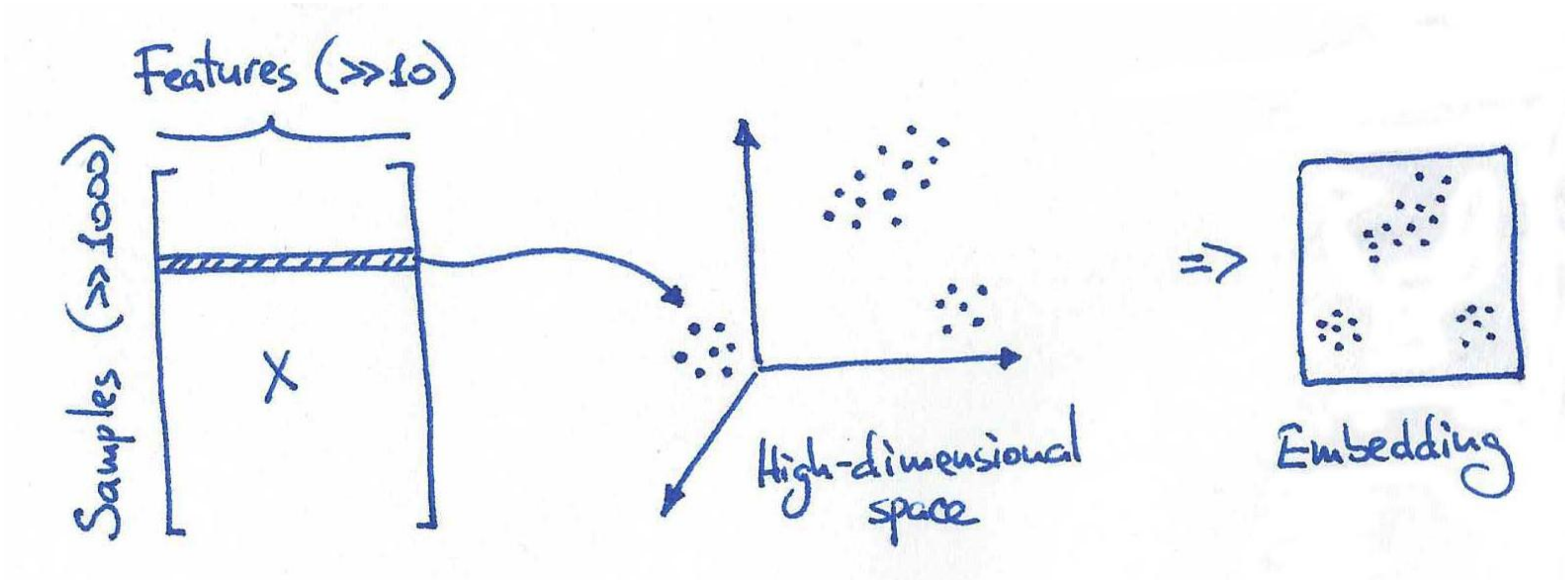
Data visualization with t-SNE



Dimensionality reduction

Dimensionality reduction algorithms can be:

- unsupervised / supervised (PCA / LDA)

Dimensionality reduction

Dimensionality reduction algorithms can be:

- unsupervised / supervised (PCA / LDA)
- linear / non-linear (PCA / kernel PCA)

Dimensionality reduction

Dimensionality reduction algorithms can be:

• unsupervised / supervised	(PCA / LDA)
• linear / non-linear	(PCA / kernel PCA)
• parametric / non-parametric	(PCA / MDS)

Dimensionality reduction

Dimensionality reduction algorithms can be:

• unsupervised / supervised	(PCA / LDA)
• linear / non-linear	(PCA / kernel PCA)
• parametric / non-parametric	(PCA / MDS)

