t-SNE and UMAP

Nava Leibovich and Matt Smart

2021-10-29

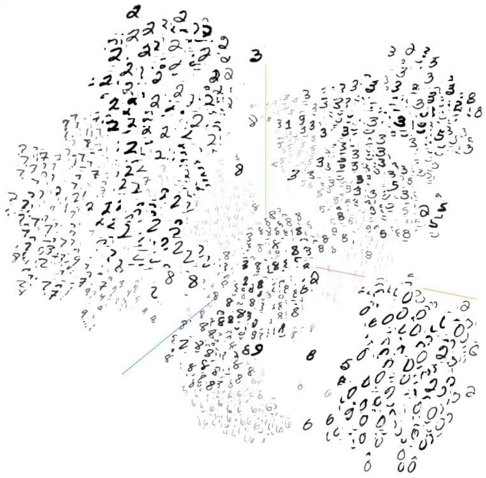
Goal: Dimension reduction

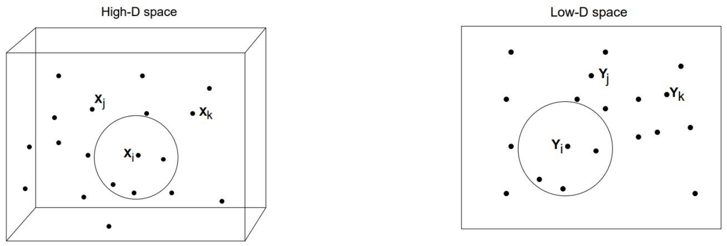
- Collection of high-dimensional data

$$\{\mathbf{x}_i\}_{i=1}^{M} \text{ with } \mathbf{x}_i \in \mathbb{R}^n$$

- Embed to lower dimension $p < n$

$$\{\mathbf{y}_i\}_{i=1}^{M} \text{ with } \mathbf{y}_i \in \mathbb{R}^p$$

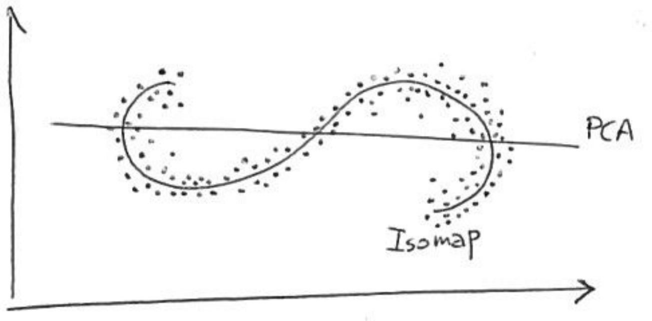


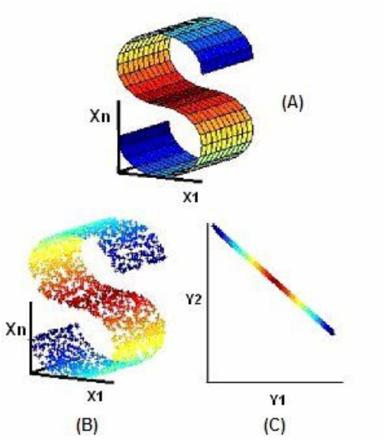


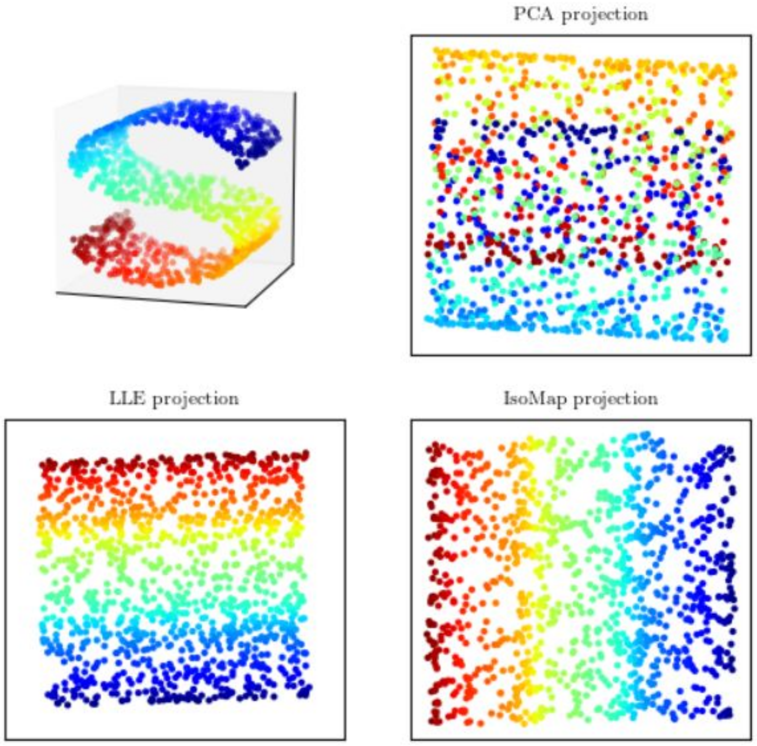
Approaches:

- Linear: PCA
- Non-linear: MDS, LLE, Laplacian Eigenmaps, t-SNE, UMAP

Linear vs. Non-linear Dimension Reduction

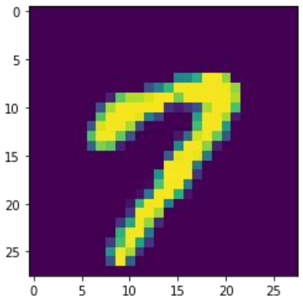


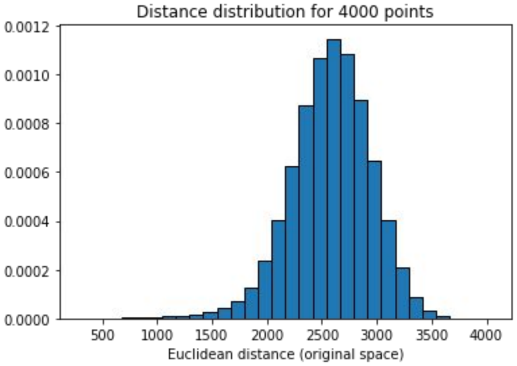


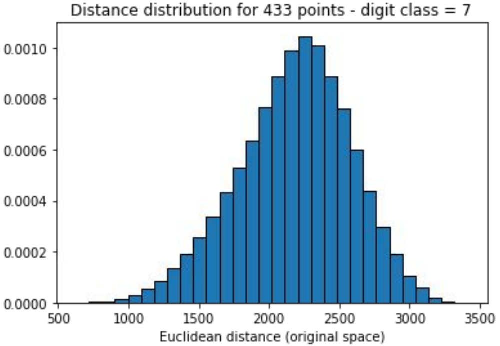


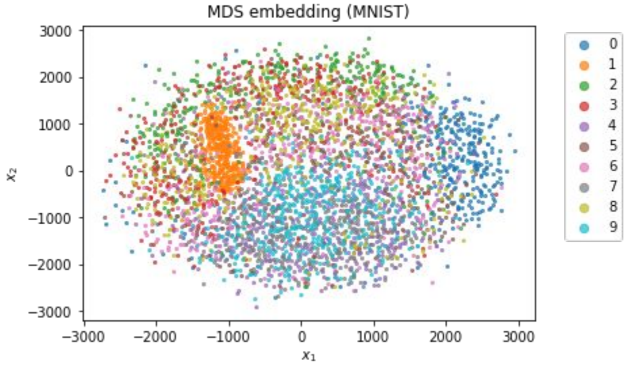
Issue: High-dimensional distance is “qualitatively different”

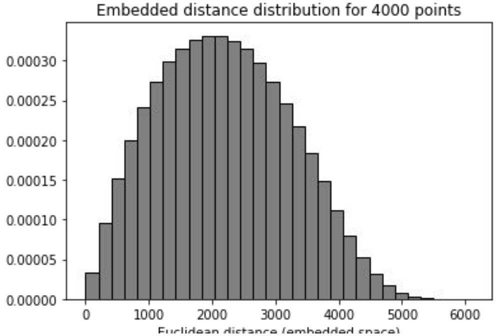
$\mathbf{x}_i \in \mathbb{R}^{784}$











Multi-dimensional scaling
- tries to match distances
- not possible to have no small distances
- fails on MNIST discrimination

Rather than embedding **distance** data to low-dimensions, embed **“neighborhood structure”**

t-distributed Stochastic Neighbor Embedding (t-SNE)

- Define probability of $\mathbf{x}_j$ being a neighbor of $\mathbf{x}_i$ $$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma_i^2)}{\sum_{i \neq k} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2 / 2\sigma_i^2)}$$
    - Each point *i* has a neighborhood parameter σ_i
    - The σ_i are computed by fixing the entropy of each distribution: $\mathrm{Perplexity} = 2^{-\sum_j p_{j|i} \log_2 p_{j|i}}$

- Define a symmetrized version: $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2M}$ and set $p_{ii} = 0$

- Define analogous distribution for the embedding (where the "*t*" comes in) $$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_k \sum_{l \neq k} (1 + \|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}}$$

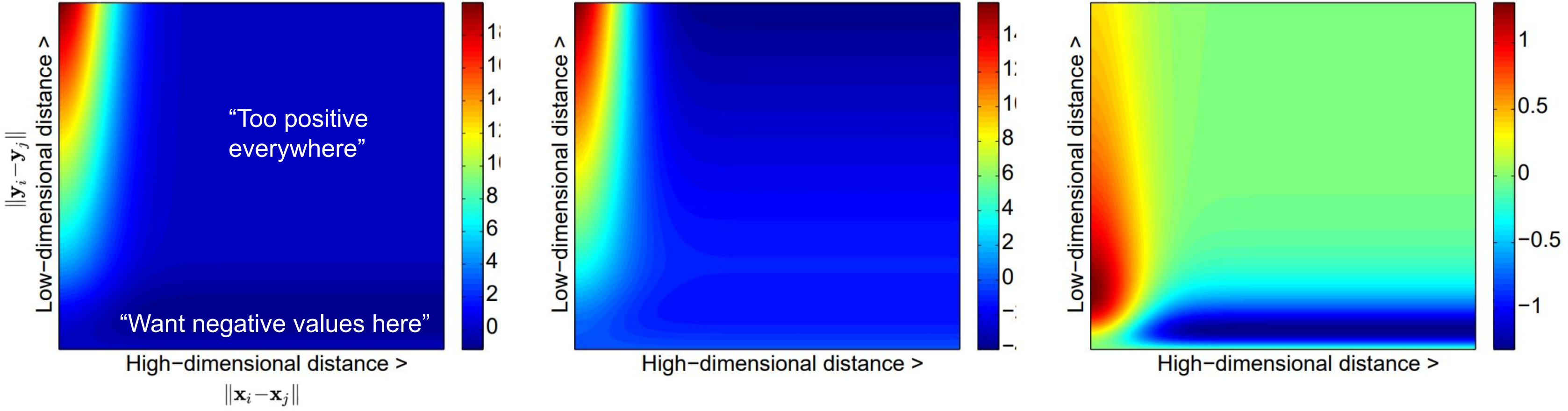
t-distributed Stochastic Neighbor Embedding (t-SNE)

- Objective: find $\{\mathbf{y}_i\}_{i=1}^{M}$ such that $q_{ij} \approx p_{ij}$

- Cost function: KL-divergence $C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$

- Optimization:
  - Compute gradient $\frac{\partial C}{\partial \mathbf{y}_i} = 4 \sum_j (p_{ij} - q_{ij})(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}(\mathbf{y}_i - \mathbf{y}_j)$
  - Random initial condition $\mathbf{y}_i^{(t=0)} \sim N(0, 10^{-4}\mathbf{I}_p)$
  - *Gradient descent $\mathbf{y}_i^{(t)} = \mathbf{y}_i^{(t-1)} + \eta \frac{\partial C}{\partial \mathbf{y}_i} + \alpha(t)(\mathbf{y}_i^{(t-1)} - \mathbf{y}_i^{(t-2)})$

*not sure if they update the vectors sequentially or in parallel via matrix updates of $\mathbf{Y} \in \mathbb{R}^{p \times M}$
**the first term should have a negative sign (consider a 1D minimization problem to recall why)



(a) Gradient of SNE. (b) Gradient of UNI-SNE. (c) Gradient of t-SNE.

