Image and Vision Computing 123 (2022) 104461

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Deep Isometric Maps

Gautam Pai [a,b,*], Alex Bronstein [b], Ronen Talmon [b], Ron Kimmel [b]

[a] Eindhoven University of Technology (TU/e), The Netherlands
[b] Technion-Israel Institute of Technology, Haifa, Israel

ARTICLE INFO

Article history:
Received 30 January 2021
Received in revised form 1 March 2022
Accepted 14 April 2022
Available online 25 April 2022

Keywords:
Multidimensional scaling
Manifold learning
Non-linear dimensionality reduction
Neural networks

ABSTRACT

Isometric feature mapping is an established time-honored algorithm in manifold learning and non-linear dimensionality reduction. Its prominence can be attributed to the output of a coherent global low-dimensional representation of data by preserving intrinsic distances. In order to enable an efficient and more applicable isometric feature mapping, a diverse set of sophisticated advancements have been proposed to the original algorithm to incorporate important factors like sparsity of computation, conformality, topological constraints and spectral geometry. However, a significant shortcoming of most approaches is the dependence on large-scale dense-spectral decompositions and the inability to generalize to points far away from the sampling of the manifold.

In this paper, we explore an unsupervised deep learning approach for computing distance-preserving maps for non-linear dimensionality reduction. We demonstrate that our framework is general enough to incorporate all previous advancements and show a significantly improved local and non-local generalization of the isometric mapping. Our approach involves training with only a few landmark points and avoids the need for population of dense matrices as well as computing their spectral decomposition.

1. Introduction

Multidimensional scaling (MDS) is a generic name for a family of techniques that estimate a configuration of points in a target metric space from the information about inter-point distances measured in some other metric space. Large-scale MDS problems are often essential in representation and visualization of high-dimensional data and therefore solving such problems efficiently is of key importance in many problems involving non-linear dimensionality reduction. In a manifold learning setup, the most prominent method to employ this distance preserving approach is the Isomap algorithm where, the target space is chosen to be euclidean and the input inter-point distances are the estimated geodesic distances computed over the nearest-neighbor graph associated with the set of points. Therefore, this technique enables recovering a near isometric embedding of the high-dimensional data points or achieving manifold flattening.

Distance preserving representations are of fundamental interest to the problem of non-linear dimensionality reduction and manifold learning. They enable a coherent global representation of data by preserving the metric structure of the data manifold. The preservation of intrinsic distances renders such a representation dependent only on the geometry of the manifold and not on how it is embedded in the ambient space, and hence makes the resulting representation meaningful.

In this paper, we use the computational infrastructure of neural networks to model maps that preserve geodesic distances on data manifolds. We revisit the classical framework of multidimensional scaling to find a configuration of points that satisfy pairwise distance constraints. We show that instead of optimizing over the individual coordinates of the points, we can optimize over the *function that generates these points* by modeling this map as a neural network. This choice of modeling the isometric map with a parametric network provides a straightforward out-of-sample extension, which is a simple forward pass of the network. We exploit efficient sampling techniques that progressively select landmark points on the manifold by maximizing the spread of their pairwise geodesic distances. We demonstrate that a small amount of these landmark points are sufficient to train a network to generate faithful low-dimensional embeddings of manifolds. Fig. 1 provides a visualization of our proposed approach.

From another viewpoint, our method can be understood as a particular case of metric learning. Metric learning refers to the set of techniques that learn features from data, where these features are trained to adhere to some notion of predefined distance. The typical input to a metric learning scheme are pairs of data-points and a label associated with each pair, that encodes the expected distance relationship between them. This relationship can be expressed as a coarse binary input (*close* and *far*) [1–3] or a tertiary input using triplets [4,5]. In either

* Corresponding author at: Eindhoven University of Technology, The Netherlands
E-mail addresses: g.pai@tue.nl (G. Pai), bron@cs.technion.ac.il (A. Bronstein), ronen@ee.technion.ac.il (R. Talmon), ron@cs.technion.ac.il (R. Kimmel).

https://doi.org/10.1016/j.imavis.2022.104461

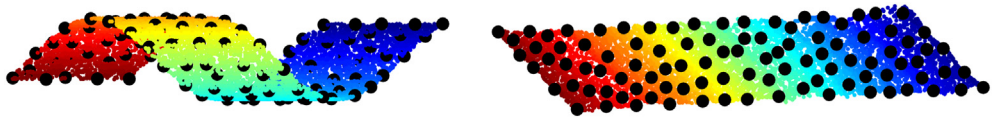


Fig. 1. Learning to unfold a ribbon: A three dimensional Helical Ribbon (left) and its two dimensional embedding - a planar parallelogram (right) learned using a two-layer MLP. The network was trained using estimated pairwise geodesic distances between only 100 points (marked in black) out of the total 8192 samples.

case, the goal is to discover features for each data-point, such that the expected distance relationship is preserved in the feature space. By training with a large number of pairs/triplets, the metric learning scheme is expected to converge to consistent features that characterize the data in terms of these expected distance relationships.

In this paper, we explore the scenario when the expected distance between pairs of points is known precisely (for example, the geodesic distance). Therefore, our approach allows us to view the concept of *generalization* from a geometric viewpoint. Algorithms that exhibit poor generalizations will yield embeddings that show a clear visual depiction of suboptimal flattening and can be objectively measured using the stress function (Eq. 4).

We begin by providing a detailed review of manifold learning and specifically, the MDS problem in high-dimensional data analysis in Section 2 and focus on a elaborate discussion on the different interpolation philosophies involved in sparse MDS regimes. In Section 3, we discuss our approach of modeling isometric maps with neural networks (DIMAL: Deep Isometric MAnifold Learning) and describe the training procedure. In Section 4 we qualitatively and quantitatively evaluate existing MDS approaches with our formulation. Finally in Section 5, we explore generalizations of our *learned* isometric mapping to problems involving non-trivial sampling and topology.

This paper extends the principle ideas proposed in [6] with essential additional discussions and experiments of our deep isometric manifold learning scheme. Specifically, we provide a detailed discussion on interpolation philosophies involved in sparse MDS regimes in Section 2.3 and show that our deep learning approach exhibits superior generalization properties in challenging *extrapolation* scenarios. These attributes stand out most significantly for the flattening of non-trivial manifolds: those that have holes, boundaries and other non-trivial structures that we additionally explore in Section 5.

2. Background

2.1. Manifold Learning

Manifold learning is the process of recovering a low-dimensional representation from a possibly non-linear high-dimensional data. The literature on manifold learning is dominated by spectral methods that have a characteristic computational pattern. The first step involves the computation of the k nearest neighbors of all N data points. Then, an $N \times N$ square matrix is populated using some geometric principle which characterizes the nature of the desired low-dimensional embedding. The eigenvalue decomposition of this matrix is then used to obtain the low-dimensional representation of the manifold. Manifold learning techniques such as Laplacian Eigenmaps [7], LLE [8], HLLE [9] and Diffusion Maps [10] are considered to be local methods, since they are designed to minimize some form of local distortion and hence result in embeddings which preserve locality. Methods like Isomap [11–13] are considered global because they enforce preserving all geodesic distances in the low-dimensional embedding. Local methods usually lead to sparse matrix eigenvalue problems and hence are computationally advantageous. However, global methods are more robust to noise and achieve globally coherent embeddings, in contrast to the local methods which can sometimes lead to excessively clustered results. All spectral techniques are non-parametric in nature and hence do not characterize the map that generates them. Therefore, the computational burden of large spectral decompositions becomes a major drawback when the number of data-points is large. Furthermore, out-of-sample extension of the map is a computationally expensive task. [14,15] address this issue by providing formulas for out-of-sample extensions to the spectral algorithms. However, these interpolating formulations are computationally inefficient and exhibit poor non-local generalization of the manifold [16].

Examining the ability of neural networks to represent data manifolds has received considerable interest in recent years and has been studied from multiple perspectives. [17–19] use neural networks specifically for solving the out-of-sample extension problem for manifold learning. However, their procedure involves training a network to follow a pre-computed non-parametric embedding rather than adopting an entirely unsupervised approach, thereby inheriting some of the deficiencies of the non-parametric methods. Another notable approach is based on the metric-learning arrangement of the Siamese configuration [1–3] using a contrastive loss. The loss enforces positive pairs to have similar outputs and negative pairs to have dissimilar outputs, cumulatively leading to a local optimum that preserves all pairwise relationships. Similarly, autoencoder networks [20–23] are very prominent to generate low-dimensional latent codes that characterize the manifold structure of data by a dual loss combination of reconstruction and regularization. However, most of these deep learning methods require considerable training effort in terms of number of examples needed for satisfactory embeddings.

In contrast to these existing approaches, our method attempts to combine the advantages of both: traditional manifold learning as well as more recent deep-learning models. Firstly, the pre-computation of manifold distances allows for a sampling strategy to select landmark points for training, thereby considerably reducing training effort. Secondly, we can incorporate various advancements made in manifold based distance computation procedures to deal with non trivial manifolds having structures like holes, boundaries and also variable sampling issues. Finally, the parametrization of the map with a network allows for improving *generalization* in contrast to this drawback in the more traditional non-parametric spectral schemes.

2.2. Multidimensional Scaling

Multidimensional scaling (MDS) is a classical algorithm for obtaining the global picture of data using pairwise distances or dissimilarities information. The core idea of MDS is to find an embedding configuration $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3} \ldots \mathbf{x_N}]$, such that all pairwise distances measured in the embedded space (typically $\|\mathbf{x_i} - \mathbf{x_j}\|$) are faithful to the given distances $\mathbf{D}_s = [d_{ij}^2]$ as much as possible. Therefore, the input to an MDS algorithm is the *distance matrix* and the output is the embedding configuration $\mathbf{X}$ that preserves these pairwise distances. In the context of isometric manifold learning, the MDS framework is enabled by imputing pairwise *geodesic* distances and recovering a low-dimensional embedding output such that pairwise Euclidean distances in this space match the corresponding geodesic distances. Putting simply, this is the action of *flattening* the manifold as shown in Fig. 1.