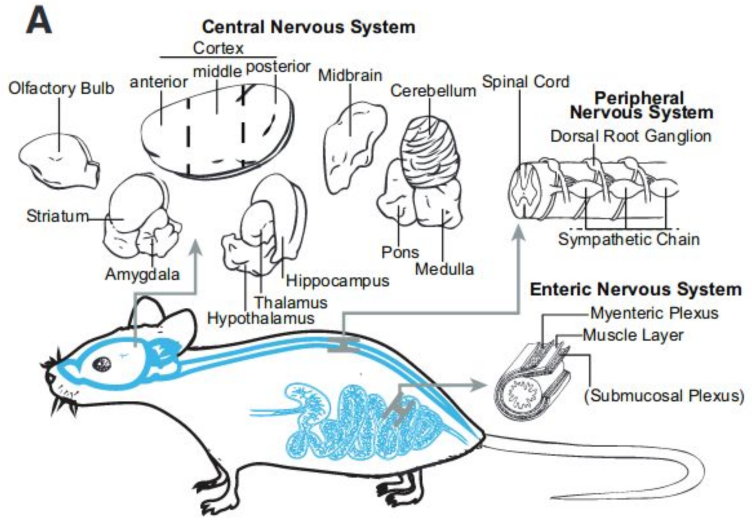
Today we are talking about unsupervised non-parametric methods (often called ‘non-linear dimensionality reduction’).

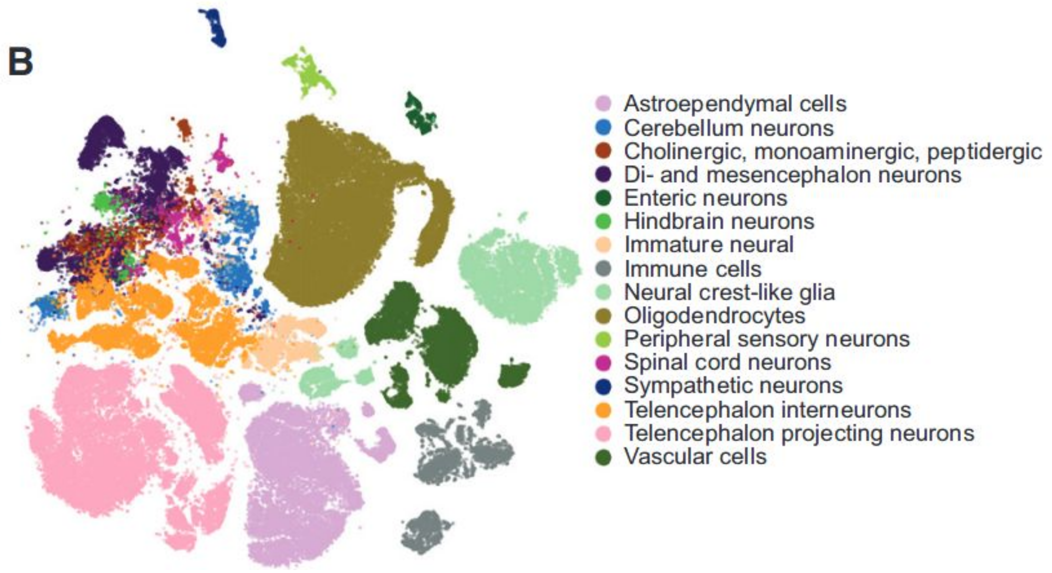
Examples: MDS, t-SNE, UMAP, etc.

Where are these algorithms used?

Where are these algorithms used?

Single-cell transcriptomics (single-cell RNA sequencing): samples are cells, features are genes.