Gaussian Distribution	**SNE + “Uniform Background”**	**Student-t distribution with $\nu = 1$**
$$q_{ij} = \frac{\exp(-\|\mathbf{y}_i - \mathbf{y}_j\|^2)}{\sum_k \sum_{l \neq k} \exp(-\|\mathbf{y}_k - \mathbf{y}_l\|^2)}$$ Issue (among others): crowding of points.	$$q_{ij} = \frac{(1-\rho)\exp(-\|\mathbf{y}_i - \mathbf{y}_j\|^2)}{\sum_k \sum_{l \neq k} \exp(-\|\mathbf{y}_k - \mathbf{y}_l\|^2)} + \frac{2\rho}{M(M-1)}$$ Tried to fix crowding issue in SNE. Add uniform background of fake points to add repulsion (negative gradient).	$$q_{ij} = \frac{(1+\|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_k \sum_{l \neq k} (1+\|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}}$$
[SNE paper] Hinton, Roweis. (2002) Stochastic Neighbor Embedding. NIPS.	***[UNI-SNE paper]*** Cook, Sutskever, Mnih, Hinton. (2007). Visualizing Similarity Data with a Mixture of Maps. PMLR.	***[t-SNE paper]*** Van der Maaten, Hinton. (2008) Visualizing Data using t-SNE. JMLR

Weaknesses: t-SNE

- 2D or 3D: method is for reduction to 2 or 3 dimensional space. (higher dimension need other similarity distribution, maybe higher degree of t-distribution)

- Curse of dimensionality:

  * the method is based on euclidean metric (assume local linearity on the manifold) → in high dimensional data the assumption cannot be fulfilled.

  * sparse data - all points might appears statistically dissimilar

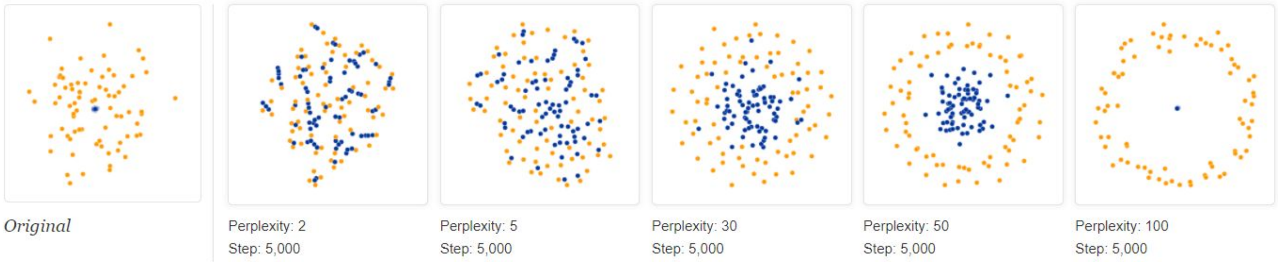
  * intuition: dimension reduction of high dimensional data will lose important information

- Non-convex cost function
  - Non-linear but convex - classical scaling, Isomap, LLE, and diffusion maps