There are two prominent, yet different versions of MDS: *Classical Scaling* and *Least-Squares Scaling*. Classical Scaling is based on the observation that the double centering of a pairwise squared distance matrix gives an inner-product matrix which can be factored to obtain the desired embedding. Therefore, if $\mathbf{H} = \mathbf{I} - \frac{1}{N}\mathbf{1}\mathbf{1}^T$ is the centering matrix,

classical scaling minimizes the strain of the embedding configuration $\mathbf{X}$ and is computed conveniently using the eigen-decomposition of the $N \times N$ matrix $-\frac{1}{2}\mathbf{H}\mathbf{D}_s\mathbf{H}$:

$$\mathbf{X}^*_{CS} = \arg\min_{\mathbf{X}} \quad \|\mathbf{X}\mathbf{X}^T + \frac{1}{2}\mathbf{H}\mathbf{D}_s\mathbf{H}\|_F^2 \tag{1}$$

$$\mathbf{X}^*_{CS} = \mathbf{V}^{\frac{1}{2}}, \text{ where } -\frac{1}{2}\mathbf{H}\mathbf{D}_s\mathbf{H} = \mathbf{V}\mathbf{V}^{\mathbf{T}} \tag{2}$$

At the other end, least squares scaling is based on minimizing the misfits between the pairwise distances of $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3} \ldots \mathbf{x_N}]$ and desired distances $[d_{ij}]$ measured by the *stress* function

$$\mathbf{X}^*_{LS} = \arg\min_{\mathbf{X}} \quad \sigma(\mathbf{X}) \tag{3}$$

$$\sigma(\mathbf{X}) = \sum_{i<j} w_{ij}(\|\mathbf{x}_i - \mathbf{x}_j\| - d_{ij})^2 \tag{4}$$

Minimization of Eq. 4 is typically handled using gradient descent iterations with $w_{ij} = 1$ for all i, j.[1] One particular case was introduced by Leeuw et al. [25] under the name SMACOF (Scaling by Majorizing a COmplicated Function). The algorithm is based on the following iterative step:

$$\mathbf{X}_{k+1} = \left(\mathbf{V} + \frac{1}{N}\mathbf{1}\mathbf{1}^T\right)^{\dagger} \mathbf{B}(\mathbf{X}_k)\ \mathbf{X}_k \tag{5}$$

where $N \times N$ matrices $\mathbf{V} = [v_{ij}]$ and $\mathbf{B}(\mathbf{X}_k) = [b_{ij}]$ are given by:

$$v_{ij} = \begin{cases} -w_{ij} & i \neq j \\ \sum_{k \neq i} w_{ik} & i = j \end{cases} \quad \text{and} \quad b_{ij} = \begin{cases} -w_{ij}\dfrac{d_{ij}}{\|\mathbf{x_i} - \mathbf{x_j}\|} & i \neq j, x_i \neq x_j \\ 0 & i \neq j, x_i = x_j \\ -\sum_{k \neq i} b_{ik} & i = j. \end{cases} \tag{6}$$

The iteration in Eq 5 guarantees a non-increasing stress, due to the principle of majorization. For more details we refer the interested reader to [25].

In practice, the MDS framework is enabled by estimating all pairwise *geodesic* distances with a shortest path algorithm like Dijkstra's [26], and choosing an MDS scaling algorithm to generate the low-dimensional embeddings. Schwartz et al. [27,28] and the Isomap algorithm [11] were the first to suggest populating the pairwise distance matrix $\mathbf{D}_s$ using Dijkstra's algorithm. [29,30] first suggested the use of consistent approximate of geodesics for the goal of flattening into 2D and [31] employed the Fast Marching method for computing the geodesic distances on 2D surfaces.

Historically, MDS was first developed in [32,33] who proposed the purely Euclidean model (input distances were euclidian) of the MDS. Later, non-metric versions of MDS were developed in [34,35] which focused on a generic dissimilarity information rather than plain inter-point distances. In the pattern recognition literature, a version of least squares scaling of Eq 3 is known as Sammon's non-linear mapping (NLM) [36]. Sammon's NLM proposed a straightforward gradient descent on a scaled stress function similar to Eq 4 for feature extraction from data. Extensions to Sammon's non-linear mapping using artifical neural networks (called SAMANN) was shown in [37,38] using plain euclidean distances. More recently, [58] proposed metric estimation and parametrization using neural networks, with an emphasis on the application to indoor localization in ad-hoc sensor networks.

[1] A case of a least squares scaling problem with non-trivial w_{ij}'s was introduced in [24] and is discussed in Section 5.1

2.3. Sparse Multidimensional Scaling

One of the major drawbacks of the plain MDS approach is its computational cost and the lack of a principled generalization framework for unseen data. The minimization of objectives 1 and 4 demand computing geodesic distances between *all* pairs of points leading to an expensive eigen-decomposition of a dense $N \times N$ matrix as in the case of classical scaling. Similarly, each step of the SMACOF iteration 5 demands computing pairwise Euclidean distances between *all* pairs of points.

This motivated various *fast* MDS algorithms that were based on sampling the manifold and computing the pairwise geodesic distances *only* between these sampled pairs of points. The input to a *sparse MDS* algorithm is a much smaller $K \times K$ matrix of distances with $K \ll N$ as compared to the entire $\mathbf{D}_s \in \mathbb{R}^{N \times N}$ matrix required in Eqs 2 and 5. The embedding of the whole dataset is then extracted using different interpolation philosophies.

For example, Aflalo et al. [39] use the spectral geometry of the manifold using the Laplace Beltrami Operator (LBO) to interpolate the distances in a classical scaling framework. The eigenfunctions of the LBO form an orthonormal basis and are infact proven to be optimal [40] for representing smooth functions on manifolds. Moreover, from a computational standpoint, unlike the dense inter-geodesic distance matrix $\mathbf{D}_s$, the discrete Laplace Beltrami Operator is sparse and computing the low-frequency eigenfunctions of the LBO is computationally tractable even for large N.

For a given problem of N points, Let $\in \mathrm{R}^{N \times p}$ be the first p eigenfunctions of the LBO (see Fig. 2), with typically $p \ll N$. Spectral MDS uses these eigenfunctions to approximate geodesic distance functions as well as the final embedding $\mathbf{X}$. Therefore the whole $\mathbf{D}_s \in \mathbb{R}^{N \times N}$ matrix is represented using a smaller $\alpha \in \mathbb{R}^{p \times p}$ (spectral co-efficients) that are implicitly computed using a much smaller input geodesic distance matrix $\mathbf{D}_K \in \mathbb{R}^{K \times K}$(we refer the interested reader to [39] for the complete formulation)

$$\begin{aligned} \mathbf{X} &\approx \Phi\beta \\ \mathbf{D}_s &\approx \Phi\alpha\Phi^T \end{aligned} \tag{7}$$

The approximations of Eq. (7) using $\alpha \in \mathbb{R}^{p \times p}$ and $\beta \in \mathbb{R}^{p \times d}$ ($d \ll N$ is the output dimension) are then used in the classical MDS framework of Eq. (2). This leads to a reduced set of equations that solve for β which approximates the final embedding $\mathbf{X}$. Similar in spirit, Boyarski et al. [41] propose to use a spectral interpolation for the SMACOF iterations of Eq 5, minimizing the stress for only a subsampled configuration of points and interpolating the rest using eigenvectors of the Laplace Beltrami operator. Shamai et al. [42,43] developed a sparse MDS using the Nystrom extension method. A similar approach is also adopted in multigrid-MDS [44] and vector extrapolation [45].

Although the use of a secondary interpolation scheme provided a speed up, these frameworks still lack the principled means to generalize to unseen data. The Landmark Isomap method [15,46] proposed a formula to compute the out-of-sample extension of MDS. This formula was based on computing geodesic distances of the new samples to existing landmarks to estimate their embedding without re-doing the entire optimization for new data. Specifically, the landmark isomap procedure is as follows. First, a classical scaling is performed for a smaller matrix $\mathbf{D}_K \in \mathbb{R}^{K \times K}$. Then define $\mathbf{L}_d \in \mathbb{R}^{K \times d}$ as

$$\mathbf{L}_d = \begin{bmatrix} \sqrt{\lambda_1} \cdot \mathbf{v}_1^{\mathrm{T}} \\ \sqrt{\lambda_2} \cdot \mathbf{v}_2^{\mathrm{T}} \\ \vdots \\ \sqrt{\lambda_d} \cdot \mathbf{v}_d^{\mathrm{T}} \end{bmatrix}^T \tag{8}$$

where $\{\lambda_1, \lambda_2 \ldots \lambda_d\}$ and $\{\mathbf{v}_1, \mathbf{v}_2, \ldots \mathbf{v}_d\}$ are the d largest eigenvalues and eigenvectors corresponding to the classical scaling of matrix $\mathbf{D}_K$ as per

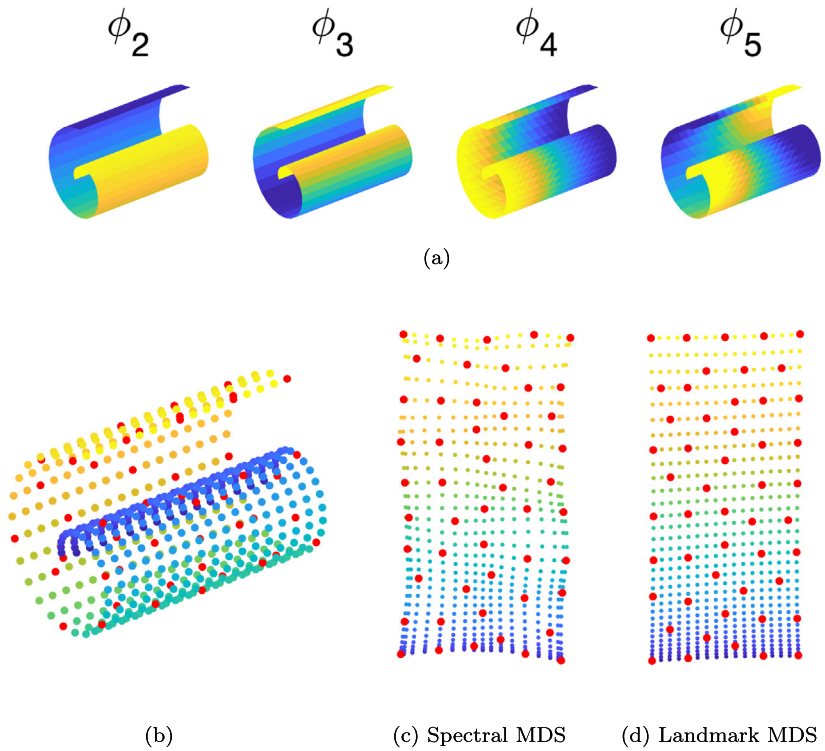


Fig. 2. Sparse Multidimensional Scaling (a) First few non-trivial eigenfunctions of the Laplace Beltrami operator on a swiss roll manifold used for spectral interpolation [39,41]. (b) A swiss roll manifold with sparse landmarks (in red). A sparse MDS algorithm outputs an embedding using a much smaller number of input distances associated with the sparse landmarks. (c)-(d) output of two prominent sparse MDS algorithms: Spectral MDS [39] and Landmark MDS [12].