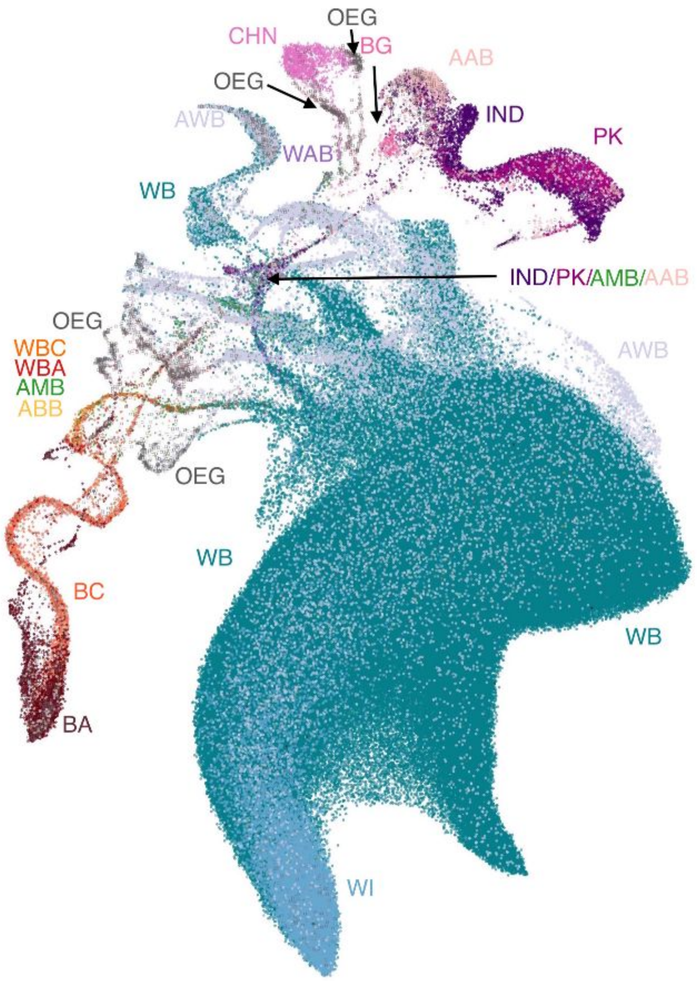




Zeisel et al. (2018)
$n \approx 500{,}000$

Where are these algorithms used?

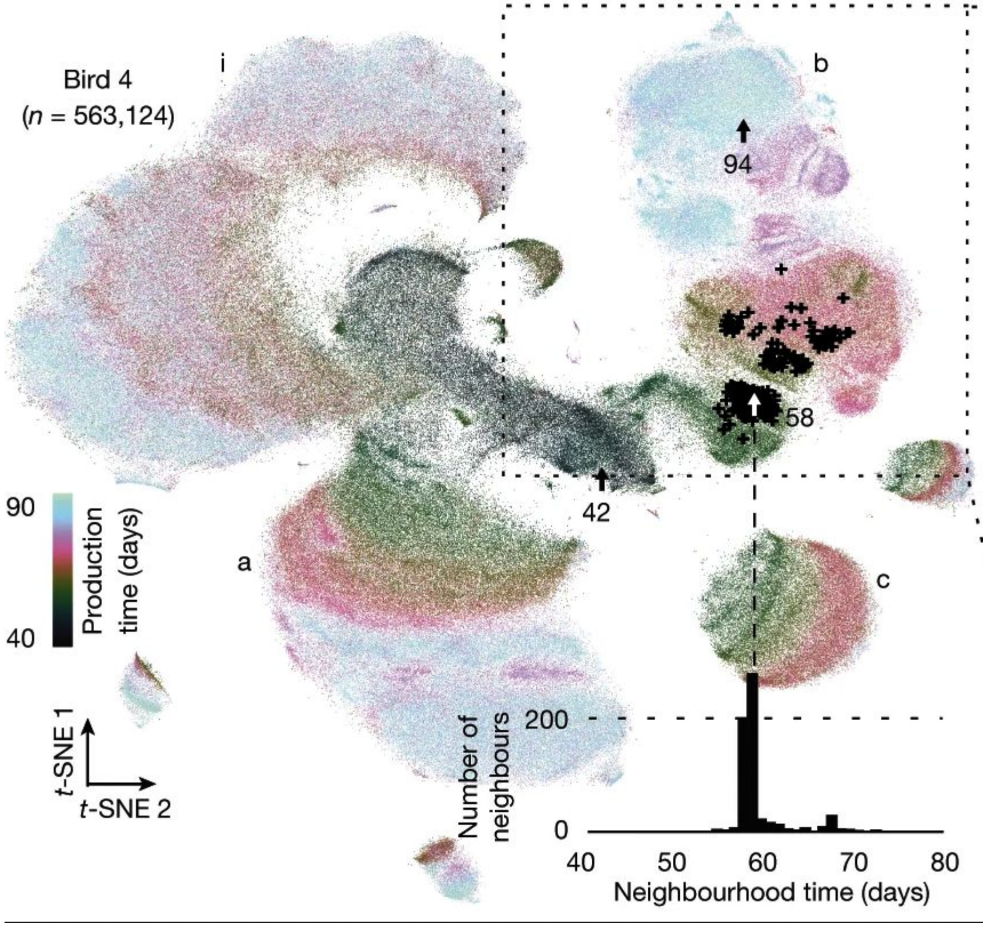
Population genomics: samples are people, features are single-nucleotide polymorphims.



