Latent Manifold Reconstruction and Representation with Topological and Geometrical Regularization

Ren Wang[1, 2][0009-0005-8637-1180] and Pengcheng Zhou[2, 3][0000-0003-1237-3931]

[1] Department of Biomedical Engineering, Southern University of Science and Technology, Shenzhen 518055, Guangdong, China
[2] Faculty of Life and Health Sciences, Shenzhen University of Advanced Technology, Shenzhen 518107, Guangdong, China
[3] Interdisciplinary Center for Brain Information, Shenzhen-Hong Kong Institute of Brain Science, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, 518055, Guangdong, China
r.wang3@siat.ac.cn

Abstract. Manifold learning aims to discover and represent low-dimensional structures underlying high-dimensional data while preserving critical topological and geometric properties. Existing methods often fail to capture local details with global topological integrity from noisy data or construct a balanced dimensionality reduction, resulting in distorted or fractured embeddings. We present an Autoencoder-based method that integrates a manifold reconstruction layer, which uncovers latent manifold structures from noisy point clouds, and further provides regularizations on topological and geometric properties during dimensionality reduction, whereas the two components promote each other during training. Experiments on point cloud datasets demonstrate that our method outperforms baselines like t-SNE, UMAP, and Topological Autoencoders in discovering manifold structures from noisy data and preserving them through dimensionality reduction, as validated by visualization and quantitative metrics. This work demonstrates the significance of combining manifold reconstruction with manifold learning to achieve reliable representation of the latent manifold, particularly when dealing with noisy real-world data.
Code repository: https://github.com/Thanatorika/mrtg.

Keywords: Manifold Learning, Topological Deep Learning, Geometric Deep Learning.

1 Introduction

1.1 Background

In the area of manifold learning, or nonlinear dimensionality reduction [1], one foundational assumption is the so-called "manifold hypothesis": high dimensional data tend to locate near a latent low-dimensional manifold [2]. One major task of manifold learning is to explicitly transform or embed high-dimensional noisy data into low-dimensional spaces, while keeping the topological and geometric properties of the underlying data manifold.

Several manifold learning methods, including t-SNE [3] and UMAP [4], use the neighborhood of a data point to approximate its tangent space, thus keeping some local structures of neighboring data points. However, these methods do not explicitly regulate the global topology of the latent manifold, and their embeddings are often fractured as clusters. To address this, Moor et. al. [5] introduced Topological AutoEncoders, which preserve the global shape by incorporating a topological regularizer, although local structures still experience distortion or stretching. Also, because of the noise in the raw data, directly applying manifold structural constraints can mislead the regularization terms in the dimensionality reduction process.

1.2 Contribution

In this paper, we propose an AutoEncoder [6] equipped with a manifold reconstruction layer, a topological regularizer, and a geometrical regularizer to achieve improved manifold reconstruction and representation. We use the manifold reconstruction layer to transform data points to the latent manifold, by contracting a point to the expected direction and distance of the closest point on the latent manifold, then we use the AutoEncoder network to learn a low-dimensional latent representation, and utilize the encode-decode loss with topological and geometrical regularizers to optimize the representation quality. As the training is conducted in an end-to-end manner, the hyperparameters for the manifold reconstruction process will also be optimized, resulting in a mutual-promotion between the manifold reconstruction and representation components.

2 Related Work

2.1 Reconstructing the latent manifold.

Manifold fitting is a challenging problem in manifold learning, and it aims to reconstruct the smooth latent manifold from a data set that lies on or near it in the ambient Euclidean space. Early manifold fitting approaches [7, 8] tackle noise-free samples with the Delaunay triangulation technique [9]. Recently, Fefferman et al. [2] validated the feasibility of manifold fitting under noise and proposed their own algorithm [10-13]. Genovese et. al. also provided a series of works from the perspective of minimax risk under Hausdorff distance [14, 15]. An alternative approach is FlatNet by Psenka et al. [16] which flattens the noisy data points can reconstruct the manifold with linearized features. The state-of-the-art manifold fitting algorithm proposed by Yao et al. [17] features local contraction of noisy data points to an estimated direction.

2.2 Topological Regularization

Homology is one of the most important topological invariants, and is a natural target for classifying manifolds. The prevailing method to obtain the homology of a point cloud is *Persistent Homology* [18], which can be computed as a *Persistent Diagram*.

Moor et al. [5] proposed a "Topological Autoencoder" with a *topological signature loss*, which computes the difference between persistent diagrams of the data set and the latent embedding, thus encouraging the latent embedding to have the same topological or global structures as the original point cloud.