Eq. (2). The low-dimensional embedding of any new point p denoted by $\mathbf{X}(p) \in \mathbb{R}^d$ is given by

$$\mathbf{X}(p) = -\frac{1}{2}\mathbf{L}_d^{\#}(\delta_p - \delta_\mu) \tag{9}$$

where $\delta_p \in \mathbb{R}^K$ is the vector of squared geodesic distances of the point p to the K landmarks, $\delta_\mu \in \mathbb{R}^K$ is a vector where each element is the average squared geodesic distance of that landmark to all other landmarks, and $\mathbf{L}_d^{\#}$ is the pseudo-inverse of $\mathbf{L}_d$. The Eqs. (8) and (9) are equivalent to a kernel formulation of the out-of-sample extension for MDS [14]:

$$\mathbf{X}(p) = \sum_{i=1}^{i=K} \mathbf{X}(p_i)K(p,p_i) \tag{10}$$

$$K(p,p_i) \propto d_{avg}^2 - d^2(p,p_i) \tag{11}$$

where d_{avg} is the average squared geodesic distance between the ith landmark to the other K landmarks. $d(p,p_i)$ is the geodesic distance between landmark p_i and the point p. Therefore the landmark Isomap requires an estimate of the geodesic distance between the out-of-sample points to the landmarks in order to implement the formula of Eq 9. This requirement is an inherent drawback of the method. For a problem of N points and K landmarks, Landmark isomap demands a computation of $\mathcal{O}(KN)$ distances in contrast to $\mathcal{O}\left(K^2\right)$ (as in spectral MDS). Also, in most practical scenarios, computing geodesic distances for new points is intractable and inaccurate especially for non-local points on the manifold.

To conclude this section, we reiterate that the problem of estimating a near isometric embedding of data from the estimated distances between a sparse and subsampled set of points, can be interpreted as an implicit interpolation/extrapolation problem. Methods like Spectral MDS [39] and Landmark MDS [12] proposed solutions, that use additional information like spectral geometry (eigenfunctions of the LBO) or additional geodesic distances. What we demonstrate in this paper is that a neural network scheme for the same objective, outperformes these previous methods with only $\mathcal{O}\left(K^2\right)$ distances required for training and without the need for any external information such as spectral geometry, nystrom projection matrices or out-of-sample geodesic distances. The superiority of our approach is most significant for challenging *extrapolation* scenarios which we demonstrate in Section 5. In the next section we outline the procedure of our main contribution: DIMAL: Deep Isometric MAnifold Learning.

3. Deep Isometric Maps

3.1. Training Configuration

We incorporate the ideas of least squares scaling into the computational infrastructure of the Siamese configuration as shown in Fig. 3. A Siamese configuration comprises of two identical networks that process two different inputs. The outputs are then combined in a loss that is typically a function of the distance between the output pairs. This configuration has been extensively used for the purposes of metric learning, descriptor learning, 3D shape correspondence etc. [1,2,47,59]. For every p^{th} pair of data-points, we estimate the geodesic distance using a shortest path algorithm and train the network by minimizing the network-parameterized stress function (see Fig. 3):

$$\mathcal{L}(\Theta) = \sum_p \left(\| \mathcal{F}_\Theta\left(\mathbf{X}_1^{(p)}\right) - \mathcal{F}_\Theta\left(\mathbf{X}_2^{(p)}\right) \| - d^{(p)} \right)^2 \tag{12}$$

The expectation here is that with a sufficient number of example pairs, the network learns to model a map $\mathcal{F}_\Theta : \mathbb{R}^M \rightarrow \mathbb{R}^m, \quad m \ll M,$

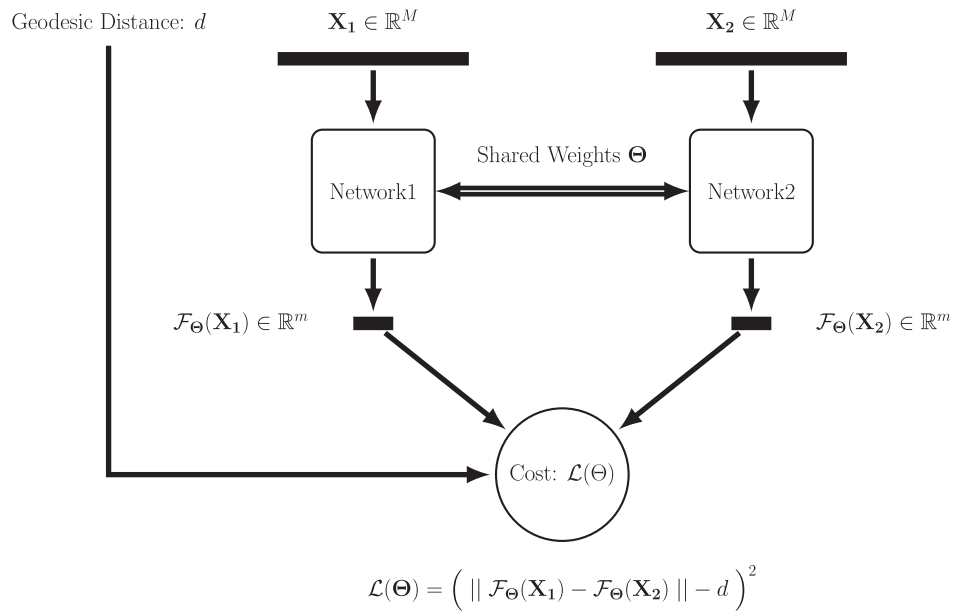


Fig. 3. Siamese configuration: Each arm of the siamese network models the map $\mathcal{F}_\theta : \mathbb{R}^M \rightarrow \mathbb{R}^m$, $m \ll M$. For each training pair $(\mathbf{X}_1, \mathbf{X}_2) \in \mathbb{R}^M$, the loss minimizes the squared difference between the Euclidean distances of the low-dimensional embedding and the geodesic distance d.

from high to low dimension, that preserves the distance. As explained in Section 4.1, we can study exactly how many examples are needed to learn this map to sufficient accuracy, by employing a sampling strategy that sufficiently covers the entire manifold uniformly.

Algorithm 1. Farthest point sampling.

Input:	metric space (X, d_X), initial point $x_1 \in X$, number of landmarks: K
Output:	sampling $X' = \{x_1, x_2, x_3, ...x_K\}$
Initialization:	$X' = \{x_1\}$, $d(x) = d_X(x, x_1)$

1. **while** $|X'| < K$ **do:**
2. Find the farthest point from X', $x' = \arg\max_{x \in X} d(x)$
3. Update set of selected samples: $X' \longleftarrow X' \cup \{x'\}$
4. Update distance function $d(x)$: $d(x) \longleftarrow \min_x \{d(x), d_X(x, x')\}$
5. **End**

3.2. Geodesic Farthest Point Sampling

The farthest point sampling strategy [48,49] (also referred to as the MinMax strategy in [15]) is a method for picking landmarks amongst the points of a discretely sampled manifold such that under certain conditions, these samples cover the manifold as uniformly as possible. Starting from a random selection, the landmarks are chosen one at a time such that each new selection from the unused samples has the largest geodesic distance to the set of the selected sample points. Fig. 4 provides a visualization of this sampling mechanism. We train the network by minimizing the loss in Eq 12 by computing the pairwise geodesic distances between K landmarks. Therefore, the pre-training computational effort is limited to $\mathcal{O}\left(K^2\right)$ distances. Given a dataset with N samples, we can summarize the proposed DIMAL algorithm as follows (see Algorithm 2). We first build a graph from the data using an approximate nearest neighbor algorithm, and obtain the set of landmarks and corresponding pairwise geodesic distances using farthest point sampling (Algorithm 1). Then, we construct a dataset of all pairs of landmarks and their corresponding geodesic distances. Using the Siamese configuration depicted in Fig. 3 we train the network parameters minimizing the MDS loss of Eq 12. The low-dimensional embedding for any high-dimensional datapoint is obtained with its forward pass through the trained network, as shown in Fig. 4. If the cost of computing one row of $N \times N$ geodesic distances (using different choices like Dijkstra/Fast Marching etc.) is f, then the resulting time complexity is $\mathcal{O}(Kf)$.

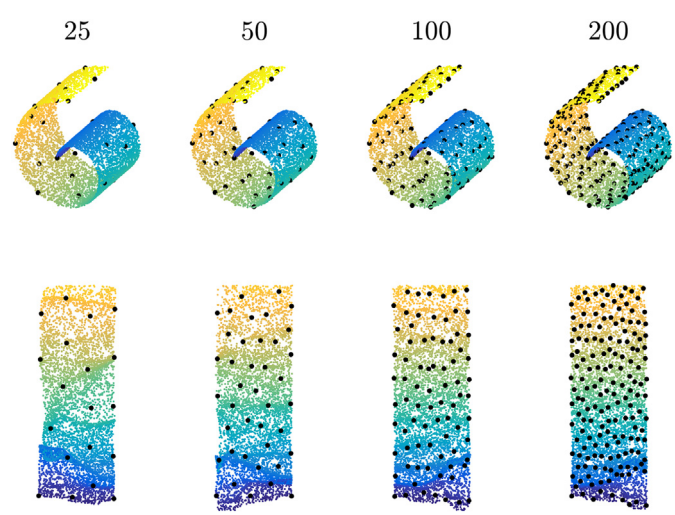


Fig. 4. Farthest point sampling: The farthest point sampling algorithm enables a sparse sub-sampling of the manifold, [48]. The network is trained using the loss of Eq. (12) and the embedding for the entire configuration is visualized.

Algorithm 2. DIMAL: Deep Isometric MAnifold Learning.

Input: High-dimensional data-points of manifold: $\{\mathbf{X}_i \in \mathbb{R}^M\}$, $i = 1, 2, ...N$, number of landmarks: K

Output: Low-dimensional embeddings of datapoints: $\{\mathbf{x}_i \in \mathbb{R}^m, m \ll M\}$, $i = 1, 2, ...N$,

1. Compute the nearest-neighbor graph from the manifold data and obtain a set of K landmark points using Algorithm 1
2. Obtain pairwise geodesic distances $\mathbf{D}_s$ between landmarks using Dijkstra's or any other numerical algorithm
3. Form a dataset of landmark pairs with corresponding geodesic distances $\left\{\mathbf{X}_1^{(p)}, \mathbf{X}_2^{(p)}, d^{(p)} = d(\mathbf{X}_1^{(p)}, \mathbf{X}_2^{(p)})\right\}$
4. **Training:** Train network $\mathcal{F}_\Theta$ with Siamese configuration of Figure 3 minimizing loss 12 for the dataset obtained in Step 3
5. **Inference:** Obtain low-dimensional embedding with forward pass of the network: $\mathbf{x}_i = \mathcal{F}_\Theta(\mathbf{X}_i)$, $\forall i \in \{1, 2, ...N\}$

4. Experiments

The core idea behind our experiments is to train on the sub-sampled landmarks and observe the effect on the entire set. Put in another way, we train to preserve only $\binom{K}{2} = \frac{1}{2}K(K-1)$ distances between the landmarks and evaluate the stress of Eq 4 over *all* $\frac{1}{2}N(N-1)$ datapoints. We perform three separate experiments using this evaluation methodology. First, in Section 4.1 we vary the number of landmarks K, and observe the overall stress as a function of the network architecture. Second, in Sections 4.3 and 4.4, we compare DIMAL to existing sparse MDS frameworks by imputing all the algorithms with the *same landmarks and same geodesic distances* and evaluating the MDS outputs using overall stress and also analyze the out-of-sample performance. Finally, in Section 5 we extend our method to challenging non-trivial conditions like boundaries, holes and arbitrary topologies.