Diaz-Papkovich et al. (2019)
$n \approx 500{,}000$

Where are these algorithms used?

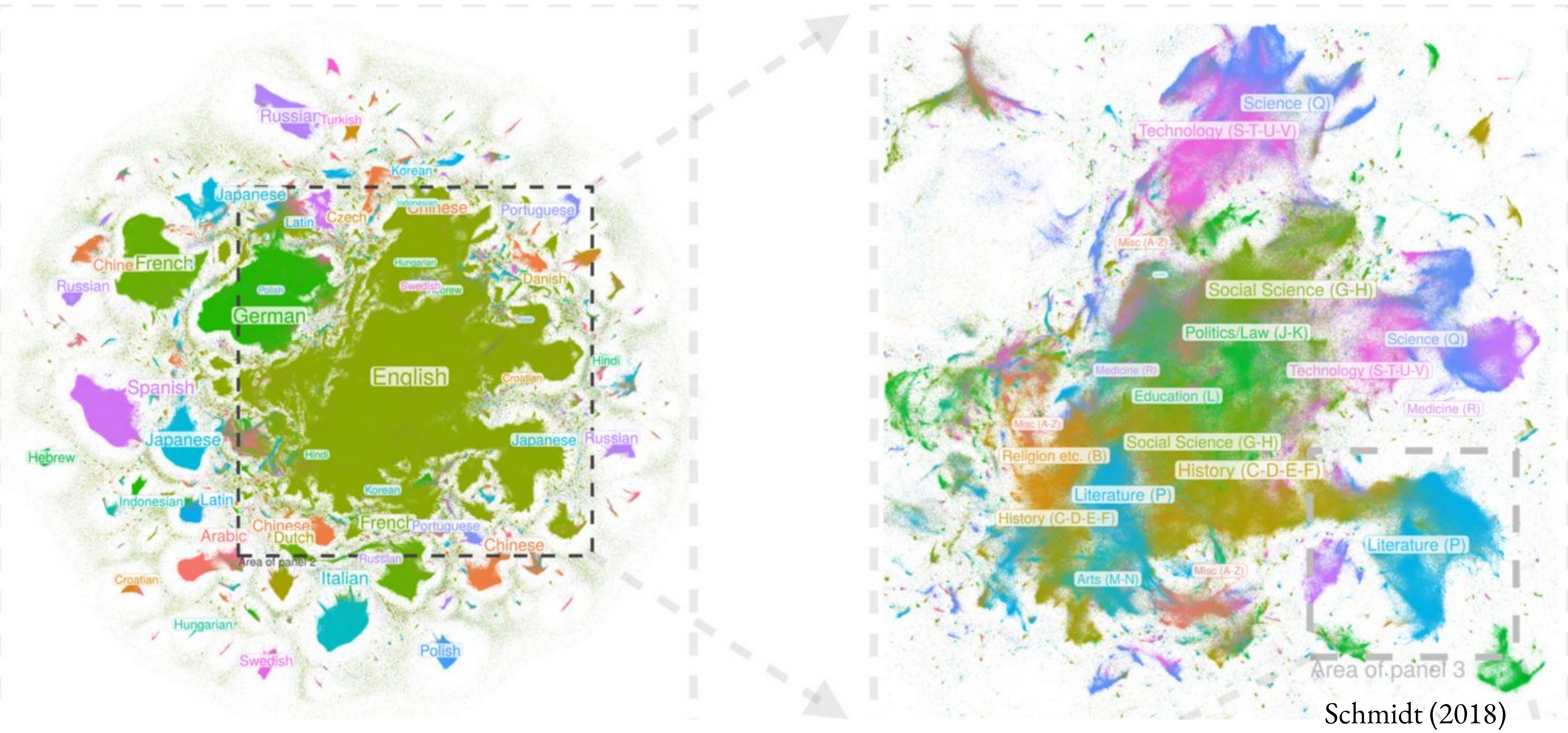
Behavioural physiology: samples are syllable renditions, features are spectrogram bins.



