# Preserving clusters and correlations: a dimensionality reduction method for exceptionally high global structure preservation

Jacob Gildenblat, Jens Pahnke

***Abstract*—We present *Preserving Clusters and Correlations* (PCC), a novel dimensionality reduction (DR) method that achieves state-of-the-art global structure (GS) preservation while maintaining competitive local structure (LS) preservation. It optimizes two objectives: a GS preservation objective that preserves an approximation of Pearson and Spearman correlations between high- and low-dimensional distances, and an LS preservation objective that ensures clusters in the high-dimensional data are separable in the low-dimensional data. PCC has a state-of-the-art ability to preserve the GS while having competitive LS preservation. In addition, we show the correlation objective can be combined with UMAP to significantly improve its GS preservation with minimal degradation of the LS. We quantitatively benchmark PCC against existing methods and demonstrate its utility in medical imaging, and show PCC is a competitive DR technique that demonstrates superior GS preservation in our benchmarks.**

***Index Terms*—dimensionality reduction, UMAP, clustering, global structure preservation**

## I. Introduction

DIMENSIONALITY reduction (DR) methods are widely used in data science, both as a pre-processing technique for machine learning and to visualize data by transforming it into 2 or 3 dimensions. These methods can broadly be categorized into methods that focus on preserving the GS of the high-dimensional data, and those that focus on preserving the LS. PCA [1] and MDS [2] are examples of the former, and t-SNE [3] or Isomap [4] are examples of the latter. UMAP is a widely adopted DR method, with increased scalability and GS preservation compared to t-SNE [5].

Despite UMAP's widespread adoption, particularly in life sciences, several studies have highlighted its limitations, including poor GS preservation and sensitivity to initialization. [6]. Low GS preservation means that the distances between point clusters, and relationships between several points in general are not ensured to be meaningful. Even within clusters of points, local distances within clusters may not reflect true high-dimensional relationships.

Modern DR methods like UMAP, with high LS but low GS preservation, primarily preserve local clusters but do not reliably maintain inter-cluster relationships. This is still highly useful for data exploration and identifying clusters in the data. However, these methods may be misleading when it is required to distinguish between close points in the clusters, analyze global trends in the data, or have a reliable pre-processing step.

To address this, we propose a simple *global correlation loss objective* that excels in preserving GS. We sample data reference points and then for each point in the data, we measure the distances from these reference points. The correlation objective demands a high correlation between these distances in the high-dimensional data and in the learned low-dimensional representation. By enforcing correlation between high- and low-dimensional distances to reference points, PCC preserves the relative positioning of data points, ensuring a faithful low-dimensional representation.

For correlation we consider the Pearson [7] correlation, and a differential approximation of the Spearman rank correlation [8] proposed in [9]. This objective achieves high GS preservation, exceeding all current methods by a very large margin. Notably, it also considerably improves upon PCA, a classical method often considered the gold standard for GS preservation.

Unlike graph-based methods like UMAP [5], t-SNE [3] or PaCMAP [10], PCC is conceptually simple, and straightforward to implement. PCC can be connected to existing DR methods to improve GS preservation.

Our main contributions are:

1. A global correlation preservation objective based on sampling reference points and for each data point maximizing the high/low dimensional distance correlations between it and all the reference points.

2. We propose a simple method for preserving LS, requiring processing all points independently without constructing a neighbor graph, based on preserving clustering observability. We use a simple LS objective that preserves cluster separability in the low-dimensional representation, by joint learning of both the transformation and a linear classifier that classifies which clusters transformed points belong to.

3. We combine the correlation objective with UMAP in two different ways. In the first, we optimize for the UMAP objective enhanced with the correlation objective and show

*J. Gildenblat and J. Pahnke are with the Translational Neurodegeneration Research and Neuropathology Lab, Department of Clinical Medicine, Medical Faculty, University of Oslo (UiO) and Section of Neuropathology Research, Department of Pathology, Clinics for Laboratory Medicine, Oslo University Hospital (OUS), Norway. Web: www.pahnkelab.eu, e-mails: jacob.gildenblat@gmail.com; jens.pahnke@gmail.com.*

*J.P. is also with the Institute of Nutritional Medicine, University of Lübeck (UzL) and University Medical Center Schleswig-Holstein (UKSH), Germany, the Department of Neuromedicine and Neuroscience, The Faculty of Medicine and Life Sciences, University of Latvia (LU), and the Department of Neurobiology, School of Neurobiology, Biochemistry and Biophysics, The Georg S. Wise Faculty of Life Sciences, Tel Aviv University (TAU), Israel.*

that it significantly improves GS with minimal LS degradation. In the second, we consider enhancing precomputed UMAP representations by running a few iterations with the correlation loss.

## II. Previous work

PCA [7] is a linear dimensionality reduction technique that finds an orthonormal linear projection of the high-dimensional data that maximizes the variance. Distances between points that are along the principle components hyper-planes are fully preserved, while for other points their proximity to the principle components determines their distance preservation. Therefore PCA is a useful method for high GS preservation and is often considered a gold standard method for this, although if the selected number of components does not cover the variation in the data, distortions are expected.

