nature communications

Article

https://doi.org/10.1038/s41467-025-60434-9

Assessing and improving reliability of neighbor embedding methods: a map-continuity perspective

Received: 11 November 2024

Accepted: 23 May 2025

Published online: 30 May 2025

Check for updates

Zhexuan Liu[1], Rong Ma[2,3] & Yiqiao Zhong[1]

Visualizing high-dimensional data is essential for understanding biomedical data and deep learning models. Neighbor embedding methods, such as t-SNE and UMAP, are widely used but can introduce misleading visual artifacts. We find that the manifold learning interpretations from many prior works are inaccurate and that the misuse stems from a lack of data-independent notions of embedding maps, which project high-dimensional data into a lower-dimensional space. Leveraging the leave-one-out principle, we introduce LOO-map, a framework that extends embedding maps beyond discrete points to the entire input space. We identify two forms of map discontinuity that distort visualizations: one exaggerates cluster separation and the other creates spurious local structures. As a remedy, we develop two types of point-wise diagnostic scores to detect unreliable embedding points and improve hyper-parameter selection, which are validated on datasets from computer vision and single-cell omics.

Data visualization plays a crucial role in modern data science, as it offers essential and intuitive insights into high-dimensional datasets by providing low-dimensional embeddings of the data. For visualizing high-dimensional data, the last two decades have witnessed the rising popularity of t-SNE[1] and UMAP[2], which are extensively used in, e.g., single-cell analysis[3–5] and feature interpretations for deep learning models[6,7].

The neighbor embedding methods[8,9] are a family of visualization methods, which include t-SNE, UMAP, and LargeVis[10] as popular examples, that determine embedding points directly by solving a complicated optimization algorithm to minimize the discrepancy between similarities of input points and those of the corresponding low-dimensional points. Given input data $\mathbf{x}_1, \ldots, \mathbf{x}_n$, a neighbor embedding algorithm $\mathcal{A}$ computes the points $(\mathbf{y}_1, \ldots, \mathbf{y}_n) = \mathcal{A}(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ in the 2D plane, aiming to preserve the essential structures of $\mathbf{x}_1, \ldots, \mathbf{x}_n$. Due to algorithmic complexity, $\mathcal{A}$ is often used as a black-box visualization tool.

These visualization methods are often interpreted as manifold learning algorithms, which extract and represent latent low-dimensional manifolds in 2D and 3D spaces[11–13]. However, unlike classical dimension reduction methods such as PCA[14], where a parametric mapping $\mathbf{f}_{\boldsymbol{\theta}}$ is determined and any input point $\mathbf{x}$ is embedded through $\mathbf{y} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x})$, there is no globally defined embedding map for neighbor embedding methods as the "embedding points" $\mathbf{y}_1, \ldots, \mathbf{y}_n$ are determined in a discrete manner.

A key conceptual difficulty is the lack of sample-independent notion of embedding maps, since the embedding points $\mathbf{y}_1, \ldots, \mathbf{y}_n$ depend on n input points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ collectively, which makes it challenging to understand the correspondence between an input point $\mathbf{x}_i$ and an embedding point $\mathbf{y}_i$. Thus, it is unclear what structures the embedding points inherit from the input points, even in ideal settings where inputs are drawn from known distributions or simple manifolds. The lack of continuous-space embedding maps leads to recent recognition that neighbor embedding methods often produce misleading results by creating severe distortion through the embedding maps and introducing spurious clusters in low-dimensional visualization[15,16]. Moreover, neighbor embedding methods are sensitive to the choice of optimization algorithms[17], initialization schemes[18], and hyperparameters[18,19], leading to inconsistent interpretations[20,21].

Some progress has been made to improve the reliability of these visualization methods, including insights on embedding stages[19,22,23],

[1]Department of Statistics, University of Wisconsin-Madison, Madison, WI, USA. [2]Department of Biostatistics, T.H. Chan School of Public Health, Harvard University, Boston, MA, USA. [3]Department of Data Science, Dana-Farber Cancer Institute, Boston, MA, USA. e-mail: yiqiao.zhong@wisc.edu

force-based interpretations[24], visualization quality[22,23,25], initialization schemes, and hyperparameter selection[3,16,26–28]. To enhance the faithfulness of neighbor embedding methods, multiple diagnostic approaches have been proposed[3,16,26,27,29–31]. However, most existing diagnostic methods offer only partial solutions and rely on ad hoc fixes, sometimes even introducing new artifacts.

In this work, we show that the manifold learning interpretation, which implicitly assumes a continuous mapping, is inaccurate. Our analyses reveal intrinsic discontinuity points in the embeddings that result in severe distortions. Our results imply that t-SNE and UMAP—which can induce topological changes to visualization—are fundamentally different from PCA and other parametric embedding methods.

We address the conceptual difficulty by proposing a notion of embedding map—which we call LOO-map—induced by a given neighbor embedding method $\mathcal{A}$. LOO-map is a mapping in the classical sense and approximates the properties of $\mathcal{A}$ around each embedding point. It is based on a well-established strategy from statistics known as the leave-one-out (LOO) method, which posits that adding, deleting, or changing a single input point has negligible effects on the overall inferential results. Using LOO, we can decouple the pairwise interaction in the algorithm $\mathcal{A}$: we add a new input point $\mathbf{x}$ to $\mathbf{x}_1$, ..., $\mathbf{x}_n$ and freeze $\mathbf{y}_1$, ..., $\mathbf{y}_n$ in the optimization problem, allowing only one free variable $\mathbf{y}$. We call the resulting minimizer $\mathbf{f}(\mathbf{x})$ the LOO-map, which satisfies the approximation $(\mathbf{y}_1, \ldots, \mathbf{y}_n, \mathbf{f}(\mathbf{x})) \approx \mathcal{A}(\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{x})$. By design, the LOO-map $\mathbf{f}$ not only satisfies $\mathbf{f}(\mathbf{x}_i) \approx \mathbf{y}_i$ for all i's, but also reveals the embedding point $\mathbf{f}(\mathbf{x})$ of a potential new input point $\mathbf{x}$. As such, LOO-map extends the mapping defined over the discrete input set $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$ to the entire input space.

LOO-map offers a unified framework for understanding known issues like distance distortion[16,32], low stability[30], and poor neighborhood preservation[16,33], while also revealing new insights into embedding discontinuity. In our view, discontinuities of $\mathbf{f}(\mathbf{x})$ represent an extreme form of distortion that accompanies topological changes in the embedding space, e.g., connected clusters become separated and a uniform shape is fractured into pieces. In contrast, classical dimension reduction methods such as PCA do not suffer from map discontinuity since a continuous parametric map $\mathbf{f}_\theta(\mathbf{x})$ is constructed explicitly. In this regard, embedding discontinuity is an innate issue of the family of neighbor embedding methods.

Using LOO-map, we identify two types of observed distortion patterns, one affecting global properties of the embedding map and the other affecting local relationships. Both types of distortion are a consequence of discontinuities in $\mathbf{f}(\mathbf{x})$ and can cause topological changes in the embedding structures.

- Overconfidence-inducing (OI) discontinuity. Overlapping clusters or data mixtures in the input space are embedded into well-separated clusters, which creates a misleading visual impression of over-confidence that there is less uncertainty in the datasets. This biased perception of uncertainty can, in turn, lead to overly confident scientific conclusions.
- Fracture-inducing (FI) discontinuity. Small spurious and artificial clusters form in the embedding space, even for non-clustered data. Unlike OI discontinuity, such spurious clusters are small, localized, and formed in arbitrary locations.

We propose two types of point-wise diagnostic scores, namely perturbation scores and singularity scores, to quantify the severity of the two types of map discontinuity at each embedding point. Our approach is flexible and works as a wrapper around many neighbor embedding algorithms (Supplementary File Section 1) without any label information.

In this work, we demonstrate the utility of our method through two use cases: detecting out-of-distribution data (or distribution shifts) in computer vision using the perturbation score, and selecting hyperparameters in single-cell data analysis using the singularity score. We evaluate our method on multiple simulated and real-world datasets (Supplementary Table 1, "Methods"). Comparisons with existing approaches show that our method achieves superior performance in detecting topological changes in embedding and hyperparameter selection. The R package implementing our method, along with a tutorial, is publicly available on GitHub: https://github.com/zhexuandliu/MapContinuity-NE-Reliability.

Results

Overview of methods

We provide an overview of LOO-map and demonstrate the proposed two diagnostic scores (Fig. 1).

First, we introduce a general strategy to discern and analyze discontinuities in neighbor embedding methods (e.g., t-SNE, UMAP). Given input points $\mathbf{x}_1$, ..., $\mathbf{x}_n$ in a potentially high-dimensional space, e.g., attribute vectors or feature vectors, an embedding algorithm $\mathcal{A}$ maps them to 2D points $\mathbf{y}_1$, ..., $\mathbf{y}_n$ by solving an optimization problem involving $O(n^2)$ pairwise interaction terms. The LOO strategy assumes no dominant interaction term so that perturbing any single input point has negligible effects on the overall embedding. We extensively verify this assumption on simulated and real datasets (Table 1, Supplementary Table 2, Methods). By adding a new input $\mathbf{x}$ and optimizing its corresponding $\mathbf{y}$ while freezing $(\mathbf{y}_j)_{j \le n}$, LOO-map reduces the optimization problem to only $O(n)$ effective interaction terms. We identify the discontinuity points of $\mathbf{f}(\mathbf{x})$ as the source of the observed distortions and artifacts.

Then, we devise two label-free point-wise diagnostic scores to quantitatively assess embedding quality (Fig. 1a). The first quantity, namely the perturbation score, quantifies how much an embedding point $\mathbf{y}_i$ moves when the input $\mathbf{x}_i$ is moderately perturbed, which helps to probe the discontinuity of $\mathbf{f}(\mathbf{x})$ from the input space. The second quantity, namely the singularity score, measures how sensitive an embedding point is to an infinitesimal input perturbation, thus providing insights into $\mathbf{f}(\mathbf{x})$ at each specific location $\mathbf{x} = \mathbf{x}_i$. The two scores, as we will show below, are motivated by different considerations and reveal qualitatively distinct features of the visualizations (Fig. 1b–d).

Finally, we demonstrate how our scores can improve the reliability of neighbor embedding methods. Following the workflow in Fig. 1a, we extract high-dimensional features of image data using a deep learning model (e.g., ResNet-18[34]) and apply t-SNE for the 2D embedding. We observe that some inputs with ambiguous (mixed) class membership are misleadingly embedded into well-separated clusters (Fig. 1c), creating overconfidence in the cluster structure. Ground-truth labels and label-informed entropy scores confirm that the visualization under-represents the uncertainty for mixed points, making them appear more distinct than they should be (Fig. 1c). Further examination of image examples confirms such an artifact of reduced uncertainty in the embedding space. As a diagnosis, we find that embedding points with high perturbation scores correlate well with such observed (OI) discontinuity.

Our second diagnostic score can help with hyperparameter selection. A practical challenge of interpreting t-SNE embeddings is that the results may be sensitive to tuning parameters. In fact, we find that a small perplexity tends to induce small spurious structures similar to fractures, visually speaking, suggesting the presence of local (FI) discontinuity in the LOO-map $\mathbf{f}$ (Fig. 1d). Our singularity score captures such FI discontinuity as more high-scoring points emerge under smaller perplexities. With this diagnosis, we recommend choosing a perplexity no smaller than the elbow point of the FI discontinuity curve.