Kollmorgen et al. (2020)
$n \approx 600{,}000$

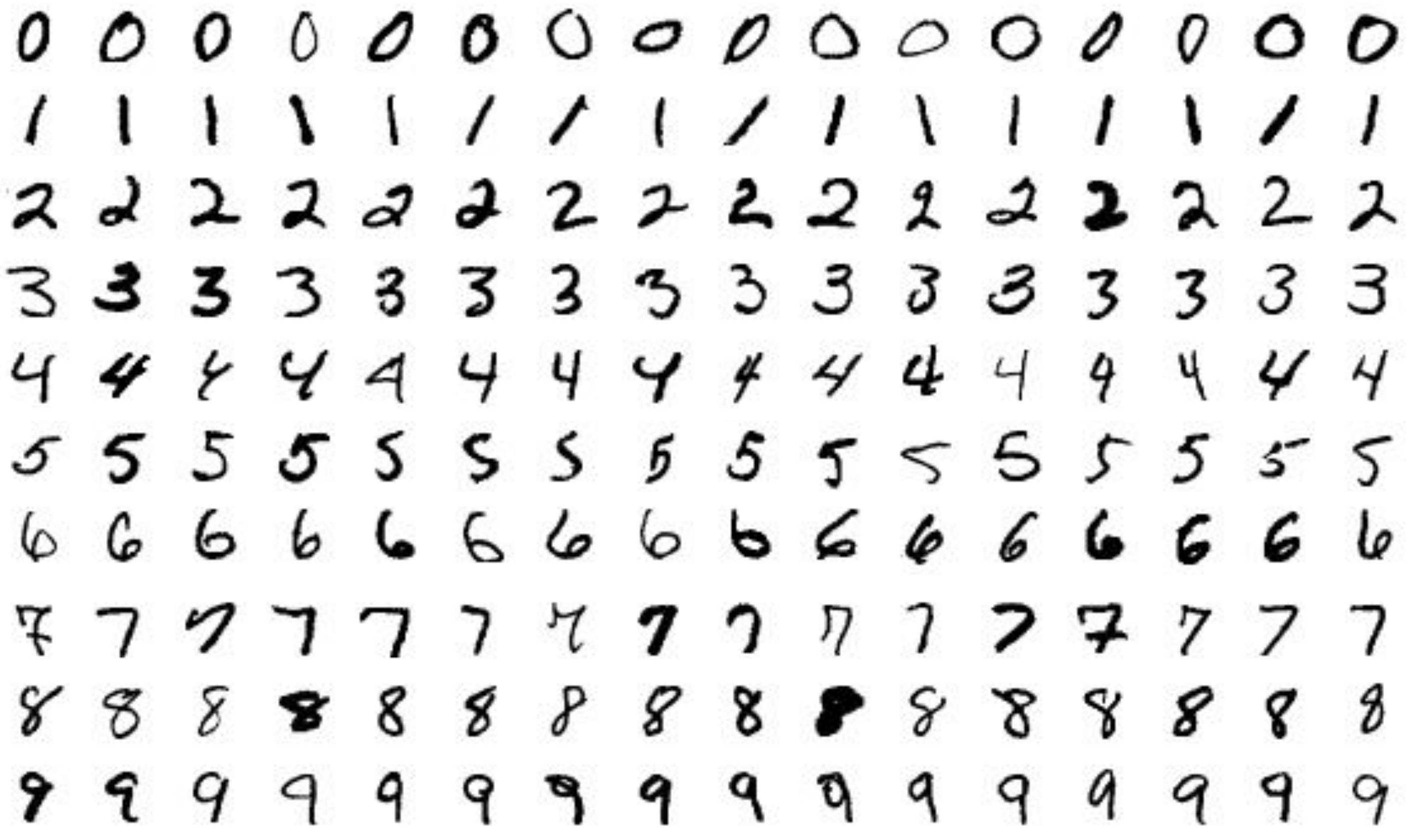
Where are these algorithms used?