UMAP [5] is a widely used non-linear dimensionality reduction technique that constructs a high-dimensional graph of data relationships and embeds it into a lower-dimensional space using a fuzzy topological framework. The focus of UMAP and similar methods is on preserving the local neighborhood of points through the neighbor graph and thus on the LS. UMAP can preserve some GS through an initialization that preserves GS, for example by initializing with PCA.

PaCMAP [10] is a method that follows up on UMAP to improve the GS preservation explicitly. It achieves this by constructing a graph with three types of edges: near pairs, mid-range pairs, and far pairs.

Finally, Parametric-UMAP [11] trains a parametric model with the UMAP objective with mini-batch gradient descent. They propose a GS preservation loss term by measuring distances between points in the batch and maximizing the Pearson correlation between the distances in the high and low dimensional spaces. This requires re-computing distances between all pairs in a batch, potentially limiting the batch size. We expand upon this idea for a non-parametric transformation, without mini batches, considering all data points at once.

## III. Method

### A. The Correlation objective for global structure preservation

The global correlation metric proposed in [12] measures the correlation of pairs of points in the high-dimensional data, and the transformed data, to evaluate how well the GS is preserved. Motivated by this metric, we aim to approximate it in a differentiable way and optimize for it directly.

Consider a group of points $\{\mathbf{x}_i\}_{i=1}^N$ that we want to transform to a lower dimensionality $y_i$. We sample a subset of K indices of points in $x$:

$$I = \{i_1, i_2, \ldots, i_K\}, \quad i_1, i_2, \ldots, i_K \in \{1, 2, \ldots, N\}$$

$I_j$ is then the index of the j'th sampled data point in $x$.

For a distance function D, e.g. the Euclidean distance, we measure the distances between each data point and all the reference points in the high dimensional data $x$ and the transformation $y$:

$$d_{ij}^x = D(x_i, x_{I_j}) = ||x_i - x_{I_j}||_2$$

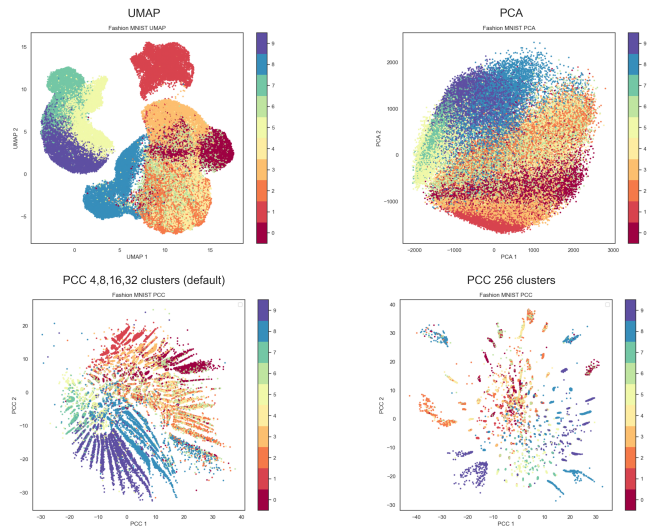


Fig. 1. Results on Fashion MNIST. In PCC, unlike UMAP, distances between different points are meaningful since GS is preserved. Unlike PCA, clusters are separated because of the higher GS. By using a higher cluster choice like 256, we get isolated groups of points belonging to those clusters.

$$d_{ij}^y = D(y_i, y_{I_j}) = ||y_i - y_{I_j}||_2$$

The correlation loss objective is to maximize the correlation between $d^x$ and $d^y$. For the Pearson correlation this is

$$L_{\text{Pearson}} = -\frac{\text{Cov}(d^x, d^y)}{\sqrt{\text{Var}(d^x)}\sqrt{\text{Var}(d^y)}}$$

$$= -\frac{\sum_{i=1}^N \sum_{j=1}^K (d_{ij}^x - \bar{d}^x)(d_{ij}^y - \bar{d}^y)}{\sqrt{\sum_{i=1}^N \sum_{j=1}^K (d_{ij}^x - \bar{d}^x)^2}\sqrt{\sum_{i=1}^N \sum_{j=1}^K (d_{ij}^y - \bar{d}^y)^2}}$$

We also consider the Spearman rank correlation (and the average of both correlations). Although the Pearson correlation implementation is differentiable and can be maximized with gradient descent, the Spearman rank correlation is not, because ranks are not differentiable. We use a differentiable approximation of the ranks, the soft ranking proposed in [9] and perform the correlation on that. For Spearman rank correlation, for each point we compute the soft rank approximations of the distances from the point to the k reference points and measure the correlation of these approximated ranks with the ranks of the same distances in the transformed lower-dimensional data.