Besides the topological signature loss, there are also other methods to extract and compare topological properties of point clouds, including Representation Topology Divergence (RTD) [19, 20] which modifies the computation of the persistent diagram, and Euler Characteristic Transform (ECT) [21, 22] which is a generalization of the Euler characteristics.

2.3 Geometric regularization

An *isometry* between Riemannian manifolds preserves the Riemannian metric, and thus is an equivalence up to geometric properties. By encouraging the mapping to approximate an isometry, one encourages the embedding not to be stretched or distorted, but keeps the geometric or local structures near each point.

The approximation is done by optimizing the Jacobian matrix of the mapping. Philipp et al. [23] proposed a "Geometric Autoencoder" based on a generalized Jacobian determinant, and Lee at al. [24] achieved similar results by a coordinate-free relaxed distortion measure. Hahm et al. [25] promoted the relaxed distortion measure to Latent Diffusion Models.

3 Methodology

Due to the limited capacity and interest of the paper, we will omit most of the math derivations in this section. A more self-contained and detailed formulation of the math notions can be found in references [5, 26, 27].

3.1 The Model Pipeline

The whole model is composed of a Manifold Reconstruction Layer (MRL), an Encoder and a Decoder, as depicted in Figure 1.

The training loss function is the Autoencoder loss between input data and decoded output, so the MRL is integrated in our training optimization. Note that we do not compute the loss between the reconstructed manifold and decoded output, otherwise the representation process will totally subject to the quality of the reconstructed manifold. In addition, two training regularizers are computed between the reconstructed manifold and the embedding, encouraging the preservation of local and global structures in the dimensionality reduction process.

The total loss function is given by

$$\mathcal{L} = \lambda_{AE}\mathcal{L}_{AE}\left(X, \hat{X}\right) + \lambda_{topo}\mathcal{L}_{topo}(Y, Z) + \lambda_{geom}\mathcal{L}_{geom}(Y, Z), \tag{1}$$

where λ_{AE}, λ_{topo} , and λ_{geom} are predefined hyperparameters, and X, Y, Z, $\hat{X}$ denotes the input noisy data set, the reconstructed manifold, the low-dimensional embedding, and the AutoEncoder output result, respectively.

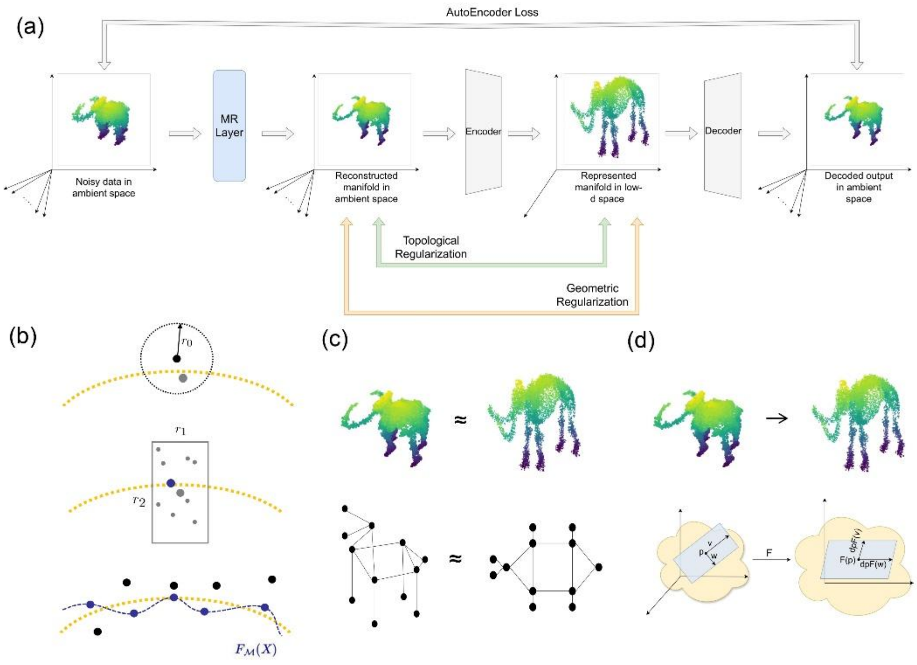
(a)
AutoEncoder Loss
Noisy data in ambient space
MR Layer
Reconstructed manifold in ambient space
Encoder
Represented manifold in low-d space
Decoder
Decoded output in ambient space
Topological Regularization
Geometric Regularization
(b)
r_0
r_1
r_2
$F_{\mathcal{M}}(X)$
(c)
≈
≈
(d)
→
p
v
w
F
F(p)
dpF(v)
dpF(w)

Fig. 1. (a) An illustration of our model pipeline, (b) an illustration of our manifold reconstruction algorithm, (c) an illustration of the topological regularizer based on persistent homology, (d) an illustration of the geometric regularizer based on scaled isometry.

