices using the Cholesky decomposition (Solomon et al., 2015). We quantify the differences between the heat kernel approximations in Appendix E.

C.2. Optimal diffusion time

Varadhan's formula suggests a small value of diffusion time t to approximate geodesic distance on a manifold. However, in the discrete data setting, geodesics are based on graph constructions, which in turn rely on nearest neighbors. Thus, small t can lead to disconnected graphs. Additionally, increasing t can serve as a way of denoising the kernel (which is often computed from noisy data) as it implements a low-pass filter over the eigenvalues, providing the additional advantage of adding noise tolerance. By computing a sequence of heat kernels $(\boldsymbol{H}_t)_t$ and evaluating their entropy $H(\boldsymbol{H}_t) := -\sum_{ij}(\boldsymbol{H}_t)_{ij}\log(\boldsymbol{H}_t)_{ij}$, we select t with the knee-point method (Satopaa et al., 2011) on the function $t \mapsto H(\boldsymbol{H}_t)$. We show in Sec. 4 that our heuristic for determining the diffusion time automatically leads to better overall results.

C.3. Weighted MDS

The loss in MDS (eq.3) is usually defined with uniform weights. Here, we optionally weight the loss by the heat kernel. For $x_i, x_j \in \boldsymbol{X}$, we minimize $(\boldsymbol{H}_t)_{ij}(d_t(x_i, x_j) - \|\phi(x_i) - \phi(x_j)\|_2)^2$. This promotes geodesic preservation of local neighbors, since more weights are given to points with higher affinities.

Diffusion maps with the heat kernel Diffusion maps (Coifman & Lafon, 2006) define an embedding with the first k eigenvectors $(\phi_i)_i$ of $\boldsymbol{P}$, while Laplacian eigenmaps uses the eigenvectors $(\psi_i)_i$ of $\boldsymbol{L}$. In the following, we recall the links between the two methods, and show that a rescaled Laplacian eigenmaps preserves the diffusion distance with the heat kernel $\boldsymbol{H}_t$.

Lemma C.1. *Rescaling the Laplacian eigenmaps embedding with* $x_i \mapsto (e^{-2t\lambda_1}\psi_{1,i}, \ldots, e^{-2t\lambda_k}\psi_{k,i})$ *preserves the*

diffusion distance $DM_{\boldsymbol{H}_t}$.

Relation to SNE The heat-geodesic method also relates to the Stochastic Neighbor Embedding (SNE) (Hinton & Roweis, 2002), and its variation using the Student distribution t-SNE (Linderman & Steinerberger, 2017). In SNE, the similarity between points is encoded via transition probabilities p_{ij}.

The objective is to learn an affinity measure q, that usually depends on the embedding distances $\|y_i - y_j\|$, such that it minimizes $D_{\mathrm{KL}}(p||q)$. Intuitively, points that have a strong affinity in the ambient space, should also have a strong affinity in the embedded space. Even though the heat-geodesic minimization is directly on the embedding distances, we can show an equivalent with SNE. In Appendix A, we provide additional comparisons between SNE and our method.

Proposition C.2. *The Heat-Geodesic embedding with squared distances minimization weighted by the heat kernel is equivalent to SNE with the heat kernel affinity in the ambient space, and a Gaussian kernel in the embedded space* $q_{ij} = \exp(-\|y_i - y_j\|^2/t)$.

D. Experiments and datasets details

Our experiments compare our approach with multiple state-of-the-art baselines for synthetic datasets (for which the true geodesic distance is known) and real-world datasets. For all models, we perform sample splitting with a 50/50 validation-test split. The validation and test sets each consist of 5 repetitions with different random initializations. The hyper-parameters are selected according to the performance on the validation set. We always report the results on the test set, along with the standard deviations computed over the five repetitions. We use the following state-of-the-art methods in our experiments: our Heat Geodesic Embedding, *diffusion maps*(Coifman & Lafon, 2006), *PHATE* (Moon et al., 2019), *Heat-PHATE* (a variation of PHATE using the Heat Kernel), *Rand-Geo* (a variation of Heat Geodesic Embedding where we use the random walk kernel), *Shortest-path* which estimates the geodesic distance by computing the shortest path between two nodes in a graph built on the point clouds, *t-SNE*(Van der Maaten & Hinton, 2008), and *UMAP*(McInnes et al., 2018).

D.1. Datasets

We consider two synthetic datasets, the well known Swiss roll and the tree datasets. The exact geodesic distance can be computed for these datasets. We additionally consider real-world datasets: PBMC, IPSC (Moon et al., 2019), EB (Moon et al., 2019), and two from the from the 2022 NeurIPS multimodal single-cell integration challenge[2].

D.1.1. Swiss Roll

The Swiss roll dataset consists of data points samples on a smooth manifold inspired by shape of the famous alpine pastry. In its simplest form, it is a 2-dimensional surface embedded in $\mathbb{R}^3$ given by

$$
\begin{aligned}
x &= t \cdot cos(t) \\
y &= h \\
z &= t \cdot sin(t)
\end{aligned}
$$

where $t \in [T_0, T_1]$ and $h \in [0, W]$. In our experiments we used $T_0 = \frac{3}{2}\pi$, $T_1 = \frac{9}{2}\pi$, and $W = 5$. We use two sampling mechanisms for generating the data points : uniformly and clustered. In the first, we sample points uniformly at random in the $[T_0, T_1] \times [0, W]$ plane. In the second, we sample according to a mixture of isotropic multivariate Gaussian distributions in the same plane with equal weights, means $[(7, W/2), (12, W/2)]$, and standard deviations $[1, 1]$. In the clustered case, data samples are given a label y according to the Gaussian mixture component from which they were sampled.

We consider variations of the Swiss roll by projecting the data samples in higher dimension using a random rotation matrix sampled from the Haar distribution. We use three different ambient dimensions: 3, 10, and 50.

Finally, we add isotropic Gaussian noise to the data points in the ambient space with a standard deviation σ.

[2] https://www.kaggle.com/competitions/open-problems-multimodal/

D.1.2. TREE

The tree dataset is created by generating K branches from a D-dimensional Brownian motion that are eventually glued together. Each branch is sampled from a multidimensional Brownian motion $d\mathbf{X_k} = 2d\mathbf{W}(t)$ at times $t = 0, 1, 2, ..., L-1$ for $k \in [K]$. The first branch is taken as the main branch and the remaining branches are glued to the main branch by setting $X_k = X_k + X_0[i_k]$ where i_k is a random index of the main branch vector. The total number of samples is thus $L \cdot K$

In our experiments, we used $L = 500$, $K = 5$, and $D = 5, 10$ (*i.e.,* two versions with different dimensions of the ambient space).

D.2. Evaluation Metrics

We compare the performance of the different methods according to several metrics. For synthetic datasets, where ground truth geodesic distance is available, we directly compare the estimated distance matrices and ground truth geodesic distance matrices. For real-world datasets, we use clustering quality and continuous interpolation as evaluation metrics.

D.2.1. DISTANCE MATRIX EVALUATION

The following methods use an explicit distance matrix: diffusion maps, Heat Geodesic Embedding, Heat-Phate, Phate, Rand-Geo and Shortest Path. For these methods, we compare their ability their ability to recover the ground truth distance matrix several metrics. Letting D and $\hat{D}$ the ground truth and inferred distance matrices respectively, and N the number of points in the dataset, we use the following metrics.

Pearson ρ We compute the average Pearson correlation between the rows of the distance matrices, $\frac{1}{N}\sum_{i=1}^{N} r_{D_i,\hat{D}_i}$, where $r_{x,y}$ is the Pearson correlation coefficient between vectors x and y. D_i stands for the i-th row of D.

Spearman ρ We compute the average Spearman correlation between the rows of the distance matrices, $\frac{1}{N}\sum_{i=1}^{N} r_{D_i,\hat{D}_i}$, where $r_{x,y}$ is the Spearman correlation coefficient between vectors x and y. D_i stands for the i-th row of D.

Frobenius Norm We use $\|D - \hat{D}\|_F$, where $\|A\|_F = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N}|A_{i,j}|^2}$

Maximum Norm We use $\|D - \hat{D}\|_\infty$, where $\|A\|_\infty = max_{i,j}|A_{i,j}|$

D.2.2. EMBEDDING EVALUATION

Some methods produce low-dimensional embeddings without using an explicit distance matrix for the data points. This is the case for UMAP and t-SNE. To compare against these methods, we use the distance matrix obtained by considering Euclidean distance between the low-dimensional embeddings. We used 2-dimensional embeddings in our experiments. For diffusion maps, we obtain these embeddings by using the first two eigenvectors of the diffusion operator only. For Heat Geodesic Embedding, Heat-PHATE, PHATE, Rand-GEO and Shortest Path, we use multidimensional scaling (MDS) on the originally inferred distance matrix.

Clustering We evaluate the ability of Heat Geodesic Embedding to create meaningful embeddings when clusters are present in the data. To this end, we run a k-means clustering on the two dimensional embeddings obtained with each method and compare them against the ground truth labels. For the Tree dataset, we use the branches as clusters. For the Swiss roll dataset, we sample data points on the manifold according to a mixture of Gaussians and use the mixture component as the ground truth cluster label.

Interpolation To quantitatively evaluate the quality of the continuous embeddings, we first embed the entire dataset and obfuscate all samples from a particular time point (*e.g.,* $t = 2$). We then estimate the distribution of the missing time point by using displacement interpolation (Villani & Villani, 2009) between the adjacent time points (*e.g.,* $t = 1$ and $t = 3$). We report the Earth Mover Distance (EMD) between the predicted distribution and true distribution. A low EMD suggests that the obfuscated embeddings are naturally located between the previous and later time points, and that the generated embedding captures the temporal evolution of the data adequately.

D.3. Hyperparameters

In Table 4, we report the values of hyperparameters used to compute the different embeddings.

Hyperparameter	Description	Values
	Heat Geodesic Embedding	
k	Number of neighbours in k-NN graph	5,10,15
order	order of the approximation	30
t	Diffusion time	0.1,1,10,50,auto
Approximation method	Approximation method for Heat Kernel	Euler, Chebyshev
Laplacian	Type of laplacian	Combinatorial
Harnack ρ	Harnack Regularization	0,0.25,0.5,0.75,1,1.5
	PHATE	
n-PCA	Number of PCA components	50,100
t	Diffusion time	1,5,10,20,auto
k	Number of neighbours	10
	Diffusion Maps	
k	Number of neighbours in k-NN graph	5,10,15
t	Diffusion time	1,5,10,20
	Shortest Path	
k	Number of neighbours in k-NN graph	5,10,15
	UMAP	
k	Number of neighbours	5,10,15
min-dist	Minimum distance	0.1,0.5,0.99
	t-SNE	
p	Perplexity	10,30,100
early exageration	Early exageration parameter	12

Table 4. Hyperparameters used in our experiments

D.4. Hardware

The experiments were performed on a compute node with 16 Intel Xeon Platinum 8358 Processors and 64GB RAM.

E. Additional results

E.1. Preservation of the inherent data structure

A crucial evaluation criteria of manifold learning methods is the ability to capture the inherent structure of the data. For instance, clusters in the data should be visible in the resulting low dimensional representation. Similarly, when the dataset consists of samples taken at different time points, one expects to be able to characterize this temporal evolution in the low dimensional embedding (Moon et al., 2019). We thus compare the different embedding methods according to their ability to retain clusters and temporal evolution of the data.