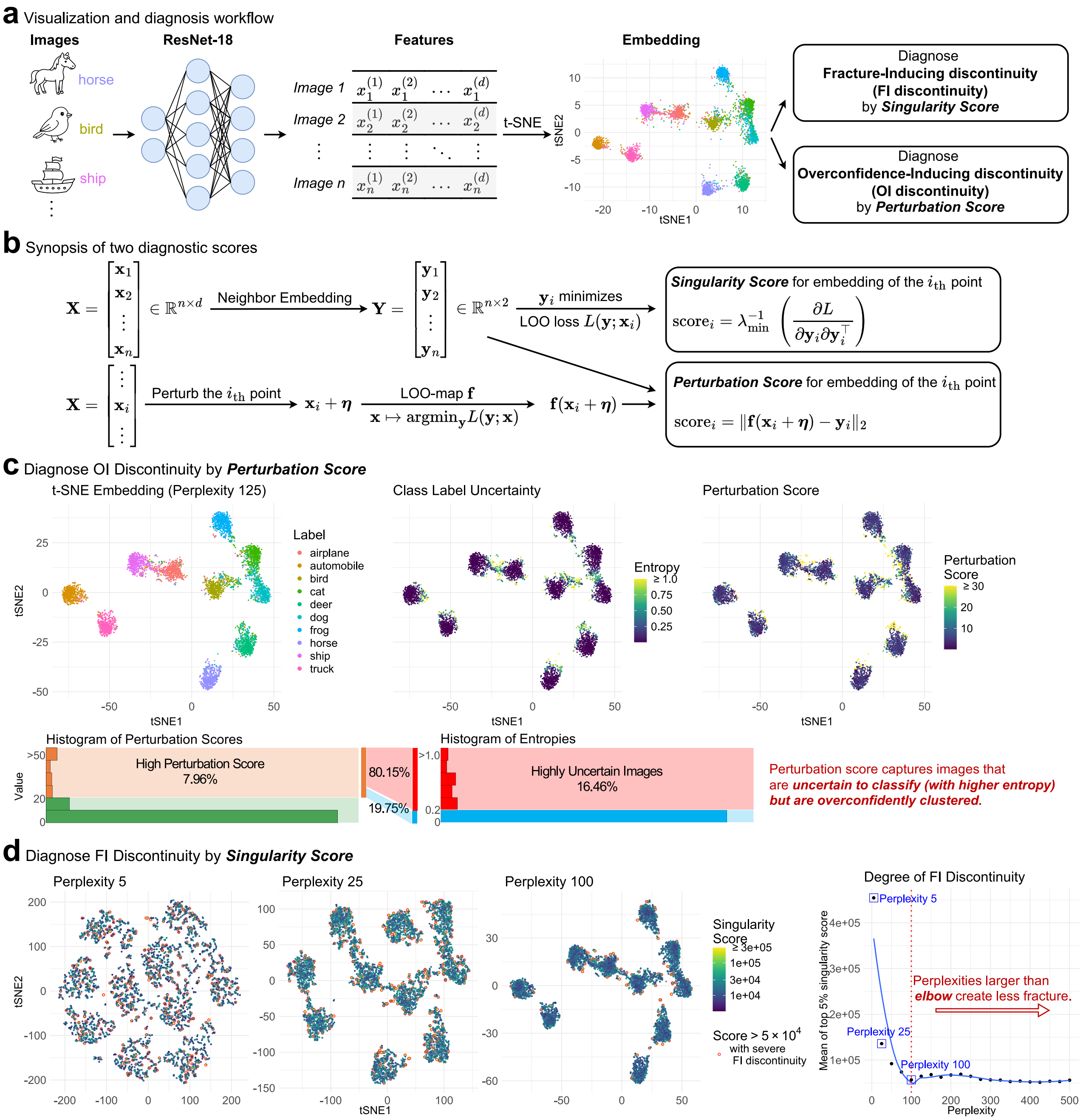

Fig. 1 | Overview: assessment of embeddings generated by neighbor embedding methods, illustrated with image data. a We use a standard pre-trained convolutional neural network (CNN) to obtain features of image samples from the CIFAR10 dataset, and then visualize the features using a neighbor embedding method, specifically t-SNE. **b** Basic ideas of singularity scores and perturbation scores. **c** t-SNE tends to embed image features into separated clusters even for images with ambiguous semantic meanings (as quantified by higher entropies of predicted class probabilities by the CNN). Perturbation scores identify the embedding points that have ambiguous class membership but less visual uncertainty. **d** An incorrect choice of perplexity leads to visual fractures (FI discontinuity), which is more severe with a smaller perplexity. We recommend choosing the perplexity no smaller than the elbow point. Source data are provided as a Source Data file.

Leave-one-out as a general diagnosis technique

We start with a generic setup for neighbor embedding methods that encompasses SNE[35], t-SNE[1], UMAP[2], LargeVis[10], PaCMAP[15], among others. First, we introduce basic mathematical concepts and their interpretations.

- Input data matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^\top \in \mathbb{R}^{n\times d}$: the input data to be visualized. Dimension d may be large (e.g., thousands).
- Embedding matrix $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n]^\top \in \mathbb{R}^{n\times p}$: the embedding points we aim to determine for visualization, where p can be 2 or 3.
- (Pairwise) similarity scores $(v_{i,j})_{i<j}$: a measure of how close two input points are in the input space, often calculated based on a Gaussian kernel.
- (Pairwise) embedding similarity scores $(w_{i,j})_{i<j}$: a measure of how close two embedding points are, which takes the form of a heavy-tailed kernel (e.g., t-distribution). The computation often requires a normalization step.
- (Pairwise) loss function $\mathcal{L}$: a measure of discrepancy between $v_{i,j}$ and $w_{i,j}$. An NE method minimizes this loss function over embedding points to preserve local neighborhood structures. The algorithms of NE methods aim to find the embedding $\mathbf{Y}$ by minimizing the total loss composed of the sum of the divergences between v_{ij} and w_{ij} of all pairs of points and a normalization factor $Z(\mathbf{Y})$.

For convenience, we introduce a generic optimization problem that neighbor embedding methods aim to solve as follows:

$$\min_{\mathbf{y}_1,\dots,\mathbf{y}_n\in\mathbb{R}^2} \sum_{1\le i\le n} \underbrace{\mathcal{L}(w(\mathbf{y}_i,\mathbf{y}_j);v_{i,j}(\mathbf{X}))}_{\text{unnormalized pair wise loss}} + \underbrace{Z(\mathbf{Y})}_{\text{normalization factor}} . \quad (1)$$

Particularly, in the t-SNE algorithm (see Supplementary Methods 2 for other algorithms), we have,

$$\mathcal{L}(w_{i,j};v_{i,j}) = -2v_{i,j}\log(w_{i,j}),$$
$$w_{i,j} = w(\mathbf{y}_i,\mathbf{y}_j) = (1+\|\mathbf{y}_i-\mathbf{y}_j\|^2)^{-1}, \quad Z(\mathbf{Y}) = \log\left(\textstyle\sum_{k\neq l} w(\mathbf{y}_k,\mathbf{y}_l)\right). \quad (2)$$

A fundamental challenge of assessing the embeddings is that we only know how discrete points—not the input space—are mapped since the optimization problem is solved numerically by a complicated algorithm. Consequently, it is unclear if underlying structures (e.g., clusters, low-dimensional manifolds) in the input space are faithfully preserved in the embedding space.

Consider adding a new point $\mathbf{x}$ to existing data points. We may wish to fix $\mathbf{x}_1, \ldots, \mathbf{x}_n$ and analyze how embedding points $\mathcal{A}(\mathbf{x}_1,\ldots,\mathbf{x}_n,\mathbf{x})$ change as we vary $\mathbf{x}$, thereby quantifying the mapping of $\mathbf{x}$ under $\mathcal{A}$. However, the embedding points would depend on all $n + 1$ input points, and require re-running the neighbor embedding algorithm for each new $\mathbf{x}$.

To address this, we use a generic decoupling technique known as leave-one-out (LOO), which enables us to isolate the changes of one embedding point versus the others[36–41]. We introduce the LOO assumption, the LOO loss function, and LOO-map as follows.

- LOO assumption: adding (or deleting/modifying) a single input point does not change embedding points significantly (Fig. 2a).
- LOO loss function $L(\mathbf{y};\mathbf{x})$: it consists of n pairwise loss terms relevant to the newly added point $\mathbf{x}$. We aim to determine the embedding $\mathbf{y}$ for $\mathbf{x}$ (Fig. 2b).
- LOO-map $\mathbf{f}$: it is defined as $\mathbf{f}:\mathbf{x}\mapsto \text{argmin}_{\mathbf{y}}L(\mathbf{y};\mathbf{x})$ for all possible input $\mathbf{x}$ (Fig. 2b). This definition allows us to examine the map property in the entire region.

Table 1 | Empirical validation of LOO on both simulated and real datasets

Number of points	n = 1000	n = 3000	n = 5000
2-GMM	0.068(0.0017)	0.044(0.0018)	0.034(0.0007)
Swissroll	0.074(0.0110)	0.043(0.0041)	0.033(0.0014)
CIFAR10	0.042(0.0081)	0.044(0.0013)	0.039(0.0006)
IFNB	0.069(0.0022)	0.049(0.0019)	0.044(0.0010)

We measured the approximation error ϵ_n defined in Eq. (4) across 20 independent trials and reported the average (and the standard deviation) of ϵ_n. We find that the errors are small and generally decreasing in n, which supports our LOO assumption.

Rooted in the stability idea[42–44], LOO assumes that adding (or deleting/modifying) a single input point does not change embedding points significantly (Fig. 2a). This assumption allows us to study the map $\mathbf{x}\mapsto\mathcal{A}(\mathbf{x}_1,\ldots,\mathbf{x}_n,\mathbf{x})$ approximately. Consider the optimization problem in Equation (1) with $n + 1$ input points $\mathbf{x}_1$, ..., $\mathbf{x}_n$, $\mathbf{x}$. Under the LOO assumption, when adding the new $(n + 1)$-th input point $\mathbf{x}$, we can freeze the embedding matrix $\mathbf{Y} = [\mathbf{y}_1,\ldots,\mathbf{y}_n]^\top$ and allow only one free variable $\mathbf{y}$ in the optimization problem. More precisely, the mathematical formulation of LOO loss function is given by

$$L(\mathbf{y};\mathbf{x}) = \sum_{1\le i\le n}\mathcal{L}\left(w(\mathbf{y}_i,\mathbf{y});v_{i,n+1}\left(\begin{bmatrix}\mathbf{X}\\ \mathbf{x}^\top\end{bmatrix}\right)\right)+Z\left(\begin{bmatrix}\mathbf{Y}\\ \mathbf{y}^\top\end{bmatrix}\right). \quad (3)$$

The LOO loss is motivated by the following observation: suppose $[\begin{smallmatrix}\widetilde{\mathbf{Y}}\\ \widetilde{\mathbf{y}}^\top\end{smallmatrix}]$ is the embedding of $\mathbf{X}_+ = [\begin{smallmatrix}\mathbf{X}\\ \mathbf{x}^\top\end{smallmatrix}]$, i.e., it reaches the minimum of the original loss, then $\mathbf{y} = \widetilde{\mathbf{y}}$ is necessarily the minimizer of a partial loss involving the embedding point of the point $\mathbf{x}$:

$$\begin{aligned}\widetilde{\mathbf{y}} &= \text{argmin}_{\mathbf{y}\in\mathbb{R}^2}\sum_{1\le i\le n}\mathcal{L}\left(w(\widetilde{\mathbf{y}}_i,\mathbf{y});v_{i,n+1}\left(\begin{bmatrix}\mathbf{X}\\ \mathbf{x}^\top\end{bmatrix}\right)\right)+Z\left(\begin{bmatrix}\widetilde{\mathbf{Y}}\\ \mathbf{y}^\top\end{bmatrix}\right)\\ &\approx \text{argmin}_{\mathbf{y}\in\mathbb{R}^2}\sum_{1\le i\le n}\mathcal{L}\left(w(\mathbf{y}_i,\mathbf{y});v_{i,n+1}\left(\begin{bmatrix}\mathbf{X}\\ \mathbf{x}^\top\end{bmatrix}\right)\right)+Z\left(\begin{bmatrix}\mathbf{Y}\\ \mathbf{y}^\top\end{bmatrix}\right)\end{aligned}$$

where the approximation is based on the LOO assumption $\widetilde{\mathbf{Y}}\approx\mathbf{Y}$. This approximation allows us to decouple the dependence of $\widetilde{\mathbf{y}}_i$ on $\mathbf{x}$. We then define the LOO-map as $\mathbf{f}:\mathbf{x}\mapsto\text{argmin}_{\mathbf{y}}L(\mathbf{y};\mathbf{x})$.

We empirically validate the LOO assumption by demonstrating that $\mathbf{Y}$ and $\widetilde{\mathbf{Y}}$ are very close for a large sample size n. Define the normalized error between embeddings before and after deleting a data point by

$$\epsilon_n = \frac{1}{\|\mathbf{Y}\|_F}\|\mathbf{Y}-\widetilde{\mathbf{Y}}\|_F, \quad (4)$$

where $\|\cdot\|_F$ means the Frobenius norm of a matrix. A sufficiently small ϵ_n will support the approximation in our derivation of the LOO-map.

We calculate this error extensively on both simulated and real datasets ("Methods"). The results support our LOO assumption (Table 1, Supplementary Table 2). We observe that the approximation errors are small and generally decreasing in n, which validates our LOO assumption.

LOO-map reveals intrinsic map discontinuities

By analyzing the LOO loss, we identify the two observed discontinuity patterns as a result of the map discontinuities of $\mathbf{f}(\mathbf{x})$. We use t-SNE as an example to illustrate the main results.

We generate mixture data by sampling 500 points from two overlapping 2D Gaussian distributions and run t-SNE with two representative choices of perplexity, 5 and 50. The resulting visualizations

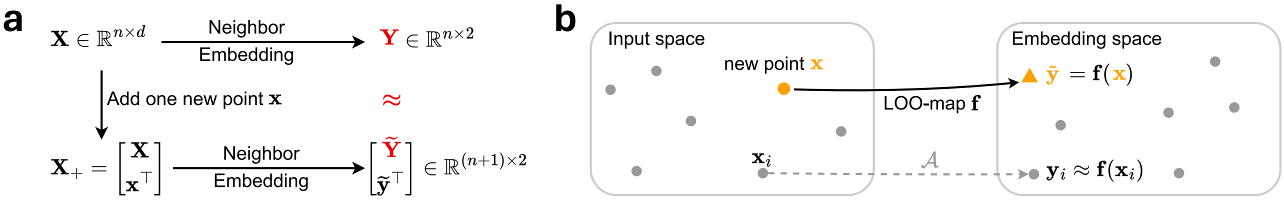
a
$\mathbf{X}\in\mathbb{R}^{n\times d}$
Neighbor Embedding
$\mathbf{Y}\in\mathbb{R}^{n\times 2}$
Add one new point $\mathbf{x}$
$\approx$
$\mathbf{X}_+ = \begin{bmatrix}\mathbf{X}\\ \mathbf{x}^\top\end{bmatrix}$
Neighbor Embedding
$\begin{bmatrix}\widetilde{\mathbf{Y}}\\ \widetilde{\mathbf{y}}^\top\end{bmatrix}\in\mathbb{R}^{(n+1)\times 2}$
b
Input space
new point $\mathbf{x}$
$\mathbf{x}_i$
LOO-map $\mathbf{f}$
$\mathcal{A}$
Embedding space
$\widetilde{\mathbf{y}} = \mathbf{f}(\mathbf{x})$
$\mathbf{y}_i\approx\mathbf{f}(\mathbf{x}_i)$

Fig. 2 | Diagrams showing the idea of Leave-one-out (LOO) and LOO-map. a Idea of LOO. Adding one input point does not significantly change the overall positions of embedding points. The assumption allows us to analyze the properties of the embedding map over the entire input space via an approximated loss which we call LOO loss. **b** We introduce a global embedding map (LOO-map) $\mathbf{f}(\mathbf{x}) = \text{argmin}_{\mathbf{y}}L(\mathbf{y};\mathbf{x})$ defined in the entire input space as an approximation to the neighbor embedding method $\mathcal{A}$.