3.2 Assumptions on the data distribution

In this paper, we treat our data as i.i.d. points residing on a latent manifold $\mathcal{M}$ with extra Gaussian noise ξ, and the manifold is regarded as a sub-manifold of Euclidean space $\mathbb{R}^D$. The set of data points $X \subset \mathbb{R}^D$ can be written in the form of

$$x = x_{\mathcal{M}} + \xi, \forall x \in X, \tag{2}$$

Where $x_{\mathcal{M}} \sim \omega_{\mathcal{M}}$ and $\xi \sim \phi_{\sigma}$. Here $\omega_{\mathcal{M}}$ is uniform distribution on $\mathcal{M}$, and ϕ_{σ} is Gaussian distribution with standard deviation σ. The probability distribution can be written as a convolution

$$v(x) = \int_{\mathcal{M}} \omega_{\mathcal{M}}(t)\phi_{\sigma}(x-t)dt. \qquad (3)$$

3.3 The Manifold Reconstruction Layer

We aim to develop an estimator for the latent manifold $\mathcal{M}$ using the data sample set X. Then for each $x \in X$, we employ a two-step procedure: (i) we identify the contraction direction, and (ii) we estimate the contracted point.

Determine the contraction direction. Suppose $x^* = argmin_{x_{\mathcal{M}} \in \mathcal{M}} \|x - x_{\mathcal{M}}\|$. We consider a D-dimensional ball $\mathcal{B}_D(x, r_0)$ with radius $r_0 \coloneqq C_0\sigma$, where C_0 is a given constant. We can estimate the direction of x^* as $F(x) = \sum \alpha_i(x)x_i$, and the weights α_i's are given as

$$\tilde{\alpha}_i(x) = \begin{cases} \left(1 - \dfrac{\|x - x_i\|_2^2}{r_0^2}\right), & \|y - y_i\|_2 \le r_0; \\ 0, & otherwise, \end{cases}$$

$$\tilde{\alpha}(x) = \sum_{i \in I_x} \tilde{\alpha}_i(x), \; \alpha_i(x) = \frac{\tilde{\alpha}_i(x)}{\tilde{\alpha}(x)}. \qquad (4)$$

where I_x is the set of indices for data points $x_i \in X \cap \mathcal{B}_D(x, r_0)$, and $k > 2$ is a fixed integer guaranteeing a twice-differentiable smoothness.

Determine the contracted point. We consider a cylinder region $\mathbb{V}_x = \mathcal{B}_{D-1}(x, r_1) \times \mathcal{B}_1(x, r_2)$ with $r_1 \coloneqq C_1\sigma$ and $r_2 \coloneqq C_2\sigma\sqrt{\log\left(\frac{1}{\sigma}\right)}$. We let $\widehat{U} = \frac{(F(x)-x)(F(x)-x)^T}{||F(x)-x||_2^2}$. For a data point $x_j \in \mathbb{V}_x$ ($j \in J_x$), we define $u_j = \widehat{U}(x_j - x)$, $v_j = x_j - x - u_i$. Then we can get an estimation $y = G(x) = \sum \beta_j(x)x_j$ with weights given as

$$w_u(u_j) = \begin{cases} 1, & \|u_j\|_2 \le \frac{r_2}{2}; \\ \left(1 - \left(\frac{2\|u_j\|_2 - r_2}{r_2}\right)^2\right)^{\mathrm{k}}, & \|v_j\|_2 \in \left(\frac{r_2}{2}, r_2\right); \\ 0, & otherwise, \end{cases}$$

$$w_v(v_j) = \begin{cases} \left(1 - \frac{\|v_j\|_2^2}{r_1^2}\right)^k, & \|v_j\|_2 \le r_1; \\ 0, & otherwise \end{cases}$$

$$\tilde{\beta}_j(x) = w_u(u_j)w_v(v_j), \; \tilde{\beta}(x) = \sum_{j \in J_x} \tilde{\beta}_j(x), \; \beta_j(x) = \frac{\tilde{\beta}_j(x)}{\tilde{\beta}(x)}. \qquad (5)$$

The MRL has four hyperparameters: initial values of radii r_0, r_1, r_2, and a smoothness parameter k, affecting the computation of contraction directions and weights of the transformation matrix of each data point. An illustration of the process is shown in Figure 1. For more details please refer to [17].