4.1. Numerical Experiments on 3D Point Clouds

Our first set of experiments is based on synthetic point-cloud manifolds, like S-curve (Fig. 5) and the Helical ribbon (Fig. 1). The architecture of the network in Fig. 3 is chosen to be a multilayer perceptron (MLP) with the PReLU

$$\mathrm{PReLU}(x) = \max(0, x) + a \min(0, x), \tag{13}$$

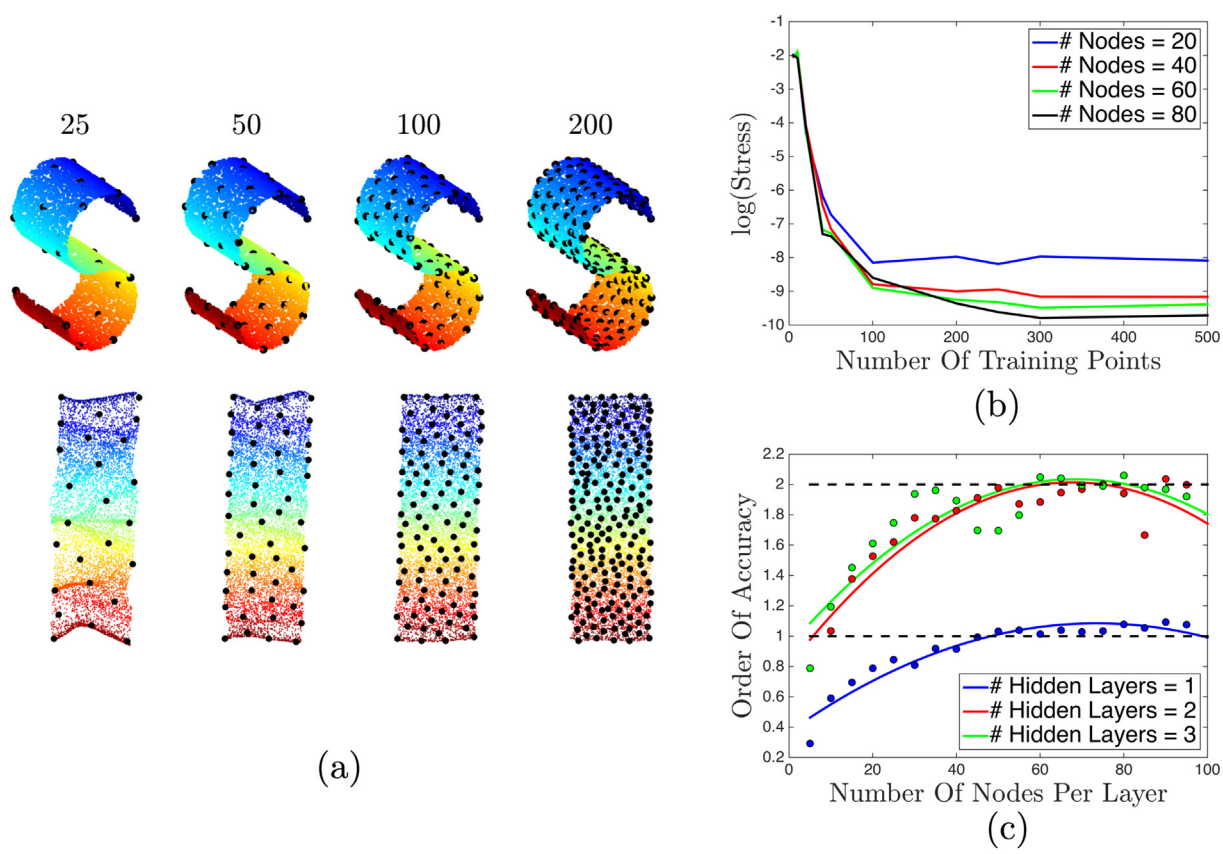


Fig. 5. Exploring the variations in architecture and the number of landmarks: (a) (top row) Three dimensional S-Curve manifold with varying number of landmark points obtained using Algorithm 1. (bottom row) The corresponding two dimensional DIMAL output (Algorithm 2) generated by a 2-Layer MLP with 70 hidden nodes per layer. (b) The logarithm of stress of all 8172 points as a function of number of landmarks. (c) Order of accuracy estimates of different architectural choices.

as the non-linear activation function, where a is a learnable parameter. The networks are trained for 1000 iterations using the ADAM optimizer with constants $(\beta_1,\beta_2) = (0.95,0.99)$ and a learning rate of 0.01. We run each optimization 5 times with random initialization to ensure convergence. All experiments were implemented in Python using the PyTorch framework [50]. We used the scikit-learn machine learning library for the nearest-neighbor and scipy-sparse for Dijkstra's shortest path algorithms.

Figs. 1, 4 and 5 show the results of our method on the Helical ribbon and S-curve respectively with varying number of training samples (in black) out of a total of 8172 data points. The number of landmarks dictates the approximation quality of the low-dimensional embedding generated by the network. Training with too few samples results in inadequate generalization which can be inferred from the corrugations of the unfolded manifold embedding in the first two parts of Fig. 5(a). Increasing the number of landmarks improves the quality of the embedding, as expected. We compute the stress function 4 of the entire point configuration to measure the quality of the MDS fit. Fig. 5 (b) shows the decay in the stress as a function of the number of training points (landmarks) in a two layer MLP.

4.1.1. Architecture Design

For 3D pointclouds like the Swiss roll (Fig. 4), Helical Ribbon (Fig. 1) or the S-curve (Fig. 5(a)), the network architecture to map $\mathcal{F}_\theta : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ is a standard feedforward multilayered perceptron with the PReLU() activation function as shown in Eq. (13). From the perspective of a simple architecture design for our setup, the natural questions to ask are *how many landmarks? how many layers? and how many hidden nodes per layer?*

We analyze the performance of the architecture, by borrowing analogous evaluation methodologies from numerical methods for partial differential equations. More specifically, given a numerical technique, the accuracy of the solution depends on the resolution of the spatial grid over which the solution is estimated. Therefore, numerical methods are ranked by an assessment of the *order of accuracy* their solutions observe [51]. This can be obtained by assuming that the relationship between the approximation error E and the resolution of the grid h is given by

$$E = C\ h^P \tag{14}$$

where P is the order of accuracy of the technique and C is some constant. Thus, for every spatial resolution h, the error of the numerical algorithm $E(h)$ is evaluated and $\log(E(h))$ is plotted as a function of $\log(h)$ for the specific algorithm. P is obtained by computing the slope of the line since,

$$\log(E) = \log(C) + P \quad \log(h). \tag{15}$$

We extend the same principle in order to evaluate network architectures (in place of numerical algorithms) for estimating the quality of isometric maps. We use the overall stress of Eq 4 as the error function E in Eq 14. We assume that due to the 2-optimal property [48] of the farthest point strategy for a two dimensional manifold, the sampling is approximately uniform and hence $h \propto \frac{1}{\sqrt{K}}$ where K is the number of landmarks. By varying the number of layers and the number of neurons per layer, we associate an order of accuracy to each architecture using Eq 15, by training with a varying number of landmarks K and evaluating the overall stress.

Fig. 5 shows the results of the described experiment. It shows that a single layer MLP has the capacity of modeling functions to the first order of accuracy. Adding a layer increases the representation power by moving to a second order result. Adding more layers does not provide any substantial gain arguably due to a larger likelihood of over-fitting as seen in the considerably noisier estimates (in green). Therefore, a two layer MLP with 70 hidden neurons per layer can be construed as a good architecture for approximating the isometric map of the S-Curve of Fig. 5(a) with 200 landmarks.

4.1.2. Robustness

In addition, we also experiment the robustness of different approaches to varying degrees of noise in the input distances in Fig. 6. Our parametric approach to the isometric embedding by *optimizing for the function that generates the points* instead of *optmizing the points* themselves makes the resultant embedding more robust to noise in the input.

4.2. Image Articulation Manifolds

Each point on a non-linear articulation manifold is a binary image that is generated by the articulation of a few parameters. In [52] it is shown that certain types of such manifolds are isometric to Euclidean space, that is, the geodesic distance between any two sample points is equal to the Euclidean distance between their articulation parameters. Therefore, one can consider such manifolds to be the multidimensional equivalents of the three dimensional Swiss-roll or S-curve. We construct a horizon articulation manifold where each image contains two distinct regions separated by a horizon which is modulated by a linear combination of two fixed sinusoidal basis elements as depicted in Fig. 7a.

$$\begin{aligned} I_{\alpha_1,\alpha_2}(u,v) &= \mathbb{1}_{\{v \le \psi_{\alpha_1,\alpha_2}(u)\}} \\ \psi_{\alpha_1,\alpha_2}(u) &= \alpha_1\ \sin(\omega_1 u) + \alpha_2\ \sin(\omega_2 u) \end{aligned} \tag{16}$$

Thus, each sample has an intrinsic dimensionality of two - the articulation parameters (α_1,α_2) which govern how the sinusoids representing the horizon are mixed. We sample the articulation parameters from a 2D uniform distribution

$$(\alpha_1,\alpha_2) \sim U([0,1] \times [0,1]). \tag{17}$$

In the context of the main narrative of this paper, which is metric preserving properties of manifolds, we find that such a dataset provides an appropriate test-bed for evaluating metric preserving algorithms. Since we are assured of isometry to Euclidean plane, we can objectively measure the performance of an MDS flattening algorithm with the stress function in Eq 4. Fig. 7b shows the comparison between DIMAL and other prominent manifold learning algorithms. Except for DIMAL and Isomap, all other methods exhibit some form of distortion indicating a suboptimal metric preservation.

4.3. Evaluation with Sparse MDS Algorithms

We compare DIMAL to existing state-of-the-art sparse MDS algorithms and present the results in Fig. 8. We generate 5000 examples of the articulation manifold of Fig. 7a with $\omega_1 = 2$ and $\omega_2 = 4$. DIMAL was trained with a CNN comprising two convolution layers, each with kernel sizes 12 and 9, number of kernels 15 and 2, respectively, along with a stride of 3 and, followed by a fully-connected layer mapping the image to a two-dimensional domain. We train the network for 500 iterations using the ADAM optimizer [53] with a learning rate of 0.01 and parameters $(\beta_1,\beta_2) = (0.95,0.99)$. Each algorithm has been imputed with the *same landmarks* and the *same corresponding pairwise geodesic distances*. For every K landmarks used, we use $\frac{1}{2}K$ eigenvectors of the Laplace-Beltrami operator for the spectral MDS algorithms as suggested in [41]. We observe that DIMAL for almost all values of K performs visually and quantitatively better that other MDS algorithms without using any external information.

