
A Heat Diffusion Perspective on Geodesic Preserving Dimensionality Reduction

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 Diffusion-based manifold learning methods have proven useful in representation
2 learning and dimensionality reduction of modern high dimensional, high through-
3 put, noisy datasets. Such datasets are especially present in fields like biology and
4 physics. While it is thought that these methods preserve underlying manifold struc-
5 ture of data by learning a proxy for geodesic distances, no specific theoretical links
6 have been established. Here, we establish such a link via results in Riemannian
7 geometry explicitly connecting heat diffusion to manifold distances. In this process,
8 we also formulate a more general heat kernel based manifold embedding method
9 that we call *heat geodesic embeddings*. This novel perspective makes clearer the
10 choices available in manifold learning and denoising. Results show that our method
11 outperforms existing state of the art in preserving ground truth manifold distances,
12 and preserving cluster structure in toy datasets. We also showcase our method on
13 single cell RNA-sequencing datasets with both continuum and cluster structure,
14 where our method enables interpolation of withheld timepoints of data. Finally, we
15 show that parameters of our more general method can be configured to give results
16 similar to PHATE (a state-of-the-art diffusion based manifold learning method) as
17 well as SNE (an attraction/repulsion neighborhood based method that forms the
18 basis of t-SNE).

19 1 Introduction

20 The advent of high throughput and high dimensional data in various fields of science have made
21 dimensionality reduction and visualization techniques an indispensable part of exploratory analysis.
22 Diffusion-based manifold learning methods, based on the data diffusion operator, first defined in
23 [5], have proven especially useful due to their ability to handle noise and density variations while
24 preserving structure. As a result, diffusion-based dimensionality reduction methods, such as PHATE
25 [21], T-PHATE [3], and diffusion maps [5], have emerged as methods for analyzing high throughput
26 noisy data in various situations. While these methods are surmised to learn manifold geodesic
27 distances, no specific theoretical links have been established. Here, we establish such a link by using
28 Varadhan’s formula [31] and a parabolic Harnack inequality [16, 23], which relate manifold distances
29 to heat diffusion directly. This lens gives new insight into existing dimensionality reduction methods,
30 including when they preserve geodesics, and suggests a new method for dimensionality reduction to
31 explicitly preserve geodesics, which we call *heat geodesic embeddings*¹. Furthermore, based on our
32 understanding of other methods [21, 5], we introduce theoretically justified parameter choices that
33 allow our method to have greater versatility in terms of distance denoising and emphasis on local
34 versus global distances.

¹Anonymized code <https://anonymous.4open.science/r/anon-heatgeo-CE2A/>

35 Generally, data diffusion operators are created by first computing distances between datapoints,
 36 transforming these distances into affinities by pointwise application of a kernel function (like a
 37 Gaussian kernel), and then row normalizing with or without first applying degree normalization into a
 38 Markovian diffusion operator P [5, 8, 13, 20, 30]. The entries of $P(x, y)$ then contain probabilities
 39 of diffusing (or random walk probabilities) from one datapoint to another. Diffusion maps and
 40 PHATE use divergences between these diffusion or random walk-based probability distributions
 41 $P(x, \cdot)$ and $P(y, \cdot)$ to design a diffusion-based distance that may not directly relate to manifold
 42 distance. Our framework directly utilizes a heat-kernel based distance, and offers a framework to
 43 study these diffusion methods from a more comprehensive perspective. By configuring parameters
 44 in our framework, we show how we can navigate a continuum of embeddings from PHATE-like to
 45 SNE-like methods.

46 In summary, our contributions are as follows:

- 47 • We define the *heat-geodesic* dissimilarity based on Varadhan’s formula.
- 48 • Based on this dissimilarity, we present a versatile geodesic-preserving method for dimensionality
 49 reduction which we call *heat geodesic embedding*.
- 50 • We establish a relationship between diffusion-based distances and the heat-geodesic dissimilarity.
- 51 • We establish connections between our method and popular dimensionality reduction techniques
 52 such as PHATE and t-SNE, shedding light on their geodesic preservation and denoising properties
 53 based on modifications of the computed dissimilarity and distance preservation losses.
- 54 • We empirically demonstrate the advantages of Heat Geodesic Embedding in preserving manifold
 55 geodesic distances in several experiments showcasing more faithful manifold distances in the
 56 embedding space, as well as our ability to interpolate data within the manifold.

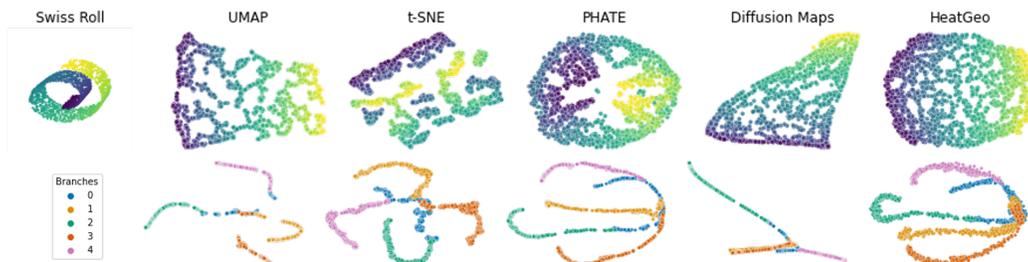


Figure 1: Embeddings of the Swiss roll (top) and Tree (bottom) datasets for different manifold learning methods. Our HeatGeo method correctly unrolls the Swiss roll while t-SNE and UMAP create undesirable artificial clusters.

57 2 Preliminaries

58 First, we introduce fundamental notions that form the basis of our manifold learning methods: Varad-
 59 han’s formula [31] on a manifold, diffusion processes on graphs, efficient heat kernel approximations,
 60 and multidimensional scaling [4, 11, 15].

61 **Varadhan’s formula** Varadhan’s formula is a powerful tool in differential geometry that establishes
 62 a connection between the heat kernel and the shortest path (geodesic) distance on a Riemannian
 63 manifold. Its versatility has led to widespread applications in machine learning [6, 9, 14, 25–27]. Let
 64 (M, g) be a closed Riemannian manifold, and Δ the Laplace-Beltrami operator on M . The heat kernel
 65 $h_t(x, y)$ on M is the minimal positive fundamental solution of the heat equation $\frac{\partial u}{\partial t} = \Delta u$ with initial
 66 condition $h_0(x, y) = \delta_x(y)$. In Euclidean space the heat kernel is $h_t(x, y) = (4\pi t)^{-n/2} e^{-d(x,y)^2/4t}$
 67 so that $-4t \log h_t(x, y) = 2nt \log(4\pi t) + d^2(x, y)$ and we observe the following limiting behavior:

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

68 Varadhan [31] (see also [19]) proved that eq. 1 (now Varadhan’s formula) holds more generally
 69 on complete Riemannian manifolds M , where $d(x, y)$ is the geodesic distance on M , and the

70 convergence is uniform over compact subsets of M . A related result for complete Riemannian
 71 manifolds that satisfy the parabolic Harnack inequality (which includes convex domains in Euclidean
 72 space and Riemannian manifolds with non-negative Ricci curvature) is the two-sided heat kernel
 73 bound [23, 16], showing that for any $\epsilon \in (0, 1)$ there exist constants $c(\epsilon)$ and $C(\epsilon)$ such that

$$\frac{c(\epsilon)}{V(x, \sqrt{t})} \exp\left(-\frac{d(x, y)^2}{4(1 + \epsilon)t}\right) \leq h_t(x, y) \leq \frac{C(\epsilon)}{V(x, \sqrt{t})} \exp\left(-\frac{d(x, y)^2}{4(1 - \epsilon)t}\right) \quad (2)$$

74 We denote this relation by $h_t(x, y) \simeq V(x, \sqrt{t})^{-1} \exp(-d(x, y)^2/t)$ and note that it again recovers
 75 eq. 1 in the $t \rightarrow 0$ limit, which is unsurprising as Varadhan’s result holds more generally. More
 76 important for our purposes is that $h_t(x, y) \simeq V(x, \sqrt{t})^{-1} \exp(-d(x, y)^2/t)$ holds for $t > 0$ which
 77 will allow us to calculate geodesic distances $d(x, y)$ from a diffusion based estimation of the heat
 78 kernel $h_t(x, y)$ and volume on point cloud data.

79 **Graph construction and diffusion** Our construction starts by creating a graph from a point cloud
 80 dataset \mathbf{X} . We use a kernel function $\kappa : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^+$, such that the (weighted) adjacency matrix is
 81 $\mathbf{W}_{ij} := \kappa(x_i, x_j)$ for all $x_i, x_j \in \mathbf{X}$. The kernel function could be a Gaussian kernel, or constructed
 82 from a nearest neighbor graph. The resulting graph \mathcal{G} is characterized by the set of nodes (an ordering
 83 of the observations), the adjacency matrix, and the set of edges, i.e. pairs of nodes with non-zero
 84 weights. The graph Laplacian is an operator acting on signals on \mathcal{G} such that it mimics the negative
 85 of the Laplace operator. The combinatorial graph Laplacian matrix is defined as $\mathbf{L} := \mathbf{Q} - \mathbf{W}$ and
 86 its normalized version as $\mathbf{L} = \mathbf{I}_n - \mathbf{Q}^{-1/2} \mathbf{W} \mathbf{Q}^{-1/2}$, where \mathbf{Q} is a diagonal degree matrix with
 87 $Q_{ii} := \sum_j \mathbf{W}_{ij}$. The Laplacian is symmetric positive semi-definite, and has an eigen-decomposition
 88 $\mathbf{L} = \Psi \Lambda \Psi^T$. Throughout the presentation, we assume that $Q_{ii} > 0$ for all $i \in [n]$. The Laplacian
 89 allows us to define the heat equation on \mathcal{G} , with respect to an initial signal $\mathbf{f}_0 \in \mathbb{R}^n$ on \mathcal{G} :

$$\frac{\partial}{\partial t} \mathbf{f}(t) + \mathbf{L} \mathbf{f}(t) = \mathbf{0}, \text{ s.t. } \mathbf{f}(0) = \mathbf{f}_0 \quad t \in \mathbb{R}^+. \quad (3)$$

90 The solution of the above differential equation is obtained with the matrix exponential $\mathbf{f}(t) = e^{-t\mathbf{L}} \mathbf{f}_0$,
 91 and we define the heat kernel on the graph as $\mathbf{H}_t := e^{-t\mathbf{L}}$. By eigendecomposition, we have
 92 $\mathbf{H}_t = \Psi e^{-t\Lambda} \Psi^T$. The matrix \mathbf{H}_t is a diffusion matrix that characterizes how a signal propagate
 93 through the graph according to the heat equations.

94 Other diffusion matrices on graphs have also been investigated in the literature. The transition matrix
 95 $\mathbf{P} := \mathbf{Q}^{-1} \mathbf{W}$ characterizing a random walk on the graph is another common diffusion matrix used
 96 for manifold learning such as PHATE and diffusion maps [5]. It is a stochastic matrix that converges
 97 to a stationary distribution $\pi_i := Q_{ii} / \sum_i Q_{ii}$, under mild assumptions.

98 **Fast computation of Heat diffusion** Exact computation of the (discrete) heat kernel \mathbf{H}_t is com-
 99 putationally costly, requiring a full eigendecomposition in $O(n^3)$ time. Fortunately, multiple fast
 100 approximations have been proposed, including using orthogonal polynomials or the Euler backward
 101 methods. In this work, we use Chebyshev polynomials, as they have been shown to converge faster
 102 than other polynomials on this problem [12].

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

$$\mathbf{H}_t \approx p_K(\mathbf{L}, t) := \frac{b_{t,0}}{2} + \sum_{k=1}^K b_{t,k} T_k(\mathbf{L} - \mathbf{I}_n), \quad (4)$$

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

115 Another possible approximation is using the Euler backward method. It requires solving K systems of
 116 linear equations $\mathbf{f}(t) = (\mathbf{I}_n + (t/K)\mathbf{L})^{-K}\mathbf{f}(0)$, which can be efficient for sparse matrices using the
 117 Cholesky decomposition [9, 26]. We quantify the differences between the heat kernel approximations
 118 in Appendix C.