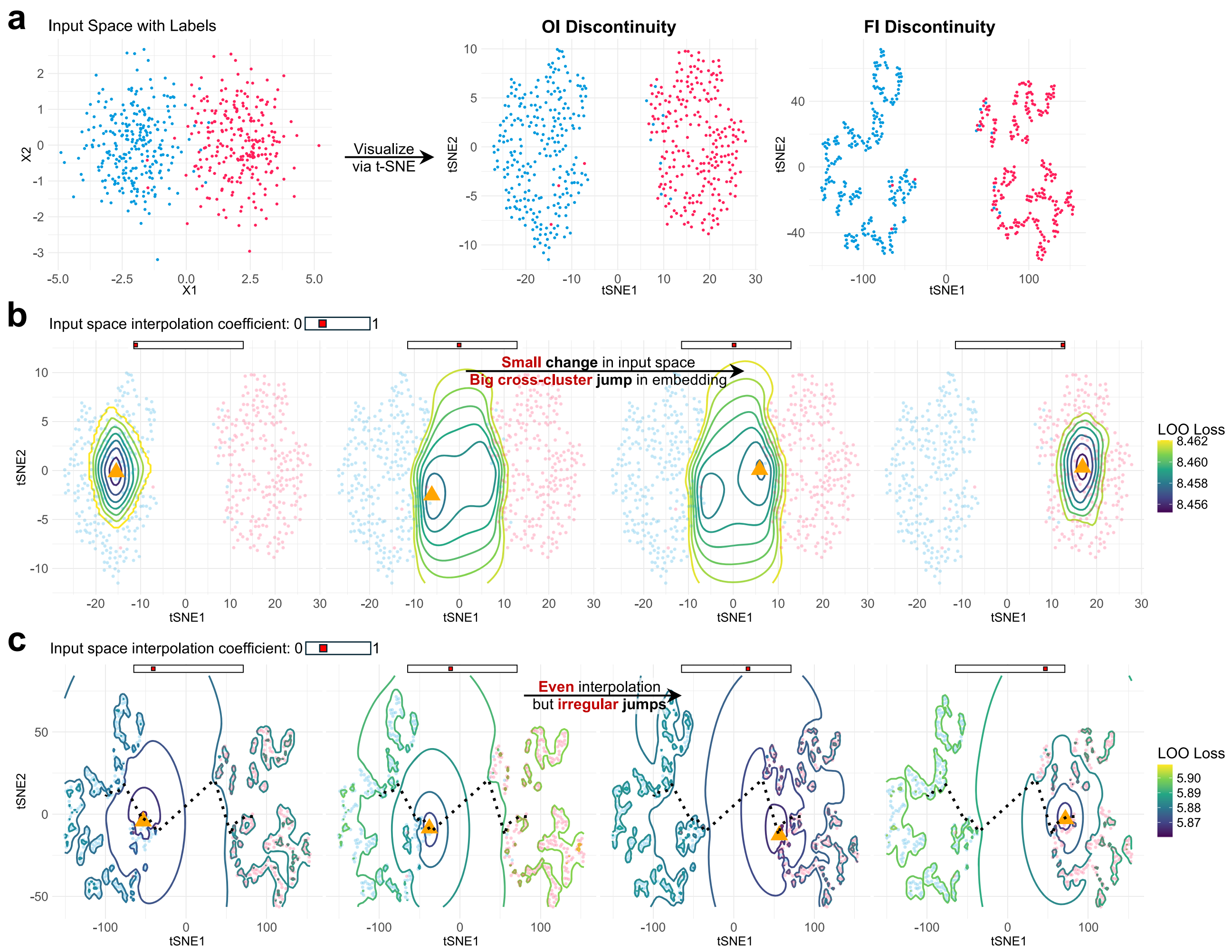

Fig. 3 | LOO loss landscape reveals the origins of two distortion patterns. a We illustrate two discontinuity patterns on simulated Gaussian mixture data. OI discontinuity: t-SNE embeds points into well-separated clusters and creates visual overconfidence. FI discontinuity: t-SNE with an inappropriate perplexity creates many artificial fractures. **b** Origin of OI discontinuity: LOO loss contour plot shows distantly separated minima. We add a new input point $\mathbf{x}$ at one of the 4 interpolated locations $\mathbf{x} = t\mathbf{c}_1 + (1 - t)\mathbf{c}_2$ where $t \in \{0, 0.47, 0.48, 1\}$ and then visualize the landscape of the LOO loss $L(\mathbf{y}; \mathbf{x})$ using contour plots in the space of $\mathbf{y}$. The middle two plots exhibit two well-separated minima (orange triangle), which cause a huge jump of the embedding point (as the minimizer of the LOO loss) under a small perturbation of $\mathbf{x}$. **c** Origin of FI discontinuity: We show LOO loss contour plots with interpolation coefficient $t \in \{0.2, 0.4, 0.6, 0.8\}$. The plots show many local minima and irregular jumps. Under an inappropriate perplexity, the loss landscape is consistently fractured. Numerous local minima cause an uneven trajectory of embedding points (dashed line) when adding $\mathbf{x}$ at evenly interpolated locations. Source data are provided as a Source Data file.

confirm the two discontinuity patterns (Fig. 3a). OI discontinuity pushes mixed points to cluster boundaries, creating overly tight structures, while FI discontinuity fragments embeddings into small pieces, leading to many sub-clusters. Similar discontinuity patterns are also common among other neighbor embedding methods (Supplementary Fig. 1).

We trace the origins of the observed discontinuity patterns by the LOO loss function. To this end, we add a single point $\mathbf{x}$ at varying locations to the input data and track how $\mathbf{x}$ is mapped. By visualizing the landscape of the LOO loss $L(\mathbf{y}; \mathbf{x})$ at four different inputs $\mathbf{x}$, we provide snapshots of the LOO-map $\mathbf{x} \mapsto \mathrm{argmin}_{\mathbf{y}} L(\mathbf{y}; \mathbf{x})$. More specifically, we choose the centers $\mathbf{c}_1$, $\mathbf{c}_2$ of the two Gaussian distributions and consider the interpolated input $\mathbf{x}(t) = t\mathbf{c}_1 + (1 - t)\mathbf{c}_2$, $t \in [0, 1]$. Since $\mathbf{x}(t)$ is mapped to the LOO loss minimizer, tracking the loss minima reveals the trajectory of the corresponding embedding point $\mathbf{y}(t)$ under varying t.

We find that the observed OI discontinuity is caused by a discontinuity point of $\mathbf{f}(\mathbf{x})$ in the midpoint of two mixtures. To demonstrate this, we visualize the LOO loss landscape and the embedding of the added point $\mathbf{x}(t)$ at four interpolated locations where $t \in \{0, 0.47, 0.48, 1\}$. There are two clearly well-separated minima in the LOO loss landscape when $t \approx 0.5$ (Fig. 3b). As a result, the embedding point $\mathbf{y}(t)$ jumps abruptly between local minima with a slight change in t. A further gradient field analysis shows a hyperbolic geometry around the discontinuity point of $\mathbf{f}(\mathbf{x})$ (see below).

We also find that the FI discontinuity is caused by numerous irregular local minima of $L(\mathbf{y}; \mathbf{x})$ under an inappropriate choice of perplexity. This conclusion is supported by the observation that the loss landscape of $L(\mathbf{y}; \mathbf{x})$ is consistently irregular and contains many local valleys under a small perplexity (Fig. 3c). Moreover, varying the interpolation coefficient t from 0 to 1 at a constant speed results in an uneven trajectory of the embedding point $\mathbf{y}(t)$. Because of many irregularities, the embedding points tend to get stuck at these local minima, thus forming spurious sub-clusters. In addition, we find that larger perplexity typically lessens FI discontinuity (Supplementary Figs. 2, 3).

LOO-map motivates diagnostic scores for capturing topological changes

OI discontinuity and FI discontinuity reflect the properties of $\mathbf{f}(\mathbf{x})$ at different levels: OI discontinuity is relatively global, while FI discontinuity is relatively local. To quantify their severity respectively, we introduce two point-wise scores (Methods): (i) perturbation scores for OI discontinuity and (ii) singularity scores for FI discontinuity. For computational efficiency, both scores are based on modifying individual input points instead of adding a new point so that we maintain n data points in total. This approach is justified by the LOO assumption, allowing using the partial loss as LOO loss.

Briefly speaking, we define the perturbation score as the amount of change of an embedding point $\mathbf{y}_i$ under the perturbation of an input point $\mathbf{x}_i$ of a moderate length. As the data distribution is not known a priori, we search the perturbation directions using the top principal directions of the data (Methods).

We define the singularity score as the inverse of the smallest eigenvalue of a Hessian matrix that represents the sensitivity of the embedding point $\mathbf{y}_i$ under infinitesimal perturbations. Our derivation (Supplementary Methods 1) reveals that small eigenvalues can produce substantial local discontinuities, whereas a singular Hessian matrix leads to the most severe discontinuity. We find that infinitesimal perturbations are particularly effective for capturing the local characteristics of FI discontinuities. Detailed expressions for the singularity scores of t-SNE, UMAP and LargeVis are provided in Supplementary Methods 2.

Generally, we recommend using the perturbation score to diagnose the trustworthiness of cluster structures, and the singular score to detect spurious local structures.

Simulation studies

We implement our proposed point-wise scores for t-SNE as an example. We evaluate our diagnostic scores on two types of simulated datasets (Methods): (i) 2D Gaussian mixture data with 5 centers (unequal mixture probabilities, n = 700) and 8 centers (equal probabilities, n = 800), and (ii) Swiss roll data, where n = 800 points are sampled from a 3D Swiss-roll manifold.

We apply perturbation scores to the 5-component Gaussian mixture data, where t-SNE creates misleadingly distinct cluster boundaries (Fig. 4a left). Without label information, our scores identify unreliable points with deceptively low uncertainty (Fig. 4a, right). Meanwhile, the entropy differences use the ground-truth labels to calculate the reduced class entropies (Methods) in the embedding space, thus providing an objective evaluation of the degree of confidence (Fig. 4a middle). Our perturbation scores are closely aligned with the entropy differences.

Next, we apply singularity scores to the 8-component Gaussian mixture and Swiss roll data under two perplexity settings (Fig. 4b–c). Each embedding is colored by ground-truth labels, singularity scores, and dichotomized singularity scores (binary thresholding). The embeddings differ visually: a low perplexity creates spurious sub-clusters, while a high perplexity preserves cluster and manifold structures. Additionally, the distributions of dichotomized scores vary: a low perplexity results in more high scores at randomly scattered locations, whereas a high perplexity yields fewer high-scoring points.

Moreover, we quantitatively assess the clustering quality for the 8-component Gaussian mixture data using three indices: DB index[45], within-cluster distance ratio (Methods), and Wilks' Λ[46]. All three indices (small values are better) indicate that t-SNE visualizations with less severe FI discontinuity, i.e., lower singularity scores, achieve better clustering quality, with the DB index dropping from 0.5982 to 0.3038, the within-cluster distance ratio from 0.0480 to 0.0024, and Wilks' Λ from 0.0028 to 9.0×10^{-6}. To further study the change in clustering quality, we generate 6 simulated datasets with varying cluster structures and dimensions. Across all datasets, tuning perplexity using singularity scores consistently improves clustering quality, reducing the DB index by approximately 50%, the within-cluster distance ratio by 65–91%, and Wilks' Λ by 57–99% (Supplementary Table 3).

Use case 1: detecting out-of-distribution image data

One common practical issue for statistical methods or machine learning algorithms is the distribution shift, where the training dataset and test dataset have different distributions, often because they are collected at different sources[47–49]. These test data are called out-of-distribution (OOD) data.

In this case study, we identify one rarely recognized pitfall of t-SNE visualization: OOD data may become harder to discern in t-SNE embeddings because they tend to be absorbed into other clusters. Our perturbation score is able to identify the misplaced OOD embedding points.

We use a standard ResNet-18 model[34] trained on the CIFAR-10 dataset[50] to extract features of its test dataset and an OOD dataset known as DTD (describable textures dataset)[51]. Ideally, visualization of the features of test images and OOD images would reveal the distribution shift. However, the t-SNE embedding shows that a fraction of OOD features are absorbed into compact, well-defined CIFAR-10 clusters (Fig. 5a). Without the label information, one may mistakenly assume that the misplaced OOD embedding points belong to the regular and well-separated classes in CIFAR-10. We find that the embedding misplacement results from OI discontinuity. Our inspection of the original feature space shows that the misplaced OOD data points appear to have mixed membership, resembling both CIFAR-10 and OOD data—thus their cluster membership is, in fact, less certain than what t-SNE suggests.

Our perturbation scores can successfully identify most of these misplaced OOD embedding points (Fig. 5b–d). The areas under the ROC curves (AUROC) are on average 0.75 for the three selected clusters. Compared with other methods aiming for OOD detection, our perturbation score demonstrates superior performance, with kernel PCA[52] achieving an average AUROC of 0.698 and the one-class support vector machine[53] achieving an average AUROC of 0.410 (Supplementary Fig. 4, Methods). Additionally, we use the prediction probabilities given by the neural network to calculate the entropy of each point and find that the entropies significantly correlate with the perturbation scores; specifically, the correlations are 0.49, 0.58, and 0.64 for the selected clusters. These findings suggest that perturbation scores are effective in detecting OOD data and can help safeguard against misinterpretation of t-SNE visualizations.

Use case 2: enhancing interpretation of single-cell data

Our second example concerns the application of singularity scores in single-cell data. In this case study, we investigate how incorrect choices of perplexity induce spurious sub-clusters. We also provide a guide of choosing perplexity based on singularity scores, thereby reducing such spurious sub-clusters.