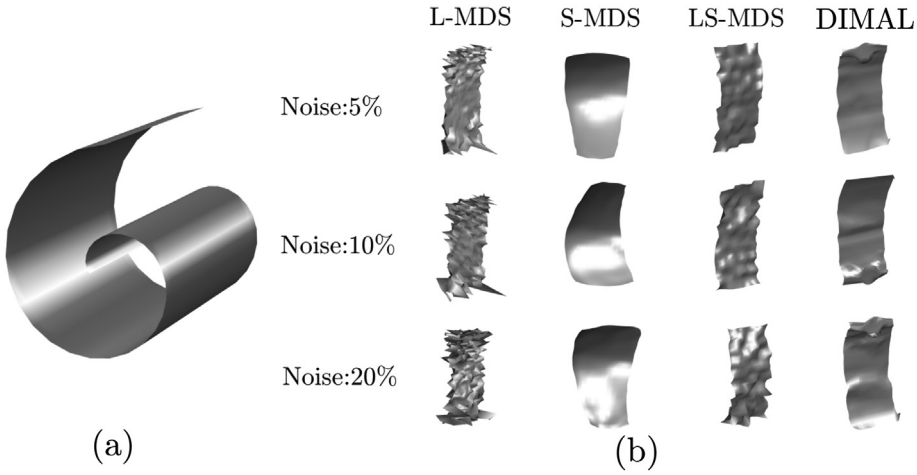


Fig. 6. Stability of the isometric embedding to varying levels of noise. (a) A Swiss-Roll embedded in 3-D. (b) The geodesic distance estimates are corrupted with increasing levels of noise and the outputs of different sparse MDS algorithms: L-MDS [12], S-MDS [39] and S-LSMDS [41] are visualized.

4.4. Comparison with Landmark Isomap

We compare DIMAL to its direct non-parametric competitor: Landmark-Isomap [15]. The main idea of Landmark-Isomap is to perform classical scaling on the inter-geodesic distance matrix of only the landmarks and then to estimate the embeddings of the remaining points using an interpolating formula. The formula uses the estimated geodesic distances of each new point to the selected landmarks in order to estimate its low dimensional embedding.

We generate a training horizon articulation dataset containing 5000 samples generated with parameters sampled from $(\alpha_1,\alpha_2) \sim U([0,0.75] \times [0,0.75])$ and evaluate the outputs on test dataset also of 5000 samples with parameters sampled from $(\alpha_1,\alpha_2) \sim U([0,1] \times [0,1])$, thereby isolating a part of the manifold during training. As in Section 4.3, both the methods are imputed with the same set of landmarks for evaluation.

As depicted in Fig. 9, the output of Landmark-Isomap shows a clustered result due to the lack of non-local data in the geodesic distance calculations for the interpolation. In contrast, our neural network clearly exhibits a better generalization property, even for parts of the manifold that were isolated during training.

4.5. Comparison to Siamese metric learning

We test our method on a more realistic dataset where the constraint of being isometric to a low-dimensional Euclidean space is not necessarily strict. We generate 1369 images obtained by smoothly varying the azimuth and elevation of the camera that is imaging a 3D object. We show in comparison, the visual results and associated training times of DrLim [3] which was trained using the hinge-loss for the Siamese architecture of Fig. 3:

$$\begin{aligned}\mathcal{L}(\Theta) = \sum_p \; & \lambda^{(p)} \; \|\mathcal{F}_\Theta\left(\mathbf{X}_1^{(p)}\right) - \mathcal{F}_\Theta\left(\mathbf{X}_2^{(p)}\right)\| \\ & + \left(1-\lambda^{(p)}\right) \; \max\left\{0, \; \mu - \|\mathcal{F}_\Theta\left(\mathbf{X}_1^{(p)}\right) - \mathcal{F}_\Theta\left(\mathbf{X}_2^{(p)}\right)\|\right\}\end{aligned} \tag{18}$$

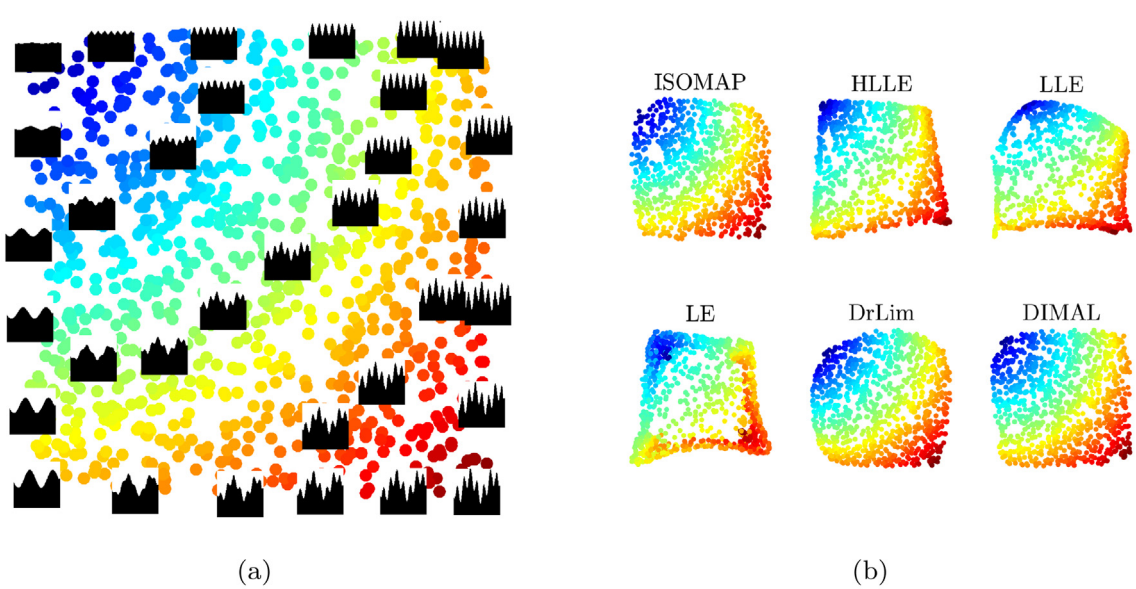


Fig. 7. (a)Visualizing a horizon articulation manifold: samples generated from the image articulation manifold as per Eqs. (16) and (17) with $\omega_1 = 2, \omega_2 = 7$. The color is proportional to the magnitude $\sqrt{\alpha_1^2 + \alpha_2^2}$ (b) Comparing metric preservation properties for different manifold learning algorithms on the image articulation manifold dataset. Isomap [11], HLLE [9], LLE [8], LE [7], DrLim [3]. The proposed method shows maximum fidelity to the ground truth shown in Figure (a).

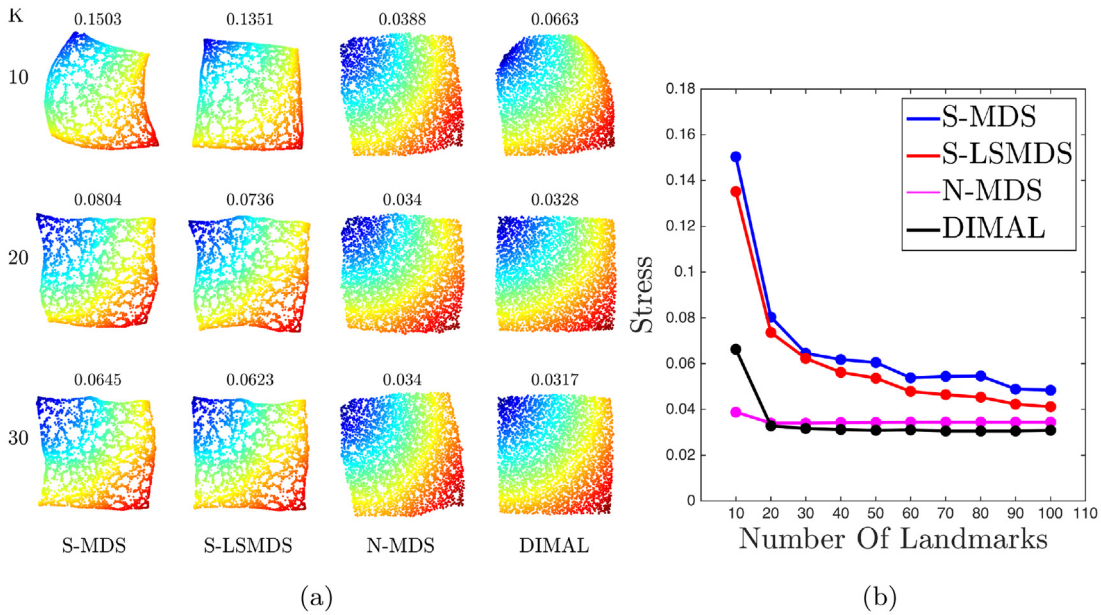


Fig. 8. Comparison with sparse MDS algorithms: (a) Visual evaluation of the interpolation behavior of different sparse MDS frameworks. The rows are in increasing order of landmarks. The titles denote the corresponding stress of that embedding. (b) Stress plots as a function of the number of landmarks K.

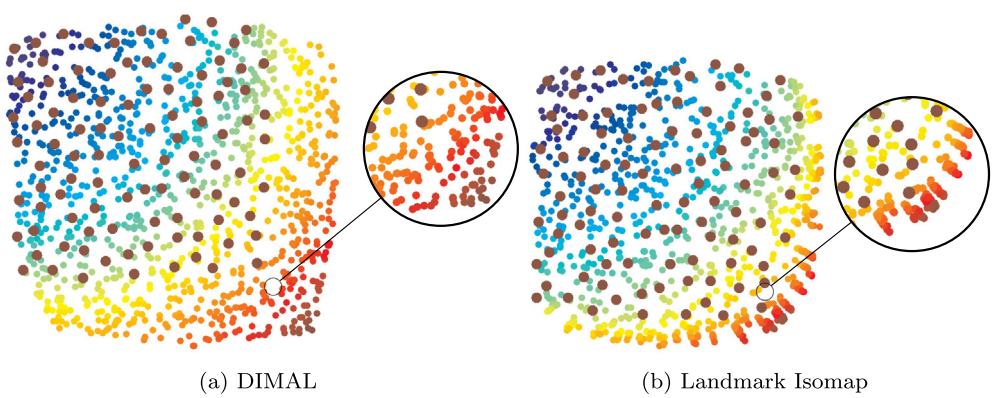


Fig. 9. Evaluation of out-of-sample extensions: Visualizing the non-local generalization properties of our method (top) and Landmark Isomap (bottom). Both algorithms were trained on the same Landmarks (in red) sampled from only a part of the manifold.