119 **Multidimensional scaling** Given a dissimilarity function d between data points, multidimensional
 120 scaling (MDS) [15] finds an embedding ϕ such that the difference between the given dissimilarity
 121 and the Euclidean distance in the embedded space is minimal across all data points. Formally, for a
 122 given function $d : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^+$, MDS minimizes the following objective:

$$L(\mathbf{X}) = \left(\sum_{ij} (d(x_i, x_j) - \|\phi(x_i) - \phi(x_j)\|_2)^2 \right)^{1/2}, \quad (5)$$

123 In metric MDS the solution is usually found by the SMACOF algorithm [28], or stochastic gradient
 124 descent [34], while classic MDS is defined by eigendecomposition.

125 3 Related Work

126 We review state-of-the-art embedding methods and contextualize them with respect to Heat Geodesic
 127 Embedding. A formal theoretical comparison of all methods is given in Section 5. Given a set of
 128 high-dimensional datapoints, the objective of embedding methods is to create a map that embeds
 129 the observations in a lower dimensional space, while preserving distances or similarities. Different
 130 methods vary by their choice of distance or dissimilarity functions, as shown below.

131 **Diffusion maps** In diffusion maps [5], an embedding in k dimensions is defined via the first k non-
 132 trivial right eigenvectors of \mathbf{P}^t weighted by their eigenvalues. The embedding preserves the *diffusion*
 133 *distance* $DM_{\mathbf{P}}(x_i, x_j) := \|(\delta_i \mathbf{P}^t - \delta_j \mathbf{P}^t)(1/\pi)\|_2$, where δ_i is a vector such that $(\delta_i)_j = 1$ if
 134 $j = i$ and 0 otherwise, and π is the stationary distribution of \mathbf{P} . Intuitively, $DM_{\mathbf{P}}(x_i, x_j)$ considers
 135 all the t -steps paths between x_i and x_j . A larger diffusion time can be seen as a low frequency
 136 graph filter, i.e. keeping only information from the low frequency transitions such as the stationary
 137 distributions. For this reason, using diffusion with $t > 1$ helps denoising the relationship between
 138 observations.

139 **PHATE** This diffusion-based method preserves the *potential distance* [21] $PH_{\mathbf{P}} := \|-\log \delta_i \mathbf{P}^t +$
 140 $\log \delta_j \mathbf{P}^t\|_2$, and justifies this approach using the log transformation to prevent nearest neighbors from
 141 dominating the distances. An alternative approach is suggested using a square root transformation.
 142 Part of our contributions is to justify the log transformation from a geometric point of view. The
 143 embedding is defined using multidimensional scaling, which we present below.

144 **SNE, t-SNE, UMAP** Well-known attraction/repulsion methods such as SNE [10], t-SNE [29], and
 145 UMAP [18] define an affinity matrix with entries p_{ij} in the ambient space, and another affinity matrix
 146 with entries q_{ij} in the embedded space. To define the embedding, a loss between the two affinity
 147 matrices is minimized. Specifically, the loss function is $D_{\text{KL}}(p||q) := \sum_{ij} p_{ij} \log p_{ij}/q_{ij}$ in SNE
 148 and t-SNE, whereas UMAP adds $D_{\text{KL}}(1-p||1-q)$ [2]. While these methods preserves affinities,
 149 they do not preserve any types of distances in the embedding.

150 4 Heat-Geodesic Embedding

151 In this section, we present our Heat Geodesic Embedding which is summarized in Alg. 1. We start by
 152 introducing the heat-geodesic dissimilarity, then present a robust transformation, and a heuristic to
 153 choose the optimal diffusion time. Proofs not present in the main text are given in the AppendixA.

154 We consider the discrete case, where we have a set of n points $\{x_i\}_{i=1}^n =: \mathbf{X}$ in a high dimensional
 155 Euclidean space $x_i \in \mathbb{R}^d$. From this point cloud, we want to define a map $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^k$ that embeds
 156 the observation in a lower dimensional space. An important property of our embedding is that we
 157 preserve manifold geodesic distances in a low dimensional space.

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

162 **Definition 4.1.** For a diffusion time $t > 0$ and tunable parameter $\sigma > 0$, we define the **heat-geodesic**
 163 **dissimilarity** between $x_i, x_j \in \mathbf{X}$ as

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

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

165 Here the log is applied elementwise, and the term $-4t \log(\mathbf{H}_t)_{ij}$ corresponds to the geodesic
 166 approximation when $t \rightarrow 0$ as in Varadhan’s formula. In practice one uses a fixed diffusion time
 167 $t > 0$, so we add a symmetric volume correction term as in the Harnack inequality, ensuring that
 168 $d_t(x_i, x_j)$ is symmetric. From Sec. 2, we have $h_t(x, x) \simeq V(x, \sqrt{t})^{-1}$, and we use the diagonal
 169 of \mathbf{H}_t to approximate the inverse of the volume. With this volume correction term and $\sigma = 1$, the
 170 dissimilarity is such that $d_t(x_i, x_i) = 0$ for all $t > 0$. When $\sigma = 0$ or the manifold has uniform
 171 volume growth (as in the constant curvature setting) we show that the heat-geodesic dissimilarity is
 172 order preserving:

173 **Proposition 4.2.** *When $\sigma = 0$ or the manifold has uniform volume growth, i.e. $(\mathbf{H}_t)_{ii} = (\mathbf{H}_t)_{jj}$, we*
 174 *have for triples $x, y, z \in \mathbf{X}$ that $|x - y| > |x - z|$ implies $d_t(x, y) > d_t(x, z)$, i.e. the heat-geodesic*
 175 *dissimilarity is order preserving.*

176 *Proof.* When $\sigma = 0$ or the manifold has uniform volume growth we need only consider the
 177 $-4t \log(\mathbf{H}_t)_{ij}$ terms. The assumption that $|x - y| > |x - z|$ implies $\mathbf{H}_t(x, y) < \mathbf{H}_t(x, z)$. We are
 178 able to conclude that $-4t \log \mathbf{H}_t(x, y) > -4t \log \mathbf{H}_t(x, z)$ and thus $d_t(x, y) > d_t(x, z)$. \square

179 **Denosing Distances with Triplet Computations** We note that both diffusion maps and PHATE
 180 compute a triplet distance between datapoints, i.e., rather than using the direct diffusion probability
 181 between datapoints, they use the a distance between corresponding rows of a diffusion operator. In
 182 particular, diffusion maps using Euclidean distance, and PHATE uses an M-divergence. Empirically,
 183 we notice that this step acts as a denoiser for distances. We formalize this observation in the
 184 following proposition. We note D_T the triplet distance. The triplet distance compares the distances
 185 relative to other points. Intuitively, this is a denoising step, since the effect of the noise is spread
 186 across the entire set of points. For a reference dissimilarity like the heat-geodesic, it is defined as
 187 $D_T(x_i, x_j) := \|d_t(x_i, \cdot) - d_t(x_j, \cdot)\|_2$. For linear perturbations of the form $d_t(x_i, x_j) + \epsilon$, where
 188 $\epsilon \in \mathbb{R}$, the effect of ϵ on $D_T(x_i, x_j)$ is less severe than on $d_t(x_i, x_j)$.

189 **Proposition 4.3.** *Denote the perturbed triplet distance by $\widetilde{D}_T(x_i, x_j) = \|\widetilde{d}_t(x_i, \cdot) - \widetilde{d}_t(x_j, \cdot)\|_2$*
 190 *where $\widetilde{d}_t(x_i, x_j) := d_t(x_i, x_j) + \epsilon$ and $\widetilde{d}_t(x_i, x_k) := d_t(x_i, x_k)$ for $k \neq j$. Then the triplet distance*
 191 *D_T is robust to perturbations, i.e., for all $\epsilon > 0$,*

$$\left(\frac{\widetilde{D}_T(x_i, x_j)}{D_T(x_i, x_j)} \right)^2 \leq \left(\frac{d_t(x_i, x_j) + \epsilon}{d_t(x_i, x_j)} \right)^2.$$

192 **Optimal diffusion time** Varadhan’s formula suggests a small value of diffusion time t to approxi-
 193 mate geodesic distance on a manifold. However, in the discrete data setting, geodesics are based on
 194 graph constructions, which in turn rely on nearest neighbors. Thus, small t can lead to disconnected
 195 graphs. Additionally, increasing t can serve as a way of denoising the kernel (which is often computed
 196 from noisy data) as it implements a low-pass filter over the eigenvalues, providing the additional
 197 advantage of adding noise tolerance. By computing a sequence of heat kernels $(\mathbf{H}_t)_t$ and evaluating
 198 their entropy $H(\mathbf{H}_t) := -\sum_{ij} (\mathbf{H}_t)_{ij} \log(\mathbf{H}_t)_{ij}$, we select t with the knee-point method [24] on
 199 the function $t \mapsto H(\mathbf{H}_t)$. We show in Sec. 6.1 that our heuristic for determining the diffusion time
 200 automatically leads to better overall results.

201 **Weighted MDS** The loss in MDS (eq.5) is usually defined with uniform weights. Here, we
 202 optionally weight the loss by the heat kernel. In Sec. 5, we will show how this modification relates our
 203 method to the embedding defined by SNE[10]. For $x_i, x_j \in \mathbf{X}$, we minimize $(\mathbf{H}_t)_{ij}(d_t(x_i, x_j) -$
 204 $\|\phi(x_i) - \phi(x_j)\|_2)^2$. This promotes geodesic preservation of local neighbors, since more weights
 205 are given to points with higher affinities.

206 **Heat-geodesic embedding** To define a lower dimensional embedding of a point cloud \mathbf{X} , we
 207 construct a matrix from the heat-geodesic dissimilarity, and then use MDS to create the embedding.
 208 Our embedding defines a map ϕ that minimizes $(d_t(x_i, x_j) - \|\phi(x_i) - \phi(x_j)\|_2)^2$, for all $x_i, x_j \in \mathbf{X}$.
 209 Hence, it preserves the heat-geodesic dissimilarity as the loss decreases to zero. In Alg. 1, we present
 210 the main steps of our algorithm using the heat-geodesic dissimilarity. A detailed version is presented
 211 in the Appendix A.

Algorithm 1 Heat Geodesic Embedding

- 1: **Input:** $N \times d$ dataset matrix \mathbf{X} , denoising parameter $\rho \in [0, 1]$, Harnack regularization $\sigma > 0$,
output dimension k .
 - 2: **Returns:** $N \times k$ embedding matrix \mathbf{E} .
 - 3: $\mathbf{H}_t \leftarrow p_K(\mathbf{L}, t)$ \triangleright Heat approximation
 - 4: $t \leftarrow \text{Kneedle}\{H(\mathbf{H}_t)\}_t$ \triangleright Knee detection e.g. [24]
 - 5: $\mathbf{D} \leftarrow -4t \log \mathbf{H}_t + t\sigma \mathbf{V}$ \triangleright log is applied elementwise
 - 6: $\mathbf{D} \leftarrow (1 - \rho)\mathbf{D} + \rho D_T$ \triangleright Triplet interpolation step
 - 7: Return $\mathbf{E} \leftarrow \text{MetricMDS}(\mathbf{D}, \|\cdot\|_2, k)$
-

212 5 Relation to other manifold learning methods

213 In this section, we elucidate theoretical connections between the Heat Geodesic Embedding and other
 214 manifold learning methods. We relate embeddings via the eigenvalues of \mathbf{H}_t or \mathbf{P}^t with Laplacian
 215 eigenmaps and diffusion maps. We then present the relation between our methods and PHATE and
 216 SNE. We provide further analysis in the Appendix A. In particular, we introduce a new definition
 217 of kernel preserving embeddings; either via kernel-based distances (diffusion maps, PHATE) or via
 218 similarities (e.g. t-SNE, UMAP).