3.4 The Topological Regularizer.

Persistent diagrams. Persistent Homology (PH) is computed first by constructing a *Vietoris-Rips complex*. Given a distance scale ϵ, the Vietoris-Rips complex of X, denoted by $\mathcal{K}_\epsilon(X)$, contains all simplices (i.e. subsets) of X whose elements have mutual distances smaller than or equal to ϵ. Vietoris-Rips complexes form a *filtration*, i.e. $\mathcal{K}_{\epsilon_1} \subseteq \mathcal{K}_{\epsilon_2}$ for $\epsilon_1 \leq \epsilon_2$. With this quality, we can track changes in homology groups of Vietoris-Rips complexes as ϵ increases from 0. In the increasing process, the connectivity among points in X changes accordingly, and each connectivity state has "birth and death times" denoted by two values (a, b) of ϵ. We can track the creation and destruction of n-dimensional topological features in a diagram $\mathcal{D}_n$, called the n-th *persistent diagram*.

The topological signature loss. In our method, the PH information of X is stored in a tuple($\mathcal{D}^X \coloneqq \{\mathcal{D}_0, \mathcal{D}_1, \cdots\}, \pi^X \coloneqq \{\pi_0, \pi_1, \cdots\}$), where the first component is the persistent diagrams, and the second component is the *persistent pairings* composing of pairs of edges (e, e') that each create or destroy a topological feature $(a, b) \in \mathcal{D}_n$ when they are newly connected as ϵ increases.

The values of the persistent diagram can be retrieved by subsetting the mutual distance matrix A^X of points in X with the edge indices provided by the persistent pairings, written as

$$A^X[\pi^X] = \{(|e|, |e'|), \forall (e, e') \in \pi^X\} \in \mathbb{R}\text{^}\{|\pi^X|\}. \tag{6}$$

For the manifold Y and embedding Z, we define a topological signature loss of two terms, each denoting the loss measuring the changes of topological features in one of the two Vietoris-Rips complexes compared to those in the other:

$$\mathcal{L}_{topo} = \mathcal{L}_{Y\to Z} + \mathcal{L}_{Z\to Y} = \frac{1}{2}\|A^Y[\pi^Y] - A^Z[\pi^Z]\|^2 + \frac{1}{2}\|A^Z[\pi^Z] - A^Y[\pi^Z]\|^2. \tag{7}$$

A lower topological signature loss means the topological features are better shared between the manifold Y and embedding Z.

3.5 The Geometric Regularizer

Scaled isometry. As its name suggests, the relaxed distortion measure indicates the degree of distortion of a mapping compared to a *scaled isometry*. Let $\mathcal{M}$ be a Riemannian manifold of dimension m with local coordinates $x \in \mathbb{R}^m$ and Riemannian metric $G(x) \in \mathbb{R}^{m\times m}$, and $\mathcal{N}$ be a Riemannian manifold of dimension n with $z \in \mathbb{R}^n$ and $H(z) \in \mathbb{R}^{n\times n}$. Let $f\colon \mathcal{M} \to \mathcal{N}$, then its differential is denoted by the Jacobian matrix $J_f(x) = \frac{\partial f}{\partial x}(x) \in \mathbb{R}^{n\times m}$. f is said to be a scaled isometry if it meets $G(x) = CJ_f(x)^T H\big(f(x)\big)J_f(x),\ \forall x \in \mathbb{R}^m$, for a constant C. An illustration is in Figure 1.

Relaxed distortion measure. At a point $x \in \mathcal{M}$, we consider the characteristic values of the *pullback metric* $J_f^T H J_f$ relative to G, i.e. the m eigenvalues $\lambda_1, \cdots, \lambda_m$ of $J_f(x)^T H\big(f(x)\big) J_f(x)^T G^{-1}(x)$. These eigenvalues are invariant under coordinate transformations, and let $S(\lambda_1, \cdots, \lambda_m)$ be any symmetric function, the integral $\mathcal{I}_S(f) \coloneqq \int_{\mathcal{M}} S\big(\lambda_1(x), \cdots, \lambda_m(x)\big) d\mu(x)$ is also coordinate-invariant. Here μ is a positive measure on $\mathcal{M}$.