$r_{ij}^x \in [1, K]$ are the ranks of the data point $i$ from reference points $j$, ranked against the K reference points. $r_{ij}^y$ is the approximated soft rank of the lower dimensional data point $i$ from reference point $j$, ranked against the K reference points.

$$L_{\text{spearman}} = -\frac{\text{Cov}(r^x, r^y)}{\sqrt{\text{Var}(r^x)}\sqrt{\text{Var}(r^y)}}$$

Finally, for the correlation loss, we use the average of the Pearson and Spearman correlation losses.

$$L_{\text{correlation}} = 0.5 L_{\text{Pearson}} + 0.5 L_{\text{Spearman}}$$

### B. The clustering observability objective for local structure preservation

DR methods with good LS behavior are often used to visualize or detect clusters in the data. In 2D scatter plots, DR points are often colorized by the clusters they belong to. We propose reversing this, directly clustering the original data, and then learning a reduced dimensionality embedding where it is possible to predict for every point which cluster (or clusters) it belongs to.

We hypothesize that given a good clustering model, the close neighborhood of a point in the high-dimensional data is more likely to belong to the same clusters. Therefore, if the LS is preserved in the transformed data, neighboring points should in most cases belong to the same clusters, and it should be possible to predict what cluster a transformed point belongs to. On the other hand, if it is not possible to predict which cluster a transformed data point belongs to, it means that points belonging to the same cluster are not grouped.

Motivated by this, we define a simple LS objective, by predicting the clusters. Given a clustering model that assigns each data point to one of k clusters, we learn a linear classifier A on top of the (also learned) transformed visualization that predicts the assigned cluster. For the linear classifier to be able to predict the clusters, the clusters have to be separable in the learned low-dimensional transformation. Thus, the joint learning of the cluster classifier and the transformation encourages separating or grouping data points according to their cluster.

$$\mathcal{L}_{\text{cluster}} = -\sum_i y_i \log((A \cdot e)_i)$$

The weights of this classifier are optimized jointly with the low-dimensional transformation.

In practice, we use several clustering models for different numbers of clusters with a multi-task loss function:

$$\mathcal{L}_{\text{cluster}} = -\frac{1}{M}\sum_{m=1}^{M}\sum_i y_i \log((A^{(m)} \cdot e)_i)$$

### C. Combining the local and global objectives

Deviating from UMAP/t-SNE, we use a random normal initialization and do not rely on initialization from PCA.

The loss is then

$$L_{PCC} = L_{\text{cluster}} + \beta \cdot L_{\text{correlation}}$$

For the cluster observability objective when applied, we use a multi-task objective, predicting all clusters, with k = 4, 8, 16, 32, or 64.

Examples for the Fashion-MNIST [13] dataset are shown in 1. PCC tends to create lines with points belonging to the same clusters created by the linear cluster assignment classifier. Different cluster choices affect the visualization output, with more clusters causing more isolated regions.

## IV. Experiments

### A. Benchmarking

We benchmarked PCC against several modern DR methods: UMAP [5], t-SNE [3], TriMap [14], PaCMAP [10] and PHATE [15]. For LS evaluation, we use the Trustworthiness and Continuity metrics [16] and the Mean Relative Rank Error metrics [17]. For GS evaluation, we use Pearson and Spearman correlations as proposed in [12]. For computing all metrics, we use the recent Zadu python library [18].

We evaluated 9 datasets covering different use cases: the MNIST [19] dataset and Fashion-MNIST [13] dataset often used as a more challenging alternative to MNIST. For samples of life sciences datasets, we used Macosko single-cell dataset [20]. For examples of deep learning embeddings, we used ResNet50 [21] embeddings of CIFAR [22], CIFAR100 [22], and the miniImageNET [23] datasets. Finally, for simple synthetic datasets employed to diagnose issues with DR methods, we used the mammoth [24] and Swiss roll [4] datasets.

### B. PCUMAP - Combining the global loss for Preserving Correlations with UMAP

We tested if we can improve the low GS in UMAP using the global correlation loss. We used a weight of 0.001 for the UMAP loss and added it to the PC loss. We used the TorchDR python package [25]. To help UMAP converge, we first run 10 iterations of the UMAP training before adding the global correlation loss.

$$L_{\text{PCUMAP}} = L_{\text{UMAP}} + \beta \cdot L_{corr}$$

Figure 2 presents the application of PCUMAP to the Macosko single-cell dataset [20], alongside PCC and UMAP. The bottom row visualizes the transformed data, with points colored based on their (thresholded) distances from a selected reference point, marked in red. In UMAP, both near (yellow) and distant (blue) points appear interspersed around the selected point, highlighting its poor global structure preservation. As a result, the relative distances between points are not reliable indicators of their true high-dimensional relationships. In contrast, PCUMAP and PCC exhibit a more structured separation, where nearby and distant points are clearly delineated, better reflecting the high-dimensional distance relationships.

### C. Initializing from UMAP and then running the global correlation objective

Here, we tested if we can add GS to an existing UMAP transformation. We initialized from UMAP and ran 3 iterations with the correlation loss with an additional mean squared error loss term that makes sure the result does not deviate too much from the initialization.

$$L_{\text{UMAP init + PC}} = L_{\text{corr}} + \lambda||e - e_{\text{UMAP}}||^2$$

$\lambda$ controls how close we want to keep to the initial embedding. We use $\lambda = 1$.