Digital humanities: samples are books, features are words.



Schmidt (2018)
$n \approx 15{,}000{,}000$

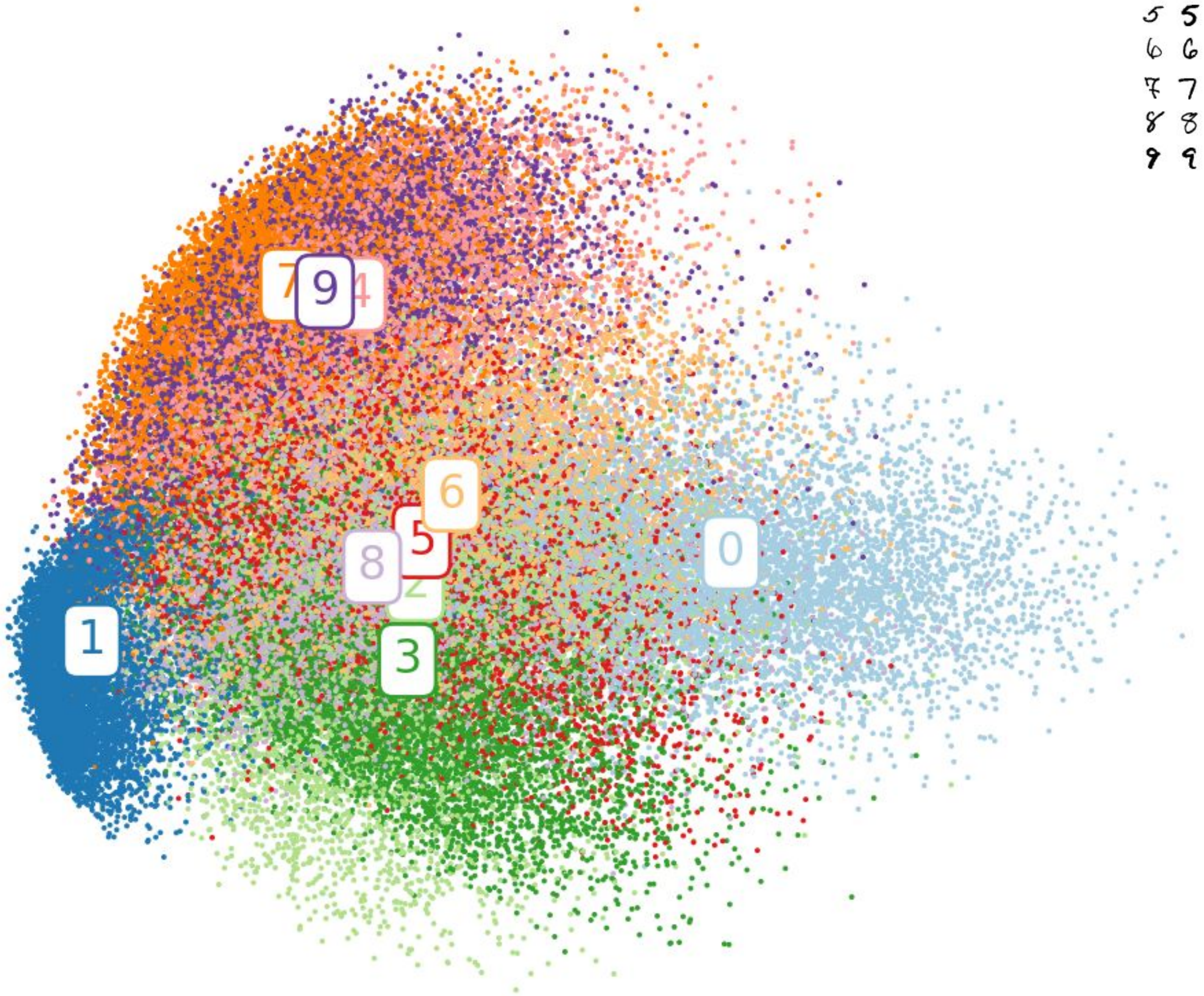
MNIST dataset

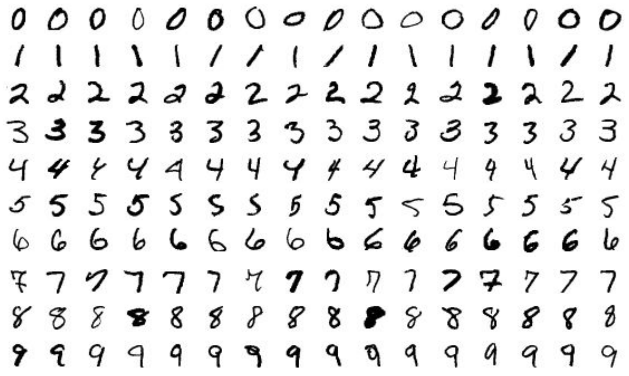


$n = 70{,}000$

28×28 images = 784 pixels

MNIST dataset: PCA





MNIST dataset: MDS