Here we design a family of coordinate-invariant functionals $\mathcal{F}$ that measures how far $f\colon \mathcal{M} \to \mathcal{N}$ is from being a scaled isometry over the support of μ:

$$\mathcal{F}(f) = \int_{\mathcal{M}} \left(\sum_{i=1}^{m} h\left(\frac{\lambda_i(x)}{\int_{\mathcal{M}} S(\lambda_1(x), \cdots, \lambda_m(x)) d\mu(x)} \right) \right) d\mu(x), \tag{8}$$

where h is some convex function and S is some symmetric function. With this given functional, we can calculate and minimize the relaxed distortion measure of our encoder function.

The geometric regularizer is given by

$$\begin{aligned} \mathcal{L}_{geom} &= \mathcal{F}\big(f_\theta; P_\phi\big) = \mathbb{E}_{y\sim P_\phi}\left(\sum_{i=1}^{D} \left(\frac{\lambda_i(y)}{\mathbb{E}_{y\sim P_\phi}\left(\sum_i \frac{\lambda_i(y)}{D} \right)} - 1 \right)^2 \right) \\ &= \mathrm{D}^2 \frac{\mathbb{E}_{y\sim P_\phi}\left(Tr\left(H_\theta^2(y) \right) \right)}{\mathbb{E}_{y\sim P_\phi}\left(Tr\left(H_\theta(y) \right) \right)^2} - \mathrm{D} \qquad (9) \end{aligned}$$

where $f_\theta\colon \mathbb{R}^D \to \mathbb{R}^d, y \mapsto z$ is the encoder function, P_ϕ is the distribution of Y in $\mathbb{R}^D$: $y \sim P_\phi$, $\lambda_i\,(y)$ are eigenvalues of $H_\theta(y) = J_{f_\theta}^T(y) H\big(f_\theta(y)\big) J_{f_\theta}(y)$. Here $H(z)$ is the metric assigned to some data point $z \in Z$ in the embedding, and $H_\theta(y)$ is the pullback metric assigned to the corresponding $y \in Y$ in the reconstructed manifold.

4 Experiments

4.1 Datasets.

In this paper, we implement our method on Euclidean point cloud datasets to validate our approach for manifold representation learning. Datasets include a noisy 3D Swiss roll manifold with a hole, produced from the dataset in Scikit-learn [28]; a noisy 3D mammoth dataset, produced from the mammoth dataset shown in the "Understanding UMAP" project [29]; a 100-dimensional spheres dataset from the "Topological Autoencoder" paper [5]; and a subset of the 3D point cloud dataset PartNet [30].

4.2 Baselines

We train the following baseline manifold learning methods on the datasets, and compare their performance with our method by showing the 2-dimensional embedding shapes, and evaluating with dimensionality reduction benchmarking metrics adopted from [5, 23].

A vanilla Autoencoder and a Topological Autoencoder are trained with the same input data, model structure, loss function, and learning rate as our model.
Standard PCA, t-SNE, and UMAP models fit-transform the same data as our model, all with default parameters.

Table 1. Local and global metrics for our method and manifold learning baselines. PartNet scores are averaged over 12 objects. Exponential digits in scientific notations are omitted. **Bold** values are winners, and underlined are runner ups.

Dataset	Model	Local			Global		
		$KL_{0.1}$(↓)	kNN(↑)	Trust(↑)	KL_{100}(↓)	RMSE(↓)	Spear(↑)
Swiss roll	Ours	**2.19**	4.69	8.98	1.12	**2.55**	**7.60**
	AE	3.07	4.17	8.84	1.13	2.56	7.41
	Topo AE	2.90	5.34	8.79	**1.11**	4.07	7.59
	t-SNE	2.95	**8.27**	**9.98**	2.68	503	4.79
	UMAP	3.20	7.80	9.96	4.77	114	2.50
Mammoth	Ours	**2.15**	4.93	9.61	**1.03**	1.63	**9.76**
	AE	21.6	2.05	9.38	4.25	1.38	7.62
	Topo AE	2.94	4.96	9.65	1.75	**1.21**	9.64
	t-SNE	27.5	**7.23**	**9.98**	12.5	871	8.13
	UMAP	10.9	6.62	9.97	8.52	149	8.51
PartNet	Ours	**8.18**	5.40	9.24	**6.02**	3.44	**8.81**
	AE	18.8	5.08	9.25	16.8	10.1	8.24
	Topo AE	18.3	5.43	9.32	15.6	**2.21**	8.61
	t-SNE	16.5	**7.85**	**9.96**	11.8	526	8.76
	UMAP	16.0	7.50	9.94	13.0	93.8	8.58
Spheres	Ours	4.78	1.76	5.78	10.1	2.81	-1.85
	AE	4.90	2.23	6.18	9.28	6.48	**27.7**
	Topo AE	3.97	1.53	5.64	**9.12**	2.82	5.21
	PCA	7.49	2.16	6.28	15.4	2.31	3.45
	t-SNE	5.95	**3.50**	**7.33**	12.7	**1.43**	7.38
	UMAP	6.17	2.59	6.83	14.2	2.40	4.26