### D. Comparing UMAP and PCC visualizations of biological data

To assess the practical performance of PCC for the visualization of life science / biological data, we used mass spectrometry imaging (MSI) lipidomics data from an Alzheimer's

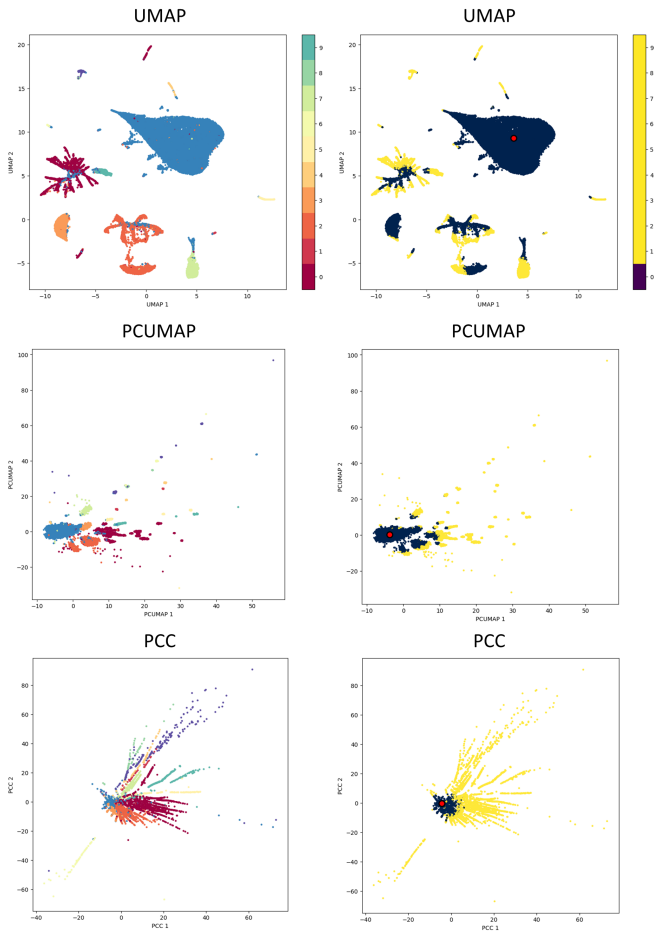


Fig. 2. Comparing UMAP (an existing method) and PCUMAP and PCC (our proposed methods) on the Macosko single cell dataset [20]. Upper row: The transformed data is colored by labels. Bottom row: colored according to distances from a selected point, in the high dimensional data. In UMAP the points in the low-dimensional data do not preserve the original distance: many points far away are close points in the high-dimensional data. In PCUMAP and PCC this is solved.

disease mouse model that was generated by us using a timsTOF fleX™ mass spectrometer in MALDI mode (Bruker Daltonics, Bremen, Germany). MSI is a challenging test case for DR methods since these images contain a large amount of detail that will not be revealed if the DR methods merely separate the data into clusters.

## V. Results

### A. Benchmarking

Figure 3 shows the benchmarking results averaged over the 9 tested datasets. As a single LS metric, we take the average of the 4 LS metrics. Similarly, for a GS metric, we used the average of the Pearson and Spearman correlation metrics.

PCA is a gold standard method for high GS preservation, but it is lower in LS preservation. Amongern methods, PaCMAP and TriMap improve over t-SNE/UMAP, with the PaCMAP method indeed performing the best, while PHATE performs the worst. PCC improves the GS preservation over all other tested methods by a large margin and is still a competitive method for LS preservation despite the clustering objective

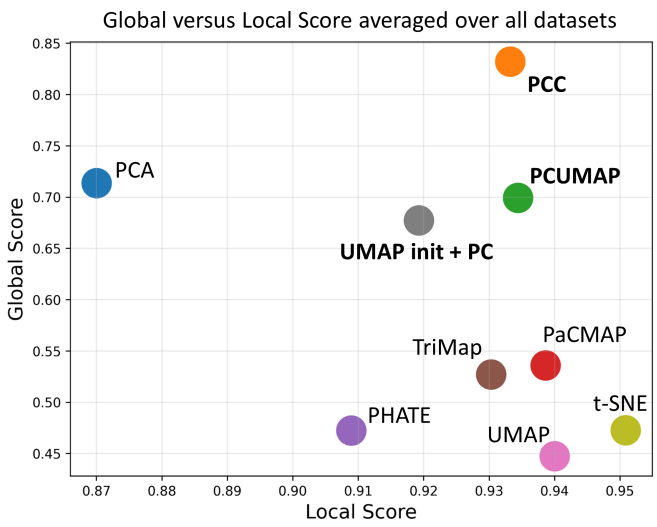


Fig. 3. Plotting the average performance of GS metrics against local structure metrics on 9 datasets. Our proposed methods: ***PCC, UMAP init+PC, PCUMAP.*** PCC improves global structure preservation over all other tested methods by a large margin (average of 0.83, while PCA gets 0.71 and UMAP 0.44), while being competitive in local structure preservation with graph methods that specialize in local structure (e.g, PCC gets 0.933 and UMAP 0.94). Among the modern graph methods, PaCMAP performs the best and slightly improves the global structure compared to UMAP. However, there is still room for improvement in global structure preservation, which we show is possible.

simplicity, making it a competitive method when the global data structure should be preserved. The full results are given in Table I.