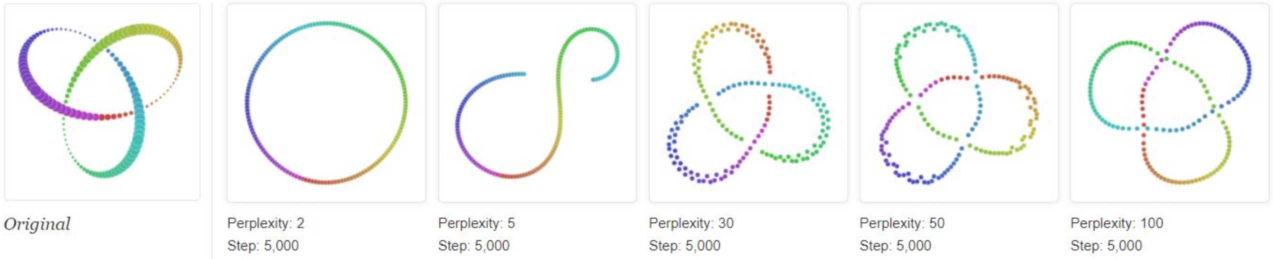
t-SNE Hyperparameters and Examples

Two nested clusters

- 75 points per cluster
- 50 dimensions

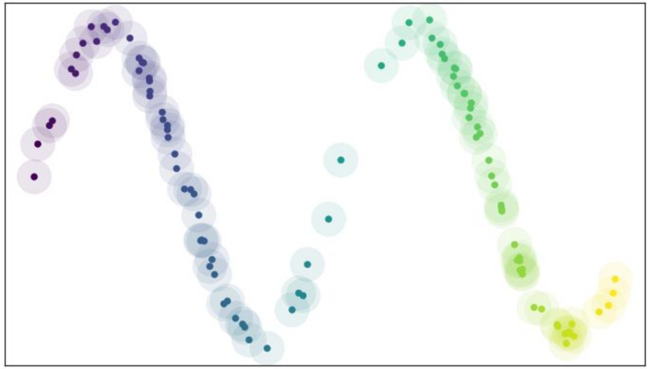


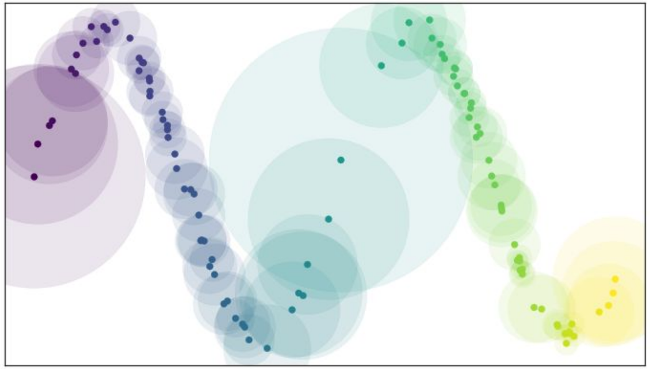
3D topology



https://distill.pub/2016/misread-tsne/

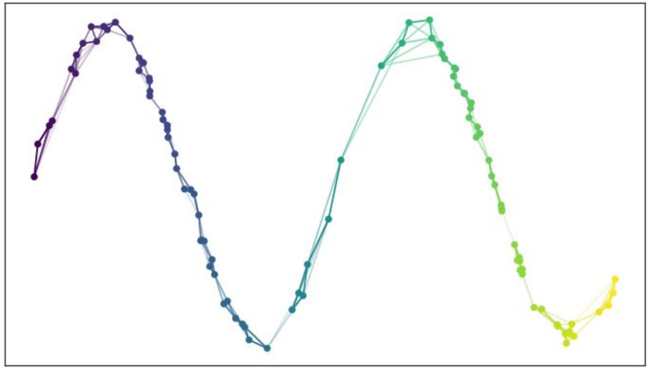
HIGH LEVEL INTUITION SLIDES MOTIVATING UMAP





1. There exists a manifold on which the data would be uniformly distributed.
2. The underlying manifold of interest is locally connected.
3. Preserving the topological structure of this manifold is the primary goal.

Weighted connected graph →



UMAP from a graph perspective

1. Construct a weighted graph representing the pairwise similarities

For each point $\mathbf{x}_i$

(a) Find its *k*-nearest neighbors (kNN) using a particular distance $d_{ij} = d(\mathbf{x}_i, \mathbf{x}_j)$
(b) Define ρ_i - the smallest *positive* distance to one of the neighbours
(c) Solve for σ_i from the SNE-like constraint

$$\log_2(k) = \sum_{j=1}^{k} \exp\left(\frac{-\max\{0,\, d_{ij} - \rho_i\}}{\sigma_i}\right)$$

Define the weights: $p_{i|j} = \exp\left(\frac{-\max\{0,\, d_{ij} - \rho_i\}}{\sigma_i}\right)$ ($p_{i|j} = 0$ for non-neighbors)

Define the symmetrized graph adjacencies: $p_{ij} = p_{i|j} + p_{j|i} - p_{i|j} \cdot p_{j|i}$

UMAP from a graph perspective

2. Compute a low-dimensional representation of the graph

- Want to find embedding $\mathbf{Y} \in \mathbb{R}^{p \times M}$ with "similar graph structure"

- Given the embedding, UMAP defines low-dimensional similarities as

$$q_{ij} = \left(1 + a\|\mathbf{y}_i - \mathbf{y}_j\|_2^{2b}\right)^{-1}$$

  - a, b are hyper-parameters solved for by a "minimum distance" constraint
  - $a = 1, b = 1$ - corresponds to t-SNE
  - $a \approx 1.9, b \approx 0.79$ - corresponds to default UMAP: $\text{min\_dist} = 0.1$

- UMAP cost function is the "fuzzy set cross-entropy" (p56)

$$C = \sum_{i \neq j} p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right) + (1 - p_{ij}) \log\left(\frac{1-p_{ij}}{1-q_{ij}}\right)$$

UMAP from a graph perspective

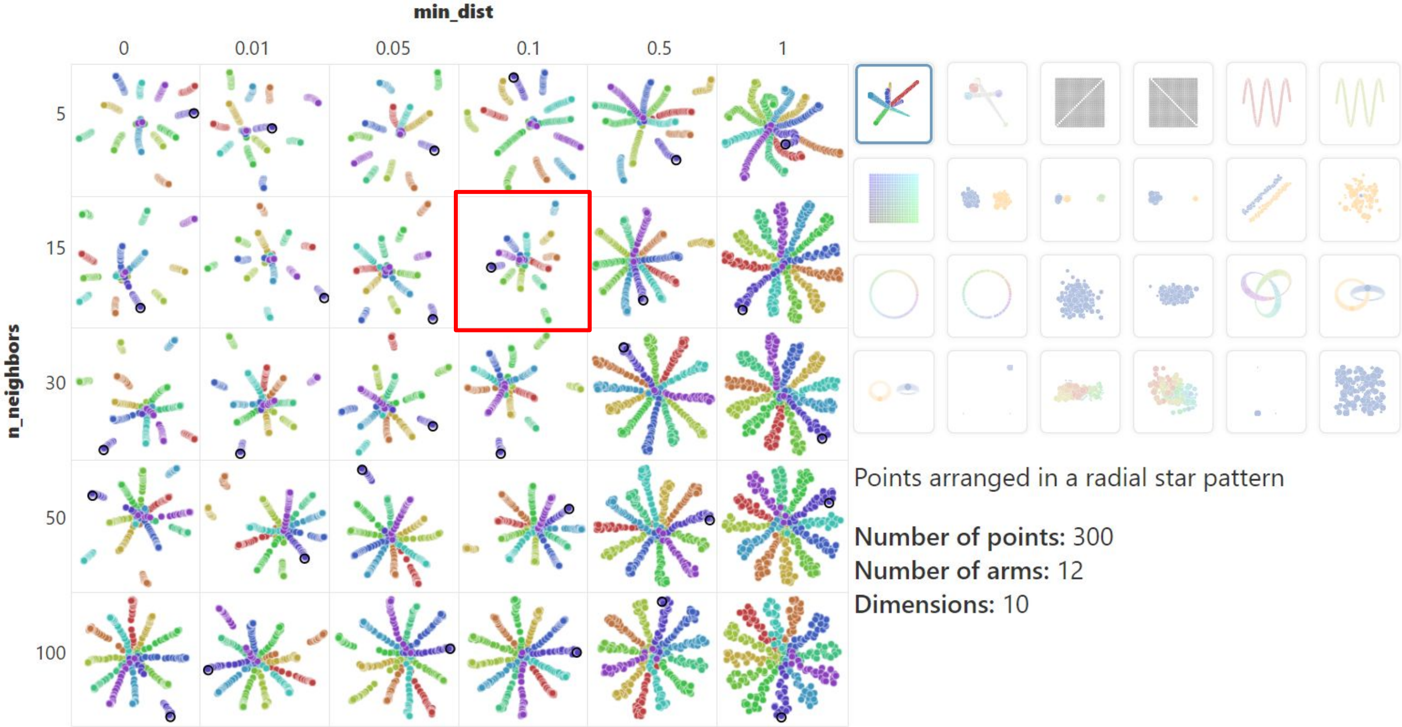
2. Compute a low-dimensional representation of the graph