Identifying clusters. We use the PBMC dataset, the Swiss roll, and the Tree dataset. The PBMC dataset consists of single-cell gene expressions from 3000 individual peripheral blood mononuclear cells. Cells are naturally clustered by their cell type. For the Tree dataset, we use the branches as clusters. For the Swiss roll dataset, we sample data points on the manifold according to a mixture of Gaussians and use the mixture component as the ground truth cluster labels. For each method, we run k-means on the two-dimensional embedding and compare the resulting cluster assignments with ground truth. Tab. 10 reports the results in terms of homogeneity and adjusted mutual information (aMI). Heat Geodesic Embedding

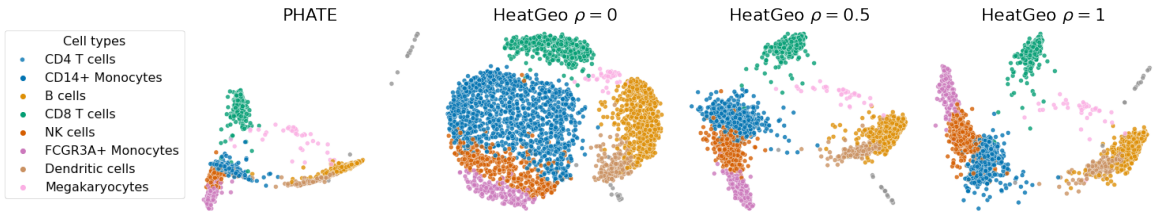

Figure 3. Embeddings on PBMC using the triplet distance with the heat-geodesic for different regularization parameter ρ.

is competitive with PHATE and outperforms t-SNE and UMAP on all metrics. Yet, we show in Appendix E that all methods tend to perform equally well when the noise level increases. In Fig. 3, we present the PBMC embeddings of PHATE and HeatGeo, showing that HeatGeo interpolates to PHATE for $\rho \to 1$.

Table 5. Clustering quality metrics for different methods. We report the homogeneity and the adjusted mutual information (aMI). Best models on average are bolded (higher is better).

	Swiss roll		Tree		PBMC	
Method	Homogeneity	aMI	Homogeneity	aMI	Homogeneity	aMI
UMAP	0.810 ± 0.036	0.726 ± 0.045	0.678 ± 0.086	0.681 ± 0.086	0.177 ± 0.037	0.148 ± 0.035
t-SNE	0.748 ± 0.067	0.668 ± 0.068	0.706 ± 0.054	0.712 ± 0.055	0.605 ± 0.019	0.544 ± 0.022
PHATE	0.731 ± 0.035	0.652 ± 0.046	0.550 ± 0.042	0.555 ± 0.042	$\mathbf{0.798 \pm 0.012}$	$\mathbf{0.785 \pm 0.01}$
Diffusion Maps	0.643 ± 0.053	0.585 ± 0.051	0.341 ± 0.103	0.358 ± 0.093	0.026 ± 0.001	0.038 ± 0.001
HeatGeo (ours)	$\mathbf{0.820 \pm 0.008}$	$\mathbf{0.740 \pm 0.018}$	$\mathbf{0.784 \pm 0.051}$	$\mathbf{0.786 \pm 0.051}$	0.734 ± 0.009	0.768 ± 0.017

E.2. HeatGeo weighted

We know that weighting the MDS loss by the heat kernel corresponds to a specific parametrization of SNE, and thus promote the identification of cluster. In Fig. 4, we show the embeddings of four Gaussian distributions in 10 dimensions (top), and the PBMC dataset (bottom). The reference embedding is using t-SNE, as it models as it also minimizes the KL between the ambient and embedded distributions. We see that HeatGeo weighted form cluster that are shaped like a Gaussian. This is expected as Prop. C.2, indicates that this is equivalent to minimizing the D_{KL} between the heat kernel and a Gaussian affinity kernel.

E.3. Truncated distance

In Fig.5, we discretize the interval $[0, 51]$ in 51 nodes, and we compute the heat-geodesic distance of the midpoint with respect to the other points, effectively approximating the Euclidean distance. Using Chebyshev polynomials of degree of 20, we see that the impact of the truncation is greater as the diffusion time increases. The backward Euler methods does not result in a truncated distance.

E.4. Harnack inequality

For complete Riemannian manifolds that satisfy the parabolic Harnack inequality (PHI) we have $h_t(x, y) \simeq V^{-1}(x, \sqrt{t})\, e^{-d(x,y)^2/t}$ so that $-t \log h_t(x, y) \simeq t \log V(x, \sqrt{t}) + d^2(x, y)$ (Saloff-Coste, 2010).

$$h_t(x, x) = \frac{1}{V(x, \sqrt{t})} \tag{7}$$

$$V(x, \sqrt{t}) = h_t(x, x)^{-1} \tag{8}$$

We then have,

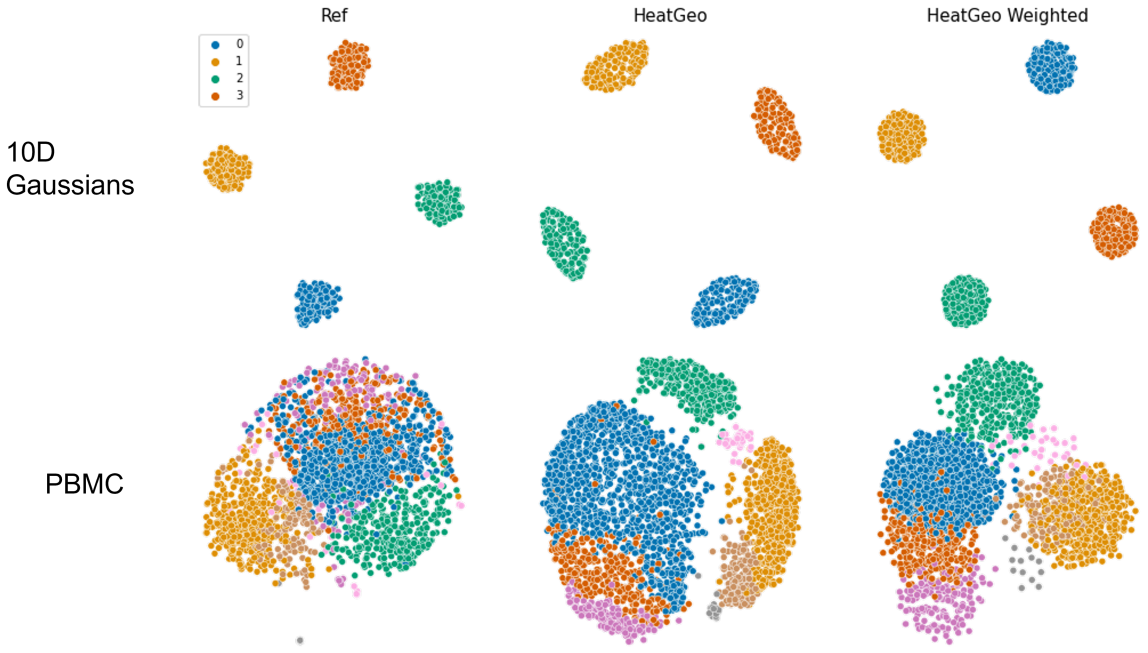

Figure 4. Embeddings of four Gaussian distributions in 10 dimensions (top), and the PBMC dataset (bottom). HeatGeo with weight is equivalent to minimizing the D_{KL} between the heat kernel and a Gaussian affinity kernel, hence produces clusters shaped similar to a Gaussian.

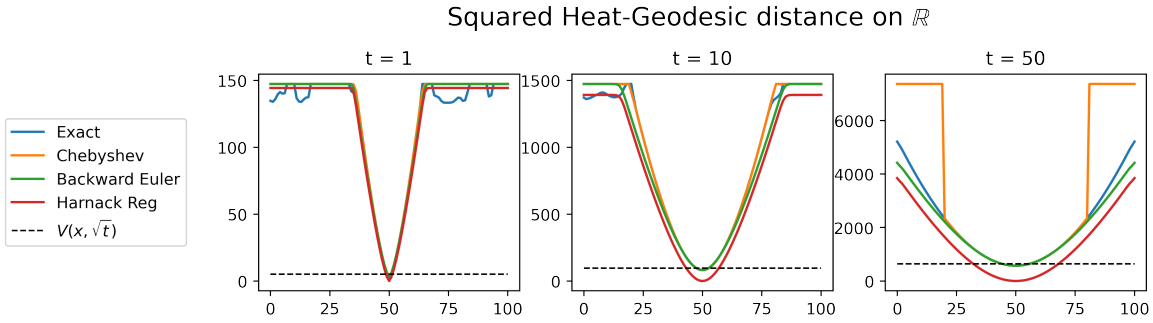

Figure 5. Approximation of the squared Euclidean distance with the Heat-geodesic for the exact computation, Backward Euler approximation, and Chebyshev polynomials. For larger diffusion time, the Chebyshev approximation results in a thresholded distance. The Harnack regularization unsures $d_t(x, x) = 0$.

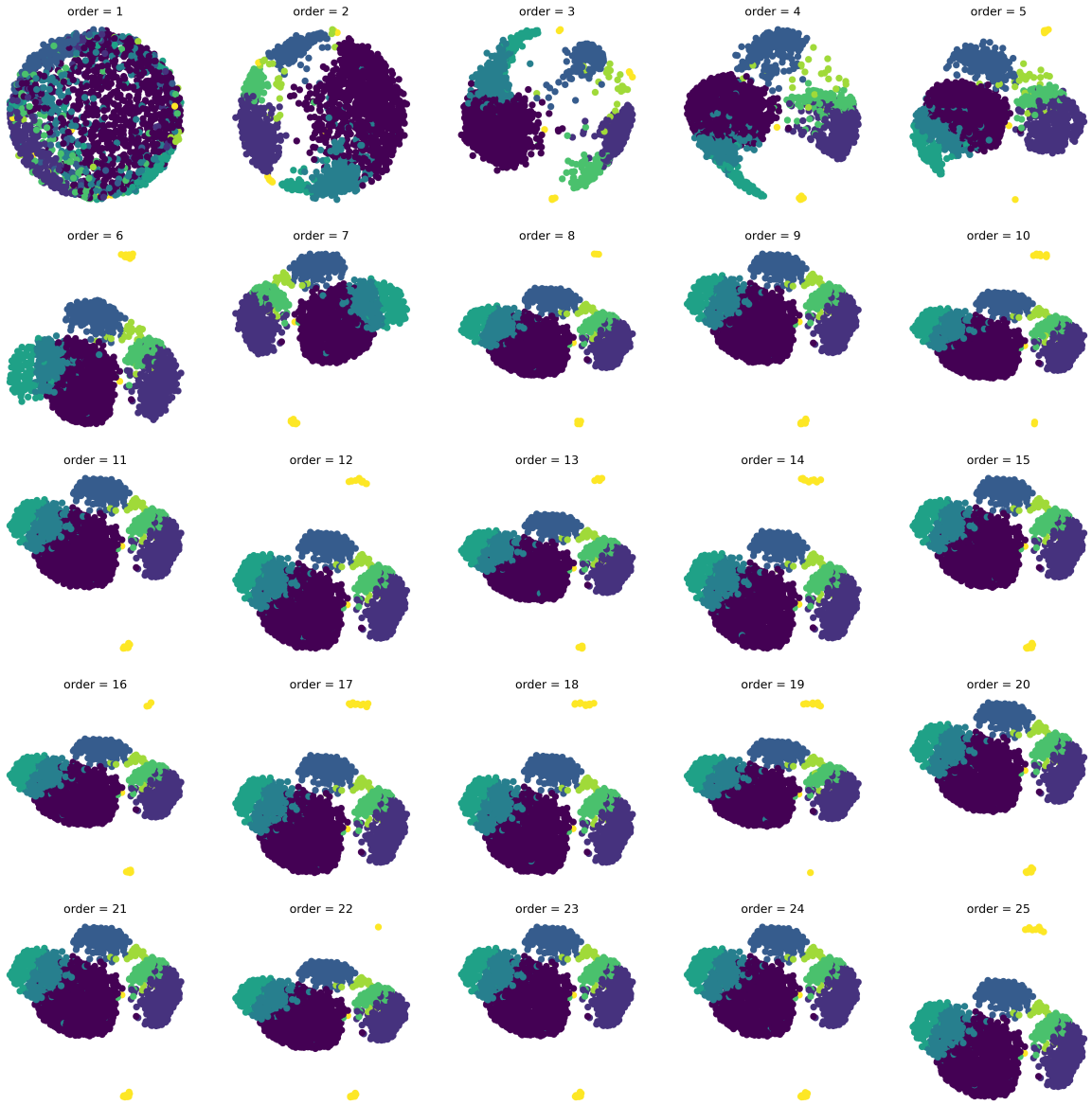

Figure 6. Impact of the Checbyshev approximation order on the embedding of HeatGeo for the PBMC dataset.