Combined optimization of UMAP and the global loss (PCUMAP) is able to achieve GS close the PCA, and a high improvement compared to UMAP alone, with a small degradation in the LS.

Initialization from UMAP and 3 PC iterations (UMAP unit + PC) is also able to improve the GS substantially, however, results in losing LS.

### B. Comparison of UMAP and PCC using MSI visualization

In addition to the quantitative improvement on the MSI mouse model dataset in the benchmark, Figure 4 shows a comparison of UMAP and PCC to visualize lipidomics MSI data of a mouse brain hemisphere. The preservation of the global data structure improves significantly the visualization of pathological tissue changes (e.g., $\beta$-amyloid plaques as found in this Alzheimer's disease mouse model) but also allows the detection of normal anatomical structures (e.g., the neuronal band of the dentate gyrus).

## VI. Discussion

This paper presents PCC, a dimensionality reduction method that focuses on improving GS preservation. Compared to previous methods, PCC achieves the highest global structure preservation by a large margin. This is achieved by maximizing the correlations of the distances of all points from a set of reference points in the high-dimensional and low-dimensional data. This deviates from modern DR methods that focus on neighbor graph construction and achieving high

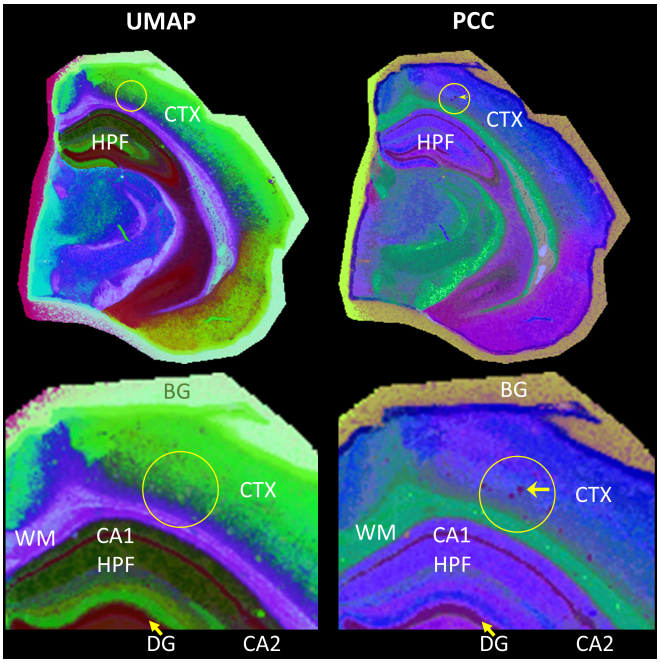


Fig. 4. Comparing UMAP and PCC using visualizations of lipidomics MSI data of mouse brain. Both methods reduce the high-dimensional image data to 3 dimensions which are then normalized and colored as RGB images. PCC reveals numerous pathological changes, so-called $\beta$-amyloid plaques (circle and arrow, brown dots) in the isocortex (CTX) of the Alzheimer's disease mouse model, but also enables visualization of normal structures, e.g. the neuronal band of the dentate gyrus (DG, arrow, band) in the hippocampus formation (HPF). Both structures are in-detectable using UMAP visualization (left images). Legend: CTX - isocortex of the cerebrum, HPF - hippocampus formation, DG - dentate gyrus, CA1 and CA2 - cornu ammonis neurons, area 1 and 2, WM - white matter, BG - background.

LS preservation. We show that follow-up methods on UMAP like TriMap and PaCMAP meant to solve the GS preservation problems of UMAP only marginally improve GS preservation as compared to our new PCC method. We also show that we can plug the global correlation loss into methods like non-parametric UMAP to improve their GS preservation. For local structure preservation, we proposed a simple multi-task cluster observability objective that is able to achieve competitive local structure preservation much better than PCA and even PHATE. However, it is still closely behind graph-based methods like UMAP. Follow-up work on PCC could include improving the clustering objective, combining the correlation objective with objectives of methods like UMAP, and adaptive sampling strategies for the reference points. Overall, when GS preservation is needed, PCC offers a DR option that is significantly improved over previously used methods.

## VII. Appendix

### A. Acknowledgements

We thank Dr Jorunn Stamnes and Thomas Brüning for tissue preparation and performing MSI measurements. Mass spectrometry-based analyses of mouse brains were performed at the Proteomics Core Facility at the University of Oslo/Oslo University Hospital, which is supported by the Core Facilities Program of the South-Eastern Norway Regional Health Authority (HSØ) and NAPI (www.napi.uio.no, NFR, Norway; 295910).