- Initialization: use the spectral embedding of the constructed graph

$$\mathbf{Y}^{(t=0)} \in \mathbb{R}^{p \times M} = \text{top } p \text{ eigenvectors of laplacian } \mathbf{L} = \mathbf{D}^{1/2}(\mathbf{D}-\mathbf{P})\mathbf{D}^{1/2}$$

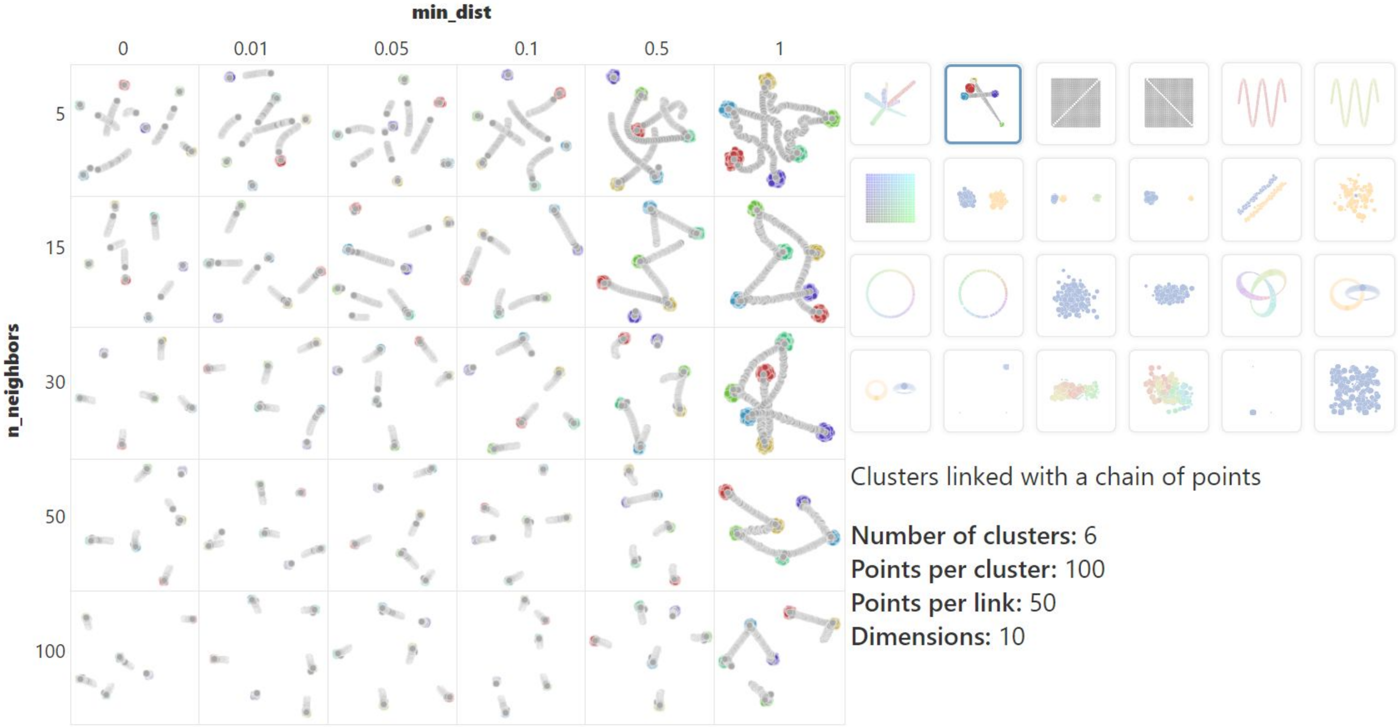
- Gradient descent $\frac{\partial C}{\partial \mathbf{y}_i}$ to update the embedding
  - However, they don't explicitly give cost gradients
  - Their optimization algorithm (Algorithm 5, p21) does not appear to incorporate the cost
  - Separately (p16), *"In practice, UMAP uses a force directed graph layout algorithm…"*
    - attractive force (*i* towards *j*) $$\mathbf{y}_i = \mathbf{y}_i + \alpha \cdot \frac{-2ab\|\mathbf{y}_i - \mathbf{y}_j\|_2^{2(b-1)}}{1 + a\left(\|\mathbf{y}_i - \mathbf{y}_j\|_2^2\right)^b} \bar{w}_{i,j}(\mathbf{y}_i - \mathbf{y}_j)$$
    - repulsive force (*i* away from *k*) $$\mathbf{y}_i = \mathbf{y}_i + \alpha \cdot \frac{b}{\left(\epsilon + \|\mathbf{y}_i - \mathbf{y}_k\|_2^2\right)\left(1 + a\left(\|\mathbf{y}_i - \mathbf{y}_k\|_2^2\right)^b\right)} (1 - \bar{w}_{i,k})(\mathbf{y}_i - \mathbf{y}_k)$$