The first dataset we examined is single-cell RNA-seq data from 421 mouse embryonic stem cells (mESCs) collected at 5 sampling time points during differentiation[54]. The second dataset is another single-cell RNA-seq data from 25,806 mouse mammary epithelial cells across 4 developmental stages[55]. We also include our analysis on a mid-sized mouse brain chromatin accessibility data in the Supplementary File (Supplementary Fig. 5, Supplementary Table 4). Through analysis of the datasets, we find that a small perplexity tends to create spurious sub-clusters (Fig. 6a, c, Supplementary Fig. 5a). Our singularity scores can provide informative insights into the spurious clusters even without the ground-truth labels, as summarized below.

- (Distribution difference) Embedding points with large singularity scores tend to appear in random and scattered locations if the perplexity is too small. In contrast, under an appropriate perplexity, embedding points with large singular scores are mostly in the periphery of clusters.
- (Elbow point) As the perplexity increases, the magnitude of large singular scores (calculated as the average of the top 5%) rapidly decreases until the perplexity reaches a threshold.

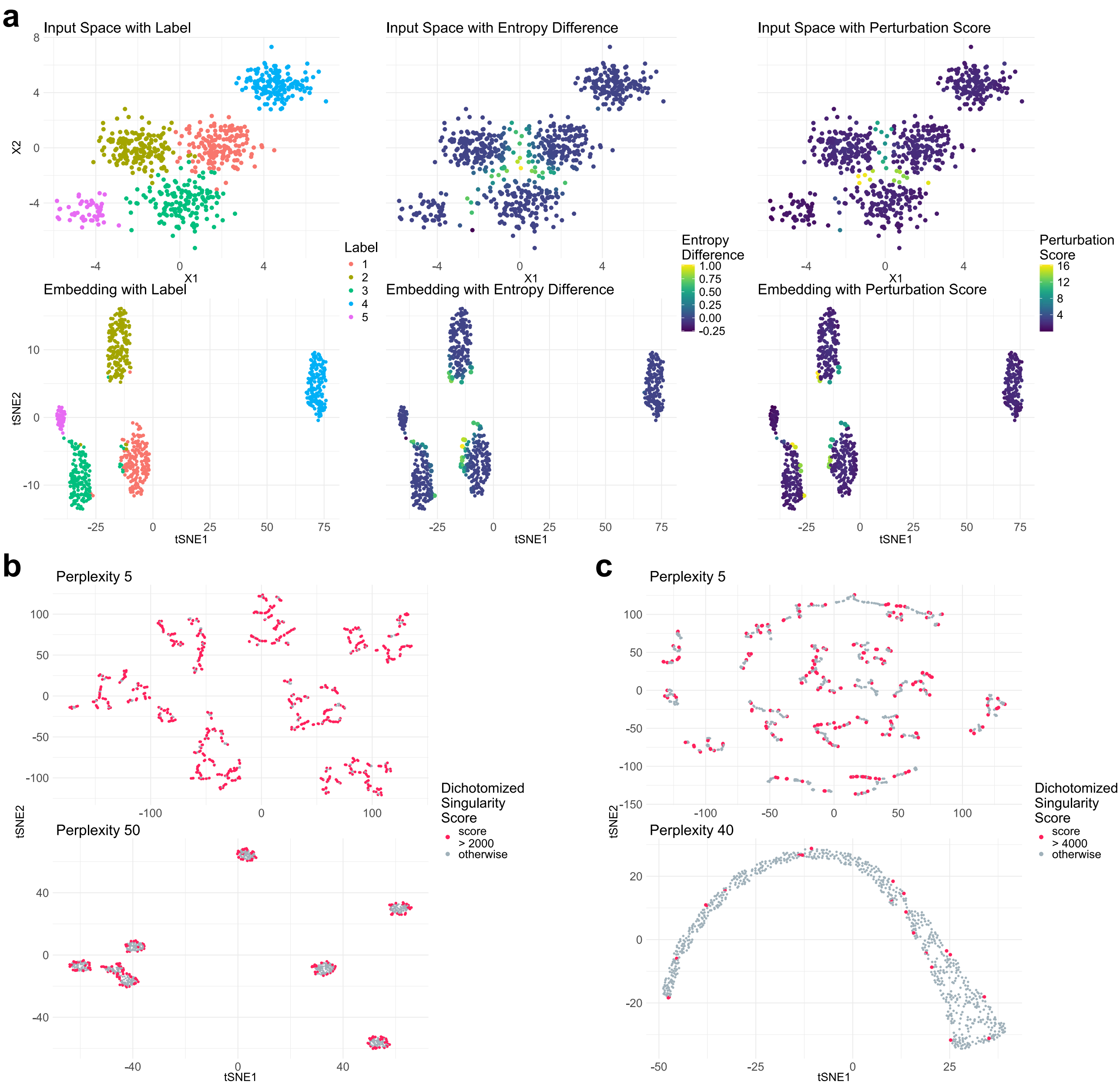

Fig. 4 | Simulation studies demonstrate the effectiveness of proposed scores. **a** Perturbation scores identify unreliable embedding points that have reduced uncertainty. Input points from 5-component Gaussian mixture data form separated clusters in the embedding space. t-SNE reduces perceived uncertainty for input points in the overlapping region (left), as captured by the label-dependent measurements, namely the entropy difference (middle). Our perturbation scores can identify the same unreliable embedding points without label information (right). Singularity scores reveal spurious sub-clusters on Gaussian mixture data (**b**) and Swiss roll data (**c**). At a low perplexity, t-SNE creates many spurious sub-clusters. Embedding points receiving high singular scores at random locations is an indication of such spurious structures. Source data are provided as a Source Data file.

The distribution of large singular values indicates spurious sub-clusters, reflecting the irregular LOO loss landscape (Supplementary Fig. 3a, c). We extensively validated the distribution difference between small and large perplexities through statistical tests, including Spearman's correlation test between singularity scores and cluster center distances, F-tests, and permutation tests for local regression models (singularity scores regressed against locations). At low perplexities, Spearman's correlation tests showed non-significant results for all five mESCs clusters and five mammary epithelial cell clusters (average p-values: 0.36 for mESCs clusters at perplexity 4 and 0.29 for mammary epithelial cell clusters at perplexity 5, Supplementary Tables 4, 6). Increasing perplexities to the singularity score elbow points (Fig. 6b, d) yielded significant correlations in four of five mESCs clusters and five of eight mammary epithelial cell classes. Similarly, F-tests and permutation tests showed p-values dropping from ~0.3 to < 0.001 (mESCs) and $<10^{-13}$ (mammary epithelial cells), confirming the dependence of singularity scores on location at higher perplexities. This transition aligns with LOO loss geometry: low perplexities create scattered local minima, forming spurious sub-clusters (Supplementary Fig. 3a, c), whereas higher perplexities smooth the loss landscape (Supplementary Fig. 3b, d), reducing artifacts.

We also observe that the degree of FI discontinuity, as indicated by the magnitude of the singularity scores, decreases rapidly until the perplexity reaches the elbow point (Fig. 6b, d). Beyond the elbow point, the spurious sub-clusters largely disappear, aligning with the improvement of neighborhood preservation (Fig. 6b, d), as measured

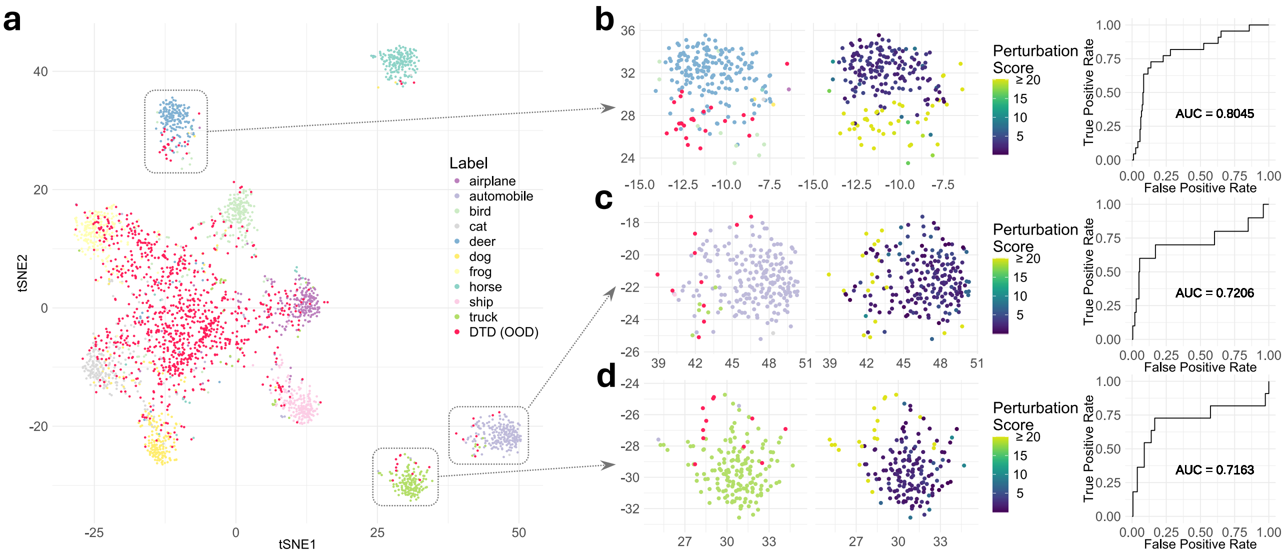

Fig. 5 | Perturbation scores detect out-of-distribution (OOD) image data. a We use a pretrained ResNet-18 model to extract features of CIFAR-10 images and, as out-of-distribution data, of DTD texture images. Then we visualize the features using t-SNE with perplexity 100. A fraction of OOD embedding points are absorbed into clusters that represent CIFAR-10 image categories such as deer, truck, and automobile. **b–d** Perturbation scores can effectively identify misplaced out-of-distribution data points. The ROC curves show the proportion of OOD points correctly identified by the perturbation scores. Source data are provided as a Source Data file.

by the nearest-neighbor distance correlation between the input and embedding spaces (Methods). However, we would not suggest increasing perplexity excessively, as it may merge clusters[26], result in the loss of certain microscopic structures[3], and often lead to longer computational running time[27]. Therefore, we suggest choosing a perplexity around the elbow point.

Computational cost of perturbation score

Theoretically, the computational complexity for solving the LOO loss optimization takes $O(n)$ flops, instead of $O(n^2)$ flops of the original loss which involves every pairwise interaction term. Practically, our R package has the following running time.

- For exact perturbation scores, it takes 35.2 s to compute the score per point for the CIFAR-10 images in Fig. 1 on a MacBook Air (Apple M2 chip).
- Leveraging pre-computed quantities, we also provide an approximation method to reduce the running time per point to 7.1 s, while preserving high accuracy relative to the exact score (Supplementary Fig. 8).
- In addition to the approximation, we introduce a pre-screening step to increase the computational efficiency by 14X for the same dataset. This pre-screening step identifies a subset of embedding points most likely to yield high scores, and thus significantly reduces computational cost while still providing a comparable assessment of OI discontinuity locations (Supplementary Fig. 7). Combining the approximation method and the pre-screening step results in an average of 0.47 s.

Computational cost of singularity score

Theoretically, the computational complexity for calculating the singularity scores for the entire dataset is $O(n^2)$ flops, primarily due to matrix operations when calculating Hessian matrices. Practically, the running time for computing singularity scores for CIFAR-10 is 15.9 s for all 5000 points on a MacBook Air (Apple M2 chip).

Comparison with other assessment metrics

There are multiple recent papers on assessing and improving the reliability of neighbor embedding methods. None of these papers view the observed artifacts as an intrinsic map discontinuity and as a result, cannot reliably identify topological changes in their proposed diagnosis. For illustration, we compare our method with EMBEDR[29], scDEED[28], and DynamicViz[30].

- EMBEDR identifies dubious embedding points by using statistical significance estimates as point-wise reliability scores. This process begins by computing point-wise KL divergences between the kernels in the input and embedding spaces, followed by a permutation test to determine whether the neighborhood preservation is significantly better than random chance. Lower p-values from the test indicate higher embedding reliability. EMBEDR selects the perplexity by minimizing the median p-values.
- scDEED calculates point-wise p-values by conducting a similar permutation test on the correlations of nearest-neighbor distances. Similarly, lower p-values indicate higher embedding reliability. ScDEED provides two approaches for parameter selection based on dubious embedding points: the first locates the elbow point, and the second selects the perplexity to minimize the number of dubious points.
- DynamicViz employs a bootstrap approach to assess the stability of embeddings. Point-wise variance scores are constructed based on resampling, defined as the average variance of distances to the neighbors. Embedding points with lower variance scores are considered more reliable. DynamicViz selects the perplexity by minimizing the median variance score.