$$d^2(x,y) \simeq -t\log h_t(x,y) - t\log V(x,\sqrt{t})$$
$$d^2(x,y) \simeq -t\log h_t(x,y) - t\log h_t(x,x)^{-1}$$
$$d^2(x,y) \simeq -t\log h_t(x,y) + t\log h_t(x,x)$$

E.4.1. CASE STUDIES FOR SPECIFIC MANIFOLDS

The circle - $\mathbb{S}_1$ We now show that our expression for the Heat Geodesic Embedding-distance is monotonically increasing with respect to the ground truth geodesic distance $d \in \mathbb{R}^+$ for a fixed diffusion time t and for any Harnack regularization in $\mathbb{S}_1$. Therefore, the

Our expression for the Heat Geodesic Embedding-distance is

$$\hat{d} = \sqrt{-4t\log(h_t(d)) + 4t\log(h_t(0))}$$

As the square-root is monotonic, and $4t\log h_t(0)$ is constant with respect to d, we need to show that $f(d) = -log(h_t(d))$ is monotonically increasing.

For $\mathbb{S}_1$, we have

$$h_t(d) = \sum_{m\in\mathbb{Z}} \frac{1}{\sqrt{4\pi t}} e^{-\frac{(d+2\pi m)^2}{4t}}$$

As log is monotonically increasing, it suffices to show that $\sum_{m\in\mathbb{Z}} e^{-\frac{(d+2\pi m)^2}{4t}}$ is monotonically *decreasing*, which is the case as for any $d' > d$, $\forall m \in \mathbb{Z}$, we have

$$e^{-\frac{(d+2\pi m)^2}{4t}} > e^{-\frac{(d'+2\pi m)^2}{4t}}.$$

In general, one can see that (1) the heat kernel depending only on the geodesic distance and (2) the heat kernel being monotonically decreasing with respect to the geodesic distance are sufficient conditions for preserving ordering of pair-wise distances with Heat Geodesic Embedding.

The sphere - $\mathbb{S}_n$ The above result can be applied to the higher-dimensional sphere $\mathbb{S}_n$. It is known that the heat kernel on manifold of constant curvatures is a function of the the geodesic distance (d) and time only. For $\mathbb{S}_n$ the heat kernel is given by

$$h_t(x,y) = \sum_{l=0}^{\infty} e^{-l(l+n)-2t} \frac{2l+n-2}{n-2} C_l^{\frac{n}{2}-1}(cos(d))$$

with I the regularized incomplete beta function and C the Gegenbauer polynomials.

Furthermore, the heat kernel of the sphere is monotonically decreasing. The distance inferred from Heat Geodesic Embedding thus preserves ordering of the pair-wise distances.

Euclidean ($\mathbb{R}^3$) For the euclidean space, we have for the volume of $\sqrt{t}$-geodesic ball and for the heat kernel:

$$V_{\sqrt{t}} = \frac{4}{3}\pi t^{3/2}$$
$$h_t(x,y) = \frac{1}{(4\pi t)^{3/2}} e^{-\frac{\rho^2}{4t}}.$$

Recalling Harnack inequality,

$$\frac{c_1}{V(x,\sqrt{t})} e^{-\frac{d(x,y)^2}{c_2 t}} \leq h_t(x,y) \leq \frac{c_3}{V(x,\sqrt{t})} e^{-\frac{d(x,y)^2}{c_4 t}}$$

With $c_2 = c_4 = 4$, we have

$$\frac{c_1}{V(x,\sqrt{t})} \leq \frac{1}{(4\pi t)^{3/2}} \leq \frac{c_3}{V(x,\sqrt{t})}$$

In this case, the bound can be made tight, by setting

$$\begin{aligned} c_1 = c_3 &= \frac{V(x,\sqrt{t})}{(4\pi t)^{3/2}} \\ &= \frac{\frac{4}{3}\pi t^{3/2}}{(4\pi t)^{3/2}} \\ &= \frac{1}{3\sqrt{4\pi}} = \frac{1}{6\sqrt{\pi}}, \end{aligned}$$

we recover the exact geodesic distance.

E.5. Quantitative results

E.5.1. Distance matrix evaluation

We report the performance of the different methods in terms of the ground truth geodesic matrix reconstruction in Table. 6 for the Swiss roll dataset and in Table. 7, for the Tree dataset.

E.5.2. Distance matrix evaluation via two-dimensional embeddings

We report the performance of the different methods in terms of the ground truth geodesic matrix reconstruction in Table 8 for the Swiss roll dataset and in Table 9, for the Tree dataset.

E.5.3. Clustering quality evaluation

On Tables 10, we report the performance on clustering quality for the synthetic datasets with different noise level.

E.6. Impact of the different hyperparameters

We investigate the impact of the different hyperparameters on the quality of the embeddings. In Figure 7, we show the embeddings of HeatGeo for different values of diffusion time, number of neighbours, order, and Harnack regularization.

In Figures 8, 9, 10, and 11, we show the impact of different hyperparameters on the Pearson correlation between the estimated distance matrix and ground truth distance matrix for different methods on the Swiss roll dataset.

data	Noise level	Method	PearsonR	SpearmanR	Norm Fro N2	Norm inf N2
Swiss roll	0.1	Diffusion Map	0.974 ± 0.01	0.983 ± 0.007	0.018 ± 0.0	0.026 ± 0.0
Swiss roll	0.1	Heat-Geo	0.992 ± 0.003	0.995 ± 0.002	0.002 ± 0.0	0.003 ± 0.0
Swiss roll	0.1	Heat-PHATE	0.99 ± 0.002	0.997 ± 0.001	0.079 ± 0.002	0.1 ± 0.003
Swiss roll	0.1	PHATE	0.621 ± 0.006	0.58 ± 0.01	0.022 ± 0.0	0.026 ± 0.0
Swiss roll	0.1	Rand-Geo	0.956 ± 0.003	0.993 ± 0.001	0.009 ± 0.0	0.012 ± 0.0
Swiss roll	0.1	Shortest Path	$\mathbf{1.0 \pm 0.0}$	$\mathbf{1.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.001 \pm 0.0}$
Swiss roll	0.5	Diffusion Map	0.982 ± 0.003	0.987 ± 0.002	0.018 ± 0.0	0.026 ± 0.0
Swiss roll	0.5	Heat-Geo	0.994 ± 0.002	0.996 ± 0.001	0.002 ± 0.0	0.004 ± 0.0
Swiss roll	0.5	Heat-PHATE	0.993 ± 0.001	0.998 ± 0.0	0.064 ± 0.001	0.083 ± 0.002
Swiss roll	0.5	PHATE	0.649 ± 0.007	0.615 ± 0.006	0.023 ± 0.0	0.028 ± 0.0
Swiss roll	0.5	Rand-Geo	0.969 ± 0.002	0.995 ± 0.001	0.009 ± 0.0	0.011 ± 0.0
Swiss roll	0.5	Shortest Path	$\mathbf{0.999 \pm 0.0}$	$\mathbf{0.999 \pm 0.0}$	$\mathbf{0.001 \pm 0.0}$	$\mathbf{0.002 \pm 0.0}$
Swiss roll	1.0	Diffusion Map	0.476 ± 0.226	0.478 ± 0.138	0.018 ± 0.0	0.026 ± 0.0
Swiss roll	1.0	Heat-Geo	$\mathbf{0.702 \pm 0.086}$	$\mathbf{0.7 \pm 0.073}$	$\mathbf{0.01 \pm 0.0}$	$\mathbf{0.012 \pm 0.0}$
Swiss roll	1.0	Heat-PHATE	0.623 ± 0.144	0.633 ± 0.114	$\mathbf{0.01 \pm 0.002}$	0.019 ± 0.004
Swiss roll	1.0	PHATE	0.457 ± 0.01	0.404 ± 0.024	0.024 ± 0.0	0.028 ± 0.0
Swiss roll	1.0	Rand-Geo	0.521 ± 0.042	0.608 ± 0.025	$\mathbf{0.01 \pm 0.0}$	0.014 ± 0.0
Swiss roll	1.0	Shortest Path	0.497 ± 0.144	0.558 ± 0.134	0.011 ± 0.001	0.015 ± 0.002
Swiss roll high	0.1	Diffusion Map	0.98 ± 0.003	0.986 ± 0.001	0.018 ± 0.0	0.026 ± 0.0
Swiss roll high	0.1	Heat-Geo	0.992 ± 0.003	0.996 ± 0.002	0.002 ± 0.0	0.003 ± 0.0
Swiss roll high	0.1	Heat-PHATE	0.991 ± 0.002	0.997 ± 0.001	0.079 ± 0.002	0.101 ± 0.004
Swiss roll high	0.1	PHATE	0.625 ± 0.013	0.582 ± 0.017	0.022 ± 0.0	0.026 ± 0.0
Swiss roll high	0.1	Rand-Geo	0.956 ± 0.002	0.993 ± 0.001	0.009 ± 0.0	0.012 ± 0.0
Swiss roll high	0.1	Shortest Path	$\mathbf{1.0 \pm 0.0}$	$\mathbf{1.0 \pm 0.0}$	$\mathbf{0.001 \pm 0.0}$	$\mathbf{0.002 \pm 0.0}$
Swiss roll high	0.5	Diffusion Map	0.98 ± 0.002	0.985 ± 0.002	0.018 ± 0.0	0.026 ± 0.0
Swiss roll high	0.5	Heat-Geo	0.997 ± 0.001	0.997 ± 0.0	$\mathbf{0.005 \pm 0.0}$	$\mathbf{0.007 \pm 0.0}$
Swiss roll high	0.5	Heat-PHATE	0.995 ± 0.0	0.997 ± 0.0	0.041 ± 0.001	0.054 ± 0.002
Swiss roll high	0.5	PHATE	0.717 ± 0.004	0.707 ± 0.005	0.026 ± 0.0	0.034 ± 0.001
Swiss roll high	0.5	Rand-Geo	0.984 ± 0.0	0.996 ± 0.0	0.008 ± 0.0	0.01 ± 0.0
Swiss roll high	0.5	Shortest Path	$\mathbf{0.999 \pm 0.0}$	$\mathbf{0.998 \pm 0.0}$	0.006 ± 0.0	0.009 ± 0.0
Swiss roll high	1.0	Diffusion Map	0.555 ± 0.155	0.526 ± 0.081	0.018 ± 0.0	0.026 ± 0.0
Swiss roll high	1.0	Heat-Geo	$\mathbf{0.705 \pm 0.065}$	$\mathbf{0.695 \pm 0.052}$	0.011 ± 0.0	$\mathbf{0.012 \pm 0.0}$
Swiss roll high	1.0	Heat-PHATE	0.63 ± 0.106	0.625 ± 0.074	0.011 ± 0.001	0.014 ± 0.002
Swiss roll high	1.0	PHATE	0.473 ± 0.026	0.419 ± 0.024	0.027 ± 0.0	0.039 ± 0.001
Swiss roll high	1.0	Rand-Geo	0.563 ± 0.05	0.644 ± 0.033	$\mathbf{0.01 \pm 0.0}$	$\mathbf{0.012 \pm 0.0}$
Swiss roll high	1.0	Shortest Path	0.384 ± 0.02	0.461 ± 0.017	0.011 ± 0.0	0.015 ± 0.0
Swiss roll very high	0.1	Diffusion Map	0.977 ± 0.005	0.984 ± 0.004	0.018 ± 0.0	0.026 ± 0.0
Swiss roll very high	0.1	Heat-Geo	0.992 ± 0.002	0.996 ± 0.001	$\mathbf{0.002 \pm 0.0}$	$\mathbf{0.003 \pm 0.0}$
Swiss roll very high	0.1	Heat-PHATE	0.991 ± 0.001	0.997 ± 0.001	0.079 ± 0.003	0.101 ± 0.003
Swiss roll very high	0.1	PHATE	0.631 ± 0.01	0.594 ± 0.011	0.023 ± 0.0	0.028 ± 0.001
Swiss roll very high	0.1	Rand-Geo	0.957 ± 0.002	0.994 ± 0.001	0.009 ± 0.0	0.012 ± 0.0
Swiss roll very high	0.1	Shortest Path	$\mathbf{0.999 \pm 0.0}$	$\mathbf{0.999 \pm 0.0}$	0.006 ± 0.0	0.007 ± 0.0
Swiss roll very high	0.5	Diffusion Map	0.978 ± 0.002	0.984 ± 0.001	0.018 ± 0.0	0.026 ± 0.0
Swiss roll very high	0.5	Heat-Geo	0.997 ± 0.0	$\mathbf{0.998 \pm 0.0}$	$\mathbf{0.008 \pm 0.0}$	0.01 ± 0.0
Swiss roll very high	0.5	Heat-PHATE	0.996 ± 0.001	0.997 ± 0.0	0.016 ± 0.0	0.02 ± 0.001
Swiss roll very high	0.5	PHATE	0.815 ± 0.002	0.823 ± 0.004	0.032 ± 0.0	0.049 ± 0.002
Swiss roll very high	0.5	Rand-Geo	0.986 ± 0.0	0.996 ± 0.0	$\mathbf{0.008 \pm 0.0}$	$\mathbf{0.009 \pm 0.0}$
Swiss roll very high	0.5	Shortest Path	$\mathbf{0.998 \pm 0.0}$	$\mathbf{0.998 \pm 0.0}$	0.019 ± 0.001	0.027 ± 0.001
Swiss roll very high	1.0	Diffusion Map	0.324 ± 0.061	0.399 ± 0.033	0.018 ± 0.0	0.026 ± 0.0
Swiss roll very high	1.0	Heat-Geo	$\mathbf{0.466 \pm 0.007}$	0.506 ± 0.006	0.011 ± 0.0	0.013 ± 0.0
Swiss roll very high	1.0	Heat-PHATE	0.369 ± 0.011	0.43 ± 0.019	0.011 ± 0.0	0.014 ± 0.0
Swiss roll very high	1.0	PHATE	0.377 ± 0.011	0.425 ± 0.009	0.036 ± 0.0	0.062 ± 0.004
Swiss roll very high	1.0	Rand-Geo	0.398 ± 0.009	$\mathbf{0.516 \pm 0.008}$	$\mathbf{0.01 \pm 0.0}$	$\mathbf{0.012 \pm 0.0}$
Swiss roll very high	1.0	Shortest Path	0.367 ± 0.018	0.443 ± 0.016	0.012 ± 0.0	0.015 ± 0.0