4.3 Architecture and training

For 3D point cloud data, we build our AE model with a "3-2-2" encoder and a "2-2-3" decoder, with GELU [31] activations. For spheres data in the 101-D space, the model has a "101-64-32-16-8-4-2" encoder and a "2-4-8-16-32-64-101" decoder. MRL hyperparameters are $(r_0, r_1, r_2, k) = (1.0,0.01,1.0,3)$ for all experiments. The loss term weights $(\lambda_{AE}\lambda_{topo}, \lambda_{geom})$ are (1,1,5) for the Swiss roll dataset, (1,0.5,0.5) for the mammoth dataset, (1,0.01,0.01) for the PartNet dataset, and (1,1,1) for the Spheres dataset. The Autoencoder loss is set as the MSE loss between the model input and output.

4.4 Evaluation

We adopted the evaluation metrics from [5, 23] for measuring the similarity of local and global structures between the original point cloud and the embedding. The metrics are separated into two groups: $KL_{0.1}$, kNN and Trust are local metrics, while KL_{100} , RMSE and Spear are global metrics.

In Table 1, we can see that our method keeps a balance between local and global structures. Worth noting is that in most of the metrics rankings, our model outperforms Vanilla AE.

4.5 Visualization

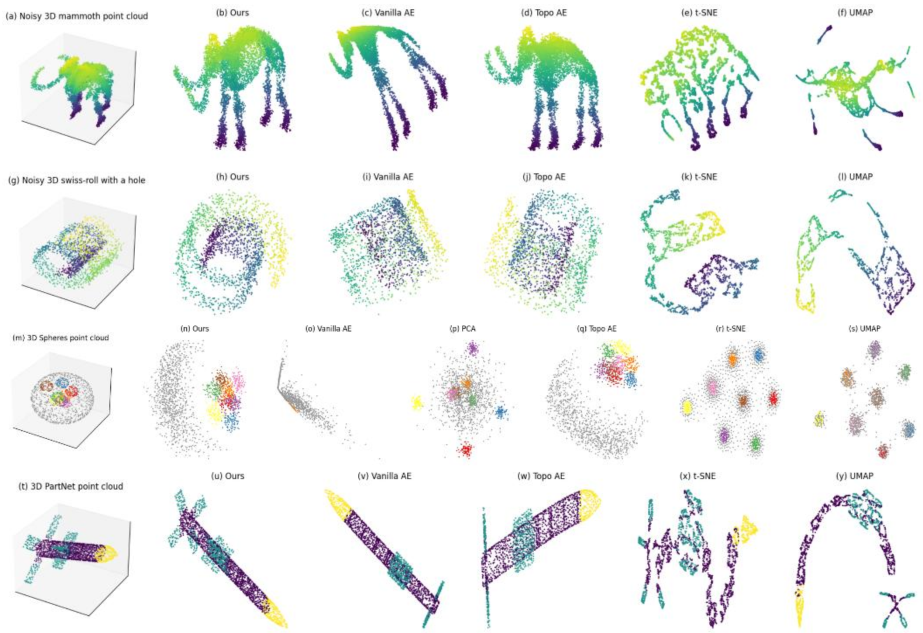

Fig. 2. Our proposed method best preserves global and local structures of the latent manifolds underlying noisy data. The Spheres dataset is composed of nine 100-dimensional spheres in 101-dimensional Euclidean space, where eight small spheres are located inside a large sphere. The PartNet dataset is composed of 12 distinct 3D point clouds, each indicating a real-life object. The points on an object are labeled by a ground truth semantic segmentation, where different colors indicate different "parts" of the object.

We can see that Vanilla AE reduces the manifolds as a whole to 2D space, but the embeddings lost many details (two tusks of the mammoth, the hole on the Swiss roll, and spreading tails of the rocket), and are heavily distorted. The Topo AE embeddings keep their global shape, but their local structures are ignored, as embeddings are "folded" into 2D space, producing non-stereoscopic self-covering embeddings. T-SNE and UMAP embeddings primarily focus on local structural information, while the manifolds are torn apart with unrecognizable global shapes. Our proposed method keeps the most local and global shape details, and preserves the proportions of the manifold not distorted or torn apart, thus leaving a stereoscopic representation of the point cloud.