219 **Diffusion maps with the heat kernel** Diffusion maps [5] define an embedding with the first k
 220 eigenvectors $(\phi_i)_i$ of \mathbf{P} , while Laplacian eigenmaps [1] uses the eigenvectors $(\psi_i)_i$ of \mathbf{L} . In the
 221 following, we recall the links between the two methods, and show that a rescaled Laplacian eigenmaps
 222 preserves the diffusion distance with the heat kernel \mathbf{H}_t .

223 **Lemma 5.1.** *Rescaling the Laplacian eigenmaps embedding with $x_i \mapsto (e^{-2t\lambda_1}\psi_{1,i}, \dots, e^{-2t\lambda_k}\psi_{k,i})$*
 224 *preserves the diffusion distance $DM_{\mathbf{H}_t}$.*

225 **Relation to PHATE** The potential distance in PHATE (Sec. 3) is defined by comparing the transition
 226 probabilities of two t -steps random walks initialized from different vertices. The transition matrix \mathbf{P}^t
 227 mimics the heat propagation on a graph. The heat-geodesic dissimilarity provides a new interpretation
 228 of PHATE. In the following proposition, we show how the heat-geodesic relates to the PHATE
 229 potential distance with a linear combination of t -steps random walks.

230 **Proposition 5.2.** *The PHATE potential distance with the heat kernel $PH_{\mathbf{H}_t}$ can be expressed in*
 231 *terms of the heat-geodesic dissimilarity with $\sigma = 0$*

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

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

234 *Proof.* We present a simplified version of the proof, more details are available in Appendix A. For
 235 $\sigma = 0$, we have $d_t(x_i, x_j) = -4t \log(\mathbf{H}_t)_{ij}$, the relation between the PHATE potential and the
 236 heat-geodesic follows from the definition

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

237 Using the heat kernel \mathbf{H}_t with the random walk Laplacian $\mathbf{L}_{rw} = \mathbf{Q}^{-1}\mathbf{L} = \mathbf{I}_n - \mathbf{Q}^{-1}\mathbf{W}$ corresponds
 238 to a multiscale random walk kernel. We can write $\mathbf{L}_{rw} = \mathbf{S}\mathbf{\Lambda}\mathbf{S}^{-1}$, where $\mathbf{S} := \mathbf{Q}^{-1/2}\mathbf{\Psi}$. Since
 239 $\mathbf{P} = \mathbf{I}_n - \mathbf{R}_{rw}$, we have $\mathbf{P}^t = \mathbf{S}(\mathbf{I}_n - \mathbf{\Lambda})^t\mathbf{S}^{-1}$. Interestingly, we can relate the eigenvalues of \mathbf{H}_t
 240 and \mathbf{P} with the Poisson distribution. The probability mass function of a Poisson distribution with

241 mean t is given by $m_t(k) := t^k e^{-t}/k!$. For $t \geq 0$, we have $e^{-t(1-\mu)} = \sum_{k \geq 0} m_t(k) \mu^k$. With this
 242 relationship, we can express \mathbf{H}_t as a linear combination of \mathbf{P}^t weighted by the Poisson distribution.
 243 Indeed, substituting $\lambda = 1 - \mu$ in yields

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

244

□

245 *Remark 5.3.* In the previous proposition, the same argument holds for the symmetric Laplacian and
 246 the affinity matrix $\mathbf{A} := \mathbf{Q}^{-1/2} \mathbf{W} \mathbf{Q}^{-1/2}$ used in other methods such as diffusion maps [5]. This is
 247 valid since we can write $\mathbf{L}_{sym} = \mathbf{Q}^{-1/2} \Psi \Lambda \Psi^T \mathbf{Q}^{-1/2}$, and $\mathbf{A} = \mathbf{I}_n - \mathbf{L}_{sym}$.

248 *Remark 5.4.* This proposition shows that, as the denoising parameter $\rho \rightarrow 1$, Heat Geodesic
 249 Embedding interpolates to the PHATE embeddings with a weighted kernel $\sum_{k=0}^{\infty} m_t(k) \mathbf{P}^k$.

250 **Relation to SNE** The heat-geodesic method also relates to the Stochastic Neighbor Embedding
 251 (SNE) [10], and its variation using the Student distribution t-SNE [17]. In SNE, the similarity between
 252 points is encoded via transition probabilities p_{ij} . The objective is to learn an affinity measure q , that
 253 usually depends on the embedding distances $\|y_i - y_j\|$, such that it minimizes $D_{\text{KL}}(p||q)$. Intuitively,
 254 points that have a strong affinity in the ambient space, should also have a strong affinity in the
 255 embedded space. Even though the heat-geodesic minimization is directly on the embedding distances,
 256 we can show an equivalent with SNE. In Appendix A, we provide additional comparisons between
 257 SNE and our method.

258 **Proposition 5.5.** *The Heat-Geodesic embedding with squared distances minimization weighted by*
 259 *the heat kernel is equivalent to SNE with the heat kernel affinity in the ambient space, and a Gaussian*
 260 *kernel in the embedded space $q_{ij} = \exp(-\|y_i - y_j\|^2/t)$.*

261 6 Results

262 In this section, we show the versatility of our method, showcasing its performance in terms of
 263 clustering and preserving the structure of continuous manifolds. We compare the performance of
 264 Heat Geodesic Embedding with multiple state-of-the-art baselines on synthetic datasets and real-
 265 world datasets. For all models, we perform sample splitting with a 50/50 validation-test split. The
 266 validation and test sets each consists of 5 repetitions with different random initializations. The
 267 hyper-parameters are selected according to the performance on the validation set. We always report
 268 the results on the test set, along with the standard deviations computed over the five repetitions. We
 269 use the following methods in our experiments: our *Heat Geodesic Embedding*, *diffusion maps* [5],
 270 *PHATE* [21], *shortest-path* which estimates the geodesic distance by computing the shortest path
 271 between two nodes in a graph built on the point clouds, *t-SNE* [29], and *UMAP* [18]. Details about
 272 each of these methods, and results for different parameters (graph type, heat approximation, etc.) are
 273 given in Appendix C.

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.

Method	Swiss roll		Tree	
	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)	0.702 ± 0.086	0.700 ± 0.073	0.822 ± 0.008	0.807 ± 0.016

274 6.1 Distance matrix comparison

275 We start by evaluating the ability of the different methods to recover the ground truth distance matrix
 276 of a point cloud. For this task, we use point clouds from the Swiss roll and Tree datasets, for which
 277 the ground truth geodesic distance is known. The Swiss roll dataset consists of data points sampled on
 278 a smooth manifold (see Fig. 1). The Tree dataset is created by connecting multiple high-dimensional
 279 Brownian motions in a tree-shape structure. In Fig. 1, we present embeddings of both datasets. Our

280 method recovers the underlying geometry, while other methods create artificial clusters or have too
 281 much denoising. Because we aim at a faithful relative distance between data points, we compare the
 282 methods according to the Pearson and Spearman correlations of the estimated distance matrices with
 283 respect to ground truth. Results are displayed in Tab. 1. We observe that Heat Geodesic Embedding
 284 typically outperforms previous methods in terms of the correlation with the ground truth distance
 285 matrix, confirming the theoretical guarantees provided in Sec. 4 & 2. Additional results with different
 286 noise levels and ambient dimensions are available in Appendix C.

287 **Optimal diffusion time** In Section 4, we described a heuristic
 288 to automatically choose the diffusion time based on the entropy
 289 of H_t . In Fig. 2, we show that the knee-point of $t \mapsto H(H_t)$,
 290 corresponds to a high correlation with the ground distance, while
 291 yielding a low approximation error of the distance matrix (measured by the Frobenius norm of the difference between D and the ground truth).

294 6.2 Preservation of the inherent data structure

295 A crucial evaluation criteria of manifold learning methods is
 296 the ability to capture the inherent structure of the data. For
 297 instance, clusters in the data should be visible in the resulting
 298 low dimensional representation. Similarly, when the dataset
 299 consists of samples taken at different time points, one expects
 300 to be able to characterize this temporal evolution in the low
 301 dimensional embedding [21]. We thus compare the different
 302 embedding methods according to their ability to retain clusters
 303 and temporal evolution of the data.

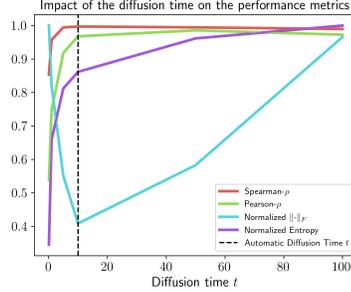


Figure 2: Evolution of the correlation between estimated and ground truth distance matrices in function of the diffusion time t .

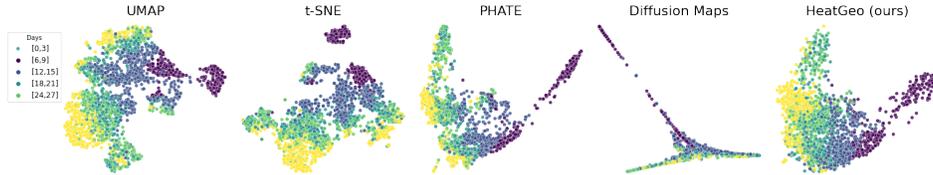


Figure 3: Embeddings of 2000 differentiating cells from embryoid body [21] over 28 days. UMAP and t-SNE do not capture the continuous manifold representing the cells’ evolution.

304 **Identifying clusters.** We use the PBMC dataset, the Swiss roll, and the Tree dataset. The PBMC
 305 dataset consists of single-cell gene expressions from 3000 individual peripheral blood mononuclear
 306 cells. Cells are naturally clustered by their cell type. For the Tree dataset, we use the branches as
 307 clusters. For the Swiss roll dataset, we sample data points on the manifold according to a mixture of
 308 Gaussians and use the mixture component as the ground truth cluster labels. For each method, we
 309 run k-means on the two-dimensional embedding and compare the resulting cluster assignments with
 310 ground truth. Tab. 10 reports the results in terms of homogeneity and adjusted mutual information (aMI). Heat Geodesic Embedding is competitive with PHATE and outperforms t-SNE and UMAP on all metrics. Yet, we show in Appendix C that all methods tend to perform equally well when the noise level increases. In Fig. 4, we present the PBMC embeddings of PHATE and HeatGeo, showing that HeatGeo interpolates to PHATE for $\rho \rightarrow 1$.

Table 2: 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).

Method	Swiss roll		Tree		PBMC	
	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	0.798 ± 0.012	0.785 ± 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)	0.820 ± 0.008	0.740 ± 0.018	0.784 ± 0.051	0.786 ± 0.051	0.734 ± 0.009	0.768 ± 0.017

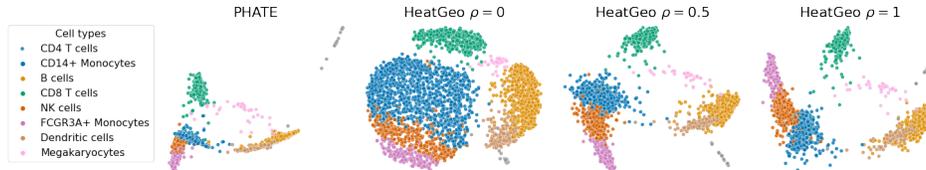


Figure 4: Embeddings on PBMC using the triplet distance with the heat-geodesic for different regularization parameter ρ .