Table 6. Comparison of the estimated distance matrices with the ground truth geodesic distance matrices on the Swiss roll dataset. Best models on average are bolded (not necessarily significant).

data	Noise level	Method	PearsonR	SpearmanR	Norm Fro N2	Norm inf N2
Tree	1.0	Diffusion Map	0.748 ± 0.125	0.733 ± 0.111	0.113 ± 0.012	0.161 ± 0.019
Tree	1.0	Heat-Geo	$\mathbf{0.976 \pm 0.019}$	$\mathbf{0.977 \pm 0.02}$	0.092 ± 0.011	0.135 ± 0.018
Tree	1.0	Heat-PHATE	0.918 ± 0.032	0.885 ± 0.04	$\mathbf{0.03 \pm 0.005}$	$\mathbf{0.044 \pm 0.007}$
Tree	1.0	PHATE	0.671 ± 0.021	0.398 ± 0.052	0.051 ± 0.008	0.084 ± 0.017
Tree	1.0	Rand-Geo	0.926 ± 0.011	0.966 ± 0.019	0.076 ± 0.01	0.117 ± 0.018
Tree	1.0	Shortest Path	0.965 ± 0.026	0.963 ± 0.027	0.039 ± 0.008	0.06 ± 0.008
Tree	5.0	Diffusion Map	0.656 ± 0.054	0.653 ± 0.057	0.113 ± 0.012	0.161 ± 0.019
Tree	5.0	Heat-Geo	$\mathbf{0.822 \pm 0.008}$	$\mathbf{0.807 \pm 0.016}$	0.1 ± 0.012	0.146 ± 0.019
Tree	5.0	Heat-PHATE	0.765 ± 0.025	0.751 ± 0.023	$\mathbf{0.043 \pm 0.006}$	$\mathbf{0.08 \pm 0.01}$
Tree	5.0	PHATE	0.766 ± 0.023	0.743 ± 0.028	0.055 ± 0.007	0.093 ± 0.008
Tree	5.0	Rand-Geo	0.806 ± 0.014	0.795 ± 0.018	0.094 ± 0.011	0.139 ± 0.018
Tree	5.0	Shortest Path	0.78 ± 0.009	0.757 ± 0.019	0.075 ± 0.009	0.117 ± 0.014
Tree	10.0	Diffusion Map	0.538 ± 0.05	0.471 ± 0.089	0.113 ± 0.012	0.161 ± 0.019
Tree	10.0	Heat-Geo	0.62 ± 0.025	$\mathbf{0.59 \pm 0.033}$	0.1 ± 0.012	0.146 ± 0.019
Tree	10.0	Heat-PHATE	$\mathbf{0.63 \pm 0.018}$	0.588 ± 0.031	$\mathbf{0.046 \pm 0.005}$	$\mathbf{0.083 \pm 0.012}$
Tree	10.0	PHATE	0.623 ± 0.016	0.583 ± 0.029	0.07 ± 0.01	0.112 ± 0.017
Tree	10.0	Rand-Geo	0.578 ± 0.043	0.558 ± 0.053	0.095 ± 0.011	0.14 ± 0.018
Tree	10.0	Shortest Path	0.539 ± 0.041	0.513 ± 0.055	0.072 ± 0.01	0.118 ± 0.017
Tree high	1.0	Diffusion Map	0.754 ± 0.049	0.741 ± 0.057	0.267 ± 0.021	0.369 ± 0.026
Tree high	1.0	Heat-Geo	0.996 ± 0.001	$\mathbf{0.999 \pm 0.001}$	0.242 ± 0.02	0.338 ± 0.026
Tree high	1.0	Heat-PHATE	0.927 ± 0.011	0.875 ± 0.032	0.062 ± 0.003	0.084 ± 0.006
Tree high	1.0	PHATE	0.528 ± 0.085	0.141 ± 0.061	0.209 ± 0.023	0.307 ± 0.027
Tree high	1.0	Rand-Geo	0.85 ± 0.014	0.944 ± 0.011	0.227 ± 0.02	0.323 ± 0.025
Tree high	1.0	Shortest Path	$\mathbf{0.998 \pm 0.001}$	$\mathbf{0.999 \pm 0.001}$	$\mathbf{0.009 \pm 0.002}$	$\mathbf{0.018 \pm 0.005}$
Tree high	5.0	Diffusion Map	0.706 ± 0.124	0.705 ± 0.113	0.267 ± 0.021	0.369 ± 0.026
Tree high	5.0	Heat-Geo	$\mathbf{0.97 \pm 0.01}$	$\mathbf{0.975 \pm 0.009}$	0.253 ± 0.021	0.353 ± 0.026
Tree high	5.0	Heat-PHATE	0.932 ± 0.022	0.919 ± 0.03	$\mathbf{0.072 \pm 0.004}$	$\mathbf{0.112 \pm 0.008}$
Tree high	5.0	PHATE	0.913 ± 0.014	0.872 ± 0.034	0.19 ± 0.017	0.278 ± 0.025
Tree high	5.0	Rand-Geo	0.968 ± 0.01	0.971 ± 0.009	0.245 ± 0.019	0.342 ± 0.024
Tree high	5.0	Shortest Path	0.952 ± 0.016	0.95 ± 0.019	0.137 ± 0.017	0.209 ± 0.024
Tree high	10.0	Diffusion Map	0.598 ± 0.117	0.613 ± 0.103	0.267 ± 0.021	0.369 ± 0.026
Tree high	10.0	Heat-Geo	$\mathbf{0.861 \pm 0.039}$	$\mathbf{0.87 \pm 0.038}$	0.254 ± 0.021	0.353 ± 0.026
Tree high	10.0	Heat-PHATE	0.844 ± 0.05	0.838 ± 0.051	0.168 ± 0.015	0.27 ± 0.025
Tree high	10.0	PHATE	0.837 ± 0.052	0.838 ± 0.049	0.204 ± 0.018	0.301 ± 0.024
Tree high	10.0	Rand-Geo	0.845 ± 0.041	0.86 ± 0.038	0.248 ± 0.02	0.346 ± 0.025
Tree high	10.0	Shortest Path	0.779 ± 0.051	0.777 ± 0.054	$\mathbf{0.159 \pm 0.018}$	$\mathbf{0.257 \pm 0.026}$

Table 7. Comparison of the estimated distance matrices with the ground truth geodesic distance matrices on the Tree roll dataset. Best models on average are bolded (not necessarily significant).