5 Ablation Study

To validate the necessity of our model ingredients, we perform an ablation study on the model pipeline with the Swiss roll dataset. We compared the performance of ablated models as:

- Final model: Manifold Reconstruction Layer + Autoencoder + two regularizers;
- Topo AE: Autoencoder + Topological Regularizer;
- Geom AE: Autoencoder + Geometric Regularizer;
- Topo-geom AE: Autoencoder + two regularizers;
- MR AE: Manifold Reconstruction Layer + Autoencoder.

Evaluation metrics are shown in Table 2. In the "Point Cloud vs Manifold" group of the table, our model beats MR AE in all the metrics, indicating that the representation component (the Autoencoder) improves the performance of the Manifold Reconstruction Layer during training. Our model's performance in "Manifold vs Embedding" also exceeds Topo-geom AE's in "Point Cloud vs Embedding" in four out of six metrics, which means the reconstructed manifold discovers local and global properties out of the noisy point cloud, which can be more easily carried to low-dimensional space under the same dimensionality reduction method.

In summary, the manifold reconstruction and representation regularization components are both virtuous in our implementation, and they promote each other's performance during the training optimization, "clearer answers are easier to be written down with the same pen, and clearer writing can improve scores earned by the same answer", as we previously claimed.

6 Limitations and Future Directions

Supervision of Manifold Reconstruction. The Manifold Reconstruction process alters the data, while the Autoencoder faithfully embeds its results. In our implementation, we improve the MRL's performance through the encode-decode feedback, but this may not be a sufficient supervision, and the learning process could be unstable. In the future, we aim to come up with better ways to supervise the manifold reconstruction process.

Non-Euclidean data. When dealing with data types other than point clouds in Euclidean space, we need to modify our manifold reconstruction method: if the coordinate axes are correlated, we need to first orthogonalize them; when the data are sequential, we need to integrate the sequence information into our distance function; and if we have external labels for data points, we should learn a distance metric with them.

Table 2. The evaluation metrics of our ablation study. Evaluation and comparison are conducted in three groups: (1) Noisy point clouds versus low-d embeddings, (2) reconstructed manifolds versus low-d embeddings, and (3) noisy point clouds versus reconstructed manifolds. Bold values indicate winners from each group, and underlined are winners from the comparison between Topo-geom AE in "" and the Final model in "Manifold vs Embedding".

Test	Model	Local			Global		
		$KL_{0.1}$(↓)	kNN(↑)	Trust(↑)	KL_{100}(↓)	RMSE(↓)	Spear(↑)
PC. vs E.	Final	**1.95**	**5.37**	**9.03**	9.69	3.56	7.75
	AE	2.54	5.03	9.02	13.3	11.9	6.98
	Topo AE	2.67	5.25	8.88	9.80	4.05	7.84
	Geom AE	2.44	5.08	8.93	**8.39**	4.05	**7.93**
	T-G AE	2.67	5.30	8.85	9.62	4.06	7.83
	MR AE	2.78	4.61	8.82	13.9	**1.72**	7.13
M. vs E.	Final	**2.00**	**5.43**	**9.02**	**1.02**	**3.43**	**7.74**
	MR AE	21.6	2.05	9.38	4.25	1.38	7.62
PC. vs M.	Final	**1.39**	**8.92**	**10.0**	**8.56**	**2.54**	**9.93**
	MR AE	27.8	4.61	8.82	139	17.2	7.13