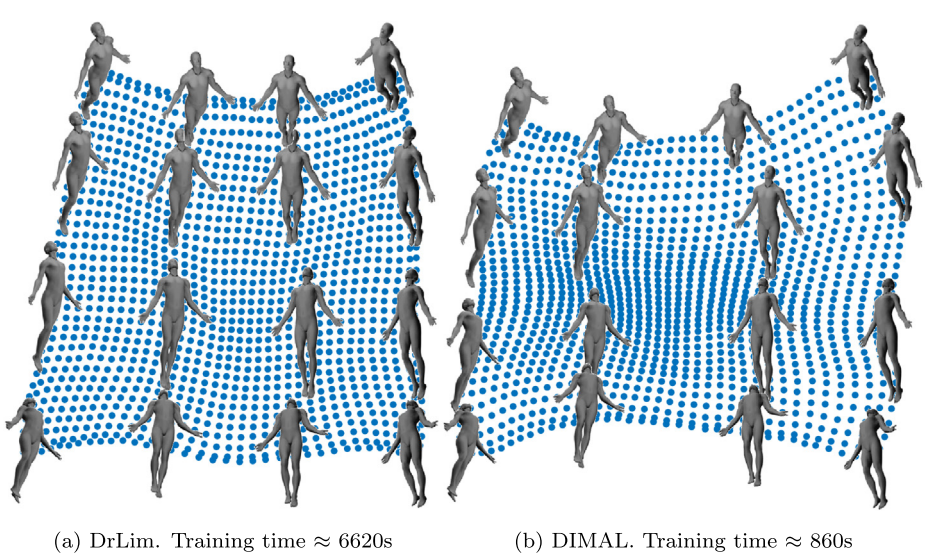


Fig. 10. Camera pose manifold: Embedding results and training times for DrLim [3] and DIMAL. DIMAL shows a faithful result for a much smaller training time.

DrLim requires a considerable training effort, requiring all possible $\binom{1369}{2} = 936396$ pairs whereas DIMAL yields a comparable result in a considerably smaller training time (geodesic distances between only $\binom{600}{2} = 179700$ pairs). We used the same architecture for DIMAL and DrLim for generating the embedding in Fig. 10. DIMAL generates a comparable result with an order of magnitude smaller training time.

5. Extensions

5.1. Topologically Constrained Deep Isometric Embedding

One of the drawbacks of using an all-distance preserving approach along the lines of Eqs. (2) and (3) is that it behaves poorly for manifolds with non-convex properties like holes and boundaries. See Fig. 11. The presence of non-convex structures like holes in the manifold, would mean that *some* of the estimated geodesic distances are not optimal in a distance preserving regime. [24] proposed a topologically constrained MDS scheme (henceforth called TCIE) which filters out potentially problematic distances between distant feature points based on the properties of the geodesics connecting those points and their relative distance to the boundary of the feature manifold. Input distances are first identified as being *consistent* or *inconsistent* (like $z_1 - z_2$ in Fig. 11). Following this identification, the optimization enforcing distance preservation is implemented only for the consistent distances. The three main steps of the TCIE algorithm are (i) detection of boundary points, (ii) detection of a set of consistent geodesics, (iii) solution of a weighted MDS problem. See Algorithm 3.

Algorithm 3. Topologically constrained isometric embedding [24].

1. Compute the $N \times N$ matrix of geodesic distances d_{ij}
2. Detect the boundary points $\partial\mathcal{M}$ of the data manifold
3. Detect a subset of consistent distances ($\overline{P} = \overline{P_1} \cup \overline{P_2}$) according to either:
$$\overline{P}_1 = \{(i,j) : c_{\mathcal{M}}(\mathbf{z}_i, \mathbf{z}_j) \cap \partial\mathcal{M} = \emptyset\}$$
(criterion 1), where $c_{\mathcal{M}}(\mathbf{z}_i, \mathbf{z}_j)$ is the geodesic connecting $\mathbf{z}_i$ and $\mathbf{z}_j$, or (criterion 2)
$$\overline{P}_2 = \{(i,j) : d_{\mathcal{M}}(\mathbf{z}_i, \mathbf{z}_j) \leq d_{\mathcal{M}}(\mathbf{z}_j, \partial\mathcal{M}) + d_{\mathcal{M}}(\mathbf{z}_i, \partial\mathcal{M})\}$$
where $d_{\mathcal{M}}(\mathbf{z}, \partial\mathcal{M}) = \inf_{\mathbf{z}' \in \partial\mathcal{M}} d_{\mathcal{M}}(\mathbf{z}, \mathbf{z}')$ denotes the distance of $\mathbf{z}$ from the boundary of $\mathcal{M}$.
4. Solve the MDS problem for consistent pairs only by minimizing
$$\sigma(\mathbf{X}) = \sum_{i<j} w_{ij}\left(||\mathbf{x}_i - \mathbf{x}_j|| - d_{ij}\right)^2$$
where $w_{ij} = 1$ if $(i,j) \in \overline{P}$ and $w_{ij} = 0$ otherwise, where
$$\overline{P} : \{(i,j) : (i,j) \in \overline{P_1} \ \vee \ (i,j) \in \overline{P_2}\}$$

We implement the TCIE algorithm using our parametric deep learning approach. Instead of minimizing the loss in Eq. (12), we apply the TCIE algorithm to filter out inconsistent distances and minimize the network loss:

$$\mathrm{L}(\Theta) = \sum_p w^{(p)} \left(\parallel \mathcal{S}_\Theta\left(\mathbf{X}_1^{(p)}\right) - \mathcal{S}_\Theta\left(\mathbf{X}_2^{(p)}\right) \parallel - d^{(p)} \right)^2 \tag{19}$$

where $w^{(p)} = 1$ if the pair is consistent and $w^{(p)} = 0$ otherwise as shown in algorithm 3. Fig. 12 shows the results. The network learns to flatten the manifold correctly, by filtering out the inconsistent distances highlighted in Algorithm 3 and [24].

The presence of non convexity's like holes in the isometric embedding problem provides a means to develop interesting and non-trivial *out-of-sample* extension scenarios for our deep learning approach. As mentioned earlier, we train the network to preserve the consistent inter-geodesic distances between a sparse set of landmarks as shown in Fig. 11. Now, in a separate test-case scenario, we generate the 3D points from within the hole, which was excluded from the entire training process. We then pass these points to the trained network and visualize the output of the network for these within-hole points. The results in Fig. 12 show that given sufficient number of landmarks, the network can learn to generalize the isometric map to challenging interpolating and extrapolating circumstances.

5.2. Conformal Isometric Mapping

[11] proposed a solution to the conformal embedding problem by linking it to the isometric embedding problem. A conformal map is locally isometric upto a scale factor. Therefore, in conformal isomap (also called C-Isomap), the authors proposed to estimate the local scale from the observed data and use it to re-scale the local metric of the manifold. After local scale has been factored out, we proceed with the regular isometric embedding problem using classical scaling to recover the low-dimensional flattened embedding. Therefore C-Isomap is essentially an application of the classical scaling algorithm applied to a modified distance matrix which takes the conformal factor into account. A typical test case for this technique is the 3D conformal fishbowl visualized in Fig. 13. The conformal fishbowl has a varying density along the manifold that is sparse at the bottom and dense near the rim of the bowl.

We extend our DIMAL framework to this scenario using the same modified distances as proposed. To test the generalization abilities of both algorithms, we trained on landmarks comprising from only 60% of the height of the fishbowl and test on the flattening of the entire sample set. Fig. 13 demonstrates that DIMAL clearly provides a much better extension than Landmark Isomap which is unable to generalize to non-local points.

5.3. Parallel Transport Unfolding

[13] proposed a scheme to estimate distances on manifolds that are more robust to arbitrary topologies and variable sampling. Specifically, instead of using the Dijkstra's algorithm, geodesic distances of discrete paths over the input pointset are evaluated through parallel transport to offer robustness to poor sampling and arbitrary topology. This refinement in geodesic distance computation is used in conjunction with the classical MDS optimization of Eq. (2), leading to a revised Isomap procedure called Parallel transport Unfolding (PTU).

The advantages of this scheme can be best visualized on the petals dataset reported in [13]. It comprises of points sampled from specific regions on the unit sphere having a petal like structure as shown in Fig. 14. Due to the non-trivial nature of the manifold, a vanilla isomap procedure will not be able to flatten (or unfurl the petal) perfectly. However, the use of modified geodesic distances, using parallel tranport allow for a robust distance computation which achieves an accurate unfurling of the petals, as demonstrated in the Reference embeddings of Fig. 14.

We implemented our method using the input distances from [13]. We trained our DIMAL framework from interpoint *parallel transport distances* for 100 landmarks of a 4-petal manifold and compare it to the

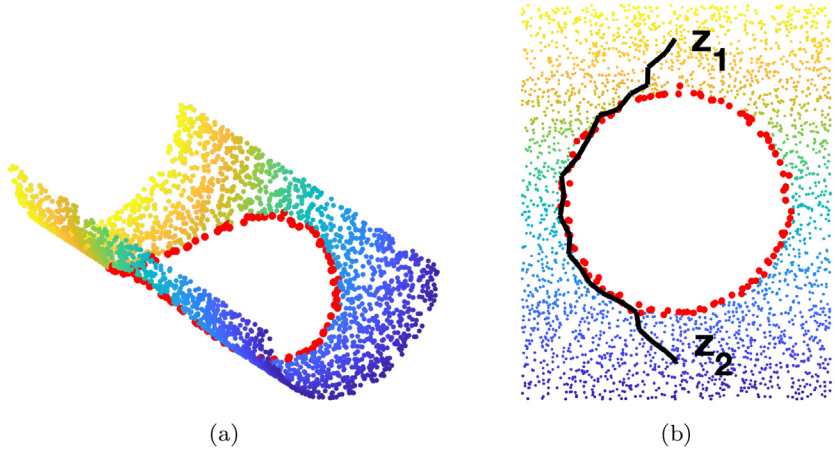


Fig. 11. Example of a manifold (in this case a developable 2D surface embedded in 3D) with holes. Estimating the discrete geodesic path on the manifold would yield curves like $z_1 - z_2$. Distances corresponding to paths like $d(z_1,z_2)$ must **not** be preserved for optimal flattening. [24] provides a scheme to detect boundaries of holes (in red) and avoid using such inconsistent geodesic distances. This is achieved by solving a weighted MDS problem by setting $w_{z1,\,z2} = 0$ in a least squares scaling framework (Eq. (3)).