data	Noise level	Method	PearsonR	SpearmanR	Norm Fro N2	Norm inf N2
Swiss roll	0.1	Diffusion Map	0.974 ± 0.01	0.983 ± 0.007	0.018 ± 0.0	0.026 ± 0.0
Swiss roll	0.1	Heat-Geo	0.995 ± 0.003	0.996 ± 0.002	0.018 ± 0.0	0.026 ± 0.0
Swiss roll	0.1	Heat-PHATE	0.99 ± 0.002	0.997 ± 0.001	0.018 ± 0.0	0.026 ± 0.0
Swiss roll	0.1	PHATE	0.677 ± 0.02	0.697 ± 0.014	0.018 ± 0.0	0.026 ± 0.0
Swiss roll	0.1	Rand-Geo	0.917 ± 0.003	0.915 ± 0.002	0.018 ± 0.0	0.026 ± 0.0
Swiss roll	0.1	Shortest Path	$\mathbf{1.0 \pm 0.0}$	$\mathbf{1.0 \pm 0.0}$	0.018 ± 0.0	0.026 ± 0.0
Swiss roll	0.1	TSNE	0.905 ± 0.005	0.897 ± 0.004	$\mathbf{0.006 \pm 0.0}$	$\mathbf{0.008 \pm 0.0}$
Swiss roll	0.1	UMAP	0.802 ± 0.013	0.79 ± 0.012	0.011 ± 0.0	0.016 ± 0.001
Swiss roll	0.5	Diffusion Map	0.982 ± 0.003	0.987 ± 0.002	0.018 ± 0.0	0.026 ± 0.0
Swiss roll	0.5	Heat-Geo	0.997 ± 0.0	0.996 ± 0.001	0.018 ± 0.0	0.026 ± 0.0
Swiss roll	0.5	Heat-PHATE	0.993 ± 0.001	0.997 ± 0.0	0.018 ± 0.0	0.026 ± 0.0
Swiss roll	0.5	PHATE	0.696 ± 0.011	0.711 ± 0.008	0.018 ± 0.0	0.026 ± 0.0
Swiss roll	0.5	Rand-Geo	0.932 ± 0.002	0.932 ± 0.002	0.018 ± 0.0	0.026 ± 0.0
Swiss roll	0.5	Shortest Path	$\mathbf{0.999 \pm 0.0}$	$\mathbf{0.999 \pm 0.0}$	0.018 ± 0.0	0.026 ± 0.0
Swiss roll	0.5	TSNE	0.899 ± 0.01	0.892 ± 0.008	$\mathbf{0.006 \pm 0.0}$	$\mathbf{0.008 \pm 0.0}$
Swiss roll	0.5	UMAP	0.838 ± 0.019	0.819 ± 0.017	0.012 ± 0.0	0.016 ± 0.001
Swiss roll	1.0	Diffusion Map	0.476 ± 0.226	0.478 ± 0.138	0.018 ± 0.0	0.026 ± 0.0
Swiss roll	1.0	Heat-Geo	0.672 ± 0.221	0.676 ± 0.193	0.018 ± 0.0	0.026 ± 0.0
Swiss roll	1.0	Heat-PHATE	0.674 ± 0.169	0.684 ± 0.134	0.018 ± 0.0	0.026 ± 0.0
Swiss roll	1.0	PHATE	0.287 ± 0.03	0.349 ± 0.028	0.018 ± 0.0	0.026 ± 0.0
Swiss roll	1.0	Rand-Geo	0.39 ± 0.029	0.43 ± 0.022	0.018 ± 0.0	0.026 ± 0.0
Swiss roll	1.0	Shortest Path	0.467 ± 0.17	0.511 ± 0.163	0.018 ± 0.0	0.026 ± 0.0
Swiss roll	1.0	TSNE	0.721 ± 0.183	$\mathbf{0.724 \pm 0.151}$	$\mathbf{0.008 \pm 0.002}$	$\mathbf{0.014 \pm 0.003}$
Swiss roll	1.0	UMAP	$\mathbf{0.727 \pm 0.181}$	0.713 ± 0.167	0.012 ± 0.001	0.018 ± 0.001
Swiss roll	5.0	Diffusion Map	0.157 ± 0.021	0.173 ± 0.015	0.018 ± 0.0	0.026 ± 0.0
Swiss roll	5.0	Heat-PHATE	0.203 ± 0.014	$\mathbf{0.239 \pm 0.013}$	0.018 ± 0.0	0.026 ± 0.0
Swiss roll	5.0	PHATE	0.201 ± 0.014	0.237 ± 0.013	0.018 ± 0.0	0.026 ± 0.0
Swiss roll	5.0	Rand-Geo	0.201 ± 0.014	0.238 ± 0.012	0.018 ± 0.0	0.026 ± 0.0
Swiss roll	5.0	Shortest Path	0.2 ± 0.011	0.233 ± 0.01	0.018 ± 0.0	0.026 ± 0.0
Swiss roll	5.0	TSNE	0.2 ± 0.011	0.233 ± 0.01	$\mathbf{0.012 \pm 0.0}$	$\mathbf{0.018 \pm 0.001}$
Swiss roll	5.0	UMAP	$\mathbf{0.205 \pm 0.013}$	$\mathbf{0.239 \pm 0.012}$	0.015 ± 0.0	0.022 ± 0.0
Swiss roll high	0.1	Diffusion Map	0.98 ± 0.003	0.986 ± 0.001	0.018 ± 0.0	0.026 ± 0.0
Swiss roll high	0.1	Heat-Geo	0.996 ± 0.002	0.997 ± 0.001	0.018 ± 0.0	0.026 ± 0.0
Swiss roll high	0.1	Heat-PHATE	0.991 ± 0.002	0.997 ± 0.001	0.018 ± 0.0	0.026 ± 0.0
Swiss roll high	0.1	PHATE	0.678 ± 0.027	0.698 ± 0.019	0.018 ± 0.0	0.026 ± 0.0
Swiss roll high	0.1	Rand-Geo	0.917 ± 0.003	0.915 ± 0.002	0.018 ± 0.0	0.026 ± 0.0
Swiss roll high	0.1	Shortest Path	$\mathbf{1.0 \pm 0.0}$	$\mathbf{1.0 \pm 0.0}$	0.018 ± 0.0	0.026 ± 0.0
Swiss roll high	0.1	TSNE	0.903 ± 0.004	0.896 ± 0.003	$\mathbf{0.006 \pm 0.0}$	$\mathbf{0.008 \pm 0.0}$
Swiss roll high	0.1	UMAP	0.806 ± 0.014	0.794 ± 0.01	0.011 ± 0.0	0.016 ± 0.001
Swiss roll high	0.5	Diffusion Map	0.98 ± 0.002	0.985 ± 0.002	0.018 ± 0.0	0.026 ± 0.0
Swiss roll high	0.5	Heat-Geo	0.998 ± 0.0	0.997 ± 0.0	0.018 ± 0.0	0.026 ± 0.0
Swiss roll high	0.5	Heat-PHATE	0.995 ± 0.0	0.997 ± 0.0	0.018 ± 0.0	0.026 ± 0.0
Swiss roll high	0.5	PHATE	0.754 ± 0.01	0.756 ± 0.006	0.018 ± 0.0	0.026 ± 0.0
Swiss roll high	0.5	Rand-Geo	0.945 ± 0.001	0.945 ± 0.002	0.018 ± 0.0	0.026 ± 0.0
Swiss roll high	0.5	Shortest Path	$\mathbf{0.999 \pm 0.0}$	$\mathbf{0.998 \pm 0.0}$	0.018 ± 0.0	0.026 ± 0.0
Swiss roll high	0.5	TSNE	0.905 ± 0.006	0.899 ± 0.003	$\mathbf{0.006 \pm 0.0}$	$\mathbf{0.008 \pm 0.0}$
Swiss roll high	0.5	UMAP	0.876 ± 0.017	0.86 ± 0.024	0.012 ± 0.0	0.017 ± 0.001
Swiss roll high	1.0	Diffusion Map	0.555 ± 0.155	0.526 ± 0.081	0.018 ± 0.0	0.026 ± 0.0
Swiss roll high	1.0	Heat-Geo	0.643 ± 0.173	0.693 ± 0.114	0.018 ± 0.0	0.026 ± 0.0
Swiss roll high	1.0	Heat-PHATE	0.609 ± 0.17	0.611 ± 0.121	0.018 ± 0.0	0.026 ± 0.0
Swiss roll high	1.0	PHATE	0.271 ± 0.025	0.343 ± 0.011	0.018 ± 0.0	0.026 ± 0.0
Swiss roll high	1.0	Rand-Geo	0.41 ± 0.038	0.446 ± 0.03	0.018 ± 0.0	0.026 ± 0.0
Swiss roll high	1.0	Shortest Path	0.343 ± 0.013	0.4 ± 0.007	0.018 ± 0.0	0.026 ± 0.0
Swiss roll high	1.0	TSNE	0.737 ± 0.124	0.723 ± 0.099	$\mathbf{0.008 \pm 0.001}$	$\mathbf{0.015 \pm 0.003}$
Swiss roll high	1.0	UMAP	$\mathbf{0.893 \pm 0.055}$	$\mathbf{0.889 \pm 0.083}$	0.014 ± 0.001	0.02 ± 0.001
Swiss roll high	5.0	Diffusion Map	0.164 ± 0.016	0.174 ± 0.009	0.018 ± 0.0	0.026 ± 0.0
Swiss roll high	5.0	Heat-PHATE	$\mathbf{0.202 \pm 0.01}$	$\mathbf{0.236 \pm 0.009}$	0.018 ± 0.0	0.026 ± 0.0
Swiss roll high	5.0	PHATE	0.201 ± 0.01	0.234 ± 0.008	0.018 ± 0.0	0.026 ± 0.0
Swiss roll high	5.0	Rand-Geo	0.192 ± 0.009	0.228 ± 0.008	0.018 ± 0.0	0.026 ± 0.0
Swiss roll high	5.0	Shortest Path	0.187 ± 0.01	0.221 ± 0.009	0.018 ± 0.0	0.026 ± 0.0
Swiss roll high	5.0	TSNE	0.182 ± 0.011	0.213 ± 0.01	$\mathbf{0.013 \pm 0.0}$	$\mathbf{0.019 \pm 0.001}$
Swiss roll high	5.0	UMAP	0.195 ± 0.009	0.227 ± 0.008	0.016 ± 0.0	0.024 ± 0.001
Swiss roll very high	0.1	Diffusion Map	0.977 ± 0.005	0.984 ± 0.004	0.018 ± 0.0	0.026 ± 0.0
Swiss roll very high	0.1	Heat-Geo	0.996 ± 0.001	0.997 ± 0.001	0.018 ± 0.0	0.026 ± 0.0
Swiss roll very high	0.1	Heat-PHATE	0.991 ± 0.001	0.997 ± 0.001	0.018 ± 0.0	0.026 ± 0.0
Swiss roll very high	0.1	PHATE	0.683 ± 0.023	0.701 ± 0.016	0.018 ± 0.0	0.026 ± 0.0
Swiss roll very high	0.1	Rand-Geo	0.918 ± 0.002	0.917 ± 0.002	0.018 ± 0.0	0.026 ± 0.0
Swiss roll very high	0.1	Shortest Path	$\mathbf{0.999 \pm 0.0}$	$\mathbf{0.999 \pm 0.0}$	0.018 ± 0.0	0.026 ± 0.0
Swiss roll very high	0.1	TSNE	0.905 ± 0.006	0.897 ± 0.004	$\mathbf{0.006 \pm 0.0}$	$\mathbf{0.008 \pm 0.0}$
Swiss roll very high	0.1	UMAP	0.785 ± 0.024	0.781 ± 0.017	0.011 ± 0.0	0.016 ± 0.001
Swiss roll very high	0.5	Diffusion Map	0.978 ± 0.002	0.984 ± 0.001	0.018 ± 0.0	0.026 ± 0.0
Swiss roll very high	0.5	Heat-Geo	0.997 ± 0.0	$\mathbf{0.998 \pm 0.0}$	0.018 ± 0.0	0.026 ± 0.0
Swiss roll very high	0.5	Heat-PHATE	0.996 ± 0.001	0.997 ± 0.0	0.018 ± 0.0	0.026 ± 0.0
Swiss roll very high	0.5	PHATE	0.827 ± 0.003	0.815 ± 0.002	0.018 ± 0.0	0.026 ± 0.0
Swiss roll very high	0.5	Rand-Geo	0.944 ± 0.001	0.944 ± 0.001	0.018 ± 0.0	0.026 ± 0.0
Swiss roll very high	0.5	Shortest Path	$\mathbf{0.998 \pm 0.0}$	0.997 ± 0.0	0.018 ± 0.0	0.026 ± 0.0