Multidimensional scaling: arrange points in 2D to approximate high-dimensional pairwise distances (1950s–1960s; Kruskal, Torgerson, etc.).

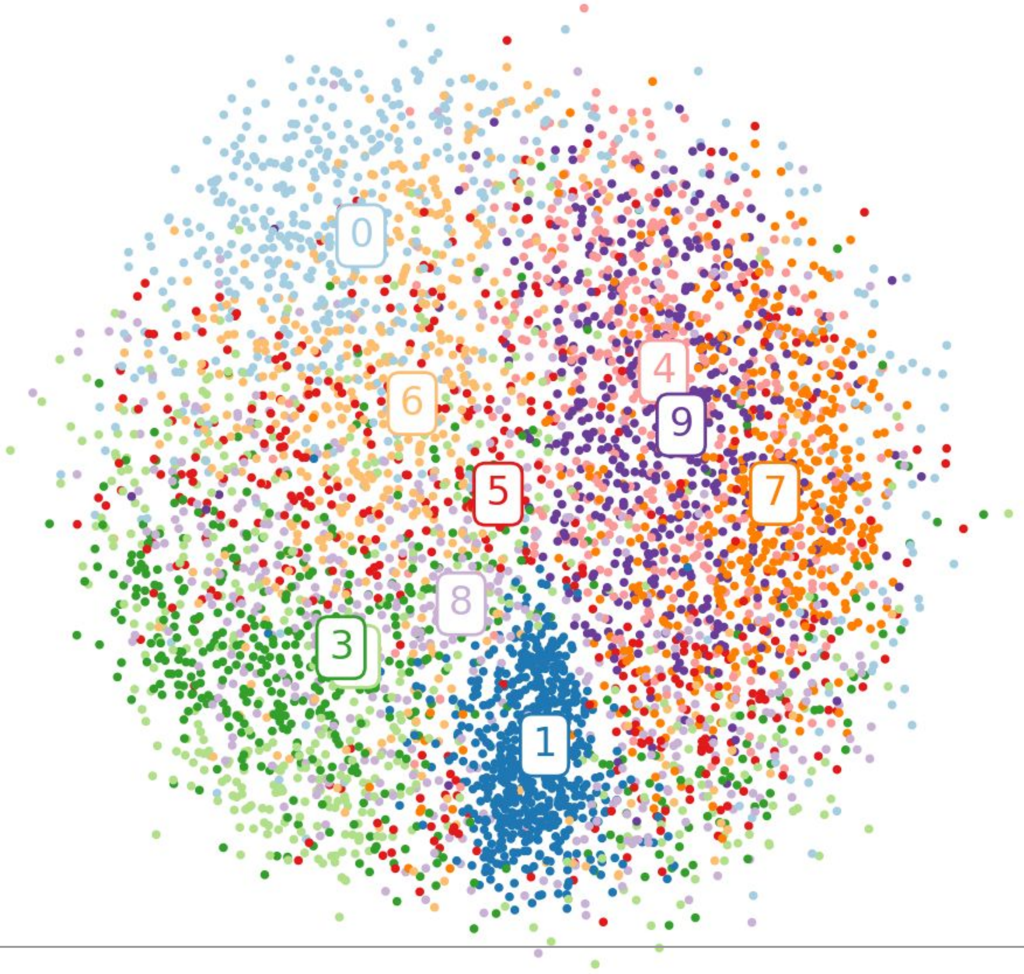
MNIST dataset: MDS

Multidimensional scaling: arrange points in 2D to approximate high-dimensional pairwise distances (1950s–1960s; Kruskal, Torgerson, etc.).