UMAP with different hyperparameters



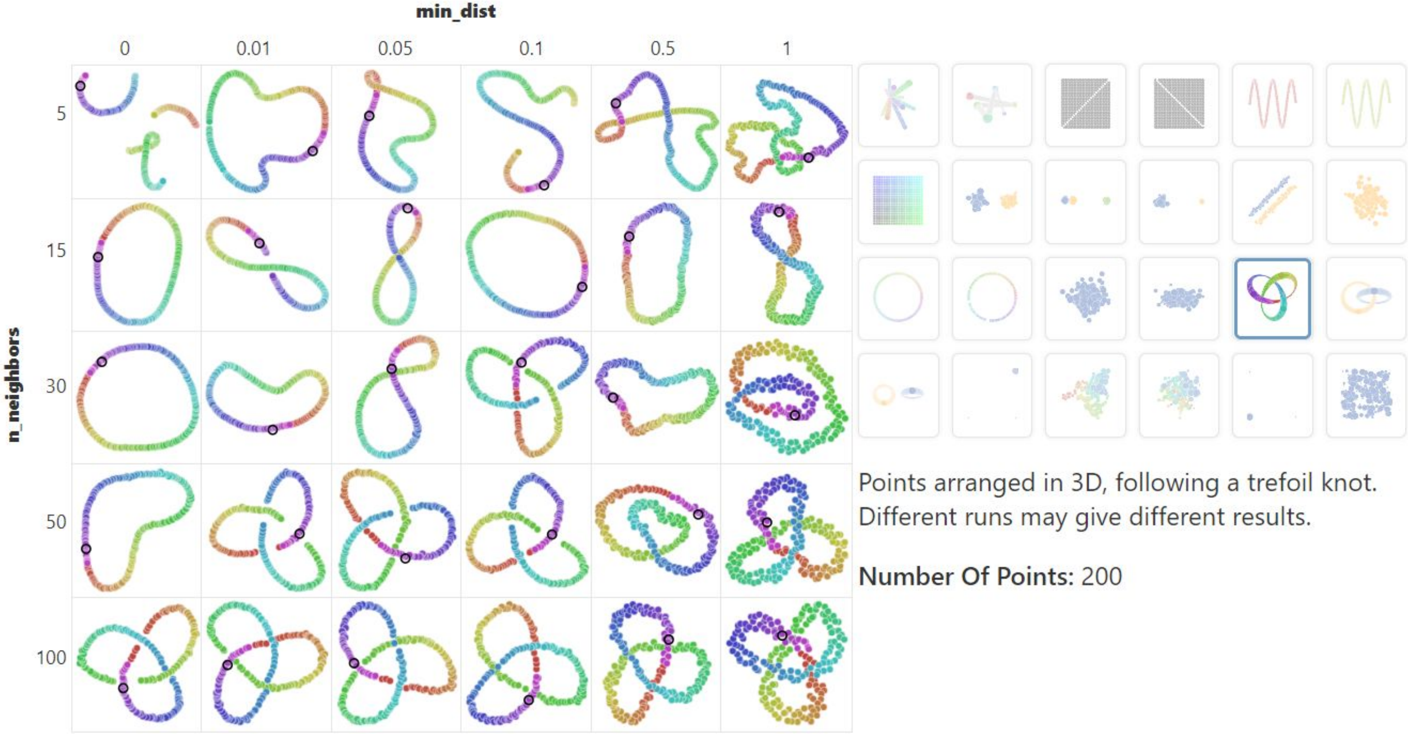
https://pair-code.github.io/understanding-umap/

UMAP with different hyperparameters



https://pair-code.github.io/understanding-umap/

UMAP with different hyperparameters

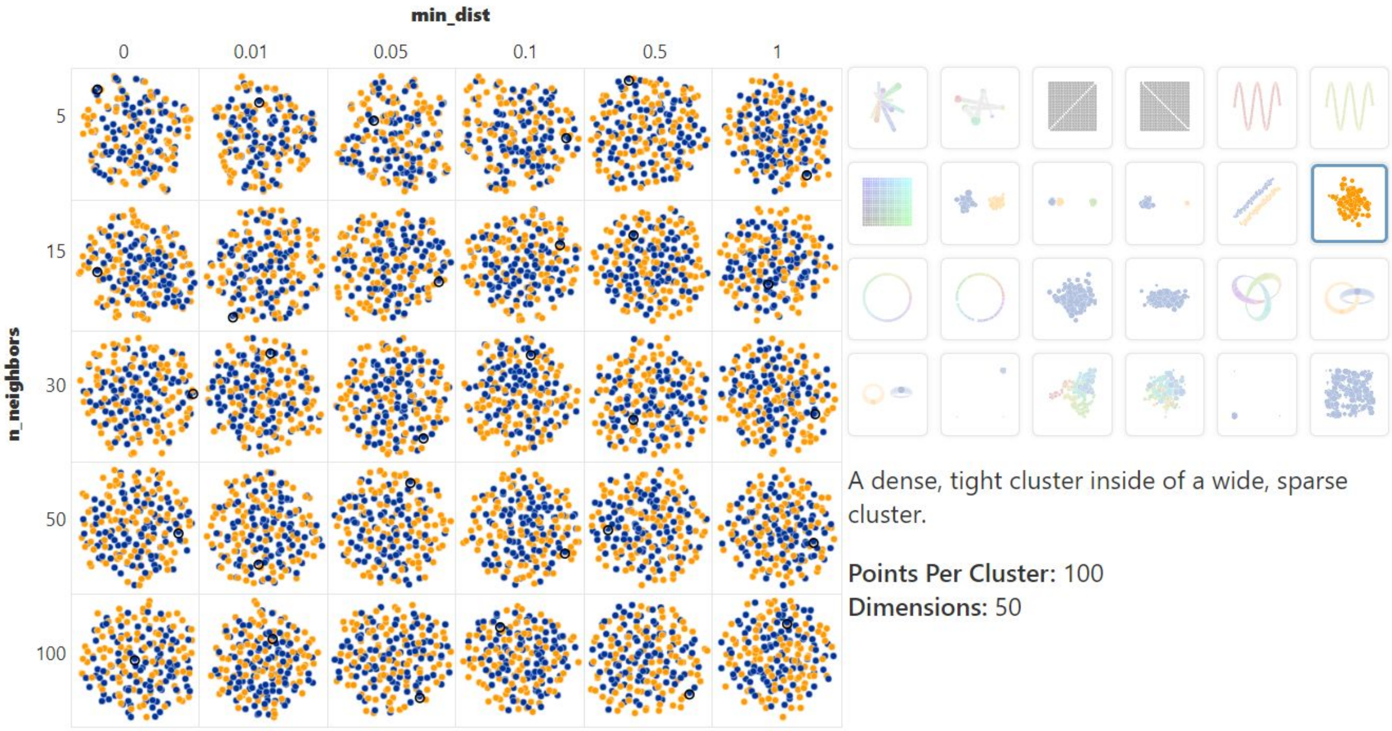


Points arranged in 3D, following a trefoil knot. Different runs may give different results.