data	Noise level	Method	PearsonR	SpearmanR	Norm Fro N2	Norm inf N2
Tree	0.1	Diffusion Map	0.748 ± 0.125	0.733 ± 0.111	0.113 ± 0.012	0.161 ± 0.019
Tree	0.1	Heat-Geo	$\mathbf{0.943 \pm 0.037}$	$\mathbf{0.94 \pm 0.037}$	0.113 ± 0.012	0.161 ± 0.019
Tree	0.1	Heat-PHATE	0.872 ± 0.04	0.83 ± 0.061	0.113 ± 0.012	0.161 ± 0.019
Tree	0.1	PHATE	0.564 ± 0.039	0.469 ± 0.052	0.113 ± 0.011	0.161 ± 0.018
Tree	0.1	Rand-Geo	0.868 ± 0.017	0.85 ± 0.019	0.113 ± 0.012	0.161 ± 0.019
Tree	0.1	Shortest Path	0.937 ± 0.037	0.931 ± 0.041	0.113 ± 0.012	0.161 ± 0.019
Tree	0.1	TSNE	0.847 ± 0.034	0.824 ± 0.045	$\mathbf{0.082 \pm 0.012}$	$\mathbf{0.123 \pm 0.022}$
Tree	0.1	UMAP	0.692 ± 0.058	0.671 ± 0.047	0.107 ± 0.012	0.153 ± 0.019
Tree	0.5	Diffusion Map	0.656 ± 0.054	0.653 ± 0.057	0.113 ± 0.012	0.161 ± 0.019
Tree	0.5	Heat-Geo	$\mathbf{0.806 \pm 0.019}$	$\mathbf{0.787 \pm 0.009}$	0.113 ± 0.012	0.161 ± 0.019
Tree	0.5	Heat-PHATE	0.746 ± 0.024	0.744 ± 0.031	0.113 ± 0.012	0.161 ± 0.019
Tree	0.5	PHATE	0.766 ± 0.023	0.746 ± 0.03	0.113 ± 0.011	0.161 ± 0.018
Tree	0.5	Rand-Geo	0.721 ± 0.024	0.694 ± 0.024	0.113 ± 0.012	0.161 ± 0.019
Tree	0.5	Shortest Path	0.765 ± 0.01	0.738 ± 0.011	0.113 ± 0.012	0.161 ± 0.019
Tree	0.5	TSNE	0.795 ± 0.046	0.766 ± 0.055	$\mathbf{0.083 \pm 0.012}$	$\mathbf{0.128 \pm 0.018}$
Tree	0.5	UMAP	0.783 ± 0.06	0.757 ± 0.054	0.11 ± 0.011	0.157 ± 0.018
Tree	1.0	Diffusion Map	0.538 ± 0.05	0.471 ± 0.089	0.113 ± 0.012	0.161 ± 0.019
Tree	1.0	Heat-Geo	0.613 ± 0.025	$\mathbf{0.58 \pm 0.036}$	0.113 ± 0.012	0.161 ± 0.019
Tree	1.0	Heat-PHATE	0.614 ± 0.02	0.571 ± 0.044	0.113 ± 0.012	0.161 ± 0.019
Tree	1.0	PHATE	$\mathbf{0.615 \pm 0.017}$	0.572 ± 0.036	0.113 ± 0.011	0.161 ± 0.018
Tree	1.0	Rand-Geo	0.487 ± 0.064	0.465 ± 0.071	0.113 ± 0.012	0.161 ± 0.019
Tree	1.0	Shortest Path	0.542 ± 0.047	0.514 ± 0.06	0.113 ± 0.012	0.161 ± 0.019
Tree	1.0	TSNE	0.583 ± 0.042	0.553 ± 0.045	$\mathbf{0.086 \pm 0.011}$	$\mathbf{0.135 \pm 0.017}$
Tree	1.0	UMAP	0.595 ± 0.032	0.562 ± 0.036	0.111 ± 0.011	0.158 ± 0.019
Tree high	0.1	Diffusion Map	0.754 ± 0.049	0.741 ± 0.057	0.267 ± 0.021	0.369 ± 0.026
Tree high	0.1	Heat-Geo	0.956 ± 0.014	$\mathbf{0.957 \pm 0.015}$	0.267 ± 0.021	0.369 ± 0.026
Tree high	0.1	Heat-PHATE	0.831 ± 0.082	0.764 ± 0.115	0.267 ± 0.021	0.369 ± 0.026
Tree high	0.1	PHATE	0.484 ± 0.036	0.4 ± 0.028	0.267 ± 0.02	0.369 ± 0.025
Tree high	0.1	Rand-Geo	0.817 ± 0.013	0.774 ± 0.022	0.267 ± 0.021	0.369 ± 0.026
Tree high	0.1	Shortest Path	$\mathbf{0.958 \pm 0.014}$	0.956 ± 0.017	0.267 ± 0.021	0.369 ± 0.026
Tree high	0.1	TSNE	0.89 ± 0.039	0.866 ± 0.043	$\mathbf{0.233 \pm 0.021}$	$\mathbf{0.327 \pm 0.026}$
Tree high	0.1	UMAP	0.8 ± 0.031	0.764 ± 0.034	0.259 ± 0.021	0.36 ± 0.028
Tree high	0.5	Diffusion Map	0.706 ± 0.124	0.705 ± 0.113	0.267 ± 0.021	0.369 ± 0.026
Tree high	0.5	Heat-Geo	$\mathbf{0.932 \pm 0.022}$	$\mathbf{0.928 \pm 0.023}$	0.267 ± 0.021	0.369 ± 0.026
Tree high	0.5	Heat-PHATE	0.923 ± 0.023	0.921 ± 0.022	0.267 ± 0.021	0.369 ± 0.026
Tree high	0.5	PHATE	0.844 ± 0.048	0.79 ± 0.07	0.267 ± 0.02	0.369 ± 0.025
Tree high	0.5	Rand-Geo	0.875 ± 0.042	0.855 ± 0.048	0.267 ± 0.021	0.369 ± 0.026
Tree high	0.5	Shortest Path	0.917 ± 0.025	0.91 ± 0.03	0.267 ± 0.021	0.369 ± 0.026
Tree high	0.5	TSNE	0.922 ± 0.035	0.91 ± 0.045	$\mathbf{0.237 \pm 0.021}$	$\mathbf{0.334 \pm 0.027}$
Tree high	0.5	UMAP	0.823 ± 0.054	0.803 ± 0.041	0.261 ± 0.021	0.361 ± 0.026
Tree high	1.0	Diffusion Map	0.598 ± 0.117	0.613 ± 0.103	0.267 ± 0.021	0.369 ± 0.026
Tree high	1.0	Heat-Geo	0.794 ± 0.066	0.805 ± 0.049	0.267 ± 0.021	0.369 ± 0.026
Tree high	1.0	Heat-PHATE	0.826 ± 0.064	0.823 ± 0.067	0.267 ± 0.021	0.369 ± 0.026
Tree high	1.0	PHATE	0.827 ± 0.059	0.82 ± 0.062	0.267 ± 0.02	0.369 ± 0.025
Tree high	1.0	Rand-Geo	0.71 ± 0.043	0.686 ± 0.045	0.267 ± 0.021	0.369 ± 0.026
Tree high	1.0	Shortest Path	0.771 ± 0.064	0.753 ± 0.07	0.267 ± 0.021	0.369 ± 0.026
Tree high	1.0	TSNE	0.84 ± 0.066	0.821 ± 0.074	$\mathbf{0.238 \pm 0.02}$	$\mathbf{0.335 \pm 0.026}$
Tree high	1.0	UMAP	$\mathbf{0.853 \pm 0.051}$	$\mathbf{0.839 \pm 0.057}$	0.264 ± 0.021	0.365 ± 0.026

Table 9. Comparison of the estimated distance matrices with the ground truth geodesic distance matrices on the Tree dataset, using a two-dimensional embedding. Best models on average are bolded (not necessarily significant).

data	Noise level	Method	Homogeneity	Adjusted Rand Score	Adjusted Mutual Info Score
Swiss roll	0.1	Heat-Geo	$\mathbf{0.82 \pm 0.008}$	$\mathbf{0.668 \pm 0.034}$	$\mathbf{0.74 \pm 0.018}$
Swiss roll	0.1	Phate	0.731 ± 0.035	0.546 ± 0.044	0.652 ± 0.046
Swiss roll	0.1	TSNE	0.748 ± 0.067	0.537 ± 0.1	0.668 ± 0.068
Swiss roll	0.1	UMAP	0.81 ± 0.036	0.611 ± 0.039	0.726 ± 0.045
Swiss roll	0.5	Heat-Geo	0.813 ± 0.026	0.656 ± 0.049	0.733 ± 0.022
Swiss roll	0.5	Phate	0.735 ± 0.048	0.543 ± 0.064	0.656 ± 0.053
Swiss roll	0.5	TSNE	0.764 ± 0.07	0.564 ± 0.097	0.684 ± 0.065
Swiss roll	0.5	UMAP	$\mathbf{0.826 \pm 0.019}$	$\mathbf{0.664 \pm 0.073}$	$\mathbf{0.744 \pm 0.032}$
Swiss roll	1.0	Heat-Geo	0.722 ± 0.051	0.548 ± 0.091	0.652 ± 0.056
Swiss roll	1.0	Phate	0.482 ± 0.014	0.317 ± 0.031	0.428 ± 0.021
Swiss roll	1.0	TSNE	$\mathbf{0.757 \pm 0.037}$	$\mathbf{0.562 \pm 0.058}$	$\mathbf{0.679 \pm 0.042}$
Swiss roll	1.0	UMAP	0.726 ± 0.041	0.51 ± 0.077	0.65 ± 0.05
Swiss roll high	0.1	Heat-Geo	$\mathbf{0.82 \pm 0.015}$	$\mathbf{0.666 \pm 0.033}$	$\mathbf{0.739 \pm 0.019}$
Swiss roll high	0.1	Phate	0.705 ± 0.03	0.518 ± 0.048	0.628 ± 0.04
Swiss roll high	0.1	TSNE	0.757 ± 0.078	0.558 ± 0.115	0.677 ± 0.08
Swiss roll high	0.1	UMAP	0.796 ± 0.03	0.624 ± 0.048	0.714 ± 0.037
Swiss roll high	0.5	Heat-Geo	$\mathbf{0.805 \pm 0.021}$	$\mathbf{0.655 \pm 0.047}$	$\mathbf{0.725 \pm 0.035}$
Swiss roll high	0.5	Phate	0.745 ± 0.04	0.562 ± 0.061	0.664 ± 0.047
Swiss roll high	0.5	TSNE	0.747 ± 0.075	0.538 ± 0.11	0.668 ± 0.075
Swiss roll high	0.5	UMAP	0.787 ± 0.041	0.573 ± 0.067	0.703 ± 0.032
Swiss roll high	1.0	Heat-Geo	0.7 ± 0.045	0.534 ± 0.057	0.644 ± 0.032
Swiss roll high	1.0	Phate	0.552 ± 0.047	0.386 ± 0.056	0.496 ± 0.04
Swiss roll high	1.0	TSNE	0.754 ± 0.034	0.548 ± 0.068	0.675 ± 0.036
Swiss roll high	1.0	UMAP	$\mathbf{0.76 \pm 0.041}$	$\mathbf{0.56 \pm 0.077}$	$\mathbf{0.68 \pm 0.05}$
Swiss roll very high	0.1	Heat-Geo	$\mathbf{0.818 \pm 0.033}$	$\mathbf{0.668 \pm 0.074}$	$\mathbf{0.738 \pm 0.039}$
Swiss roll very high	0.1	Phate	0.688 ± 0.043	0.497 ± 0.053	0.614 ± 0.053
Swiss roll very high	0.1	TSNE	0.741 ± 0.07	0.544 ± 0.101	0.662 ± 0.075
Swiss roll very high	0.1	UMAP	0.816 ± 0.042	0.65 ± 0.069	0.733 ± 0.054
Swiss roll very high	0.5	Heat-Geo	0.73 ± 0.045	$\mathbf{0.605 \pm 0.093}$	0.701 ± 0.028
Swiss roll very high	0.5	Phate	0.758 ± 0.034	0.55 ± 0.037	0.676 ± 0.014
Swiss roll very high	0.5	TSNE	0.77 ± 0.054	0.557 ± 0.093	$\mathbf{0.708 \pm 0.031}$
Swiss roll very high	0.5	UMAP	$\mathbf{0.789 \pm 0.052}$	0.574 ± 0.101	0.707 ± 0.061
Swiss roll very high	1.0	Heat-Geo	0.592 ± 0.033	0.427 ± 0.063	0.545 ± 0.031
Swiss roll very high	1.0	Phate	0.531 ± 0.042	0.377 ± 0.046	0.486 ± 0.045
Swiss roll very high	1.0	TSNE	$\mathbf{0.738 \pm 0.019}$	$\mathbf{0.551 \pm 0.039}$	$\mathbf{0.662 \pm 0.025}$
Swiss roll very high	1.0	UMAP	0.736 ± 0.057	0.542 ± 0.102	0.66 ± 0.061
Tree	0.1	Heat-Geo	$\mathbf{0.784 \pm 0.051}$	$\mathbf{0.734 \pm 0.07}$	$\mathbf{0.786 \pm 0.051}$
Tree	0.1	Phate	0.55 ± 0.042	0.409 ± 0.064	0.555 ± 0.042
Tree	0.1	TSNE	0.706 ± 0.054	0.61 ± 0.075	0.712 ± 0.055
Tree	0.1	UMAP	0.678 ± 0.086	0.584 ± 0.12	0.681 ± 0.086
Tree	0.5	Heat-Geo	0.545 ± 0.121	0.411 ± 0.154	0.577 ± 0.094
Tree	0.5	Phate	0.529 ± 0.111	0.404 ± 0.151	0.555 ± 0.095
Tree	0.5	TSNE	$\mathbf{0.647 \pm 0.049}$	$\mathbf{0.591 \pm 0.065}$	0.65 ± 0.048
Tree	0.5	UMAP	0.645 ± 0.051	0.565 ± 0.058	$\mathbf{0.652 \pm 0.05}$
Tree	1.0	Heat-Geo	0.398 ± 0.07	0.3 ± 0.077	0.42 ± 0.07
Tree	1.0	Phate	0.418 ± 0.08	0.337 ± 0.093	0.43 ± 0.075
Tree	1.0	TSNE	0.405 ± 0.077	0.378 ± 0.074	0.405 ± 0.077
Tree	1.0	UMAP	$\mathbf{0.432 \pm 0.086}$	$\mathbf{0.395 \pm 0.098}$	$\mathbf{0.432 \pm 0.085}$