### B. Funding Information

J.P. received funding from Norges forskningsråd [NFR, Norway; 327571 (PETABC), 295910 (NAPI, www.napi.uio.no)], DFG (Germany; 263024513), HelseSØ (Norway; 2022046), and the EIC Pathfinder Open Challenges program (EU commission; 101185769).

### C. Data availability

The mass spectrometry imaging dataset of the mouse brain hemisphere is available at ProteomeXchange PRIDE repository with dataset number PXD056609.

TABLE I
Benchmarking results of DR methods for visualization on 9 datasets.

| Dataset | Method | Trustworthiness | Continuity | Mrre-False | Mrre-Missing | Pearson | Spearman |
|---|---|---|---|---|---|---|---|
| fashion_mnist | UMAP | 0.972 | 0.976 | 0.972 | 0.981 | 0.599 | 0.586 |
| fashion_mnist | pca | 0.915 | 0.971 | 0.913 | 0.974 | 0.880 | 0.879 |
| fashion_mnist | t-SNE | 0.980 | 0.974 | 0.984 | 0.979 | 0.646 | 0.640 |
| fashion_mnist | PaCMAP | 0.968 | 0.976 | 0.968 | 0.981 | 0.629 | 0.621 |
| fashion_mnist | TriMap | 0.964 | 0.981 | 0.965 | 0.984 | 0.682 | 0.677 |
| fashion_mnist | Phate | 0.942 | 0.974 | 0.945 | 0.979 | 0.641 | 0.651 |
| mouse_msi | UMAP | 0.963 | 0.961 | 0.963 | 0.966 | 0.695 | 0.770 |
| mouse_msi | pca | 0.921 | 0.968 | 0.923 | 0.970 | 0.988 | 0.985 |
| mouse_msi | t-SNE | 0.992 | 0.977 | 0.994 | 0.982 | 0.296 | 0.313 |
| mouse_msi | PaCMAP | 0.962 | 0.960 | 0.962 | 0.962 | 0.816 | 0.830 |
| mouse_msi | TriMap | 0.938 | 0.914 | 0.940 | 0.879 | 0.100 | 0.791 |
| mouse_msi | Phate | 0.967 | 0.983 | 0.969 | 0.986 | 0.874 | 0.872 |
| mammoth | UMAP | 0.990 | 0.978 | 0.992 | 0.984 | 0.774 | 0.800 |
| mammoth | pca | 0.967 | 0.992 | 0.959 | 0.991 | 0.992 | 0.991 |
| mammoth | t-SNE | 0.984 | 0.979 | 0.990 | 0.985 | 0.775 | 0.788 |
| mammoth | PaCMAP | 0.978 | 0.982 | 0.981 | 0.986 | 0.875 | 0.876 |
| mammoth | TriMap | 0.972 | 0.989 | 0.973 | 0.990 | 0.962 | 0.965 |
| mammoth | Phate | 0.941 | 0.968 | 0.953 | 0.976 | 0.236 | 0.287 |
| swiss_roll | UMAP | 0.997 | 0.981 | 0.996 | 0.988 | 0.423 | 0.377 |
| swiss_roll | pca | 0.876 | 0.977 | 0.891 | 0.977 | 0.847 | 0.845 |
| swiss_roll | t-SNE | 0.986 | 0.976 | 0.992 | 0.983 | 0.641 | 0.618 |
| swiss_roll | PaCMAP | 0.984 | 0.985 | 0.988 | 0.989 | 0.695 | 0.672 |
| swiss_roll | TriMap | 0.962 | 0.988 | 0.971 | 0.989 | 0.824 | 0.813 |
| swiss_roll | Phate | 0.884 | 0.977 | 0.889 | 0.982 | 0.409 | 0.367 |
| macosko | UMAP | 0.921 | 0.960 | 0.933 | 0.969 | 0.592 | 0.769 |
| macosko | pca | 0.745 | 0.904 | 0.744 | 0.912 | 0.924 | 0.937 |
| macosko | t-SNE | 0.923 | 0.958 | 0.943 | 0.965 | 0.442 | 0.591 |
| macosko | PaCMAP | 0.920 | 0.964 | 0.929 | 0.970 | 0.638 | 0.798 |
| macosko | TriMap | 0.907 | 0.967 | 0.918 | 0.972 | 0.671 | 0.795 |
| macosko | Phate | 0.829 | 0.923 | 0.838 | 0.937 | 0.724 | 0.859 |
| mnist | UMAP | 0.942 | 0.933 | 0.945 | 0.951 | 0.315 | 0.283 |
| mnist | pca | 0.739 | 0.908 | 0.737 | 0.922 | 0.536 | 0.505 |
| mnist | t-SNE | 0.954 | 0.937 | 0.968 | 0.953 | 0.362 | 0.332 |
| mnist | PaCMAP | 0.935 | 0.928 | 0.938 | 0.947 | 0.332 | 0.296 |
| mnist | TriMap | 0.919 | 0.932 | 0.922 | 0.950 | 0.196 | 0.190 |
| mnist | Phate | 0.855 | 0.935 | 0.860 | 0.951 | 0.313 | 0.287 |
| cifar | UMAP | 0.900 | 0.920 | 0.904 | 0.932 | 0.402 | 0.409 |