Number Of Points: 200

https://pair-code.github.io/understanding-umap/

UMAP with different hyperparameters

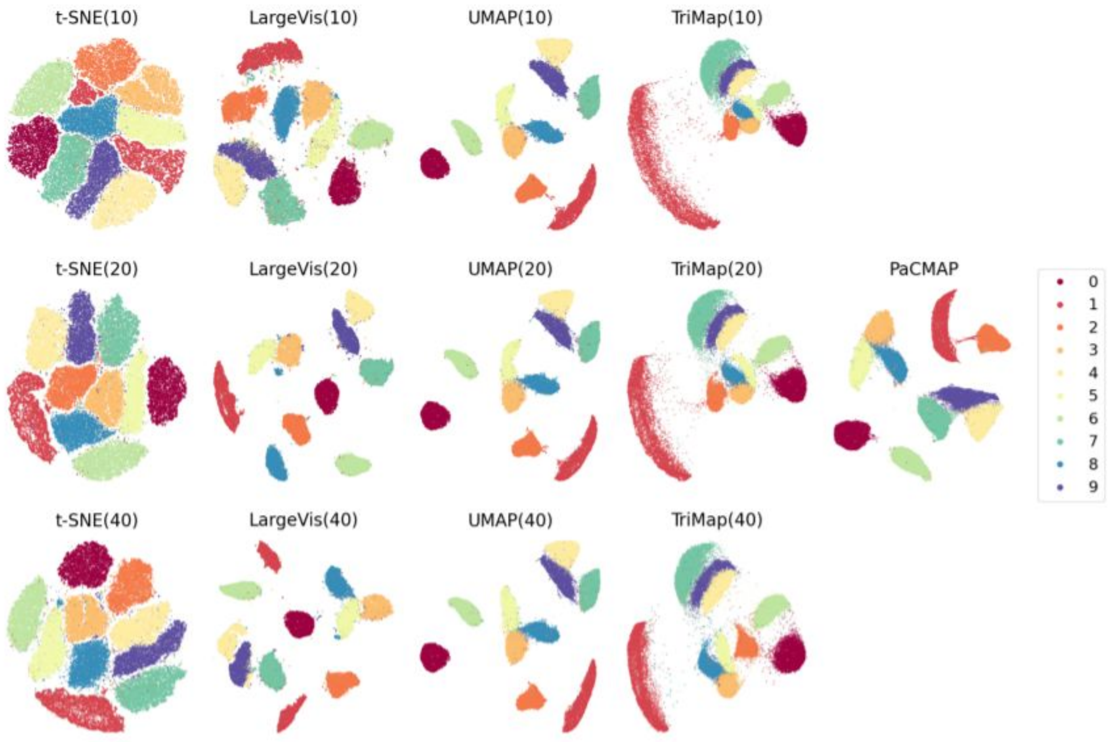


A dense, tight cluster inside of a wide, sparse cluster.

Points Per Cluster: 100
Dimensions: 50

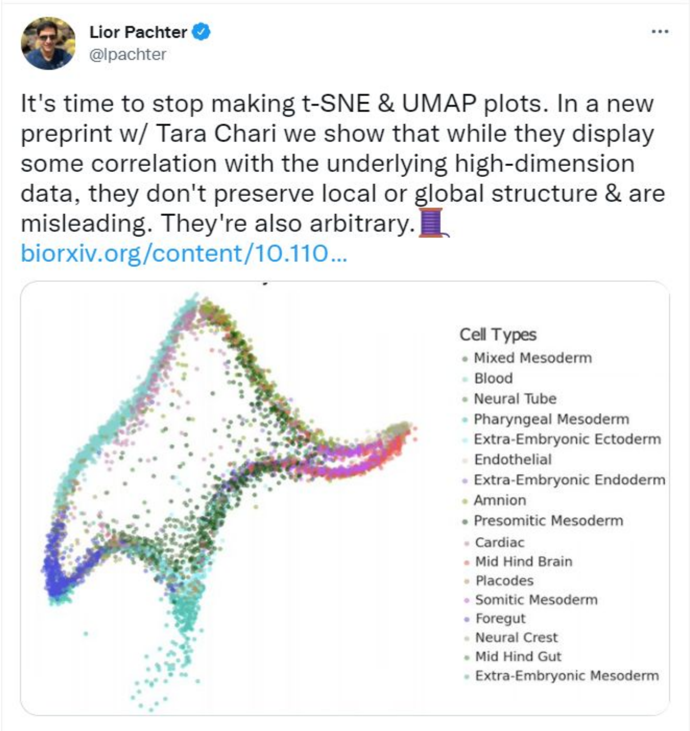
https://pair-code.github.io/understanding-umap/

Some comparison between UMAP, t-SNE, others



See also https://arxiv.org/pdf/2012.04456.pdf and https://pair-code.github.io/understanding-umap/

Criticisms: It's time to stop making t-SNE & UMAP plots.



Lior Pachter (Caltech) - 2021 Twitter rant

- https://twitter.com/lpachter/status/1431325969411821572?s=20
- https://threadreaderapp.com/thread/1431325969411821572.html