Compared with existing methods, our perturbation score offers the following advantages in detecting distortions of global structure. First, perturbation scores are better at locating the topological changes of global structures by pinpointing the exact points. By design, they capture embedding points close to the intrinsic discontinuity of the embedding map. In a simulated Swiss roll dataset, t-SNE erroneously splits the smooth manifold into two disconnected pieces (Fig. 7a), which is a severe visualization artifact caused by OI discontinuity. Our perturbation scores accurately highlight unreliable points exactly at the disconnection location (Fig. 7b). In contrast, EMBEDR and scDEED label most points as unreliable, failing to

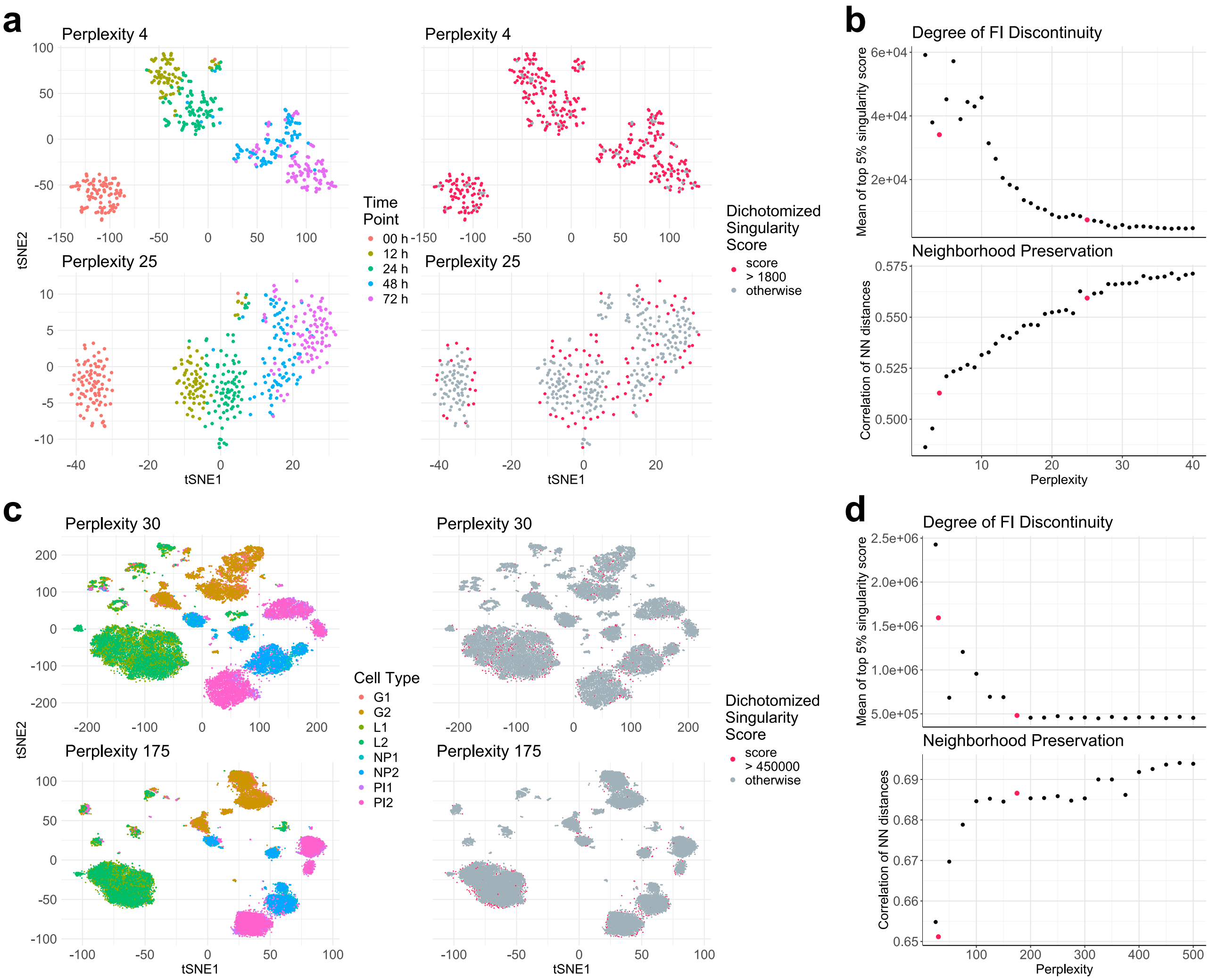

Fig. 6 | Singularity scores inform the selection of the perplexity parameter. Comparative t-SNE embeddings and the corresponding singularity scores at two different perplexities in mouse embryonic cell differentiation data (**a**) and in mouse mammary epithelial cell data (**c**). The perplexity as a tuning parameter has a large impact on t-SNE visualization qualitatively. At a small perplexity, there are many spurious sub-clusters. Embeddings with high singular scores appear in random locations, which indicates the presence of such spurious structures and severe FI discontinuity. Plots of the degree of FI discontinuity and neighborhood preservation versus perplexity are shown for mouse embryonic cell differentiation data (**b**) and for mouse mammary epithelial cell data (**d**). We recommend choosing a perplexity no smaller than the elbow point, as this ensures that randomly positioned points with high singularity scores largely disappear, remaining only at cluster peripheries. Consequently, the neighborhoods of most points are embedded more faithfully, resulting in better neighborhood preservation score. Source data are provided as a Source Data file.

pinpoint the discontinuity (Fig. 7c, d), as they emphasize neighborhood preservation rather than topological changes. DynamicViz identifies the general region but lacks precision (Fig. 7e).

Second, perturbation scores are also robust to low-density regions. In the simulated Gaussian mixture dataset (Supplementary Fig. 9a), DynamicViz fails to accurately characterize discontinuity locations in areas with a lower point density, as these areas are prone to insufficient sampling (Supplementary Fig. 9c). In contrast, our perturbation scores are more robust to the low-density regions (Supplementary Fig. 9b).

Compared with existing methods, our singularity score consistently selects a perplexity that is neither too small nor too large, thereby reducing sub-clusters while preserving fine-grained structures. We illustrate the advantage of this consistency in aiding hyperparameter selection using three datasets (Supplementary Table 7).

For the mouse embryonic cell differentiation data (Fig. 6a), scDEED recommends two approaches for perplexity selection; the first is based on the elbow point and yields 3, and the second is based on minimizing the number of dubious points and does not produce a unique value (Supplementary Fig. 10a). EMBEDR fails to suggest a valid hyperparameter because we encountered errors potentially due to a small dataset size. DynamicViz and singular scores select moderate perplexity (20 and 25), reducing spurious sub-clusters compared to perplexity of 3 (Supplementary Fig. 10b) and achieving the higher neighborhood preservation score (0.5594 (singularity score, highest), 0.4955 (scDEED), 0.5524 (DynamicViz)).

For the mouse brain chromatin accessibility data (Supplementary Fig. 5a), scDEED selects 10 (elbow point) and 145 (minimizing dubious points). EMBEDR chooses perplexity of 145, showing a tendency of favoring larger perplexity that is also observed by Xia et al.[28]. DynamicViz selects perplexity of 10. Our singularity score selects perplexity of 95 (Supplementary Fig. 11a). By visual inspection, perplexity of 10 is inappropriately small because visualization exhibits numerous spurious sub-clusters. In contrast, perplexities of 95 and 145 avoid spurious sub-clusters while maintaining fine-grained structures (Supplementary Fig. 11b). Quantitatively, the perplexities suggested by singular scores, scDEED, and EMBEDR lead to similar neighborhood preservation scores (0.4108, 0.4223, 0.4223).

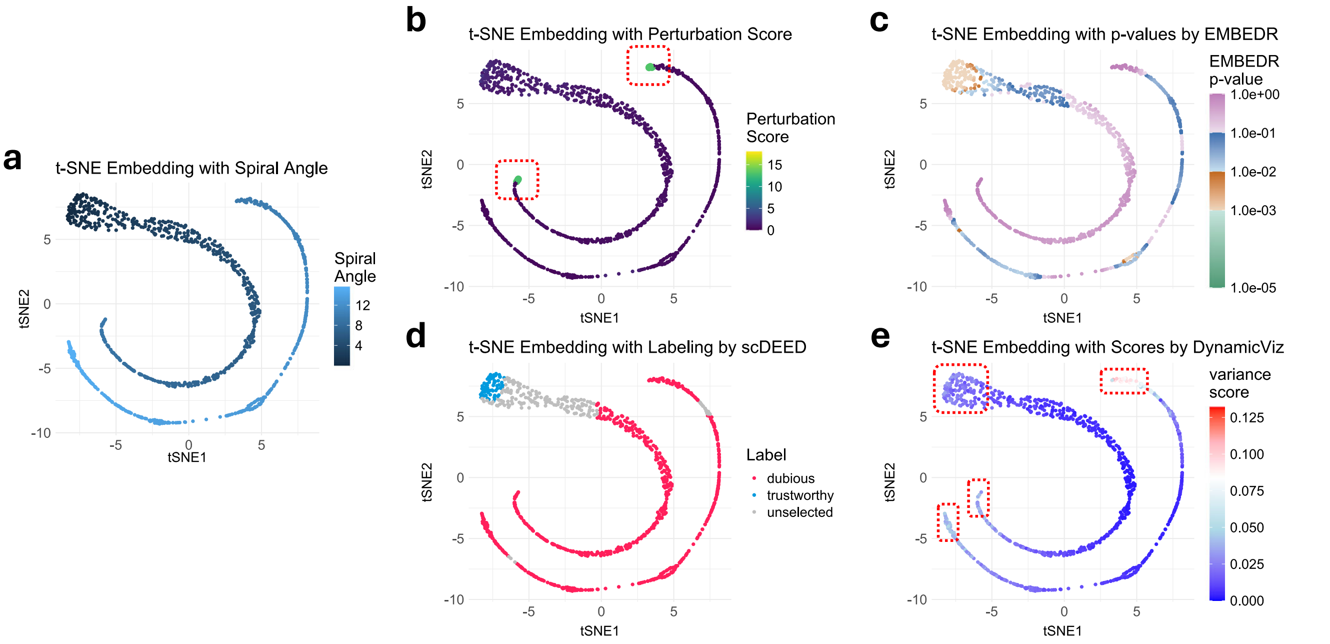

Fig. 7 | Comparing perturbation scores with three diagnostic scores for the t-SNE embedding on the simulated Swiss roll dataset. a The t-SNE embedding of $n = 1000$ simulated points from the Swiss roll manifold under perplexity 150. The colors correspond to the ground-truth spiral angles of the points. t-SNE algorithm erroneously breaks the smooth manifold into two disconnected parts, which indicates OI discontinuity. **b** Perturbation scores clearly mark the unreliable embedding points where disconnection (discontinuity) occurs. **c** EMBEDR uses the p-values from one-sided permutation tests to identify unreliable embedding points. It suggests that most embedding points are unreliable (lower p-values are more reliable). But it does not identify the discontinuity location. **d** ScDEED evaluates most embedding points as dubious, but similar to EMBEDR, it does not identify the discontinuity location. **e** DynamicViz marks both the discontinuity location and the areas at both ends of the Swiss roll as unstable, making it difficult to distinguish the actual discontinuity locations. Furthermore, while it can roughly identify the discontinuity location, it still fails to pinpoint the exact points where the split occurs. Source data are provided as a Source Data file.

In the mouse mammary epithelial cell dataset, similar phenomena are observed: our singularity score selects a balanced perplexity while scDEED and EMBEDR select perplexities that are either too small or too large, and DynamicViz lacks scalability for large datasets due to its bootstrap-based approach, which requires repeated execution of visualization algorithms (Supplementary Fig. 12). Overall, the singularity score offers robust guardrail perplexities that significantly reduce spurious sub-clusters while producing informative visualization.

Theoretical insights: landscape of LOO loss

By analyzing the LOO loss function in Equation (3) under a simple setting, we will show that OI discontinuity is caused by a hyperbolic saddle point in the LOO loss function, thereby theoretically justifying Fig. 3b.

Suppose that n input points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are generated from a data mixture with two well-separated and balanced groups, where the first group is represented by the index set $\mathcal{I}_+ \subset \{1, 2, \ldots, n\}$ with $|\mathcal{I}_+| = n/2$ and the second group represented by $\mathcal{I}_- = \{1, 2, \ldots, n\} \setminus \mathcal{I}_+$. Without loss of generality, we assume that the mean vectors of $(\mathbf{y}_i)_{i\in\mathcal{I}_+}$ and $(\mathbf{y}_i)_{i\in\mathcal{I}_-}$ are $\boldsymbol{\theta}$ and $-\boldsymbol{\theta}$, respectively, since embeddings are invariant to global shifts and rotations. Equivalently, we write

$$\mathbf{y}_i = \begin{cases} \boldsymbol{\theta} + \boldsymbol{\delta}_i & i \in \mathcal{I}_+ \\ -\boldsymbol{\theta} + \boldsymbol{\delta}_i & i \in \mathcal{I}_- \end{cases}$$

where $\sum_{i\in\mathcal{I}_+} \boldsymbol{\delta}_i = \sum_{i\in\mathcal{I}_-} \boldsymbol{\delta}_i = \mathbf{0}$. To simplify the loss function, we make an asymptotic assumption: consider (implicitly) a sequence of problems where input data have increasing distances between the two groups, so we expect an increasing separation of clusters in the embedding space:

$$\| \boldsymbol{\theta} \| \to \infty, \quad \max_{i\le n} \| \boldsymbol{\delta}_i \| = O(1).$$

Now consider adding an input point ('mixed' point) to a location close to the midpoint of the two groups. We assume that its similarity to the other inputs is

$$v_{i,n+1} = \begin{cases} p_0 + \varepsilon + o(\varepsilon) \\ p_0 - \varepsilon + o(\varepsilon) \end{cases}$$

for $1 \le i \le n$, where $p_0 > 0$ and ε is a small perturbation parameter. This assumption is reasonable because the similarity of the added point $\mathbf{x} := \mathbf{x}^\varepsilon$ has roughly equal similarities to existing inputs up to a small perturbation. We make the asymptotic assumption $\|\boldsymbol{\theta}\|^{-1} \asymp \varepsilon$, namely $\varepsilon\|\boldsymbol{\theta}\| = O(1)$ and $[\varepsilon\|\boldsymbol{\theta}\|]^{-1} = O(1)$.

Theorem 1. Consider the LOO loss function for t-SNE given in Eqs. (2) and (3). Under the assumptions stated above, the negative gradient of the loss is

$$-\nabla_{\mathbf{y}} L(\mathbf{y}; \mathbf{x}^\varepsilon) = (1 + o(1)) \left(\underbrace{\frac{\mathbf{y}_{//} - \mathbf{y}_\perp}{\|\boldsymbol{\theta}\|^2}}_{\text{hyperbolic term}} + \underbrace{\frac{\varepsilon\boldsymbol{\theta}}{\|\boldsymbol{\theta}\|^2}}_{\text{perturbation term}} \right)$$

where $\mathbf{y}_{//} = \boldsymbol{\theta}\boldsymbol{\theta}^\top\mathbf{y}/\|\boldsymbol{\theta}\|^2$ is projection of $\mathbf{y}$ in the direction of $\boldsymbol{\theta}$, and $\mathbf{y}_\perp = \mathbf{y} - \mathbf{y}_{//}$.