315 **Temporal data representation.** For this task, we aim at representing data points from population
 316 observed at consecutive points in time. We use single cell gene expression datasets collected across
 317 different time points, including the Embryoid Body (EB), IPSC [21], and two from the 2022 NeurIPS
 318 multimodal single-cell integration challenge (Cite & Multi). To quantitatively evaluate the quality
 319 of the continuous embeddings, we first embed the entire dataset and obfuscate all samples from a
 320 particular time point (*e.g.*, $t = 2$). We then estimate the distribution of the missing time point by
 321 using displacement interpolation [32] between the adjacent time points (*e.g.*, $t = 1$ and $t = 3$). We
 322 report the Earth Mover Distance (EMD) between the predicted distribution and true distribution.
 323 A low EMD suggests that the obfuscated embeddings are naturally located between the previous
 324 and later time points, and that the generated embedding captures the temporal evolution of the data
 325 adequately. Results are presented in Tab. 3. Heat Geodesic Embedding outperforms other methods
 326 on the EB, Multi, and IPSC datasets and is competitive with other approaches on Cite. We show a
 327 graphical depiction of the different embeddings for the embryoid (EB) dataset in Fig. 3.

Table 3: 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

328 7 Conclusion and Limitations

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

339 Despite the strong theoretical and empirical properties, our work presents some limitations. For
 340 instance, our method is based on a similarity measure, which is a relaxation of a distance metric.
 341 Additionally, the Harnack equation suggests that our parameters for the volume correction could be
 342 tuned depending on the underlying manifold. We envision that further analysis of this regularization
 343 is a fruitful direction for future work.

344 References

- 345 [1] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data
346 representation. *Neural computation*, 15(6):1373–1396, 2003.
- 347 [2] Jan Niklas Böhm, Philipp Berens, and Dmitry Kobak. Attraction-repulsion spectrum in neighbor
348 embeddings. *Journal of Machine Learning Research*, 23(95):1–32, 2022.
- 349 [3] Erica L Busch, Jessie Huang, Andrew Benz, Tom Wallenstein, Guillaume Lajoie, Guy Wolf,
350 Smita Krishnaswamy, and Nicholas B Turk-Browne. Multi-view manifold learning of human
351 brain-state trajectories. *Nature Computational Science*, pages 1–14, 2023.
- 352 [4] J Douglas Carroll and Phipps Arabie. Multidimensional scaling. *Measurement, judgment and*
353 *decision making*, pages 179–250, 1998.
- 354 [5] Ronald R. Coifman and Stéphane Lafon. Diffusion maps. *Applied and Computational Harmonic*
355 *Analysis*, 21(1):5–30, 2006.
- 356 [6] Keenan Crane, Clarisse Weischedel, and Max Wardetzky. Geodesics in heat: A new approach
357 to computing distance based on heat flow. *ACM Transactions on Graphics (TOG)*, 32(5):1–11,
358 2013.
- 359 [7] Michaël Defferrard, Lionel Martin, Rodrigo Pena, and Nathanaël Perraudin. Pygsp: Graph
360 signal processing in python.
- 361 [8] Laleh Haghverdi, Maren Büttner, F. Alexander Wolf, Florian Buettner, and Fabian J. Theis.
362 Diffusion pseudotime robustly reconstructs lineage branching. *Nature Methods*, 13(10):845–848,
363 2016.
- 364 [9] Matthieu Heitz, Nicolas Bonneel, David Coeurjolly, Marco Cuturi, and Gabriel Peyré. Ground
365 metric learning on graphs. *Journal of Mathematical Imaging and Vision*, 63:89–107, 2021.
- 366 [10] Geoffrey E Hinton and Sam Roweis. Stochastic neighbor embedding. *Advances in neural*
367 *information processing systems*, 15, 2002.
- 368 [11] Michael C Hout, Megan H Papesh, and Stephen D Goldinger. Multidimensional scaling. *Wiley*
369 *Interdisciplinary Reviews: Cognitive Science*, 4(1):93–103, 2013.
- 370 [12] Shih-Gu Huang, Ilwoo Lyu, Anqi Qiu, and Moo K Chung. Fast polynomial approximation of
371 heat kernel convolution on manifolds and its application to brain sulcal and gyral graph pattern
372 analysis. *IEEE transactions on medical imaging*, 39(6):2201–2212, 2020.
- 373 [13] Guillaume Hugué, Alexander Tong, Bastian Rieck, Jessie Huang, Manik Kuchroo, Matthew
374 Hirn, Guy Wolf, and Smita Krishnaswamy. Time-inhomogeneous diffusion geometry and
375 topology. *arXiv preprint arXiv:2203.14860*, 2022.
- 376 [14] Guillaume Hugué, Alexander Tong, María Ramos Zapatero, Guy Wolf, and Smita Krish-
377 naswamy. Geodesic Sinkhorn: optimal transport for high-dimensional datasets. *arXiv preprint*
378 *arXiv:2211.00805*, 2022.
- 379 [15] Joseph B Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric
380 hypothesis. *Psychometrika*, 29(1):1–27, 1964.
- 381 [16] Peter Li and Shing Tung Yau. On the parabolic kernel of the Schrödinger operator. *Acta*
382 *Mathematica*, 156(none):153 – 201, 1986.
- 383 [17] George C. Linderman and Stefan Steinerberger. Clustering with t-SNE, provably.
384 *arXiv:1706.02582 [cs, stat]*, 2017.
- 385 [18] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation
386 and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.
- 387 [19] Stanislav A Molchanov. Diffusion processes and riemannian geometry. *Russian Mathematical*
388 *Surveys*, 30(1):1, 1975.

- 389 [20] Kevin R. Moon, Jay S. Stanley, Daniel Burkhardt, David van Dijk, Guy Wolf, and Smita
390 Krishnaswamy. Manifold learning-based methods for analyzing single-cell RNA-sequencing
391 data. *Current Opinion in Systems Biology*, 7:36–46, 2018.
- 392 [21] Kevin R. Moon, David van Dijk, Zheng Wang, Scott Gigante, Daniel B. Burkhardt, William S.
393 Chen, Kristina Yim, Antonia van den Elzen, Matthew J. Hirn, Ronald R. Coifman, Natalia B.
394 Ivanova, Guy Wolf, and Smita Krishnaswamy. Visualizing structure and transitions in high-
395 dimensional biological data. *Nat Biotechnol*, 37(12):1482–1492, 2019.
- 396 [22] Adam Nowak, Peter Sjögren, and Tomasz Z Szarek. Sharp estimates of the spherical heat kernel.
397 *Journal de Mathématiques Pures et Appliquées*, 129:23–33, 2019.
- 398 [23] Laurent Saloff-Coste. The heat kernel and its estimates. *Probabilistic approach to geometry*,
399 57:405–436, 2010.
- 400 [24] Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. Finding a "kneedle" in a
401 haystack: Detecting knee points in system behavior. In *2011 31st international conference on*
402 *distributed computing systems workshops*, pages 166–171. IEEE, 2011.
- 403 [25] Nicholas Sharp, Yousuf Soliman, and Keenan Crane. The vector heat method. *ACM Transactions*
404 *on Graphics (TOG)*, 38(3):1–19, 2019.
- 405 [26] Justin Solomon, Fernando De Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher, Andy
406 Nguyen, Tao Du, and Leonidas Guibas. Convolutional wasserstein distances: Efficient optimal
407 transportation on geometric domains. *ACM Transactions on Graphics (ToG)*, 34(4):1–11, 2015.
- 408 [27] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A concise and provably informative multi-
409 scale signature based on heat diffusion. In *Computer graphics forum*, volume 28, pages
410 1383–1392. Wiley Online Library, 2009.
- 411 [28] Yoshio Takane, Forrest W Young, and Jan De Leeuw. Nonmetric individual differences
412 multidimensional scaling: An alternating least squares method with optimal scaling features.
413 *Psychometrika*, 42:7–67, 1977.
- 414 [29] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine*
415 *learning research*, 9(11), 2008.
- 416 [30] David Van Dijk, Roshan Sharma, Juozas Nainys, Kristina Yim, Pooja Kathail, Ambrose J
417 Carr, Cassandra Burdziak, Kevin R Moon, Christine L Chaffer, Diwakar Pattabiraman, et al.
418 Recovering gene interactions from single-cell data using data diffusion. *Cell*, 174(3):716–729,
419 2018.
- 420 [31] Sathamangalam R Srinivasa Varadhan. On the behavior of the fundamental solution of the heat
421 equation with variable coefficients. *Communications on Pure and Applied Mathematics*, 20(2):
422 431–455, 1967.
- 423 [32] Cédric Villani and Cédric Villani. Displacement interpolation. *Optimal Transport: Old and*
424 *New*, pages 113–162, 2009.
- 425 [33] F Alexander Wolf, Philipp Angerer, and Fabian J Theis. Scanpy: large-scale single-cell gene
426 expression data analysis. *Genome biology*, 19:1–5, 2018.
- 427 [34] Jonathan X Zheng, Samraat Pawar, and Dan FM Goodman. Graph drawing by stochastic
428 gradient descent. *IEEE transactions on visualization and computer graphics*, 25(9):2738–2748,
429 2018.

431 **Appendix****Table of Contents**

434	A Theory and algorithm details	13
435	A.1 Kernel preserving embeddings	13
436	A.2 Proofs	14
437	A.3 Algorithm details	15
438	B Experiments and datasets details	16
439	B.1 Datasets	16
440	B.2 Evaluation Metrics	17
441	B.3 Hyperparameters	18
442	B.4 Hardware	18
443	C Additional results	19
444	C.1 HeatGeo weighted	19
445	C.2 Truncated distance	19
446	C.3 Harnack inequality	20
447	C.4 Quantitative results	22
448	C.5 Impact of the different hyperparameters	22
449	C.6 Graph construction	22

453 A Theory and algorithm details

454 A.1 Kernel preserving embeddings

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

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

$$L := f(T^x(H_t^x), T^y(H_t^y)), \quad (6)$$

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

469 *Example 1.* We formulate MDS as a kernel feature-preserving embedding. Suppose we want to
 470 preserve the Euclidean distance, we have $H_t^x(x_i, x_j) = \|x_i - x_j\|_2$, $H_t^y(y_i, y_j) = \|y_i - y_j\|_2$,
 471 $f(a, b) = \|a - b\|_2$, and $T^x = T^y = I$.

472 In the following, we present popular dimensionality reduction methods that are kernel features
 473 preserving embeddings. With this definition, we can distinguish between methods that preserve a
 474 kernel via affinities or distances. For the methods considered in this work, H_t^x is an affinity kernel,
 475 but its construction varies from one method to another. In PHATE and Diffusion maps, H_t^x is a
 476 random walk P , while in Heat Geodesic Embedding we use the heat kernel H_t . t-SNE defines H_t^x
 477 as a symmetrized random walk matrix from a Gaussian kernel, while UMAP uses an unnormalized
 478 version. Methods such as PHATE and diffusion maps define a new distance matrix from a kernel in
 479 the ambient space and preserve these distances in the embedded space. Other methods like t-SNE
 480 and UMAP define similarities from a kernel and aim to preserve these similarities in the ambient
 481 space and embedded space via an entropy-based loss. We note the Kullback–Leibler divergence
 482 $D_{\text{KL}}(a, b) = \sum_{ij} a_{ij} \log[a_{ij}/b_{ij}]$.

483 **Proposition A.2.** *The embeddings methods HeatGeo, PHATE, Diffusion Maps, SNE, t-SNE, and*
 484 *UMAP are kernel feature-preserving embeddings.*

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

487 We start with the distance preserving embeddings; HeatGeo, PHATE, and Diffusion Maps. For these
 488 methods, the kernel in the embed space is simply $H_t^y(y_i, y_j) = \|y_i - y_j\|_2$, without transformation,
 489 i.e. $T^y = I$. Since they preserve a distance, the loss is $f(T^x(H_t^x), T^y(H_t^y)) = \|H_t^x - H_t^y\|_2$.

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

492 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
 493 Diffusion Maps, the diffusion distance is $(T^x(H_t^x))_{ij} = \| (H_t^x)_i - (H_t^x)_j \|_2$.