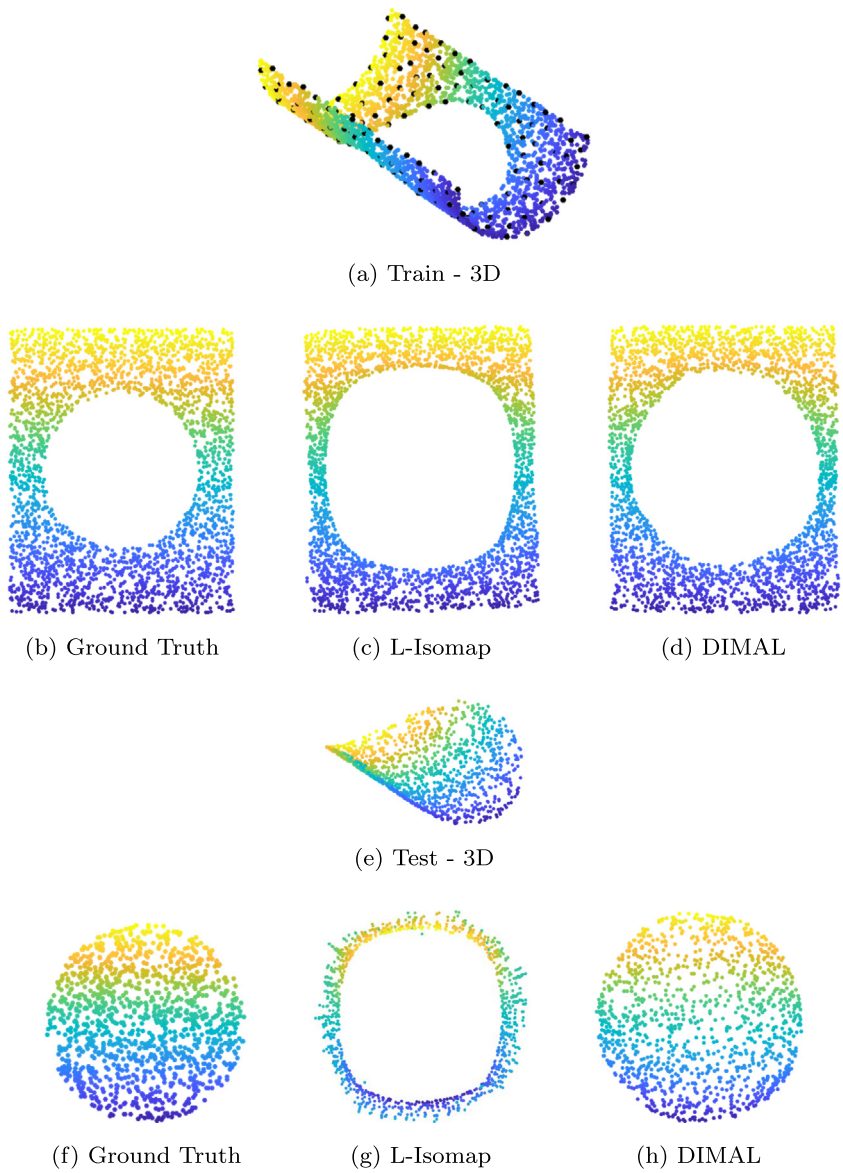
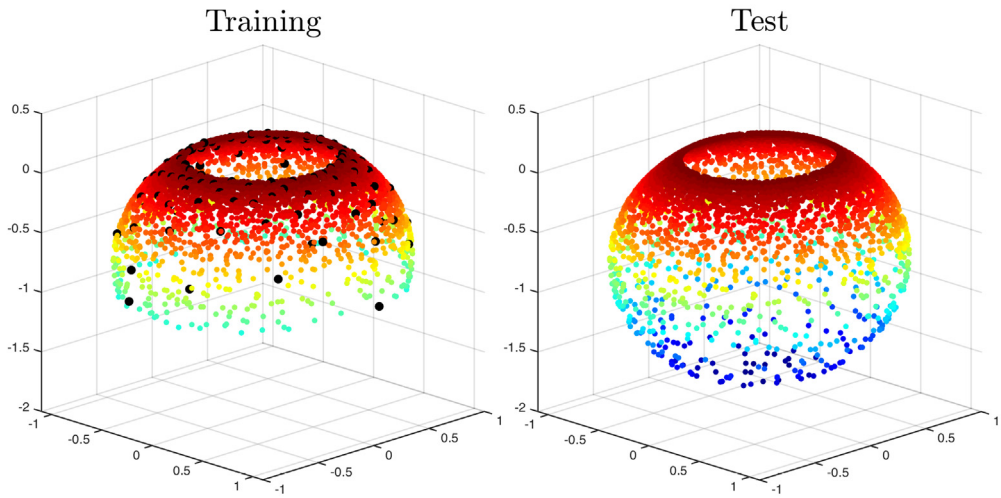
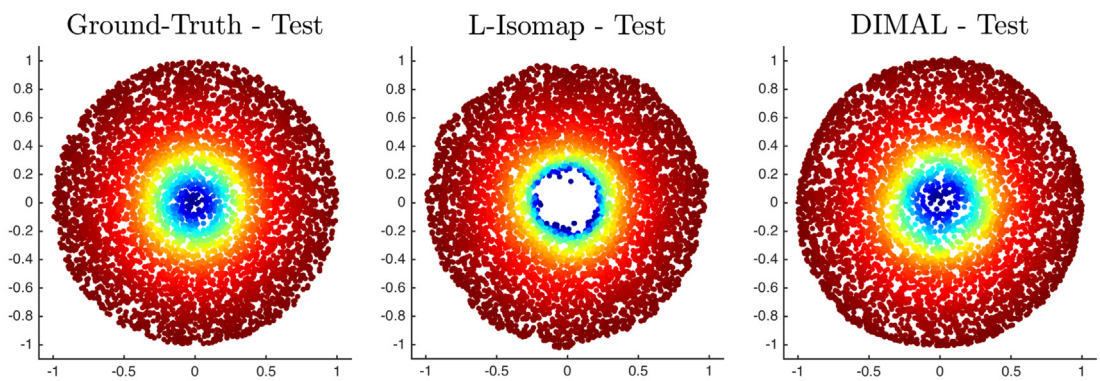


Fig. 12. Topologically Constrained Deep Isometric Embedding: (a) the 3D embedding of a manifold with a hole. The black markers denote the farthest-point samples. The network is trained to minimize the inter-geodesic distance between these markers by **excluding** the inconsistent geodesic distances as mentioned in Algorithm 3. (b)-(d) Visualizing the MDS outputs of the network and L-Isomap with the ground truth. (f)-(h) Visualizing the output of the network for *out-of-sample* points, chosen to be those within the hole. The network generalizes and extrapolates the map learned from (a) to non-local regions in (e).



(a)



(b)

Fig. 13. (a) The Conformal Fishbowl dataset [12], shown with the training and test cases. The training is done on landmarks obtained from only 60% of the height of the fishbowl. (b) The planar embeddings of Landmark Isomap and DIMAL.

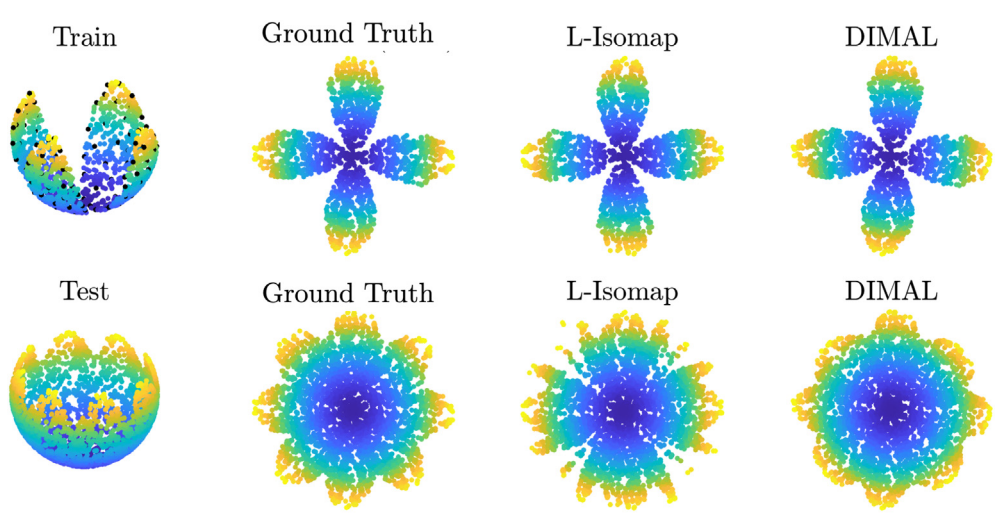


Fig. 14. Unfurling petals: The Petals dataset from [13]. The interpoint distances between the landmark points (in black) generated during an initial *training* phase were computed using the method outlined in [13] using notions of parallel transport. We compare the outputs of the landmark isomap method [12] with DIMAL in two scenarios: training and test. The network shows much improved *generalization* properties in an extrapolation setup, where it can unfurl petals that were not present during training. Both methods are shown the *same* 100 landmark points and their inter-point distances for computation.

Landmark Isomap method using the same input. As shown in the first row of Fig. 14, both the Landmark Isomap algorithm as well DIMAL show good flattening of the training data, faithful to the ground-truth reference.

In order to evaluate *generalization* of the flattening, we prepared a separate, test-case comprising of points from an 8-petal manifold. We then extrapolate the embeddings of this 8-petal manifold using the landmarks of the 4-petal manifold with L-Isomap as well as our DIMAL framework. We see that in significant contrast to the landmark isomap scheme (which was also trained with the same input distances), DIMAL shows much better unfurling of the petals, especially the ones located in non-local regions in the test case.

6. Conclusion

In conclusion, we explored an unsupervised deep learning approach to the isometric embedding problem and rediscovered the MDS procedure using optimization with parametric neural networks. Our approach can be trained with a few landmarks, thereby circumventing computationally intensive formulaic operations like spectral decompositions or out-of-sample geodesic distances. We integrate our approach with a comprehensive list of existing Isomap regimes, and demonstrate improved generalization properties as shown in the various tests that validate the interpolation and extrapolation performance.

CRediT authorship contribution statement

Gautam Pai: Conceptualization, Methodology, Software, Investigation, Writing – review & editing, Formal analysis. **Alex Bronstein:** Conceptualization, Writing – original draft, Investigation, Supervision, Formal analysis. **Ronen Talmon:** Conceptualization, Writing – original draft, Investigation, Supervision, Formal analysis. **Ron Kimmel:** Conceptualization, Writing – original draft, Resources, Investigation, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research is partially supported by the Israel Ministry of Science, grant number 3-14719, the Technion Hiroshi Fujiwara Cyber Security Research Center and the Israel Cyber Directorate. The first author was supported by a fellowship from the Hiroshi Fujiwara Cyber Security Research Center at the Technion. The authors thank anonymous reviewers for insightful feedback, especially for highlighting recent advances in non-linear dimensionality reduction from axiomatic [54,55] as well as a learning [56,57] point of view and applicable to other challenging datasets like hyperspectral images.