This result explains how the hyperbolic geometry creates OI discontinuity.

- The hyperbolic term indicates the unstable saddle point of the loss at $\mathbf{y} = \mathbf{0}$. Indeed, it is exactly the tangent vector of a hyperbola, so in the embedding force (negative gradient) field there is a pull force towards the x-axis and a push force away from the y-axis (Fig. 8).

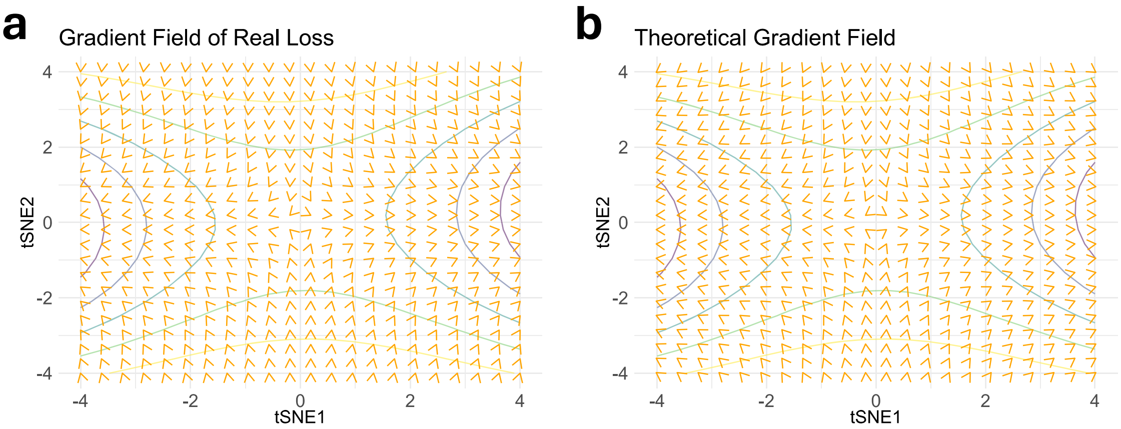

Fig. 8 | Negative gradient fields of the real/theoretical LOO loss. a We draw the negative gradient fields (force fields) $-\nabla_{\mathbf{y}}L(\mathbf{y};\mathbf{x}^{\varepsilon})$ based on the LOO loss under the same setting as in Fig. 3b. **b** We draw a similar field plot based on the hyperbolic term $\frac{y_{\parallel}-y_{\perp}}{\|\boldsymbol{\theta}\|^2}$ from Theorem 1, where we take $\boldsymbol{\theta} = (\mathbf{c}_1 - \mathbf{c}_2)/2$ and $\mathbf{c}_1$, $\mathbf{c}_2$ are the centers of two clusters in the embedding. In addition, we add loss contours to both plots, which show hyperbolic paraboloids around the origin. We observe a strong alignment between the negative gradient field of the LOO loss and that of the theoretical analysis. Both field plots show a pull force towards the x-axis and a push force away from the y-axis. Source data are provided as a Source Data file.

- The perturbation term reflects the effects of input point $\mathbf{x}^{\varepsilon}$. It tilts the negative gradients slightly in the direction of $\boldsymbol{\theta}$ if $\varepsilon > 0$ or $-\boldsymbol{\theta}$ if $\varepsilon < 0$, which causes the algorithm to jump between widely separated local minima of $L(\mathbf{y};\mathbf{x})$ under small perturbations.

Discussion

We developed a framework to interpret distortions in neighbor embedding methods as map discontinuities by leveraging the LOO strategy. Based on our LOO-map, we introduce two diagnostic scores to identify OI and FI discontinuities. While being generally effective, our method may not capture all distortion patterns, as factors like initialization, iterative algorithms, and other hyperparameters can introduce different types of distortions. We also recognize the absence of a formal mathematical framework for rigorously characterizing the LOO-map.

In future research, we aim to explore links between classical parametric and implicit embedding maps to fully address topological issues and improve interpretability. We also aim to enhance the scalability of our methods through efficient optimization, sparsity, tree-based approximations, and parallel computation.

Methods

Verify leave-one-out assumption empirically

Our LOO approach assumes that adding (or deleting/modifying) a single input point does not change the embeddings of other points on average significantly. To verify the LOO assumption, we conduct the following experiment.

Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^{\top}$ be the input data matrix, and $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n]^{\top}$ be the matrix of embedding points. We then add one point $\mathbf{x}$ to $\mathbf{X}$ to have the new input data $\mathbf{X}_+ = [\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{x}]^{\top}$. We then run the t-SNE algorithm to obtain the embedding of $\mathbf{X}_+$ as $[\tilde{\mathbf{y}}_1, \ldots, \tilde{\mathbf{y}}_n, \tilde{\mathbf{y}}]^{\top}$. Denoted $[\tilde{\mathbf{y}}_1, \ldots, \tilde{\mathbf{y}}_n]$ as $\tilde{\mathbf{Y}}$. To verify LOO empirically, we keep track of the difference between $\mathbf{Y}$ and $\tilde{\mathbf{Y}}$:

$$\epsilon_n = \frac{1}{\|\mathbf{Y}\|_F}\|\mathbf{Y} - \tilde{\mathbf{Y}}\|_F$$

and expect ϵ_n to be small.

We initialize the t-SNE algorithm in the second run by the embedding points we obtain from the first run: when calculating the embedding of $\mathbf{X}_+$, we use $\mathbf{Y}$ as the initialization for the first n points. This initialization scheme aims to address two issues: (i) the loss function in a neighbor embedding method is invariant to a global rotation and a global shift of all embedding points, so it is reasonable to choose embedding points with an appropriate initialization. (ii) There are potentially multiple local minima of the loss function due to non-convexity. We verify the LOO assumption at a given local minimum (namely $\mathbf{Y}$) obtained from the first run.

The experiment is conducted with different sample sizes n and with different types of datasets (simulated cluster data, simulated manifold data, real single-cell data, deep learning feature data). The comprehensive results showing the values of ϵ_n under different settings are presented in Supplementary Table 2. We observe that the approximation errors ϵ_n are small and generally decreasing in n, which supports our LOO assumption.

Perturbation score

For implementation convenience, our calculation of the perturbation score and the singularity score is based on modifying an input point instead of adding a new input point. According to the LOO assumption, the difference is negligible.

Given an input data matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^{\top}$ and its embedding matrix $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n]^{\top}$, we view $\mathbf{y}_i$ as the mapping of $\mathbf{x}_i$ by the partial LOO-map $\mathbf{f}_i$:

$$\begin{aligned} \mathbf{f}_i(\mathbf{x}) &= \operatorname{argmin}_{\mathbf{y}\in\mathbb{R}^2} L_i(\mathbf{y};\mathbf{x}), \quad \text{where} \\ L_i(\mathbf{y};\mathbf{x}) &= \textstyle\sum_{k\neq i}\mathcal{L}\big(w(\mathbf{y}_k,\mathbf{y}); v_{i,k}(\bar{\mathbf{X}})\big) + Z(\bar{\mathbf{Y}}), \end{aligned} \tag{5}$$

where $\bar{\mathbf{X}} = [\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_n]^{\top}$ differs from $\mathbf{X}$ only at the i-th input point, and $\bar{\mathbf{Y}} = [\mathbf{y}_1, \ldots, \mathbf{y}_{i-1}, \mathbf{y}, \mathbf{y}_{i+1}, \ldots, \mathbf{y}_n]^{\top}$ has frozen embedding points except for the i-th point which is the decision variable in the optimization problem. This partial LOO-map $\mathbf{f}_i$ is based on perturbing (or modifying) a single input point rather than adding a new point, thus maintaining n points in total. According to the LOO assumption, $\mathbf{f}_i \approx \mathbf{f}$, so we calculate the perturbation score for the i-th point based on $\mathbf{f}_i$.

To assess the susceptibility of $\mathbf{y}_i$ under moderate perturbations in $\mathbf{x}_i$, we apply a perturbation of length λ in the direction of $\mathbf{e}$ to $\mathbf{x}_i$ and measure the resulting change in the embedding map determined by the partial LOO-map $\mathbf{f}_i$. In our implementation, we search the perturbation directions among the first 3 principal directions of the data $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ and their opposites $\{-\mathbf{e}_1, -\mathbf{e}_2, -\mathbf{e}_3\}$, and the perturbation length λ is specified by the user. In this way, we can define the perturbation score of the i-th data point as

$$\max_{\mathbf{e}\in\{\pm\mathbf{e}_1,\pm\mathbf{e}_2,\pm\mathbf{e}_3\}} \|\mathbf{f}_i(\mathbf{x}_i+\lambda\mathbf{e})-\mathbf{y}_i\|_2. \tag{6}$$

In general, perturbation scores are not sensitive to perturbation lengths. Supplementary Fig. 6 illustrates the perturbation scores of the CIFAR10 deep learning feature data for three perturbation lengths ($\lambda \in \{1, 2, 3\}$). Points with high perturbation scores remain consistent across different perturbation lengths. In practice, we recommend that users run perturbation scores on a subset of data points and test with a few different perturbation lengths. Conceptually, the perturbation score detects points that fall within a radius of λ around the location of the OI discontinuity.

Moreover, we provide two approximation algorithms to accelerate the calculation of the perturbation score for t-SNE along with a strategy for users to pre-screen points for which the perturbation score should be computed.

Approximation method 1. For high-dimensional input data, often PCA as a pre-processing step is implemented before calculating the similarity scores. As similarity scores are recalculated for each perturbation we consider, PCA is repeated numerous times, leading to a significant increase in computation. Since PCA is robust to perturbing a single input point, we reuse the pre-processed input points after one PCA calculation based on the original input data. This approximation avoids multiple calculations of PCA. We find that this approximation is sufficiently accurate, as the differences between perturbation scores by approximation method 1 and the exact perturbation scores are empirically negligible (Supplementary Fig. 8a).

Approximation method 2. Besides reducing PCA computations, we can further accelerate the calculation of perturbation scores by approximating the similarity scores.

Given the input data matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^\top$ and perplexity $\mathcal{P}$, the computation of (exact) similarity scores $(v_{i,j}(\mathbf{X}))_{i<j}$ in the t-SNE algorithm follows the steps below.

- Calculate the pairwise distance $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2$ for $i, j = 1, \ldots, n$.
- Find σ_i, $i = 1, \ldots, n$ that satisfies

$$-\sum_{j\neq i}\frac{\exp(-d_{ij}^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-d_{ik}^2/2\sigma_i^2)}\log_2\left(\frac{\exp(-d_{ij}^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-d_{ik}^2/2\sigma_i^2)}\right)=\log_2(\mathcal{P}). \tag{7}$$

- Calculate $p_{j|i} = \frac{\exp(-d_{ij}^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-d_{ik}^2/2\sigma_i^2)}$, $i, j = 1, \ldots, n$. And

$$v_{i,j}(\mathbf{X}) = \frac{p_{j|i}+p_{i|j}}{2n}.$$

The main computational bottleneck is at step 2, where we conduct a binary search algorithm for n times to solve $(\sigma_i)_{1\le i\le n}$.

To provide an approximation method, we note that when perturbing the k-th point, for $i \neq k$, Equation (7) still approximately holds for the original standard deviation σ_i since only one of the terms has been changed. Therefore, we can set $\widetilde{\sigma}_i \approx \sigma_i$ for $i \neq k$ as an approximation to $(\widetilde{\sigma}_i)_{1\le i\le n}$, the standard deviations after perturbation. In this way, we only need to conduct the binary search once to solve $\widetilde{\sigma}_k$, which significantly speeds up the calculation of the similarity scores after perturbation.

In terms of computational performance, approximation method 2 leads to a reduction of running time by nearly 80% for a dataset of size 5000. We also find that approximation method 2 is highly accurate. As shown in Supplementary Fig. 8b, perturbation scores based on approximation method 2 are approximately equal to the exact perturbation scores for most of the points.

Pre-screening of points. To further speed up the computation, we use the heuristic that embedding points receiving high perturbation scores are often found at the peripheries of clusters. This heuristic motivates us to calculate the perturbation scores only for the peripheral points in the embedding space, as these points are most likely to be unreliable. We find that applying this pre-screening step tends to find most of the unreliable points (Supplementary Fig. 7) with significantly increased computational speed.

We use the function `dbscan` in the R package dbscan (version 1.2-0) to identify embeddings on the periphery of clusters.

Singularity score

Given an input data matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^\top$ and its embedding matrix $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n]^\top$, we describe our derivation of singularity scores. If we add an infinitesimal perturbation $\epsilon\mathbf{e}$ to $\mathbf{x}_i$, then by the Taylor expansion of the partial LOO-map $\mathbf{f}_i$, the resulting change in the i-th embedding point is expressed as

$$\mathbf{f}_i(\mathbf{x}_i+\epsilon\mathbf{e})-\mathbf{y}_i = -\epsilon\mathbf{H}_i^{-1}\sum_{k:k\neq i}\frac{\partial^2\mathcal{L}(w(\mathbf{y}_i,\mathbf{y}_k);v_{i,k}(\mathbf{X}))}{\partial\mathbf{y}_i\partial\mathbf{x}_k^\top}\mathbf{e}+o(\epsilon), \tag{8}$$

where $\mathbf{H}_i$ denotes the Hessian matrix of the partial LOO loss $L_i(\mathbf{y}; \mathbf{x}_i)$ with respect to $\mathbf{y}$ at $\mathbf{y} = \mathbf{y}_i$. Notably, when $\epsilon = 0$ (no perturbation), we have $\mathbf{f}_i(\mathbf{x}_i) = \mathbf{y}_i$. Denote the total loss as

$$\mathfrak{L}(\mathbf{y}_1,\ldots,\mathbf{y}_n;\mathbf{X}) = \sum_{1\le i<j\le n}\mathcal{L}(w(\mathbf{y}_i,\mathbf{y}_j);v_{i,j}(\mathbf{X}))+Z(\mathbf{Y}).$$

Then, $\mathbf{H}_i$ can be written as

$$\mathbf{H}_i = \frac{\partial^2 L_i(\mathbf{y};\mathbf{x}_i)}{\partial\mathbf{y}\partial\mathbf{y}^\top}\Big|_{\mathbf{y}=\mathbf{y}_i} = \frac{\partial^2\mathfrak{L}(\mathbf{y}_1,\ldots,\mathbf{y}_n;\mathbf{X})}{\partial\mathbf{y}_i\partial\mathbf{y}_i^\top},$$

i.e., $\mathbf{H}_i$ is also equal to the Hessian matrix of the total loss $\mathfrak{L}$ with respect to the i-th variable taking value at $\mathbf{y}_i$.