| cifar | pca | 0.771 | 0.894 | 0.769 | 0.903 | 0.544 | 0.534 |
| cifar | t-SNE | 0.923 | 0.911 | 0.939 | 0.924 | 0.479 | 0.467 |
| cifar | PaCMAP | 0.887 | 0.913 | 0.886 | 0.924 | 0.395 | 0.415 |
| cifar | TriMap | 0.877 | 0.916 | 0.880 | 0.927 | 0.410 | 0.408 |
| cifar | Phate | 0.836 | 0.905 | 0.840 | 0.919 | 0.412 | 0.417 |
| cifar100 | UMAP | 0.893 | 0.901 | 0.899 | 0.921 | 0.424 | 0.412 |
| cifar100 | pca | 0.721 | 0.863 | 0.717 | 0.877 | 0.416 | 0.393 |
| cifar100 | t-SNE | 0.909 | 0.896 | 0.930 | 0.915 | 0.417 | 0.403 |
| cifar100 | PaCMAP | 0.881 | 0.885 | 0.884 | 0.906 | 0.261 | 0.266 |
| cifar100 | TriMap | 0.851 | 0.903 | 0.856 | 0.921 | 0.376 | 0.369 |
| cifar100 | Phate | 0.815 | 0.901 | 0.820 | 0.919 | 0.475 | 0.468 |
| imagenetmini | UMAP | 0.852 | 0.883 | 0.885 | 0.918 | 0.141 | 0.131 |
| imagenetmini | pca | 0.665 | 0.837 | 0.661 | 0.851 | 0.330 | 0.320 |
| imagenetmini | t-SNE | 0.849 | 0.881 | 0.900 | 0.917 | 0.154 | 0.144 |
| imagenetmini | PaCMAP | 0.830 | 0.884 | 0.850 | 0.919 | 0.122 | 0.111 |
| imagenetmini | TriMap | 0.817 | 0.891 | 0.838 | 0.924 | 0.132 | 0.124 |
| imagenetmini | Phate | 0.758 | 0.878 | 0.775 | 0.912 | 0.107 | 0.107 |
| fashion_mnist | UMAP init + PC | 0.954 | 0.976 | 0.954 | 0.979 | 0.800 | 0.794 |
| mouse_msi | UMAP init + PC | 0.960 | 0.960 | 0.960 | 0.965 | 0.775 | 0.814 |
| mammoth | UMAP init + PC | 0.985 | 0.983 | 0.986 | 0.985 | 0.864 | 0.893 |
| swiss_roll | UMAP | 0.996 | 0.979 | 0.996 | 0.987 | 0.393 | 0.339 |
| swiss_roll | UMAP init + PC | 0.985 | 0.976 | 0.985 | 0.983 | 0.553 | 0.510 |
| macosko | UMAP init + PC | 0.898 | 0.944 | 0.911 | 0.953 | 0.694 | 0.813 |
| mnist | UMAP init + PC | 0.889 | 0.937 | 0.890 | 0.951 | 0.519 | 0.498 |
| cifar | UMAP init + PC | 0.837 | 0.910 | 0.837 | 0.917 | 0.659 | 0.668 |
| cifar100 | UMAP | 0.892 | 0.901 | 0.899 | 0.921 | 0.421 | 0.410 |
| cifar100 | UMAP init + PC | 0.778 | 0.881 | 0.783 | 0.891 | 0.715 | 0.728 |
| imagenetmini | UMAP | 0.847 | 0.883 | 0.883 | 0.918 | 0.141 | 0.130 |
| imagenetmini | UMAP init + PC | 0.751 | 0.888 | 0.765 | 0.907 | 0.452 | 0.443 |
| fashion_mnist | PCC | 0.962 | 0.973 | 0.965 | 0.975 | 0.873 | 0.878 |
| mouse_msi | PCC | 0.956 | 0.955 | 0.958 | 0.956 | 0.978 | 0.975 |
| mammoth | PCC | 0.970 | 0.987 | 0.975 | 0.985 | 0.981 | 0.979 |
| swiss_roll | PCC | 0.957 | 0.977 | 0.970 | 0.980 | 0.835 | 0.830 |
| macosko | PCC | 0.898 | 0.945 | 0.906 | 0.949 | 0.967 | 0.946 |
| mnist | PCC | 0.878 | 0.929 | 0.885 | 0.936 | 0.726 | 0.759 |
| cifar | PCC | 0.864 | 0.916 | 0.871 | 0.923 | 0.670 | 0.656 |
| cifar100 | PCC | 0.828 | 0.893 | 0.837 | 0.902 | 0.650 | 0.611 |
| imagenetmini | PCUMAP | 0.813 | 0.891 | 0.830 | 0.920 | 0.287 | 0.234 |
| cifar100 | PCUMAP | 0.857 | 0.908 | 0.859 | 0.924 | 0.524 | 0.508 |
| cifar | PCUMAP | 0.873 | 0.920 | 0.873 | 0.930 | 0.555 | 0.539 |
| mnist | PCUMAP | 0.929 | 0.916 | 0.931 | 0.933 | 0.504 | 0.496 |
| macosko | PCUMAP | 0.910 | 0.962 | 0.923 | 0.966 | 0.888 | 0.894 |
| swiss_roll | PCUMAP | 0.921 | 0.984 | 0.930 | 0.986 | 0.864 | 0.858 |
| mammoth | PCUMAP | 0.969 | 0.992 | 0.966 | 0.990 | 0.989 | 0.988 |
| fashion_mnist | PCUMAP | 0.965 | 0.979 | 0.965 | 0.981 | 0.792 | 0.790 |
| mouse_msi | PCUMAP | 0.985 | 0.986 | 0.985 | 0.987 | 0.937 | 0.943 |