References

[1] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, R. Shah, Signature verification using a "siamese" time delay neural network, in, Adv. Neural Inf. Proces. Syst. (1994) 737–744.
[2] S. Chopra, R. Hadsell, Y. LeCun, Learning a similarity metric discriminatively, with application to face verification, Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, 1, IEEE 2005, pp. 539–546.
[3] R. Hadsell, S. Chopra, Y. LeCun, Dimensionality reduction by learning an invariant mapping, Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, 2, IEEE 2006, pp. 1735–1742.
[4] E. Hoffer, N. Ailon, Deep metric learning using triplet network, International Workshop on Similarity-Based Pattern Recognition, Springer 2015, pp. 84–92.
[5] A. Hermans, L. Beyer, B. Leibe, In defense of the triplet loss for person re-identification, arXiv (2021) preprint arXiv:1703.07737.
[6] G. Pai, R. Talmon, A. Bronstein, R. Kimmel, Dimal: Deep isometric manifold learning using sparse geodesic sampling, 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE 2019, pp. 819–828.
[7] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Comput. 15 (6) (2003) 1373–1396.
[8] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, science 290 (5500) (2000) 2323–2326.
[9] D.L. Donoho, C. Grimes, Hessian eigenmaps: locally linear embedding techniques for high-dimensional data, Proc. Natl. Acad. Sci. 100 (10) (2003) 5591–5596.
[10] R.R. Coifman, S. Lafon, A.B. Lee, M. Maggioni, B. Nadler, F. Warner, S.W. Zucker, Geometric diffusions as a tool for harmonic analysis and structure definition of data: diffusion maps, Proc. Natl. Acad. Sci. U. S. A. 102 (21) (2005) 7426–7431.
[11] J.B. Tenenbaum, V. De Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, science 290 (5500) (2000) 2319–2323.
[12] V.D. Silva, J.B. Tenenbaum, Global versus local methods in nonlinear dimensionality reduction, Advances in Neural Information Processing Systems 2003, pp. 721–728.
[13] M. Budninskiy, G. Yin, L. Feng, Y. Tong, M. Desbrun, Parallel transport unfolding: a connection-based manifold learning approach, SIAM J. Appl. Algebra Geom. 3 (2) (2019) 266–291.
[14] Y. Bengio, J.-f. Paiement, P. Vincent, O. Delalleau, N.L. Roux, M. Ouimet, Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering, Advances in Neural Information Processing Systems 2004, pp. 177–184.
[15] V. De Silva, J.B. Tenenbaum, Sparse multidimensional scaling using landmark points, Tech. Rep. 120 (2004).
[16] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828.
[17] H. Gong, C. Pan, Q. Yang, H. Lu, S. Ma, Neural network modeling of spectral embedding, BMVC 2006, pp. 227–236.
[18] G. Mishne, U. Shaham, A. Cloninger, I. Cohen, Diffusion nets, Appl. Comput. Harmon. Anal. 47 (2021) 259–285.
[19] C.K. Chui, H.N. Mhaskar, Deep nets for local manifold learning, Front. Appl. Math. Stat. 4 (2018) 12.
[20] D.P. Kingma, M. Welling, Auto-encoding variational bayes, arXiv (2021) preprint arXiv:1312.6114.
[21] M. Ranzato, C. Poultney, S. Chopra, Y. LeCun, et al., Efficient learning of sparse representations with an energy-based model, Adv. Neural Inf. Proces. Syst. 19 (2007) 1137.
[22] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, L. Bottou, Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion, J. Mach. Learn. Res. 11 (12) (2021).
[23] M. Atzmon, A. Gropp, Y. Lipman, Isometric autoencoders, arXiv (2021) preprint arXiv:2006.09289.
[24] G. Rosman, M.M. Bronstein, A.M. Bronstein, R. Kimmel, Nonlinear dimensionality reduction by topologically constrained isometric embedding, Int. J. Comput. Vis. 89 (1) (2010) 56–68.
[25] J. De Leeuw, P. Mair, Multidimensional Scaling Using Majorization: Smacof in R, Department of Statistics, UCLA, 2021.
[26] E.W. Dijkstra, A note on two problems in connexion with graphs, Numer. Math. 1 (1) (1959) 269–271.
[27] E.L. Schwartz, A. Shaw, E. Wolfson, A numerical solution to the generalized mapmaker's problem: flattening nonconvex polyhedral surfaces, IEEE Trans. Pattern Anal. Mach. Intell. 11 (9) (1989) 1005–1008.
[28] E. Wolfson, E.L. Schwartz, Computing minimal distances on polyhedral surfaces, IEEE Trans. Pattern Anal. Mach. Intell. 11 (9) (1989) 1001–1005.
[29] R. Grossmann, N. Kiryati, R. Kimmel, Computational surface flattening: a voxel-based approach, IEEE Trans. Patt. Analys. Mach. Intellig. 4 (2002) 433–441.
[30] G. Zigelman, R. Kimmel, N. Kiryati, Texture mapping using surface flattening via multidimensional scaling, IEEE Trans. Vis. Comput. Graph. 8 (2) (2002) 198–207.
[31] A. Elad, R. Kimmel, Bending invariant representations for surfaces, Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, vol. 1, IEEE 2001, p. I.
[32] W.S. Torgerson, Multidimensional scaling: I. theory and method, Psychometrika 17 (4) (1952) 401–419.
[33] G. Young, A.S. Householder, Discussion of a set of points in terms of their mutual distances, Psychometrika 3 (1) (1938) 19–22.
[34] J.B. Kruskal, Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis, Psychometrika 29 (1) (1964) 1–27.
[35] R.N. Shepard, The analysis of proximities: multidimensional scaling with an unknown distance function. I, Psychometrika 27 (2) (1962) 125–140.
[36] J.W. Sammon, A nonlinear mapping for data structure analysis, IEEE Trans. Comput. 100 (5) (1969) 401–409.
[37] D. De Ridder, R.P. Duin, Sammon's mapping using neural networks: a comparison, Pattern Recogn. Lett. 18 (11–13) (1997) 1307–1316.
[38] J. Mao, A.K. Jain, Artificial neural networks for feature extraction and multivariate data projection, IEEE Trans. Neural Netw. 6 (2) (1995) 296–317.
[39] Y. Aflalo, R. Kimmel, Spectral multidimensional scaling, Proc. Natl. Acad. Sci. 110 (45) (2013) 18052–18057.
[40] Y. Aflalo, H. Brezis, R. Kimmel, On the optimality of shape and data representation in the spectral domain, SIAM J. Imag. Sci. 8 (2) (2015) 1141–1160.
[41] A. Boyarski, A.M. Bronstein, M.M. Bronstein, Subspace least squares multidimensional scaling, International Conference on Scale Space and Variational Methods in Computer Vision, Springer 2017, pp. 681–693.
[42] G. Shamai, Y. Aflalo, M. Zibulevsky, R. Kimmel, Classical scaling revisited, Proceedings of the IEEE International Conference on Computer Vision 2015, pp. 2255–2263.
[43] G. Shamai, M. Zibulevsky, R. Kimmel, Accelerating the computation of canonical forms for 3d nonrigid objects using multidimensional scaling, Proceedings of the 2015 Eurographics Workshop on 3D Object Retrieval, Eurographics Association 2015, pp. 71–78.

[44] M.M. Bronstein, A.M. Bronstein, R. Kimmel, I. Yavneh, Multigrid multidimensional scaling, Num. Linear Algebra Appl. 13 (2–3) (2006) 149–171.
[45] G. Rosman, A.M. Bronstein, M.M. Bronstein, A. Sidi, R. Kimmel, Fast Multidimensional Scaling Using Vector Extrapolation, Tech. Rep., Computer Science Department, Technion, 2008.
[46] Y. Bengio, Y. LeCun, Scaling learning algorithms towards AI, Large Scale Kernel Machines, MIT Press, 2007.
[47] J. Masci, D. Boscaini, M. Bronstein, P. Vandergheynst, Geodesic convolutional neural networks on riemannian manifolds, Proceedings of the IEEE International Conference on Computer Vision Workshops 2015, pp. 37–45.
[48] D.S. Hochbaum, D.B. Shmoys, A best possible heuristic for the k-center problem, Math. Oper. Res. 10 (2) (1985) 180–184.
[49] A.M. Bronstein, M.M. Bronstein, R. Kimmel, Numerical Geometry of Non-rigid Shapes, Springer Science & Business Media, 2008.
[50] A. Paszke, S. Gross, S. Chintala, Pytorch (2017).
[51] R.J. LeVeque, Finite Difference Methods for Differential Equations, 2021.
[52] D.L. Donoho, C. Grimes, Image manifolds which are isometric to euclidean space, J. Math. Imag. Vision 23 (1) (2005) 5–24.
[53] D. Kingma, J. Ba, Adam: a method for stochastic optimization, International Conference On Learning Representations, 2021.
[54] D. Hong, N. Yokoya, J. Chanussot, J. Xu, X.X. Zhu, Joint and progressive subspace analysis (jpsa) with spatial–spectral manifold alignment for semisupervised hyperspectral dimensionality reduction, IEEE Trans. Cybern. 51 (7) (2020) 3602–3615.
[55] D. Hong, N. Yokoya, J. Chanussot, J. Xu, X.X. Zhu, Learning to propagate labels on graphs: an iterative multitask regression framework for semi-supervised hyperspectral dimensionality reduction, ISPRS J. Photogramm. Remote Sens. 158 (2019) 35–49.
[56] D. Hong, L. Gao, N. Yokoya, J. Yao, J. Chanussot, Q. Du, B. Zhang, More diverse means better: multimodal deep learning meets remote-sensing imagery classification, IEEE Trans. Geosci. Remote Sens. 59 (5) (2020) 4340–4354.
[57] D. Hong, W. He, N. Yokoya, J. Yao, L. Gao, L. Zhang, J. Chanussot, X. Zhu, Interpretable hyperspectral artificial intelligence: when nonconvex modeling meets hyperspectral remote sensing, IEEE Geosci. Rem. Sens. Magaz. 9 (2) (2021) 52–87.
[58] A. Schwartz, R. Talmon, Intrinsic isometric manifold learning with application to localization, SIAM J. Imag. Sci. 12 (3) (2019) 1347–1391.
[59] Gautam Pai, Aaron Wetzler, Ron Kimmel, Learning Invariant Representations of Planar Curves, International Conference on Learning Representations (ICLR) (2017).