494 SNE, t-SNE, and UMAP preserve affinities from a kernel. For these three methods, the loss is
 495 a divergence between distributions, namely $f = D_{\text{KL}}$. They vary by defining different affinity
 496 kernel and transformation in the embedded space. SNE uses the unnormalized kernel $H_t^y(y_i, y_j) =$
 497 $\exp(-(1/t)\|y_i - y_j\|_2^2)$, with $T^x = T^y = I$. Whereas, t-SNE uses $(H_1^y)_{ij} = (1 + \|y_i - y_j\|_2^2)^{-1}$,
 498 and $T^x = T^y = I$. UMAP define a pointwise transformation in the embedded space with $(H_1^y)_{ij} =$
 499 $(1 + \|y_i - y_j\|_2^2)^{-1}$, $(T^x(H_t^x))_{ij} = (H_1^y)_{ij}/(1 - (H_1^y)_{ij})$, and $T^x = I$.

500 We summarize the choice of kernels and functions in Tab. 4 □

Table 4: 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	$(1 + \ y_i - y_j\ _2^2)^{-1}$	H_t^x	H_t^y	D_{KL}
UMAP	$(1 + \ y_i - y_j\ _2^2)^{-1}$	H_t^x	$\frac{(H_1^y)_{ij}}{(1 - (H_1^y)_{ij})}$	D_{KL}

501 **A.2 Proofs**

502 **Proposition 4.3.** Denote the perturbed triplet distance by $\widetilde{D}_T(x_i, x_j) = \|\widetilde{d}_t(x_i, \cdot) - \widetilde{d}_t(x_j, \cdot)\|_2$
503 where $\widetilde{d}_t(x_i, x_j) := d_t(x_i, x_j) + \epsilon$ and $\widetilde{d}_t(x_i, x_k) := d_t(x_i, x_k)$ for $k \neq j$. Then the triplet distance
504 D_T is robust to perturbations, i.e., for all $\epsilon > 0$,

$$\left(\frac{\widetilde{D}_T(x_i, x_j)}{D_T(x_i, x_j)}\right)^2 \leq \left(\frac{d_t(x_i, x_j) + \epsilon}{d_t(x_i, x_j)}\right)^2.$$

505 *Proof of Proposition 4.3.* 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}_T ,
506 we have
507

$$\frac{\widetilde{D}_T(x_i, x_j)^2}{D_T(x_i, x_j)^2} = \frac{\sum_{k \neq i, j} (d_t(x_i, x_k) - d_t(x_j, x_k))^2 + 2(d_t(x_i, x_j) + \epsilon)^2}{D_T(x_i, x_j)^2} = 1 + \frac{4\epsilon d(x_i, x_j) + 2\epsilon^2}{D_T(x_i, x_j)^2},$$

508 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

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

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

511 □

512 **Lemma 5.1.** Rescaling the Laplacian eigenmaps embedding with $x_i \mapsto (e^{-2t\lambda_1}\psi_{1,i}, \dots, e^{-2t\lambda_k}\psi_{k,i})$
513 preserves the diffusion distance $DM_{\mathbf{H}_t}$.

514 *Proof of Lemma 5.1.* Since the eigendecomposition of \mathbf{H}_t form an orthonormal basis of \mathbb{R}^n , and
515 since its first eigenvector is constant, we can write the diffusion distance $\|\delta_i \mathbf{H}_t - \delta_j \mathbf{H}_t\|_2^2 =$
516 $\sum_{k \geq 0} e^{-2t\lambda_k} (\psi_{ki} - \psi_{kj})^2 = \sum_{k \geq 1} e^{-2t\lambda_k} (\psi_{ki} - \psi_{kj})^2$. In particular, this defines the k dimensional
517 embedding $x \mapsto (e^{-t\lambda_1}\psi_1(x), \dots, e^{-t\lambda_k}\psi_k(x))$. □

518 **Proposition 5.2.** The PHATE potential distance with the heat kernel $PH_{\mathbf{H}_t}$ can be expressed in
519 terms of the heat-geodesic dissimilarity with $\sigma = 0$

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

520 and it is equivalent to a multiscale random walk distance with kernel $\sum_{k > 0} m_t(k) \mathbf{P}^k$, where
521 $m_t(k) := t^k e^{-t/k}!$.

522 *Proof of Proposition 5.2.* For $\sigma = 0$, we have $d_t(x_i, x_j) = -4t \log(\mathbf{H}_t)_{ij}$, the relation between the
 523 PHATE potential and the heat-geodesic follows from the definition

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

524 Using the heat kernel \mathbf{H}_t with the random walk Laplacian $\mathbf{L}_{rw} = \mathbf{Q}^{-1}\mathbf{L} = \mathbf{I}_n - \mathbf{Q}^{-1}\mathbf{W}$ corresponds
 525 to a multiscale random walk kernel. Recall that we can write \mathbf{L}_{rw} in terms of the symmetric
 526 Laplacian $\mathbf{L}_{rw} = \mathbf{Q}^{-1/2}\mathbf{L}_s\mathbf{Q}^{1/2}$, meaning that the two matrices are similar, hence admit the same
 527 eigenvalues Λ . We also know that \mathbf{L}_s is diagonalizable, since we can write $\mathbf{L}_s = \mathbf{Q}^{-1/2}\mathbf{L}\mathbf{Q}^{-1/2} =$
 528 $\mathbf{Q}^{-1/2}\Psi\Lambda\Psi^T\mathbf{Q}^{-1/2}$. In particular, we have $\mathbf{L}_{rw} = \mathbf{S}\Lambda\mathbf{S}^{-1}$, where $\mathbf{S} := \mathbf{Q}^{-1/2}\Psi$. The random
 529 walk matrix can be written as $\mathbf{P} = \mathbf{I}_n - \mathbf{R}_{rw}$, hence its eigenvalues are $(\mathbf{I}_n - \Lambda)$, and we can write
 530 $\mathbf{P}^t = \mathbf{S}(\mathbf{I}_n - \Lambda)^t\mathbf{S}^{-1}$. Similarly, the heat kernel with the random walk Laplacian can be written
 531 as $\mathbf{H}_t = \mathbf{S}e^{-t\Lambda}\mathbf{S}^{-1}$. Interestingly, we can relate the eigenvalues of \mathbf{H}_t and \mathbf{P} with the Poisson
 532 distribution. Note the probability mass function of a Poisson as $m_t(k) := t^k e^{-t}/k!$, for $t \geq 0$, we
 533 have

$$e^{-t(1-\mu)} = e^{-t} \sum_{k \geq 0} \frac{(t\mu)^k}{k!} = \sum_{k \geq 0} m_t(k) \mu^k. \quad (7)$$

534 We note that (7) is the probability generating function of a Poisson distribution with parameter t , i.e.
 535 $\mathbb{E}[\mu^X]$, where $X \sim \text{Poisson}(t)$. With this relationship, we can express \mathbf{H}_t as a linear combination of
 536 \mathbf{P}^t weighted by the Poisson distribution. Indeed, substituting $\lambda = 1 - \mu$ in (7) links the eigenvalues
 537 of \mathbf{H}_t and \mathbf{P} . We write the heat kernel as a linear combination of random walks weighted by the
 538 Poisson distribution, we have

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

539 □

540 **Proposition 5.5.** *The Heat-Geodesic embedding with squared distances minimization weighted by*
 541 *the heat kernel is equivalent to SNE with the heat kernel affinity in the ambient space, and a Gaussian*
 542 *kernel in the embedded space $q_{ij} = \exp(-\|y_i - y_j\|^2/t)$.*

543 *Proof of Proposition 5.5.* The squared MDS weighted by the heat kernel corresponds to

$$\begin{aligned} \sum_{ij} h_t(x_i, x_j) (d_{ij}^2 - \|y_i - y_j\|^2)^2 &= \sum_{ij} h_t(x_i, x_j) (-t \log h_t(x_i, x_j) - \|y_i - y_j\|^2)^2 \\ &= \sum_{ij} h_t(x_i, x_j) t^2 (\log h_t(x_i, x_j) - \log \exp(-\|y_i - y_j\|^2/t))^2. \end{aligned}$$

544 If there exists an embedding that attain a zero loss, then it is the same as
 545 $\sum_{ij} h_t(x_i, x_j) (\log h_t(x_i, x_j) - \log \exp(-\|y_i - y_j\|^2/t)) = D_{\text{KL}}(h_t \| q)$. □

546 A.3 Algorithm details

547 We present a detailed version of the Heat Geodesic Embedding algorithm in Alg.2.

548 For the knee-point detection we use the Kneede algorithm [24]. It identifies a knee-point as a point
 549 where the curvature decreases maximally between points (using finite differences). We summarize
 550 the four main steps of the algorithm for a function $f(x)$, and we refer to [24] for additional details.

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

Algorithm 2 Heat Geodesic Embedding

```
1: Input:  $N \times d$  dataset matrix  $\mathbf{X}$ , denoising parameter  $\rho \in [0, 1]$ , Harnack regularization  $\sigma > 0$ ,  
   output dimension  $k$ .  
2: Returns:  $N \times e$  embedding matrix  $\mathbf{E}$ .  
3:  $\triangleright$  1. Calculate Heat Operator  $\mathbf{H}_t$  ◁  
4: if  $t$  is "auto" then  
5:    $t \leftarrow \text{Kneedle}\{H(\mathbf{H}_t)\}_t$   $\triangleright$  Knee detection e.g. [24]  
6:    $\mathbf{W} \leftarrow \text{kernel}(\mathbf{X})$   
7:    $\mathbf{L} \leftarrow \mathbf{Q} - \mathbf{W}$   
8:   if Exact then  
9:      $\mathbf{H}_t \leftarrow \Psi e^{-t\Lambda} \Psi^T$   
10:  else  
11:     $\mathbf{H}_t \leftarrow p_K(\mathbf{L}, t)$   
12:  $\triangleright$  2. Calculate Pairwise Distances  $\mathbf{D}$  ◁  
13:  $\mathbf{D} \leftarrow -4t \log \mathbf{H}_t$   $\triangleright$  log is applied elementwise  
14:  $\mathbf{D} \leftarrow (1 - \rho)\mathbf{D} + \rho D_T$   $\triangleright$  Triplet interpolation step  
15: Return  $\mathbf{E} \leftarrow \text{MetricMDS}(\mathbf{D}, \|\cdot\|_2, k)$ 
```

556 B Experiments and datasets details

557 Our experiments compare our approach with multiple state-of-the-art baselines for synthetic datasets
558 (for which the true geodesic distance is known) and real-world datasets. For all models, we perform
559 sample splitting with a 50/50 validation-test split. The validation and test sets each consist of 5
560 repetitions with different random initializations. The hyper-parameters are selected according to
561 the performance on the validation set. We always report the results on the test set, along with the
562 standard deviations computed over the five repetitions. We use the following state-of-the-art methods
563 in our experiments: our Heat Geodesic Embedding, *diffusion maps*[5], *PHATE* [21], *Heat-PHATE* (a
564 variation of PHATE using the Heat Kernel), *Rand-Geo* (a variation of Heat Geodesic Embedding
565 where we use the random walk kernel), *Shortest-path* which estimates the geodesic distance by
566 computing the shortest path between two nodes in a graph built on the point clouds, *t-SNE*[29], and
567 *UMAP*[18].

568 B.1 Datasets

569 We consider two synthetic datasets, the well known Swiss roll and the tree datasets. The exact
570 geodesic distance can be computed for these datasets. We additionally consider real-world datasets:
571 PBMC, IPSC [21], EB [21], and two from the from the 2022 NeurIPS multimodal single-cell
572 integration challenge².