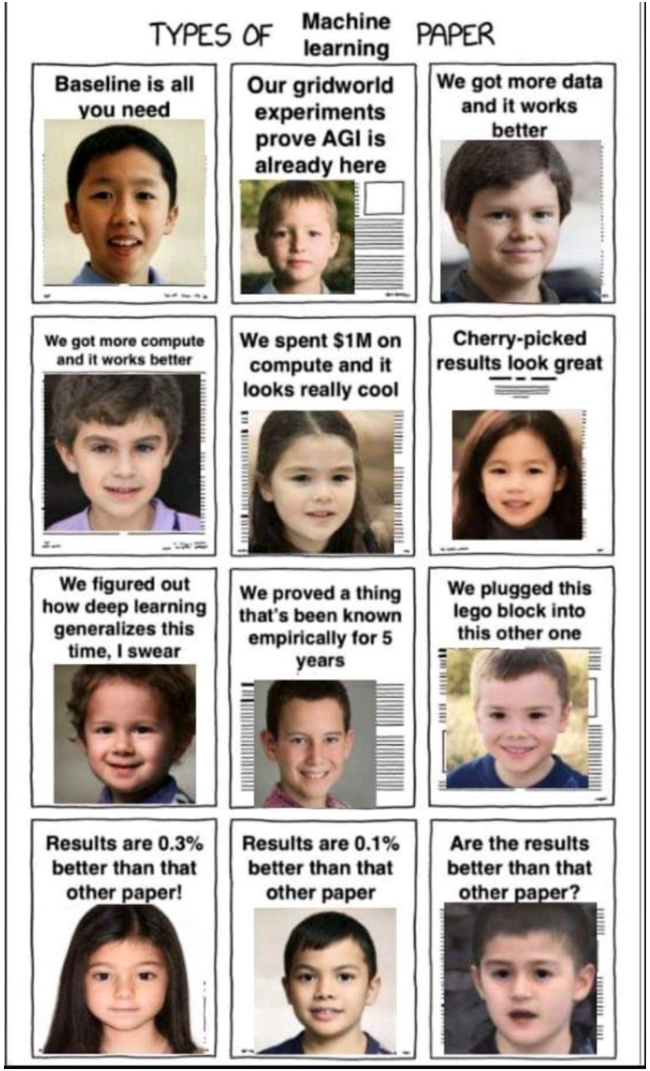
Pre-print (updated Sept. 27, 2021)

- https://www.biorxiv.org/content/10.1101/2021.08.25.457696v3
- Mostly upset about PCA pre-conditioning?

Cites heavily: https://www.nature.com/articles/s41467-019-13056-x.pdf and https://www.biorxiv.org/content/10.1101/689851v4.full.pdf

	k	t-SNE	UMAP	LargeVis	Eigenmaps	PCA
Shuttle	100	**0.994** (± 0.002)	0.993 (± 0.002)	0.992 (± 0.003)	0.962 (± 0.004)	0.833 (± 0.013)
	200	**0.992** (± 0.002)	0.990 (± 0.002)	0.987 (± 0.003)	0.957 (± 0.006)	0.821 (± 0.007)
	400	**0.990** (± 0.002)	0.988 (± 0.002)	0.976 (± 0.003)	0.949 (± 0.006)	0.815 (± 0.007)
	800	0.969 (± 0.005)	**0.988** (± 0.002)	0.957 (± 0.004)	0.942 (± 0.006)	0.804 (± 0.003)
	1600	0.927 (± 0.005)	**0.981** (± 0.002)	0.904 (± 0.007)	0.918 (± 0.006)	0.792 (± 0.003)
	3200	0.828 (± 0.004)	**0.957** (± 0.005)	0.850 (± 0.008)	0.895 (± 0.006)	0.786 (± 0.001)
MNIST	100	**0.967** (± 0.015)	**0.967** (± 0.014)	0.962 (± 0.015)	0.668 (± 0.016)	0.462 (± 0.023)
	200	0.966 (± 0.015)	**0.967** (± 0.014)	0.962 (± 0.015)	0.667 (± 0.016)	0.467 (± 0.023)
	400	0.964 (± 0.015)	**0.967** (± 0.014)	0.961 (± 0.015)	0.664 (± 0.016)	0.468 (± 0.024)
	800	0.963 (± 0.016)	**0.967** (± 0.014)	0.961 (± 0.015)	0.660 (± 0.017)	0.468 (± 0.023)
	1600	0.959 (± 0.016)	**0.966** (± 0.014)	0.947 (± 0.015)	0.651 (± 0.014)	0.467 (± 0.0233)
	3200	0.946 (± 0.017)	**0.964** (± 0.014)	0.920 (± 0.017)	0.639 (± 0.017)	0.459 (± 0.022)
Fashion-MNIST	100	**0.818** (± 0.012)	0.790 (± 0.013)	0.808 (± 0.014)	0.631 (± 0.010)	0.564 (± 0.018)
	200	**0.810** (± 0.013)	0.785 (± 0.014)	0.805 (± 0.013)	0.624 (± 0.013)	0.565 (± 0.016)
	400	**0.801** (± 0.013)	0.780 (± 0.013)	0.796 (± 0.013)	0.612 (± 0.011)	0.564 (± 0.017)
	800	**0.784** (± 0.011)	0.767 (± 0.014)	0.771 (± 0.014)	0.600 (± 0.012)	0.560 (± 0.017)
	1600	**0.754** (± 0.011)	0.747 (± 0.013)	0.742 (± 0.013)	0.580 (± 0.014)	0.550 (± 0.017)
	3200	0.727 (± 0.011)	**0.730** (± 0.011)	0.726 (± 0.012)	0.542 (± 0.014)	0.533 (± 0.017)

Table 2: kNN Classifier accuracy for varying values of k over the embedding spaces of Shuttle, MNIST and Fashion-MNIST datasets. Average accuracy scores are given over a 10-fold or 20-fold cross-validation for each of PCA, Laplacian Eigenmaps, LargeVis, t-SNE and UMAP.



References

- Original t-SNE: https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf
  - (JMLR, 2008) Visualizing Data using t-SNE
  - Note: (2002) Original SNE: https://www.cs.toronto.edu/~hinton/absps/sne.pdf
- Original UMAP: https://arxiv.org/pdf/1802.03426.pdf
  - (2018) UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction.
- Misc:
  - https://arxiv.org/abs/2012.04456 (published in JMLR, 2021)
    Understanding How Dimension Reduction Tools Work: An Empirical Approach to Deciphering t-SNE, UMAP, TriMAP, and PaCMAP for Data Visualization
  - https://arxiv.org/pdf/2007.08902.pdf (published in JMLR, 2021)
    A Unifying Perspective on Neighbor Embeddings along the Attraction-Repulsion Spectrum
  - https://youtu.be/CsUqmug7ZMc (Neighbour embeddings for scientific visualization, Dmitry Kobak, 2021)

Interactive visualizations

- Distill - How to Use t-SNE Effectively: https://distill.pub/2016/misread-tsne/
- Google PAIR - Understanding UMAP: https://pair-code.github.io/understanding-umap/
- Tensorflow - Various Datasets (PCA, t-SNE, UMAP): https://projector.tensorflow.org/
- UMAP on Fashion MNIST: https://observablehq.com/@stwind/exploring-fashion-mnist