References

1. M. Meila and H. Zhang. Manifold learning: what, how, and why. Annual Review of Statistics and Its Application, 11:393–417, 2024.
2. C. Fefferman, S. Mitter, and H. Narayanan. Testing the manifold hypothesis. Journal of the American Mathematical Society, 29:983–1049, 2016.
3. L. van der Maaten and G. Hinton. Visualizing data using t-sne. Journal of Machine Learning Research, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.
4. L. McInnes, J. Healy, N. Saul, and L. Großberger. Umap: Uniform manifold approximation and projection. Journal of Open Source Software, 2018.
5. M. Moor, M. Horn, B. Rieck, and K. Borgwardt. Topological autoencoders. In Proceedings of the 37th International Conference on Machine Learning, volume 119, pages 7045–7054. PMLR, 13–18 Jul 2020.
6. I. Goodfellow, Y. Bengio, and A. Courville. Deep Learning. MIT Press, 2016. http://www.deeplearningbook.org.
7. J.-D. Boissonnat, L. J. Guibas, and S. Y. Oudot. Manifold reconstruction in arbitrary dimensions using witness complexes. Discrete & Computational Geometry, 42:37–70, 2009.
8. Y.-C. Chen, C. R. Genovese, and L. Wassersman. Asymptotic theory for density ridges. The Annals of Statistics, 43:1896–1928, 2015.
9. D.-T. Lee and B. J. Schachter. Two algorithms for constructing a delaunay triangulation. International Journal of Computer and Information Sciences, 9:219–242, 1980.
10. C. Fefferman, S. Ivanov, Y. Kurylev, M. Lassas, and H. Narayanan. Fitting a putative manifold to noisy data. Conference On Learning Theory, pages 688–720, 2018.
11. C. Fefferman, S. Ivanov, M. Lassas, J. Lu, and H. Narayanan. Reconstruction and interpolation of manifolds. i: The geometric whitney problem. oundations of Computational Mathematic, 20:1035–1133, 2020.

12. C. Fefferman, S. Ivanov, M. Lassas, J. Lu, and H. Narayanan. Reconstruction and interpolation of manifolds ii: Inverse problems for riemannian manifolds with partial distance data, 2021. URL https://arxiv.org/abs/2111.14528.
13. C. Fefferman, S. Ivanov, M. Lassas, and H. Narayanan. Fitting a manifold of large reach to noisy data. Journal of Topology and Analysis, 0(0):1–82, 0. doi: 10.1142/S1793525323500012.
14. C. Genovese, M. Perone-Pacifico, I. Verdinelli, and L. Wasserman. Minimax manifold estimation. Journal of Machine Learning Research, 13:1263–1291, 2012.
15. C. Genovese, M. Perone-Pacifico, I. Verdinelli, and L. Wasserman. Manifold estimation and singular deconvolution under hausdorff loss. The Annals of Statistics, 40:941–963, 2012.
16. M. Psenka, D. Pai, V. Raman, S Sastry, and Y Ma. Representation learning via manifold flattening and reconstruction. Journal of Machine Learning Research, 25(132):1–47, 2024. URL http://jmlr.org/papers/v25/23-0615.html.
17. Z. Yao, J. Su, B. Li, and S.-T. Yau. Manifold fitting, 2023. URL https://arxiv.org/abs/2304.07680.
18. H. Edelsbrunner and J. Harer. Persistent homology—a survey. Discrete & Computational Geometry, 453, 01 2008.
19. S. Barannikov, I. Trofimov, N. Balabin, and E. Burnaev. Representation topology divergence: A method for comparing neural network representations. In International Conference on Machine Learning, 2022.
20. I. Trofimov, D. Cherniavskii, E. Tulchinskii, N. Balabin, S. Barannikov, and E. Burnaev. Learning topology-preserving data representations. In International Conference on Learning Representations, 2023. URL https://openreview.net/forum?id=lIu-ixf-Tzf.
21. E. Munch. An invitation to the euler characteristic transform, 2023. URL https://arxiv.org/abs/2310.10395.
22. E. Röell and B. Rieck. Differentiable euler characteristic transforms for shape classification. In The Twelfth International Conference on Learning Representations, 2024. URL https://openreview.net/forum?id=MO632iPq3I.
23. N. Philipp, D. Sebastian, and F. A. Hamprecht. Geometric autoencoders - what you see is what you decode. In Proceedings of the 40th International Conference on Machine Learning, volume 202, pages 25834–25857. PMLR, 23–29 Jul 2023.
24. Y. Lee, S. Yoon, M. Son, and F. C. Park. Regularized autoencoders for isometric representation learning. In International Conference on Learning Representations, 2022. URL https://openreview.net/forum?id=mQxt8l7JL04.
25. J. Hahm, J. Lee, S. Kim, and J. Lee. Isometric representation learning for disentangled latent space of diffusion models, 2024. URL https://arxiv.org/abs/2407.11451.
26. J. M. Lee. Introduction to Smooth Manifolds. Springer, 2012.
27. H. Federer. Geometric Measure Theory. Springer, 2014.
28. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.
29. A. Coenen and A. Pearce. Understanding umap. URL https://pair-code.github.io/understanding-umap/.
30. K. Mo, S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas, and H. Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.
31. H. Dan and G. Kevin. Bridging nonlinearities and stochastic regularizers with gaussian error linear units, 2017. URL https://openreview.net/forum?id=Bk0MRI5lg.