573 B.1.1 Swiss Roll

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

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

576 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 $W = 5$.
577 We use two sampling mechanisms for generating the data points : uniformly and clustered. In the
578 first, we sample points uniformly at random in the $[T_0, T_1] \times [0, W]$ plane. In the second, we sample
579 according to a mixture of isotropic multivariate Gaussian distributions in the same plane with equal
580 weights, means $[(7, W/2), (12, W/2)]$, and standard deviations $[1, 1]$. In the clustered case, data
581 samples are given a label y according to the Gaussian mixture component from which they were
582 sampled.

²<https://www.kaggle.com/competitions/open-problems-multimodal/>

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

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

588 B.1.2 Tree

589 The tree dataset is created by generating K branches from a D -dimensional Brownian motion that
590 are eventually glued together. Each branch is sampled from a multidimensional Brownian motion
591 $d\mathbf{X}_k = 2d\mathbf{W}(t)$ at times $t = 0, 1, 2, \dots, L - 1$ for $k \in [K]$. The first branch is taken as the main
592 branch and the remaining branches are glued to the main branch by setting $X_k = X_k + X_0[i_k]$ where
593 i_k is a random index of the main branch vector. The total number of samples is thus $L \cdot K$

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

596 B.2 Evaluation Metrics

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

601 B.2.1 Distance matrix evaluation

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

607 **Pearson ρ** We compute the average Pearson correlation between the rows of the distance matrices,
608 $\frac{1}{N} \sum_{i=1}^N r_{D_i, \hat{D}_i}$, where $r_{x,y}$ is the Pearson correlation coefficient between vectors x and y . D_i stands
609 for the i -th row of D .

610 **Spearman ρ** We compute the average Spearman correlation between the rows of the distance
611 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
612 y . D_i stands for the i -th row of D .

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

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

615 B.2.2 Embedding evaluation

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

623 **Clustering** We evaluate the ability of Heat Geodesic Embedding to create meaningful embeddings
624 when clusters are present in the data. To this end, we run a k-means clustering on the two dimensional
625 embeddings obtained with each method and compare them against the ground truth labels. For the

626 Tree dataset, we use the branches as clusters. For the Swiss roll dataset, we sample data points on the
 627 manifold according to a mixture of Gaussians and use the mixture component as the ground truth
 628 cluster label.

629 **Interpolation** To quantitatively evaluate the quality of the continuous embeddings, we first embed
 630 the entire dataset and obfuscate all samples from a particular time point (*e.g.*, $t = 2$). We then
 631 estimate the distribution of the missing time point by using displacement interpolation [32] between
 632 the adjacent time points (*e.g.*, $t = 1$ and $t = 3$). We report the Earth Mover Distance (EMD) between
 633 the predicted distribution and true distribution. A low EMD suggests that the obfuscated embeddings
 634 are naturally located between the previous and later time points, and that the generated embedding
 635 captures the temporal evolution of the data adequately.

636 B.3 Hyperparameters

637 In Table 5, 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 exaggeration	Early exaggeration parameter	12

Table 5: Hyperparameters used in our experiments

638 B.4 Hardware

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

641 **C Additional results**

642 **C.1 HeatGeo weighted**

643 Following Sec. 5, we know that weighting the MDS loss by the heat kernel corresponds to a specific
 644 parametrization of SNE, and thus promote the identification of cluster. In Fig. 5, we show the
 645 embeddings of four Gaussian distributions in 10 dimensions (top), and the PBMC dataset (bottom).
 646 The reference embedding is using t-SNE, as it models as it also minimizes the KL between the
 647 ambient and embedded distributions. We see that HeatGeo weighted form cluster that are shaped like
 648 a Gaussian. This is expected as Prop. 5.5, indicates that this is equivalent to minimizing the D_{KL}
 649 between the heat kernel and a Gaussian affinity kernel.

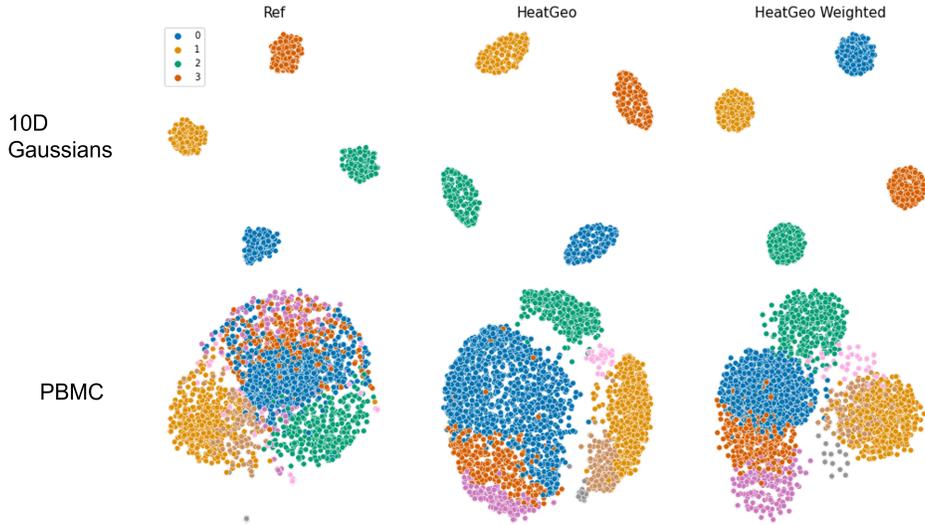


Figure 5: Embeddings of four Gaussian distributions in 10 dimensions (top), and the PBMC dataset (bottom). HeatGeo with weight is equivalent to minimizing the D_{KL} between the heat kernel and a Gaussian affinity kernel, hence produces clusters shaped similar to a Gaussian.

650 **C.2 Truncated distance**

651 In Fig.6, we discretize the interval $[0, 51]$ in 51 nodes, and we compute the heat-geodesic distance
 652 of the midpoint with respect to the other points, effectively approximating the Euclidean distance.
 653 Using Chebyshev polynomials of degree of 20, we see that the impact of the truncation is greater as
 654 the diffusion time increases. The backward Euler methods does not result in a truncated distance.

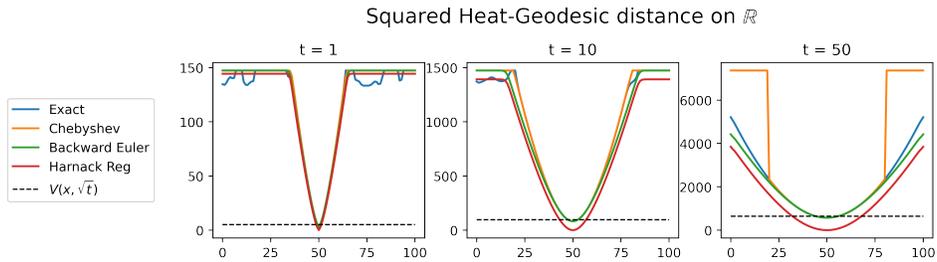


Figure 6: Approximation of the squared Euclidean distance with the Heat-geodesic for the exact computation, Backward Euler approximation, and Chebyshev polynomials. For larger diffusion time, the Chebyshev approximation results in a thresholded distance. The Harnack regularization ensures $d_t(x, x) = 0$.

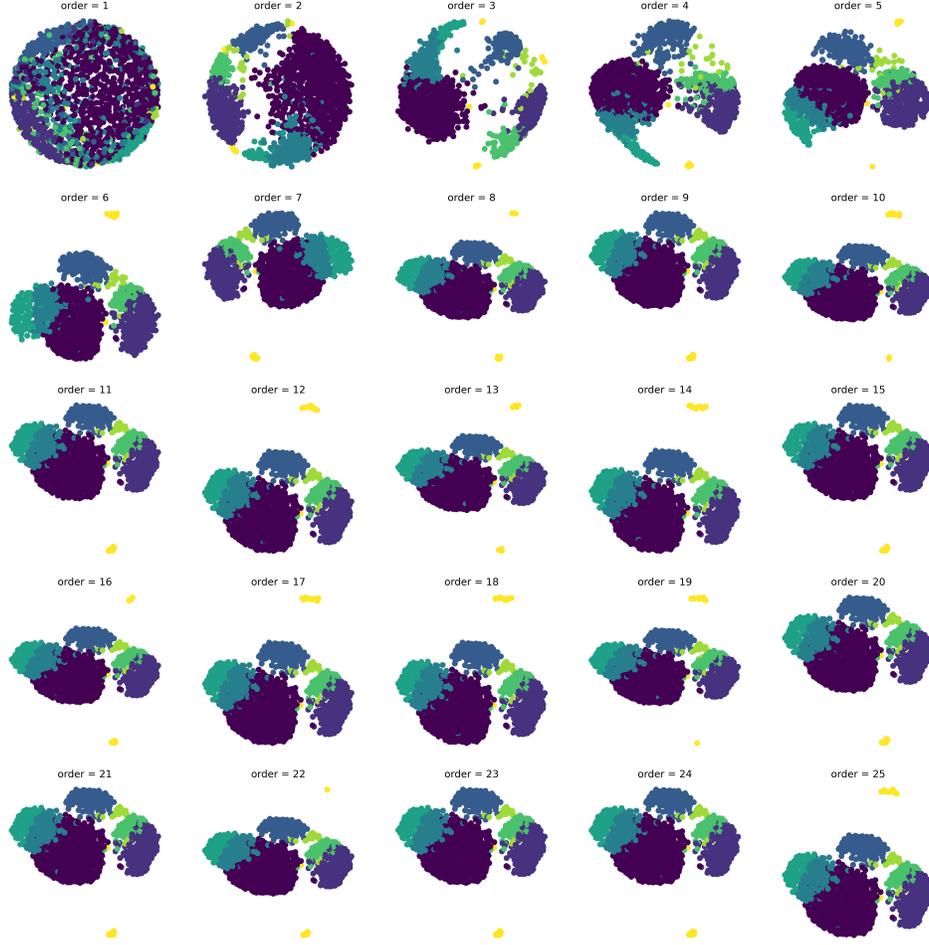


Figure 7: Impact of the Chebyshev approximation order on the embedding of HeatGeo for the PBMC dataset.

655 **C.3 Harnack inequality**

656 For complete Riemannian manifolds that satisfy the parabolic Harnack inequality (PHI) we have
 657 $h_t(x, y) \simeq V^{-1}(x, \sqrt{t}) e^{-d(x,y)^2/t}$ so that $-t \log h_t(x, y) \simeq t \log V(x, \sqrt{t}) + d^2(x, y)$ [23].

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

$$V(x, \sqrt{t}) = h_t(x, x)^{-1} \tag{9}$$

658 We then have,

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

659 **C.3.1 Case studies for specific manifolds**

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

663 Our expression for the Heat Geodesic Embedding-distance is

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

664 As the square-root is monotonic, and $4t \log h_t(0)$ is constant with respect to d , we need to show that
 665 $f(d) = -\log(h_t(d))$ is monotonically increasing.

666 For \mathbb{S}_1 , we have

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

667 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
 668 decreasing, which is the case as for any $d' > d, \forall m \in \mathbb{Z}$, we have

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

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

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

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

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

676 Furthermore, Nowak et al. [22] showed that the heat kernel of the sphere is monotonically decreasing.
 677 The distance inferred from Heat Geodesic Embedding thus preserves ordering of the pair-wise
 678 distances.

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

$$V_{\sqrt{t}} = \frac{4}{3}\pi t^{3/2}$$

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

681 Recalling Harnack inequality,

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

682 With $c_2 = c_4 = 4$, we have

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

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

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

684 we recover the exact geodesic distance.

685 C.4 Quantitative results

686 C.4.1 Distance matrix evaluation

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

689 C.4.2 Distance matrix evaluation via two-dimensional embeddings

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

692 C.4.3 Clustering quality evaluation

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

695 C.5 Impact of the different hyperparameters

696 We investigate the impact of the different hyperparameters on the quality of the embeddings. In
697 Figure 8, we show the embeddings of HeatGeo for different values of diffusion time, number of
698 neighbours, order, and Harnack regularization.