Importantly, $\mathbf{H}_i$ is independent of the perturbation direction $\mathbf{e}$. The more singular $\mathbf{H}_i$ is, the more sensitive the embedding point of $\mathbf{x}_i$ becomes to infinitesimal perturbations. Thus, we define the singularity score of the i-th data point as the inverse of the smallest eigenvalue of the Hessian matrix of $\mathfrak{L}$, that is $\lambda_{\min}^{-1}(\mathbf{H}_i)$. Supplementary Methods 1 provides detailed derivations of Eq. (8), and Supplementary Methods 2 provides expressions of singularity scores for t-SNE, UMAP, and LargeVis.

Scoring metrics and statistical tests

Entropy of class probabilities. For a classification task, a statistical or machine learning algorithm outputs predicted class probabilities for a test data point. For example, in neural networks, the probabilities are typically obtained through a softmax operation in the final layer. Often, the model predicts a class with the largest probability among all classes. The entropy of the probabilities can quantify how confident the model is in its prediction.

For a classification task of k classes, if we denote the output class probabilities for one data point $\mathbf{x}$ as $\mathbf{p} = (p_1, \ldots, p_k)$, then we define the entropy as $E(\mathbf{p}) = -\sum_{j=1}^{k} p_j \log(p_j)$. This quantity is widely used for measuring class uncertainty.

Entropy difference. We will describe an uncertainty measurement given access to the labels of input points. For a dataset $(\mathbf{x}_i)_{i\le n}$ with clustering structures, we posit the following k-component Gaussian mixture model (GMM) from which each $\mathbf{x}_i$ is sampled. Consider a uniform prior on the k clusters, i.e., $p(A_j) = \frac{1}{k}, j = 1, 2, \ldots, k$. Given cluster membership A_j, we define the conditional probability density function

$$p(\mathbf{x}|A_j) = g(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j),$$

where $\boldsymbol{\mu}_j$, $\boldsymbol{\Sigma}_j$ are the mean and covariance matrix in the j-th component, and $g(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_i)$, $j = 1, 2, ..., k$ are the Gaussian density functions with mean $\boldsymbol{\mu}_j$ and covariance matrix $\boldsymbol{\Sigma}_j$. We then have the posterior probability of A_j given an observation $\mathbf{x}$ as

$$p(A_j|\mathbf{x}) = \frac{p(\mathbf{x}|A_j)}{\sum_{j=1}^{k} p(\mathbf{x}|A_j)}. \tag{9}$$

In the analysis of neighbor embedding methods, we will use the posterior probabilities as an uncertainty measurement. Given the ground-truth labels of the data points, we can fit two GMMs, one in the input space and the other in the embedding space, yielding estimated parameters $(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)_{j \le k}$ for each fitted GMM. Then we can calculate the posterior probabilities of each data point belonging to the k components by Equation (9) with fitted parameters, in both the input space and the embedding space. For any data point, denote the posterior probabilities in input space as $\mathbf{p} = (p_1, p_2, ..., p_k)$ and in embedding space as $\mathbf{q} = (q_1, q_2, ..., q_k)$. Finally, we define the entropy difference for each point as the difference between the entropy of $\mathbf{p}$ and the entropy of $\mathbf{q}$, i.e., $E(\mathbf{p}) - E(\mathbf{q}) = -\sum_{j=1}^{k} p_j \log(p_j) + \sum_{j=1}^{k} q_j \log(q_j)$.

The entropy difference measures the amount of decreased uncertainty of cluster membership. A positive entropy difference means $E(\mathbf{q}) < E(\mathbf{p})$, so the associated data point appears to be less ambiguous in cluster membership after embedding. Vice versa, a negative entropy difference means increased uncertainty after embedding.

Since calculating entropy differences is based on ground-truth labels and fitting a clear statistical model, we believe that entropy differences are a relatively objective evaluation of visual uncertainty. If a diagnostic score without label information is aligned with the entropy difference, then the diagnostic score is likely to be reliable.

Evaluation score of neighborhood preservation. We calculate pointwise neighborhood preservation scores to evaluate how well the local structures are preserved by an embedding algorithm. Given the input matrix $\mathbf{X}$ and the embedding matrix $\mathbf{Y}$, to calculate the neighborhood preservation score for the i-th point, we first identify its k-nearest neighbors in the input space, with their indices denoted as $\mathcal{N}_i = \{i_1, i_2, \ldots, i_k\}$. Then, we compute the distances from the i-th point to its neighbors in both the input and embedding spaces:

$$\begin{aligned} \mathbf{d}_i^{\text{input}} &= [d(\mathbf{x}_i, \mathbf{x}_{i_1}), \ldots, d(\mathbf{x}_i, \mathbf{x}_{i_k})]^\top \\ \mathbf{d}_i^{\text{embedding}} &= [d(\mathbf{y}_i, \mathbf{y}_{i_1}), \ldots, d(\mathbf{y}_i, \mathbf{y}_{i_k})]^\top. \end{aligned}$$

The neighborhood preservation score for the i-th point is defined as the correlation between $\mathbf{d}_i^{\text{input}}$ and $\mathbf{d}_i^{\text{embedding}}$. A higher correlation indicates better preservation of the neighborhood structure.

We use the median neighborhood preservation score across all points in the dataset to assess the overall neighborhood preservation of the embedding. For hyperparameters, we choose $k = [n/5]$ and use the Euclidean distance as the metric d in implementation.

Davies-Bouldin index. We calculate the DB index[45] using the R function `index.DB` in the R package clusterSim (version 0.51-3) with $p = q = 2$, i.e., using the Euclidean distance.

Within-cluster distance ratio. Consider m clusters and in each cluster i, there are n_i data points, denoted as $\{\mathbf{x}_{ij}\}_{1 \le j \le n_i}$. The centroid for each cluster is denoted as $\mathbf{x}_{i\cdot} = \frac{1}{n_i}\sum_{j=1}^{n_i} \mathbf{x}_{ij}$ and the mean of all data points is denoted as $\mathbf{x}_{\cdot\cdot} = \frac{1}{n}\sum_{i=1}^{m}\sum_{j=1}^{n_i} \mathbf{x}_{ij}$.

Denote the total sum of squares (TSS) and the within-cluster sum of squares (WSS) by

$$\mathrm{TSS} = \sum_{\mathrm{i}=1}^{\mathrm{m}} \sum_{\mathrm{j}=1}^{\mathrm{n_i}} \| \mathbf{x}_{\mathrm{ij}} - \mathbf{x}_{\cdot\cdot} \|_2^2, \quad \mathrm{WSS} = \sum_{\mathrm{i}=1}^{\mathrm{m}} \sum_{\mathrm{j}=1}^{\mathrm{n_i}} \| \mathbf{x}_{\mathrm{ij}} - \mathbf{x}_{\mathrm{i}\cdot} \|_2^2.$$

The within-cluster distance ratio is defined as $\mathrm{WCDR} = \frac{\mathrm{WSS}}{\mathrm{TSS}}$. A smaller within-cluster distance ratio WCDR indicates a more pronounced clustering effect.

Wilks' Λ. We compute Wilks' Λ statistic[46] by performing a multivariate analysis of variance using the `manova` function from the R package stats (version 4.2.1), followed by a statistical test.

Statistical tests for distribution difference of singularity scores. We have claimed that embedding points with large singularity scores tend to appear in random locations under small perplexities but appear in the periphery of clusters under large perplexities. To quantitatively verify such distinction, we conduct several statistical tests and find that the results of the tests support our claim about the distribution difference (Supplementary Tables 4–6). We provide the details of the tests as follows.

We first conducted Spearman's rank correlation tests. Given the embedding $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n]^\top$ and the cluster label of each point as well as their singularity scores $\mathbf{s} = [s_1, \ldots, s_n]^\top$, we can first calculate the distance of each point to its cluster center. The distance vector is denoted as $\mathbf{d} = [d_1, \ldots, d_n]^\top$. We then conduct the Spearman's rank correlation test[56] on the singularity scores $\mathbf{s}$ and the distances to cluster center $\mathbf{d}$. The tests show that there is no significant correlation under low perplexity but a significant correlation under larger perplexity.

We use the function `cor.test` in the R package stat (version 4.2.1) to perform Spearman's rank correlation tests.

We then conducted tests for the local regression models. To test for distribution differences, we first fit a local regression model[57] using the singularity scores as the response variables and the coordinates of embedding points as predictors. Next, we fit a null model with the singularity scores as the response and only the intercept as the predictor. An F-test is then conducted to determine whether the magnitude of the singularity scores is associated with the locations of the embedding points.

We also perform permutation tests by shuffling the singularity scores and fitting a local regression model for each shuffle to approximate a null distribution for the residual sum of squares. Empirical p-values are then computed to assess whether the singularity scores are distributed randomly. Lower p-values suggest rejecting the null hypothesis of random distribution.

We use the `loess` function from the R package stat (version 4.2.1) to fit the local regression models.

Benchmark methods for OOD detection

Kernel PCA. We implemented the state-of-the-art kernel PCA method for out-of-distribution detection[52] to benchmark against the perturbation score. Since our perturbation score does not require separate training and testing steps and was directly applied to the dataset, kernel PCA was trained on the dataset and then evaluated on the same dataset to ensure a fair comparison. Additionally, to maintain consistency with the default PCA preprocessing step in the t-SNE algorithm, we applied PCA before training, retaining the first 50 principal components.

One-class support vector machine. We implemented the one-class support vector machine (SVM)[53] using the `OneClassSVM` function from the Python package `scikit-learn` (version 1.2.0), employing a polynomial kernel for optimal performance. Since our perturbation score does not require separate training and testing steps and was directly applied to the dataset, one-class SVM was also trained on the dataset and then evaluated on the same dataset to ensure a fair comparison. To align with the preprocessing step in t-SNE, we first applied PCA, reducing the data to its top 50 principal components before training.

Datasets

Gaussian mixture data. A Gaussian mixture model with k components is a linear combination of k-component Gaussian densities. The probability density function of the random variable $\mathbf{x}$ generated by Gaussian mixture model[58] is

$$p(\mathbf{x}) = \sum_{i=1}^{k} \pi_i g(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i),$$

where $\boldsymbol{\mu}_i$, $\boldsymbol{\Sigma}_i$ are the mean and covariance matrix in the i-th component, the scalars π_i, $i = 1, 2, \ldots, k$ are the mixture weights satisfying $\sum_{i=1}^{k} \pi_i = 1$, and $g(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, $i = 1, 2, \ldots, n$ are the probability density functions of the Gaussian distribution family with mean $\boldsymbol{\mu}_i$ and covariance matrix $\boldsymbol{\Sigma}_i$.

We randomly generated Gaussian mixture datasets with various numbers of components and mixture weights using the function `rGMM` in the R package MGMM (version 1.0.1.1).

Swiss roll data. The Swiss roll data is a classical manifold data. Usually, the dataset consists of three-dimensional i.i.d. data points, denoted as $(x, y, z)^\top \in \mathbb{R}^3$, where

$$x = t\cos(t),\ y = t\sin(t), z = z.$$

Here, t is the parameter controlling the spiral angle and is uniformly distributed in a chosen range $[a, b]$. And z is the height parameter and is also uniformly distributed in the chosen span of heights $[c, d]$.

We randomly generated Swiss roll datasets and used the function `Rtsne` in the R package Rtsne (version 0.17) to obtain the t-SNE embeddings of the datasets. We computed the perturbation scores with perturbation length 1 in Fig. 7b.

Deep learning feature data. We used the pretrained ResNet-18 model to perform a forward pass on the CIFAR-10 dataset to extract features of dimension 512. We also performed the forward pass using the same pre-trained model on the Describable Textures Dataset (DTD) dataset[51] as our out-of-distribution data in Fig. 5. We also randomly subsampled both datasets to reduce computational load. Specifically, in Fig. 1, we sampled 5000 images from the CIFAR-10 test dataset as our deep learning feature data and obtained the t-SNE embedding under perplexity 125. We then computed the perturbation scores with perturbation length 2. In Fig. 5, we sampled 2000 CIFAR-10 images and 1000 DTD images, combining them into a dataset that includes OOD data points. We obtained the t-SNE embedding under perplexity 100 and computed the perturbation scores with perturbation length 2.

Mouse brain single-cell ATAC-seq data. The ATAC-seq dataset was created to capture the gene activity of mouse brain cells. The dataset has been preprocessed by Luecken et al.[59]. We applied the R functions `CreateSeuratObject`, `FindVariableFeatures` and `NormalizeData` in R package Seurat to identify 1000 most variable genes for 3618 cells. The dataset was subsampled when being used to verify the LOO assumption.

Mouse embryonic stem cell differentiation data. The single-cell RNA-seq dataset was constructed to investigate the dynamics of gene expression of mouse embryonic stem cells (mESCs) undergoing differentiation[54]. The dataset was preprocessed, normalized, and scaled by following the standard procedures by R package Seurat using functions `CreateSeuratObject`, `NormalizeData` and `ScaleData`. We also used R function `FindVariableFeatures` to identify the 2000 most variable genes for all 421 cells.