Table 10. Clustering results on swiss roll (with distribution) and tree. Best models on average are bolded (not necessarily significant).

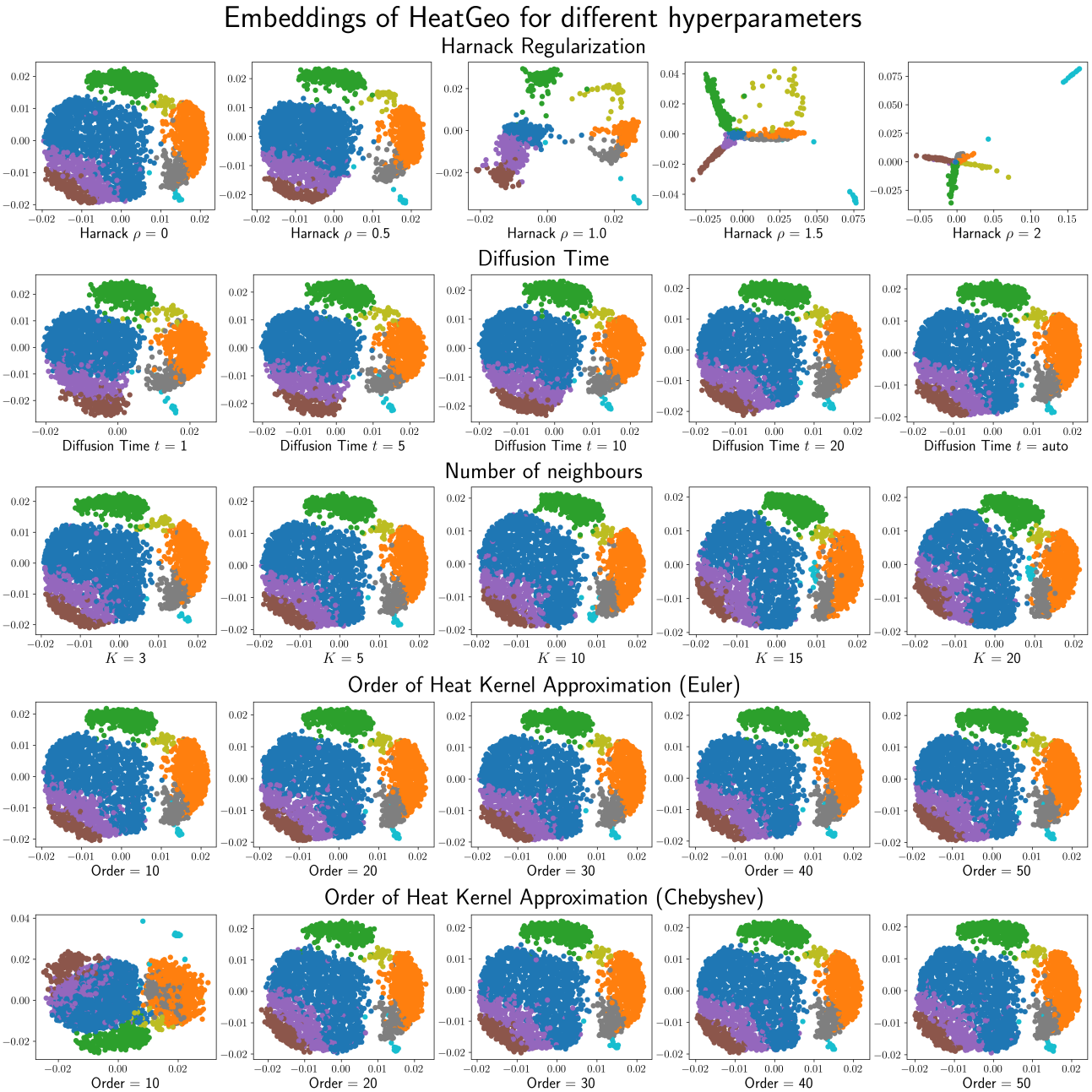

Figure 7. Embeddings of Heat Geodesic Embedding for different choices of hyperparameters on the EB dataset. We evaluate the impact of the Harnack regularization, the diffusion time, the number of neighbours in the kNN, and the order of the approximation for Euler and Checbyshev approximations.

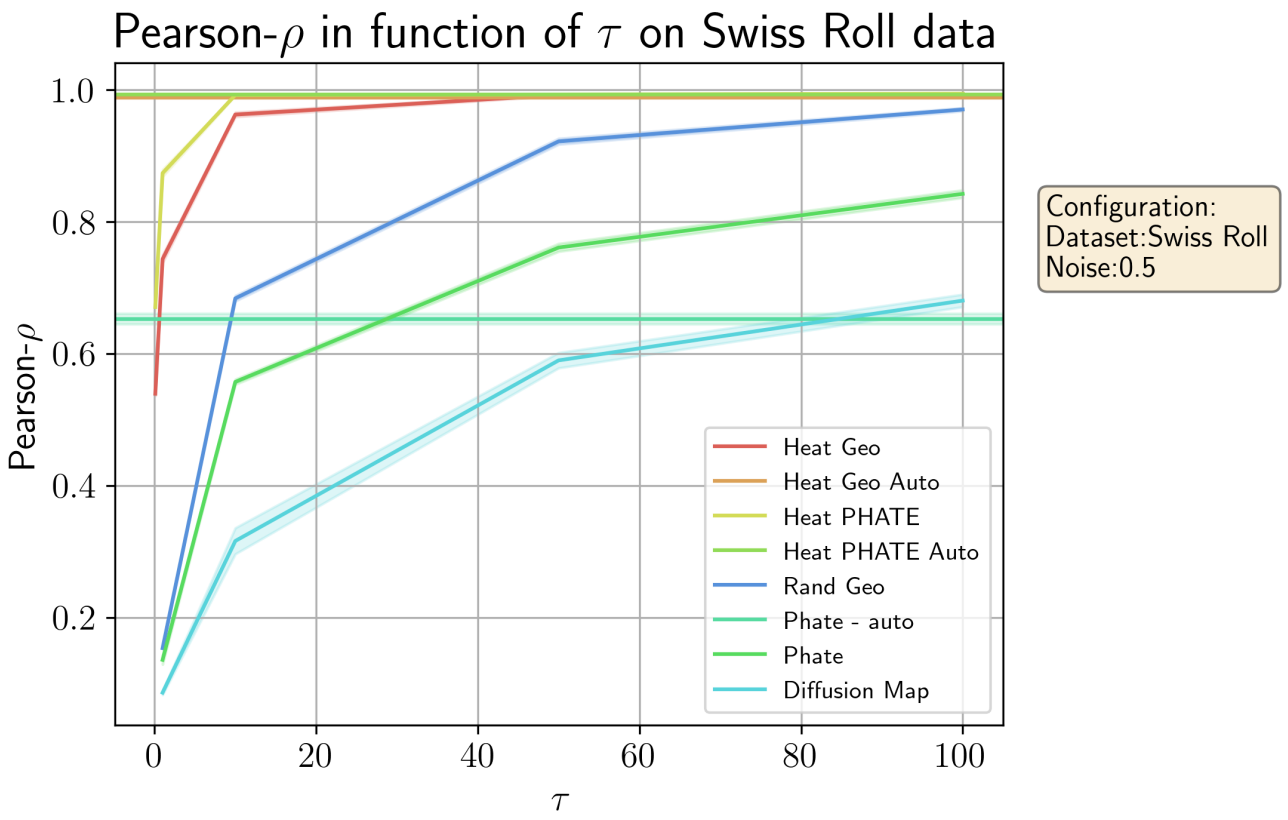

Figure 8. Impact of diffusion time on the Pearson correlation between the estimated distance matrix and ground truth distance matrix for different methods on the Swiss roll dataset.