699 In Figures 9, 10, 11, and 12, we show the impact of different hyperparameters on the Pearson
700 correlation between the estimated distance matrix and ground truth distance matrix for different
701 methods on the Swiss roll dataset.

702 C.6 Graph construction

703 We compare the embeddings of the heat-geodesic distance for different graph construction. Through-
704 out the paper we used the graph construction from PHATE [21]. In the following we present additional
705 results depending on the choice of kernel to construct the graph. Specifically, we use a simple nearest
706 neighbor (kNN) graph implemented in [7], the graph from UMAP [18], and the implementation in
707 the package Scanpy [33] for single-cell analysis. In figure, we present the embeddings 2500 points
708 of a tree with five branches in 10 dimensions, where the observations are perturbed with a standard
709 Gaussian noise. All methods used five nearest neighbors and a diffusion time of 20. In Figure 13, we
710 show the evolution of the Pearson correlation between estimated and ground truth distance matrices
711 for the 10-dimensional Swiss roll dataset for various graph constructions. We note that the results are
712 stable across different graph construction strategies.

713

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	1.0 ± 0.0	1.0 ± 0.0	0.0 ± 0.0	0.001 ± 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	0.999 ± 0.0	0.999 ± 0.0	0.001 ± 0.0	0.002 ± 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	0.702 ± 0.086	0.7 ± 0.073	0.01 ± 0.0	0.012 ± 0.0
Swiss roll	1.0	Heat-PHATE	0.623 ± 0.144	0.633 ± 0.114	0.01 ± 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	0.01 ± 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	1.0 ± 0.0	1.0 ± 0.0	0.001 ± 0.0	0.002 ± 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	0.005 ± 0.0	0.007 ± 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	0.999 ± 0.0	0.998 ± 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	0.705 ± 0.065	0.695 ± 0.052	0.011 ± 0.0	0.012 ± 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	0.01 ± 0.0	0.012 ± 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	0.002 ± 0.0	0.003 ± 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	0.999 ± 0.0	0.999 ± 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	0.998 ± 0.0	0.008 ± 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	0.008 ± 0.0	0.009 ± 0.0
Swiss roll very high	0.5	Shortest Path	0.998 ± 0.0	0.998 ± 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	0.466 ± 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	0.516 ± 0.008	0.01 ± 0.0	0.012 ± 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	0.976 ± 0.019	0.977 ± 0.02	0.092 ± 0.011	0.135 ± 0.018
Tree	1.0	Heat-PHATE	0.918 ± 0.032	0.885 ± 0.04	0.03 ± 0.005	0.044 ± 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	0.822 ± 0.008	0.807 ± 0.016	0.1 ± 0.012	0.146 ± 0.019
Tree	5.0	Heat-PHATE	0.765 ± 0.025	0.751 ± 0.023	0.043 ± 0.006	0.08 ± 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	0.59 ± 0.033	0.1 ± 0.012	0.146 ± 0.019
Tree	10.0	Heat-PHATE	0.63 ± 0.018	0.588 ± 0.031	0.046 ± 0.005	0.083 ± 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	0.999 ± 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	0.998 ± 0.001	0.999 ± 0.001	0.009 ± 0.002	0.018 ± 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	0.97 ± 0.01	0.975 ± 0.009	0.253 ± 0.021	0.353 ± 0.026
Tree high	5.0	Heat-PHATE	0.932 ± 0.022	0.919 ± 0.03	0.072 ± 0.004	0.112 ± 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	0.861 ± 0.039	0.87 ± 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	0.159 ± 0.018	0.257 ± 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	1.0 ± 0.0	1.0 ± 0.0	0.018 ± 0.0	0.026 ± 0.0
Swiss roll	0.1	TSNE	0.905 ± 0.005	0.897 ± 0.004	0.006 ± 0.0	0.008 ± 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	0.999 ± 0.0	0.999 ± 0.0	0.018 ± 0.0	0.026 ± 0.0
Swiss roll	0.5	TSNE	0.899 ± 0.01	0.892 ± 0.008	0.006 ± 0.0	0.008 ± 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	0.724 ± 0.151	0.008 ± 0.002	0.014 ± 0.003
Swiss roll	1.0	UMAP	0.727 ± 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	0.239 ± 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	0.012 ± 0.0	0.018 ± 0.001
Swiss roll	5.0	UMAP	0.205 ± 0.013	0.239 ± 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	1.0 ± 0.0	1.0 ± 0.0	0.018 ± 0.0	0.026 ± 0.0
Swiss roll high	0.1	TSNE	0.903 ± 0.004	0.896 ± 0.003	0.006 ± 0.0	0.008 ± 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	0.999 ± 0.0	0.998 ± 0.0	0.018 ± 0.0	0.026 ± 0.0
Swiss roll high	0.5	TSNE	0.905 ± 0.006	0.899 ± 0.003	0.006 ± 0.0	0.008 ± 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	0.008 ± 0.001	0.015 ± 0.003
Swiss roll high	1.0	UMAP	0.893 ± 0.055	0.889 ± 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	0.202 ± 0.01	0.236 ± 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	0.013 ± 0.0	0.019 ± 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	0.999 ± 0.0	0.999 ± 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	0.006 ± 0.0	0.008 ± 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	0.998 ± 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	0.998 ± 0.0	0.997 ± 0.0	0.018 ± 0.0	0.026 ± 0.0
Swiss roll very high	0.5	TSNE	0.917 ± 0.009	0.917 ± 0.007	0.006 ± 0.0	0.008 ± 0.001
Swiss roll very high	0.5	UMAP	0.928 ± 0.01	0.929 ± 0.012	0.012 ± 0.0	0.017 ± 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	0.364 ± 0.008	0.425 ± 0.015	0.018 ± 0.0	0.026 ± 0.0
Swiss roll very high	1.0	Heat-PHATE	0.352 ± 0.022	0.411 ± 0.018	0.018 ± 0.0	0.026 ± 0.0
Swiss roll very high	1.0	PHATE	0.326 ± 0.009	0.388 ± 0.007	0.018 ± 0.0	0.026 ± 0.0
Swiss roll very high	1.0	Rand-Geo	0.357 ± 0.007	0.404 ± 0.005	0.018 ± 0.0	0.026 ± 0.0
Swiss roll very high	1.0	Shortest Path	0.335 ± 0.014	0.39 ± 0.011	0.018 ± 0.0	0.026 ± 0.0
Swiss roll very high	1.0	TSNE	0.515 ± 0.014	0.522 ± 0.01	0.012 ± 0.0	0.016 ± 0.0
Swiss roll very high	1.0	UMAP	0.765 ± 0.059	0.737 ± 0.058	0.015 ± 0.0	0.021 ± 0.0
Swiss roll very high	5.0	Diffusion Map	0.151 ± 0.011	0.161 ± 0.008	0.018 ± 0.0	0.026 ± 0.0
Swiss roll very high	5.0	Heat-PHATE	0.175 ± 0.009	0.208 ± 0.009	0.018 ± 0.0	0.026 ± 0.0
Swiss roll very high	5.0	PHATE	0.181 ± 0.006	0.212 ± 0.006	0.018 ± 0.0	0.026 ± 0.0
Swiss roll very high	5.0	Rand-Geo	0.005 ± 0.002	0.004 ± 0.002	0.018 ± 0.0	0.026 ± 0.0
Swiss roll very high	5.0	Shortest Path	0.145 ± 0.011	0.173 ± 0.011	0.018 ± 0.0	0.026 ± 0.0
Swiss roll very high	5.0	TSNE	0.155 ± 0.008	0.188 ± 0.008	0.015 ± 0.0	0.022 ± 0.001
Swiss roll very high	5.0	UMAP	0.155 ± 0.003	0.183 ± 0.005	0.017 ± 0.0	0.024 ± 0.0

Table 8: Comparison of the estimated distance matrices with the ground truth geodesic distance matrices on the Swiss roll dataset, using a two-dimensional embedding. Best models on average are bolded (not necessarily significant).

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	0.943 ± 0.037	0.94 ± 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	0.082 ± 0.012	0.123 ± 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	0.806 ± 0.019	0.787 ± 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	0.083 ± 0.012	0.128 ± 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	0.58 ± 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	0.615 ± 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	0.086 ± 0.011	0.135 ± 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	0.957 ± 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	0.958 ± 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	0.233 ± 0.021	0.327 ± 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	0.932 ± 0.022	0.928 ± 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	0.237 ± 0.021	0.334 ± 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	0.238 ± 0.02	0.335 ± 0.026
Tree high	1.0	UMAP	0.853 ± 0.051	0.839 ± 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	0.82 ± 0.008	0.668 ± 0.034	0.74 ± 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	0.826 ± 0.019	0.664 ± 0.073	0.744 ± 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	0.757 ± 0.037	0.562 ± 0.058	0.679 ± 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	0.82 ± 0.015	0.666 ± 0.033	0.739 ± 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	0.805 ± 0.021	0.655 ± 0.047	0.725 ± 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	0.76 ± 0.041	0.56 ± 0.077	0.68 ± 0.05
Swiss roll very high	0.1	Heat-Geo	0.818 ± 0.033	0.668 ± 0.074	0.738 ± 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	0.605 ± 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	0.708 ± 0.031
Swiss roll very high	0.5	UMAP	0.789 ± 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	0.738 ± 0.019	0.551 ± 0.039	0.662 ± 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	0.784 ± 0.051	0.734 ± 0.07	0.786 ± 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	0.647 ± 0.049	0.591 ± 0.065	0.65 ± 0.048
Tree	0.5	UMAP	0.645 ± 0.051	0.565 ± 0.058	0.652 ± 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	0.432 ± 0.086	0.395 ± 0.098	0.432 ± 0.085

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

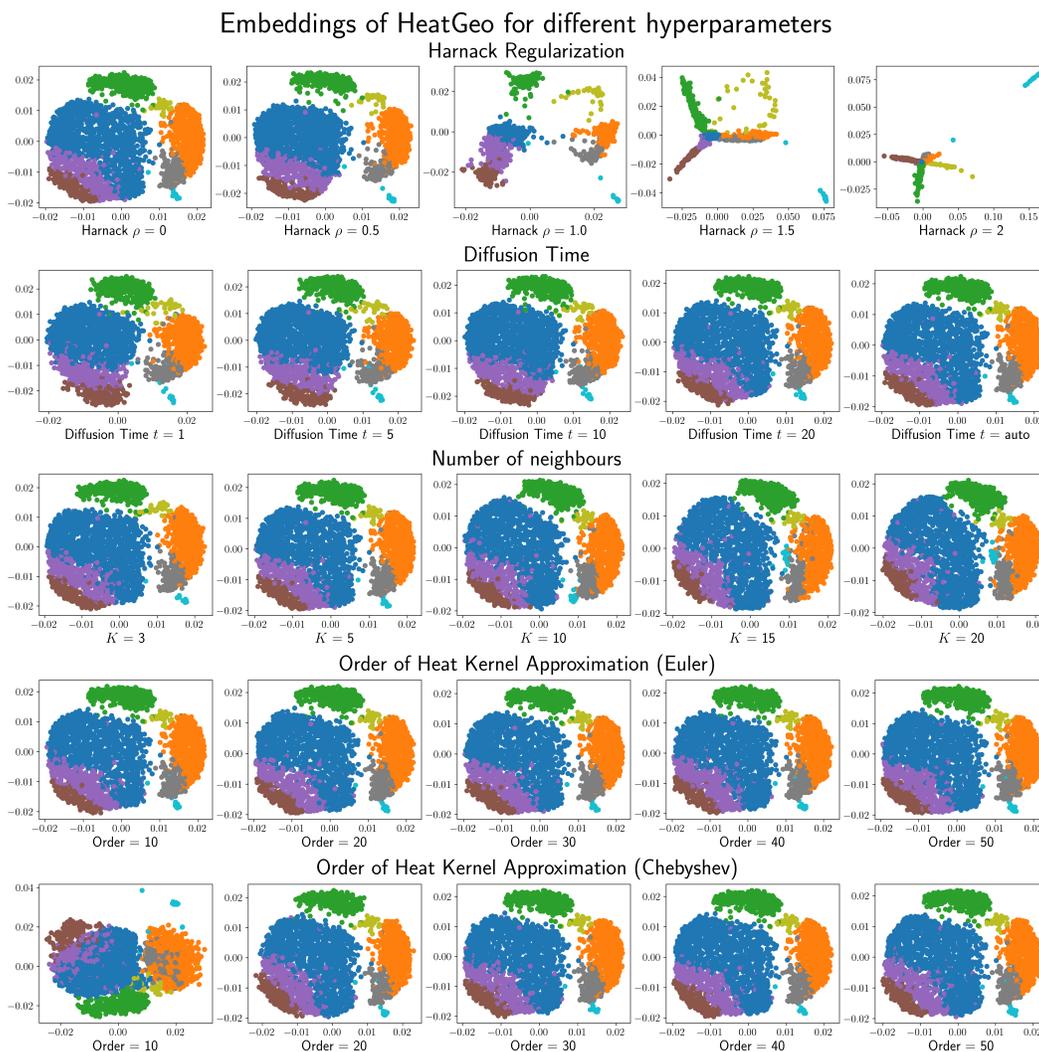


Figure 8: Embeddings of Heat Geodesic Embedding for different choices of hyperparameters on the EB dataset. We evaluate the impact of the Harnack regularization, the diffusion time, the number of neighbours in the kNN, and the order of the approximation for Euler and Chebyshev approximations.