## References


[1] K. Pearson, "On lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[2] W. S. Torgerson, *Multidimensional scaling: Theory and method*. Psychometrika, Volume 17, Issue 4, pp. 401–419, 1952.

[3] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 11, pp. 2579–2605, 2008.

[4] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

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

[6] D. Kobak and G. Linderman, "Initialization is critical for preserving global data structure in both t-sne and umap," *Nature Biotechnology*, vol. 39, no. 2, pp. 156–157, 2021.

[7] K. Pearson, "Note on regression and inheritance in the case of two parents," *Proceedings of the Royal Society of London*, vol. 58, pp. 240–242, 1895.

[8] C. Spearman, "The proof of the existence of general intelligence," *The American Journal of Psychology*, vol. 15, no. 2, pp. 201–293, 1904.

[9] M. Blondel *et al.*, "Fast differentiable sorting and ranking," *arXiv preprint arXiv:2002.08871*, 2020. [Online]. Available: https://arxiv.org/abs/2002.08871

[10] Y. Wang, H. Huang, C. Rudin, and Y. Shaposhnik, "Understanding how dimension reduction tools work: An empirical approach to deciphering t-sne, umap, trimap, and pacmap for data visualization," *Journal of Machine Learning Research*, vol. 22, no. 201, pp. 1–73, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-1061.html

[11] T. Sainburg, L. McInnes, and T. Q. Gentner, "Parametric umap: learning embeddings with deep neural networks for representation and semi-supervised learning," *arXiv preprint arXiv:2009.12981*, 2021. [Online]. Available: https://arxiv.org/abs/2009.12981

[12] X. Geng, D. chuan Zhan, and Z. hua Zhou, "Supervised nonlinear dimensionality reduction for visualization and classification," *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 35, no. 5, pp. 1162–1172, 2005.

[13] Google Research, "Fashion mnist dataset," 2017. [Online]. Available: https://github.com/zalandoresearch/fashion-mnist

[14] E. Amid and M. K. Warmuth, "Trimap: Large-scale dimensionality reduction using triplets," *arXiv preprint arXiv:1910.00204*, 2019.

[15] K. R. Moon, D. van Dijk, W. Zheng, and et al., "Phate: A dimensionality reduction method for visualizing trajectory structures in high-dimensional biological data," *bioRxiv*, 2017. [Online]. Available: https://doi.org/10.1101/120378

[16] J. A. Lee and M. Verleysen, *Nonlinear Dimensionality Reduction*. Springer-Verlag New York, 2007.

[17] M. Verleysen and J. A. Lee, "Quality assessment of dimensionality reduction: Rank-based criteria," *Neurocomputing*, vol. 72, no. 7-9, pp. 1431–1443, 2009.

[18] H. Jeon, A. Cho, J. Jang, S. Lee, J. Hyun, H.-K. Ko, J. Jo, and J. Seo, "Zadu: A python library for evaluating the reliability of dimensionality reduction embeddings," 2023. [Online]. Available: https://arxiv.org/abs/2308.00282

[19] Y. LeCun, C. Cortes, and C. Burges, "Mnist database of handwritten digits," 1998. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[20] E. Z. Macosko *et al.*, "Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets," *Cell*, vol. 161, no. 5, pp. 1202–1214, 2015.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[22] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," in *Tech Report*, 2009. [Online]. Available: http://www.cs.toronto.edu/~kriz/cifar.html

[23] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, and R. S. Zemel, "Meta-learning for semi-supervised few-shot classification," in *Proceedings of 6th International Conference on Learning Representations (ICLR)*, 2018.

[24] P. Code, "Understanding umap," 2020, accessed: 2024-11-18. [Online]. Available: https://pair-code.github.io/understanding-umap/

[25] H. Van Assel, N. Courty, R. Flamary, A. Garivier, M. Massias, T. Vayer, and C. Vincent-Cuaz, "Torchdr: Pytorch dimensionality reduction library," 2024. [Online]. Available: https://github.com/TorchDR/TorchDR