Human pancreatic tissue single-cell RNA-seq data. The single-cell RNA-seq data generated from human pancreatic tissues[60] provides a comprehensive view of gene expression across 8 different cell types in pancreatic tissue. The dataset was preprocessed, normalized, and scaled by following the standard procedures described above. We also used R function `FindVariableFeatures` to identify the 2000 most variable genes for all 2364 cells. The dataset was subsampled when being used to verify the LOO assumption.

Single-cell RNA-seq data of PBMCs with treatment of interferon-beta. This single-cell RNA-seq dataset profiles gene expression in peripheral blood mononuclear cells (PBMCs) following interferon-β (IFNB) treatment, capturing cellular responses to immune stimulation[61]. The dataset was preprocessed, normalized, and scaled by following the standard procedures described above. We used R function `FindVariableFeatures` to identify the 2000 most variable genes for all 6,548 cells. The dataset was subsampled when being used to verify the LOO assumption.

Mouse mammary epithelial single-cell data. This dataset contains the gene expression profile of mammary epithelial cells across from two mice at four developmental stages: nulliparous, mid-gestation, lactation, and post-involution[55]. The dataset was preprocessed, normalized, and scaled by following the standard procedures described above. We used R function `FindVariableFeatures` to identify the 2000 most variable genes for all 25,806 cells.

Implementation of t-SNE

We used the function `Rtsne` in the R package Rtsne (version 0.17) to perform the t-SNE algorithm. We choose `theta = 0` to perform exact t-SNE. We also adjusted the code in Rtsne to access the similarity scores $(v_{i,j}(\mathbf{X}))_{i<j}$. The adjusted function `Rtsne` can be found in https://github.com/zhexuandliu/MapContinuity-NE-Reliability.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

The datasets used in this study are publicly available and can be accessed through the following sources. The CIFAR-10 dataset used in this study is available at [https://www.cs.toronto.edu/~kriz/cifar.html]. The Describable Textures dataset used in this study is available at [https://www.robots.ox.ac.uk/~vgg/data/dtd/]. The pretrained ResNet-18 model is available at Hugging Face [https://huggingface.co/edadaltocg/resnet18_cifar10]. The mouse brain single-cell ATAC-seq data can be downloaded from Figshare [https://doi.org/10.6084/m9.figshare.12420968]. Mouse embryonic stem cell differentiation data is available with NCBI GEO accession code GSE98664. The single-cell RNA-seq dataset generated from PBMCs treated with interferon-β is available from the R package Seurat (version 5.0.3) under the name `ifnb`, and is also available with NCBI GEO accession code GSE96583. The single-cell RNA-seq data of human pancreatic tissues is available from the `smartseq2` dataset in the R package Seurat (version 5.0.3) under the name `panc8`, and is also available in EMBL-EBI with accession code E-MTAB-5061. Mouse mammary epithelial single-cell data is available from the R package scRNAseq (version 2.20.0) under the name `BachMammaryData`, and is also available with NCBI GEO accession code GSE106273. Source data are provided with this paper[62]. Source data are provided with this paper.

Code availability

The code for calculating the two diagnostic scores (as an R package), and the code for reproducing the simulation and analysis of this paper are available at the GitHub repository [https://github.com/zhexuandliu/MapContinuity-NE-Reliability] (Zenodo DOI: [https://doi.org/10.5281/zenodo.15384393])[62].

References

1. Van der Maaten, L. & Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**, 2579–2605 (2008).
2. McInnes, L., Healy, J., Saul, N. & Großberger, L. UMAP: Uniform Manifold Approximation and Projection. *J. Open Source Softw*, **3**, 861 (2018).
3. Kobak, D. & Berens, P. The art of using t-SNE for single-cell transcriptomics. *Nat. Commun.* **10**, 5416 (2019).
4. Linderman, G. C., Rachh, M., Hoskins, J. G., Steinerberger, S. & Kluger, Y. Fast interpolation-based t-SNE for improved visualization of single-cell RNA-seq data. *Nat. Methods* **16**, 243–245 (2019).
5. Luecken, M. D. & Theis, F. J. Current best practices in single-cell RNA-seq analysis: a tutorial. *Mol. Syst. Biol.* **15**, e8746 (2019).
6. Jing, R., Xue, L., Li, M., Yu, L. & Luo, J. layerumap: a tool for visualizing and understanding deep learning models in biological sequence classification using UMAP. *Iscience* **25**, 12 (2022).
7. Islam, M. T. et al. Revealing hidden patterns in deep neural network feature space continuum via manifold learning. *Nat. Commun.* **14**, 8506 (2023).
8. Van Assel, H., Espinasse, T., Chiquet, J. & Picard, F. A probabilistic graph coupling view of dimension reduction. *Adv. Neural Inf. Process. Syst.* **35**, 10696–10708 (2022).
9. Agrawal, A., Ali, A. & Boyd, S. et al. Minimum-distortion embedding. *Found. Trends Mach. Learn.* **14**, 211–378 (2021).
10. Tang, J., Liu, J., Zhang, M. & Mei, Q. Visualizing large-scale and high-dimensional data. In *Proceedings of the 25th international conference on world wide web*, 287–297 (2016).
11. Ghojogh, B., Crowley, M., Karray, F. & Ghodsi, A. *Elements of dimensionality reduction and manifold learning* (Springer, 2023).
12. Wei, J. et al. Diffusive topology preserving manifold distances for single-cell data analysis. *Proc. Natl Acad. Sci. USA* **122**, e2404860121 (2025).
13. Kim, J. & Wang, X. Inductive global and local manifold approximation and projection. *Transactions on Machine Learning Research* (2024).
14. Pearson, K. On lines and planes of closest fit to systems of points in space. *Lond. Edinb. Dublin Philos. Mag. J. Sci.* **2**, 559–572 (1901).
15. Wang, Y., Huang, H., Rudin, C. & Shaposhnik, Y. Understanding how dimension reduction tools work: an empirical approach to deciphering t-sne, umap, trimap, and pacmap for data visualization. *J. Mach. Learn. Res.* **22**, 1–73 (2021).
16. Chari, T. & Pachter, L. The specious art of single-cell genomics. *PLOS Comput. Biol.* **19**, e1011288 (2023).
17. Yang, Z., Peltonen, J. & Kaski, S. Majorization-minimization for manifold embedding. In *Artificial Intelligence and Statistics*, 1088–1097 (PMLR, 2015).
18. Kobak, D. & Linderman, G. C. Initialization is critical for preserving global data structure in both t-SNE and UMAP. *Nat. Biotechnol.* **39**, 156–157 (2021).
19. Cai, T. T. & Ma, R. Theoretical foundations of t-SNE for visualizing high-dimensional clustered data. *J. Mach. Learn. Res.* **23**, 1–54 (2022).
20. of Us Research Program Genomics Investigators, A. Genomic data in the All of US Research Program. *Nature* **627**, 340–346 (2024).
21. Marx, V. Seeing data as t-SNE and UMAP do. *Nat. Methods* **21**, 930–933 (2024).
22. Arora, S., Hu, W. & Kothari, P. K. An analysis of the t-SNE algorithm for data visualization. In *Conference on learning theory*, 1455–1462 (PMLR, 2018).
23. Linderman, G. C. & Steinerberger, S. Clustering with t-SNE, provably. *SIAM J. Math. Data Sci.* **1**, 313–332 (2019).
24. Steinerberger, S. & Zhang, Y. t-sne, forceful colorings, and mean field limits. *Res. Math. Sci.* **9**, 42 (2022).
25. Shaham, U. & Steinerberger, S. Stochastic neighbor embedding separates well-separated clusters. *arXiv preprint arXiv:1702.02670* (2017).
26. Wattenberg, M., Viégas, F. & Johnson, I. How to use t-SNE effectively. *Distill* (2016).
27. Belkina, A. C. et al. Automated optimized parameters for t-distributed stochastic neighbor embedding improve visualization and analysis of large datasets. *Nat. Commun.* **10**, 5415 (2019).
28. Xia, L., Lee, C. & Li, J. J. Statistical method scDEED for detecting dubious 2d single-cell embeddings and optimizing t-SNE and UMAP hyperparameters. *Nat. Commun.* **15**, 1753 (2024).
29. Johnson, E. M., Kath, W. & Mani, M. EMBEDR: distinguishing signal from noise in single-cell omics data. *Patterns* **3**, 100465 (2022).
30. Sun, E. D., Ma, R. & Zou, J. Dynamic visualization of high-dimensional data. *Nat. Comput. Sci.* **3**, 86–100 (2023).
31. Heiter, E. et al. Pattern or artifact? interactively exploring embedding quality with trace. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 379–382 (Springer, 2024).
32. Zhou, Y. & Sharpee, T. O. Using global t-SNE to preserve intercluster data structure. *Neural Comput.* **34**, 1637–1651 (2022).
33. Cooley, S. M., Hamilton, T., Aragones, S. D., Ray, J. C. J. & Deeds, E. J. A novel metric reveals previously unrecognized distortion in dimensionality reduction of scrna-seq data. *bioRxiv preprint bioRxiv: 689851* (2019).
34. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778 (2016).
35. Hinton, G. E. & Roweis, S. Stochastic neighbor embedding. *Advances in neural information processing systems* **15** (2002).
36. Quenouille, M. H. Notes on bias in estimation. *Biometrika* **43**, 353–360 (1956).
37. Stone, M. Cross-validatory choice and assessment of statistical predictions. *J. R. Stat. Soc.: Ser. B (Methodol.)* **36**, 111–133 (1974).
38. Stone, M. An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion. *J. R. Stat. Soc. Ser. B (Methodol.)* **39**, 44–47 (1977).
39. Geisser, S. The predictive sample reuse method with applications. *J. Am. Stat. Assoc.* **70**, 320–328 (1975).
40. Wahba, G. Smoothing noisy data with spline functions. *Numerische Math.* **24**, 383–393 (1975).
41. Golub, G. H., Heath, M. & Wahba, G. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics* **21**, 215–223 (1979).
42. Breiman, L. Heuristics of instability and stabilization in model selection. *Ann. Stat.* **24**, 2350–2383 (1996).
43. Breiman, L. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Stat. Sci.* **16**, 199–231 (2001).
44. Yu, B. Stability. *Bernoulli* **19**, 1484 – 1500 (2013).
45. Davies, D. L. & Bouldin, D. W. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence* 224–227 (1979).
46. Wilks, S. S. Certain generalizations in the analysis of variance. *Biometrika* **24**, 471–494 (1932).
47. Yang, J., Zhou, K., Li, Y. & Liu, Z. Generalized out-of-distribution detection: a survey. *Int. J. Comput. Vis.* **132**, 5635–5662 (2024).
48. Koh, P. W. et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International conference on machine learning*, 5637–5664 (PMLR, 2021).

49. Zech, J. R. et al. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. *PLoS Med.* **15**, e1002683 (2018).
50. Krizhevsky, A., Hinton, G. et al. Learning multiple layers of features from tiny images (2009).
51. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S. & Vedaldi, A. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3606–3613 (2014).
52. Fang, K. et al. Kernel PCA for out-of-distribution detection. In *Advances in Neural Information Processing Systems*, vol. 37, 134317–134344 (Curran Associates, Inc., 2024).
53. Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J. & Williamson, R. C. Estimating the support of a high-dimensional distribution. *Neural Comput.* **13**, 1443–1471 (2001).
54. Hayashi, T. et al. Single-cell full-length total RNA sequencing uncovers dynamics of recursive splicing and enhancer RNAs. *Nat. Commun.* **9**, 619 (2018).
55. Bach, K. et al. Differentiation dynamics of mammary epithelial cells revealed by single-cell RNA sequencing. *Nat. Commun.* **8**, 1–11 (2017).
56. Best, D. & Roberts, D. Algorithm as 89: the upper tail probabilities of Spearman's rho. *J. R. Stat. Soc. Ser. C. (Appl. Stat.)* **24**, 377–379 (1975).
57. Cleveland, W. S., Grosse, E. & Shyu, W. M. Local regression models. In *Statistical models in S*, 309–376 (Routledge, 2017).
58. Reynolds, D. A. et al. Gaussian mixture models. *Encycl. Biometrics* **741**, 3 (2009).
59. Luecken, M. D. et al. Benchmarking atlas-level data integration in single-cell genomics. *Nat. Methods* **19**, 41–50 (2022).
60. Satija Lab. *panc8.SeuratData: Eight Pancreas Datasets Across Five Technologies* R package version 3.0.2 (2019).
61. Kang, H. M. et al. Multiplexed droplet single-cell RNA-sequencing using natural genetic variation. *Nat. Biotechnol.* **36**, 89–94 (2018).
62. Liu, Z., Ma, R. & Zhong, Y. Assessing and improving reliability of neighbor embedding methods: a map-continuity perspective. zhexuandliu/MapContinuity-NE-Reliability, https://doi.org/10.5281/zenodo.15384393 (2025).

Acknowledgements

Y.Z. is supported by NSF-DMS grant 2412052 and by the Office of the Vice Chancellor for Research and Graduate Education at the UW Madison with funding from the Wisconsin Alumni Research Foundation. Z.L. and Y.Z. would like to thank Yixuan Li for suggesting out-of-distribution detection, and thank Sebastien Roch, Zexuan Sun, Xinyu Li, and Jingyang Lyu for helpful discussions. R.M. would like to thank Jonas Fischer, Dmitry Kobak, Stefan Steinerberger, and Bin Yu for helpful discussions on t-SNE and UMAP.

Author contributions

Y.Z. and R.M. conceived the study. Z.L. designed and implemented the method with input from Y.Z. and R.M. Z.L. contributed to the numerical analysis and software implementation. Y.Z. and R.M. designed and developed the theoretical results for the study. Z.L. prepared a draft of the manuscript. Y.Z. and R.M. edited the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41467-025-60434-9.

Correspondence and requests for materials should be addressed to Yiqiao Zhong.

Peer review information *Nature Communications* thanks the anonymous reviewer(s) for their contribution to the peer review of this work. A peer review file is available.

Reprints and permissions information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.