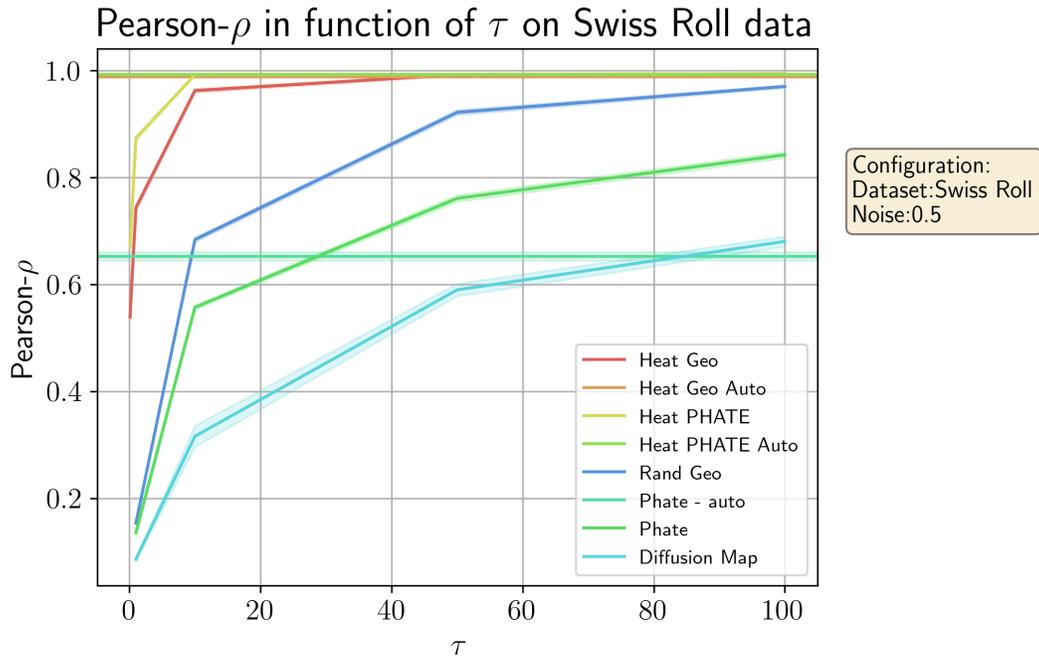


Figure 9: Impact of diffusion time on the Pearson correlation between the estimated distance matrix and ground truth distance matrix for different methods on the Swiss roll dataset.

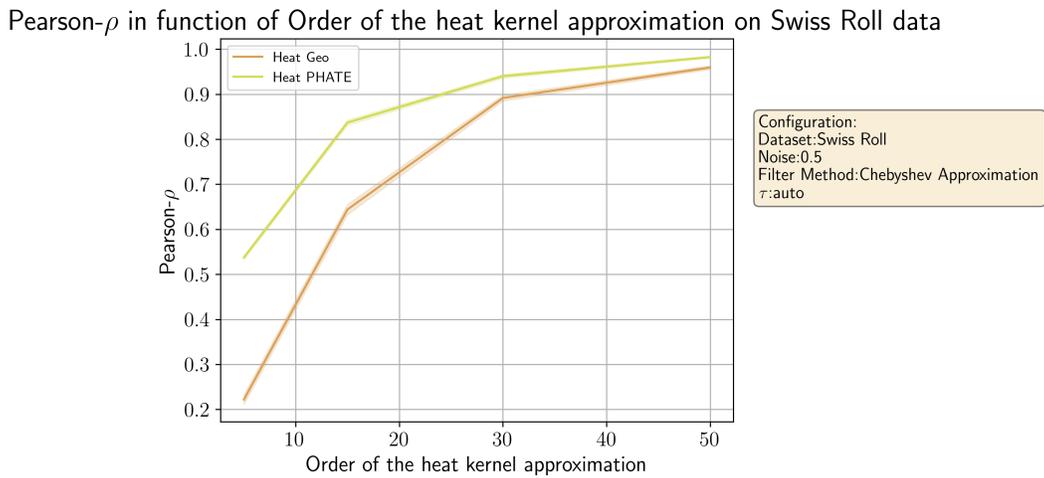


Figure 10: Impact of Chebyshev approximation order on the Pearson correlation between the estimated distance matrix and ground truth distance matrix for different methods on the Swiss roll dataset.

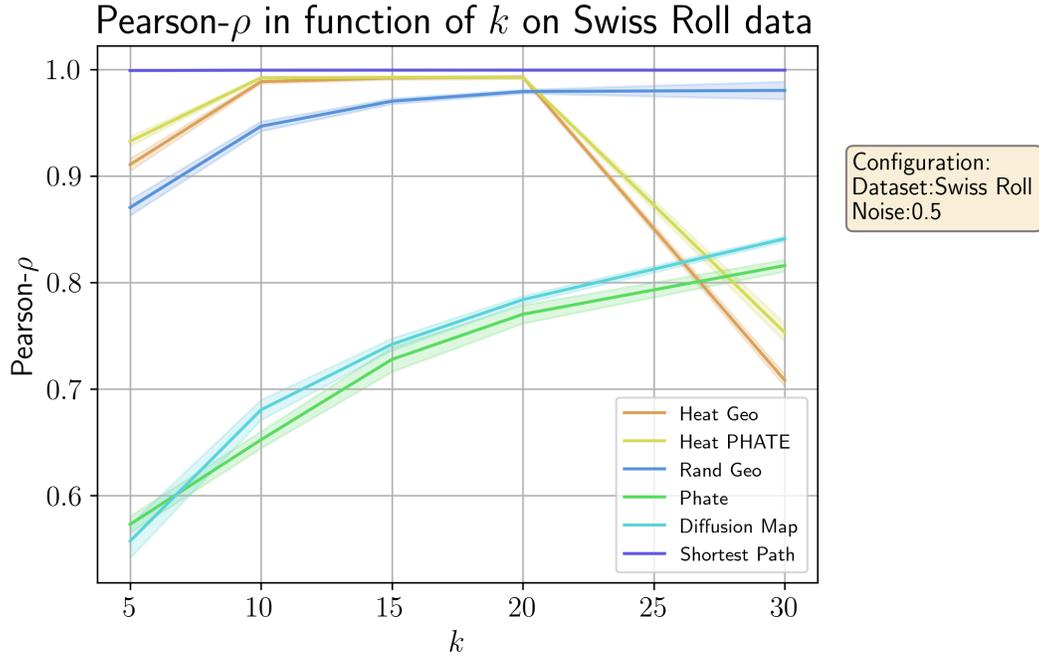


Figure 11: Impact of number of neighbours on the Pearson correlation between the estimated distance matrix and ground truth distance matrix for different methods on the Swiss roll dataset.

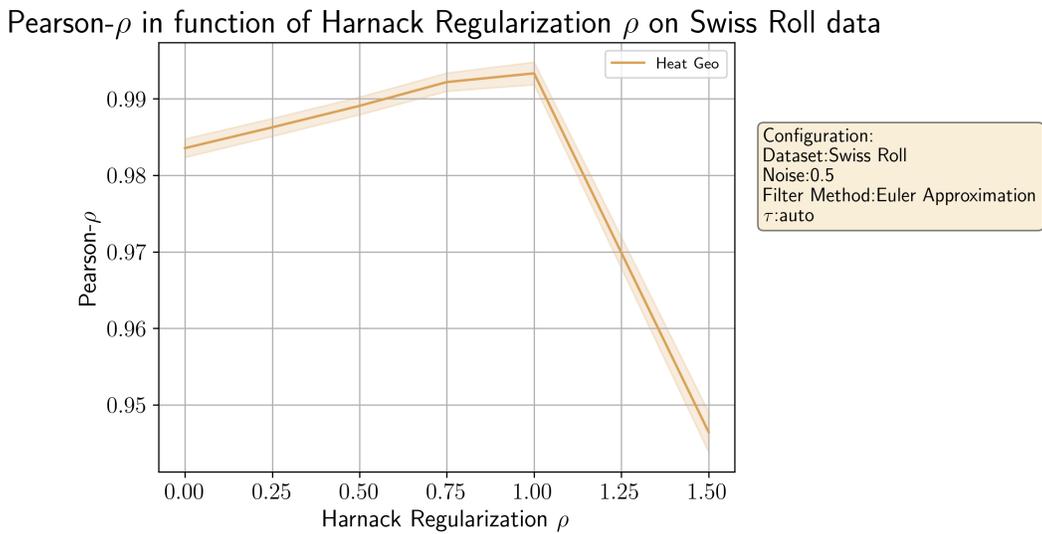


Figure 12: Impact of Harnack regularization on the Pearson correlation between the estimated distance matrix and ground truth distance matrix for HeatGeo on the Swiss roll dataset.

Pearson ρ in function of Graph type on Swiss Roll 10D data

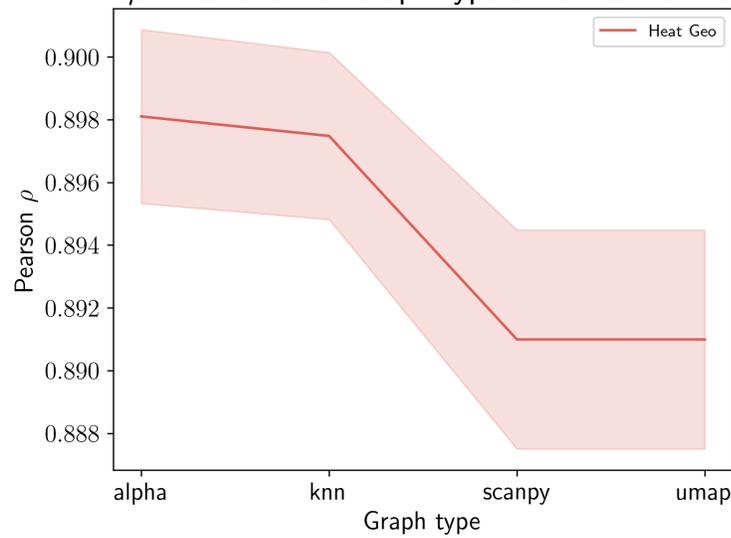


Figure 13: Pearson correlation between estimated and ground truth distance matrices for the 10-dimensional Swiss roll dataset for various graph constructions. Standard deviations are computed over the 5 test folds.