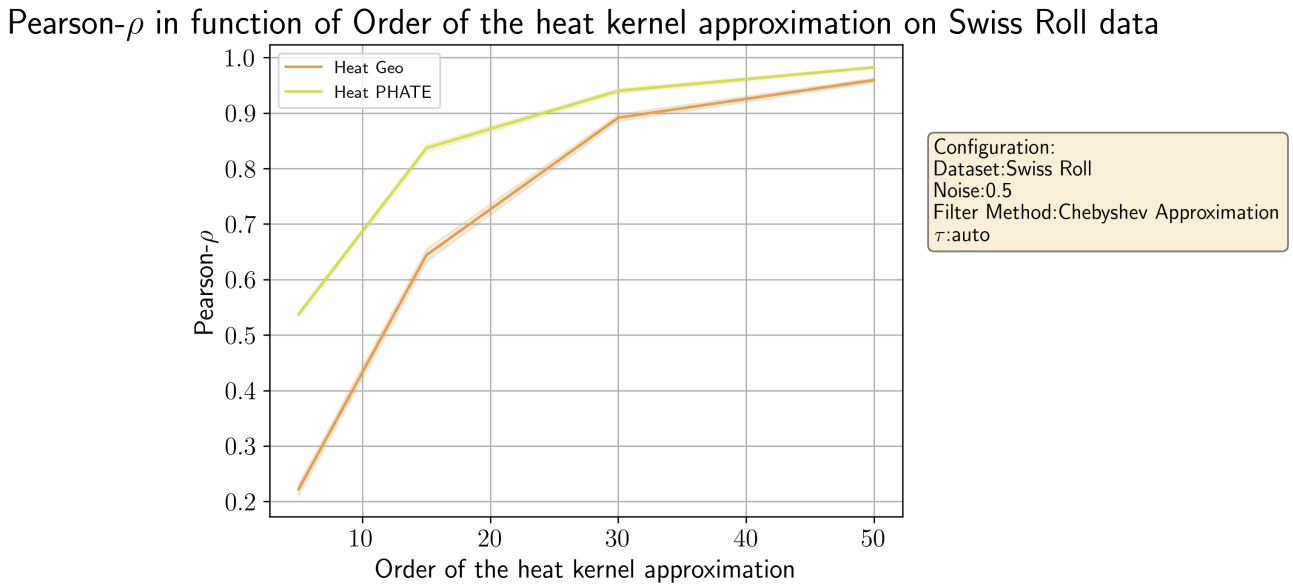

Figure 9. Impact of Checbyshev approximation order on the Pearson correlation between the estimated distance matrix and ground truth distance matrix for different methods on the Swiss roll dataset.

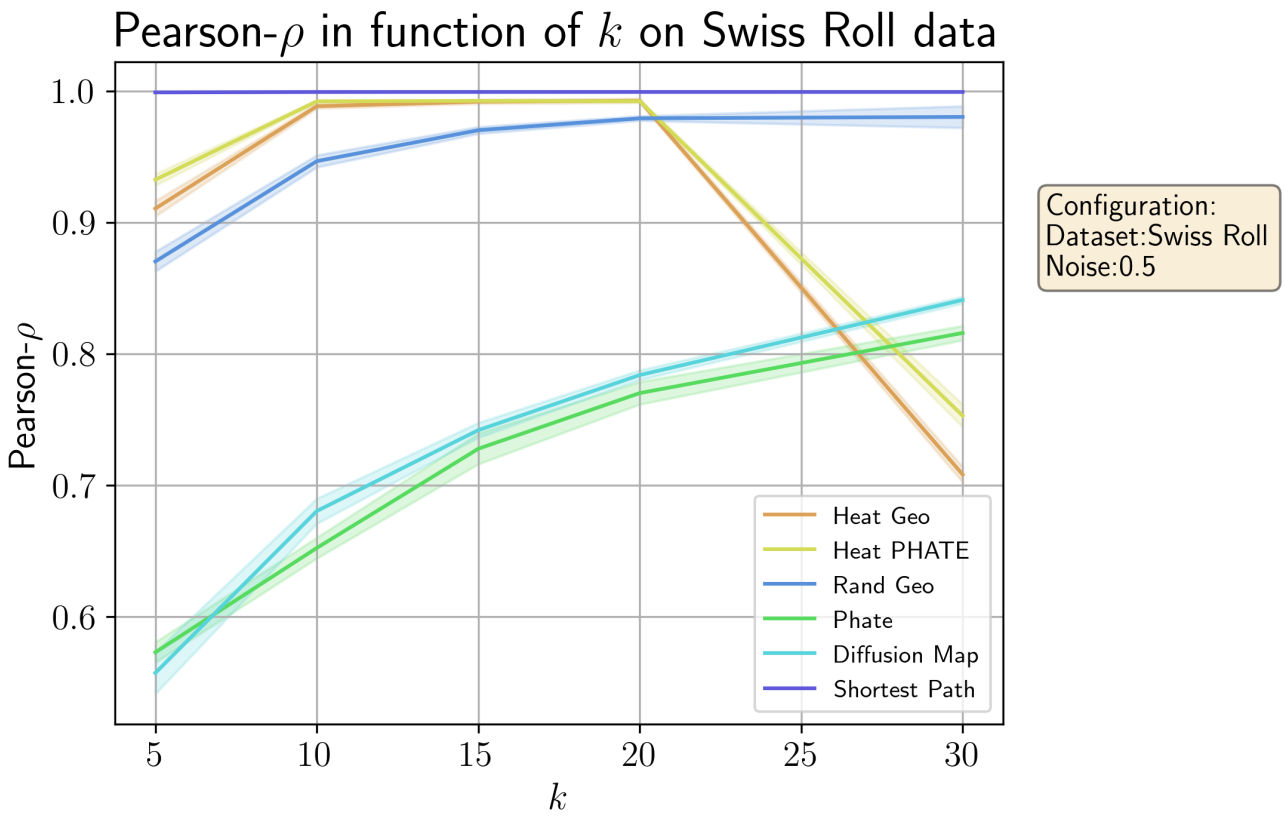

Figure 10. Impact of number of neighbours on the Pearson correlation between the estimated distance matrix and ground truth distance matrix for different methods on the Swiss roll dataset.

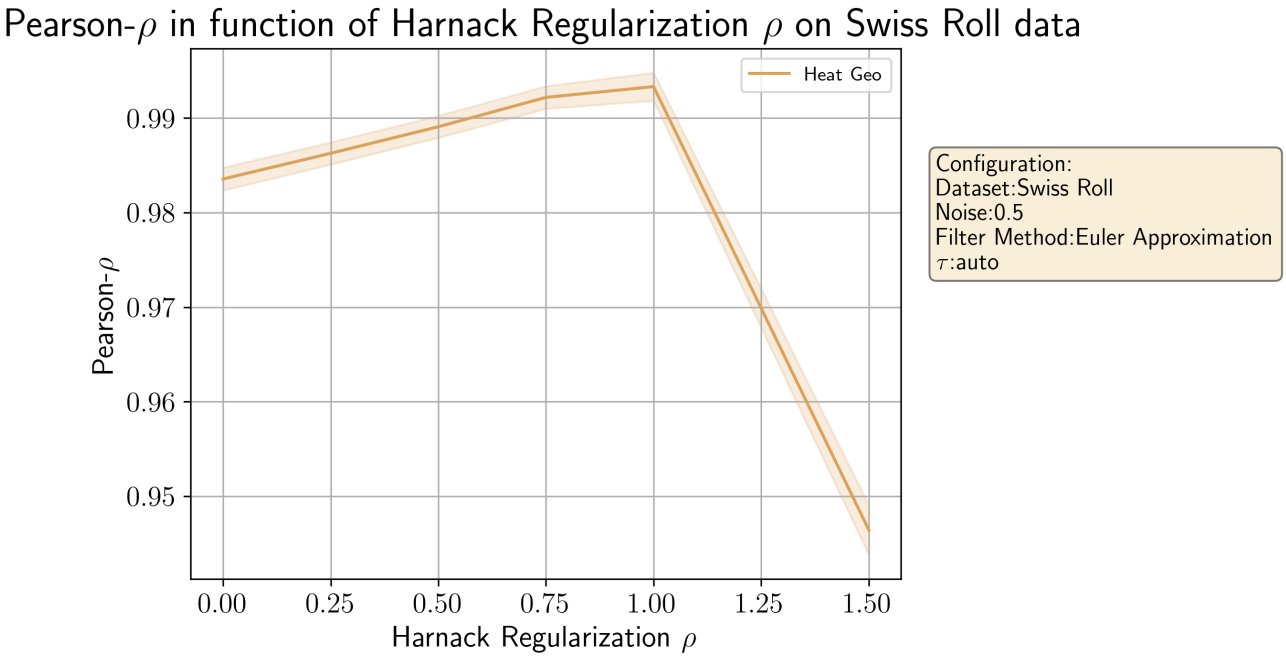

Figure 11. Impact of Harnack regularization on the Pearson correlation between the estimated distance matrix and ground truth distance matrix for HeatGeo on the Swiss roll dataset.

E.7. Graph construction

We compare the embeddings of the heat-geodesic distance for different graph construction. Throughout the paper we used the graph construction from PHATE (Moon et al., 2019). In the following we present additional results depending on the choice of kernel to construct the graph. Specifically, we use a simple nearest neighbor (kNN) graph , the graph from UMAP (McInnes et al., 2018), and the implementation in the package Scanp for single-cell analysis. In figure, we present the embeddings 2500 points of a tree with five branches in 10 dimensions, where the observations are perturbed with a standard Gaussian noise. All methods used five nearest neighbors and a diffusion time of 20. In Figure 12, we show the evolution of the Pearson correlation between estimated and ground truth distance matrices for the 10-dimensional Swiss roll dataset for various graph constructions. We note that the results are stable across different graph construction strategies.

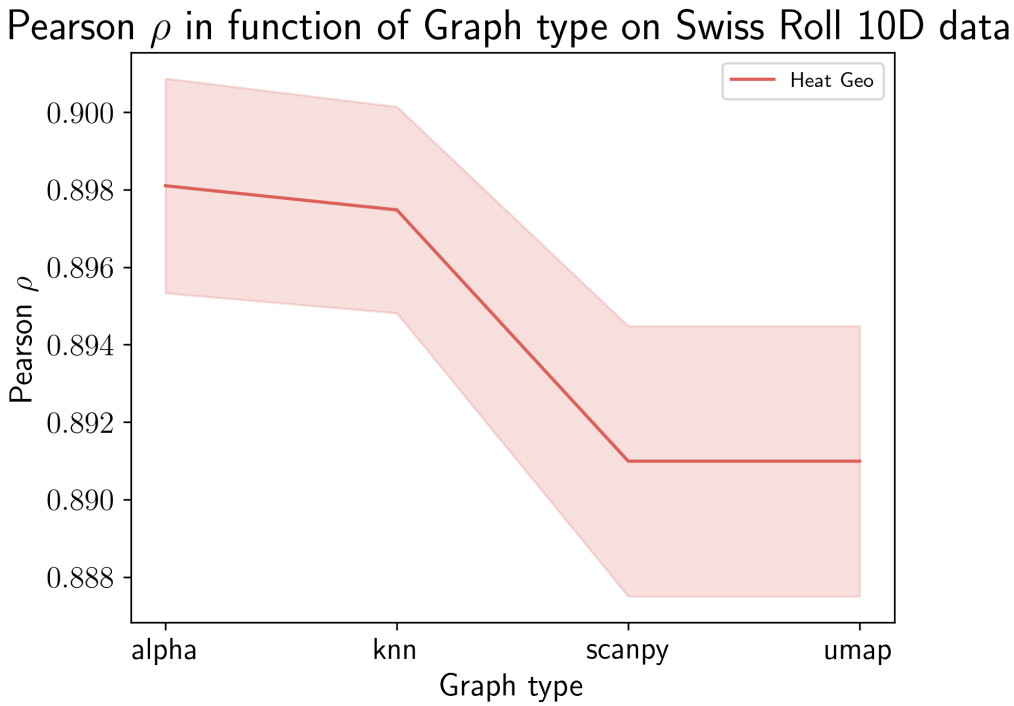

Figure 12. Pearson correlation between estimated and ground truth distance matrices for the 10-dimensional Swiss roll dataset for various graph constructions. Standard deviations are computed over the 5 test folds.