$$\mathcal{L} = \sum_{i<j} (d_{ij} - \|\mathbf{y}_i - \mathbf{y}_j\|)^2$$

MNIST dataset: MDS

Multidimensional scaling: arrange points in 2D to approximate high-dimensional pairwise distances (1950s–1960s; Kruskal, Torgerson, etc.).

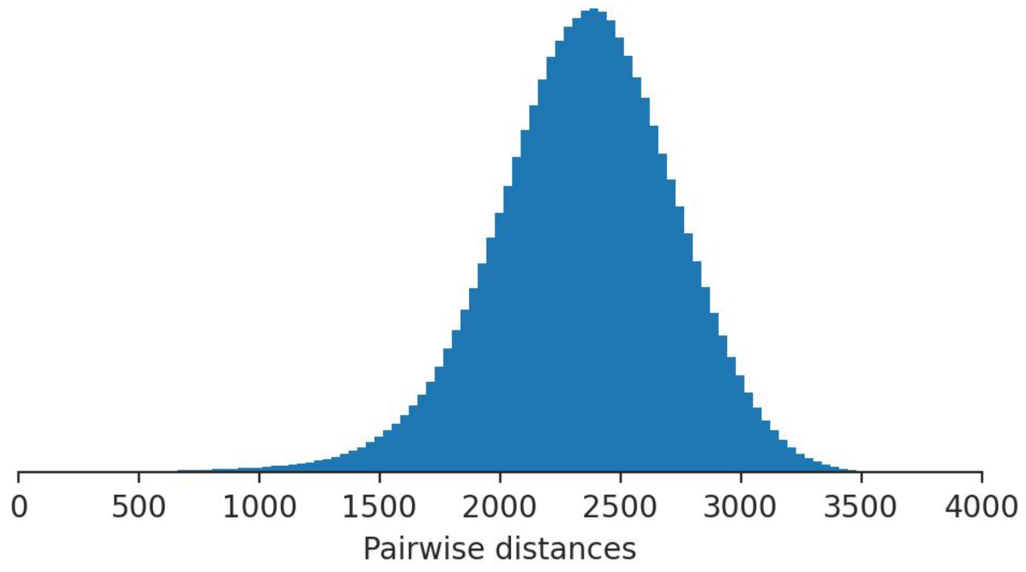


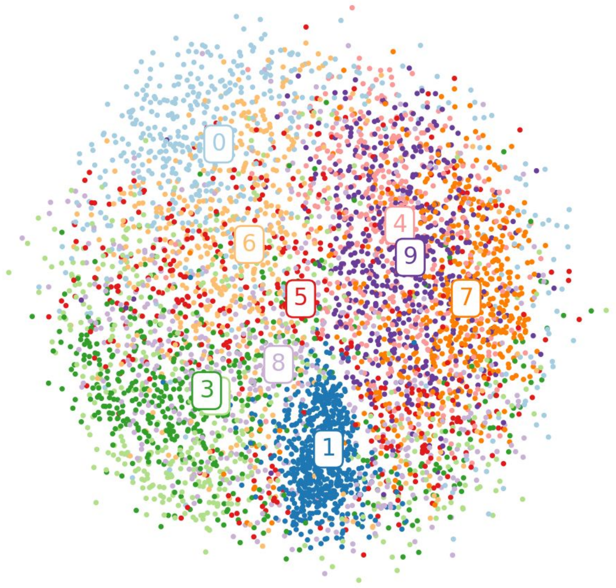
$$\mathcal{L} = \sum_{i<j} (d_{ij} - \|\mathbf{y}_i - \mathbf{y}_j\|)^2$$

Here n = 5,000.

Why does MDS fail?

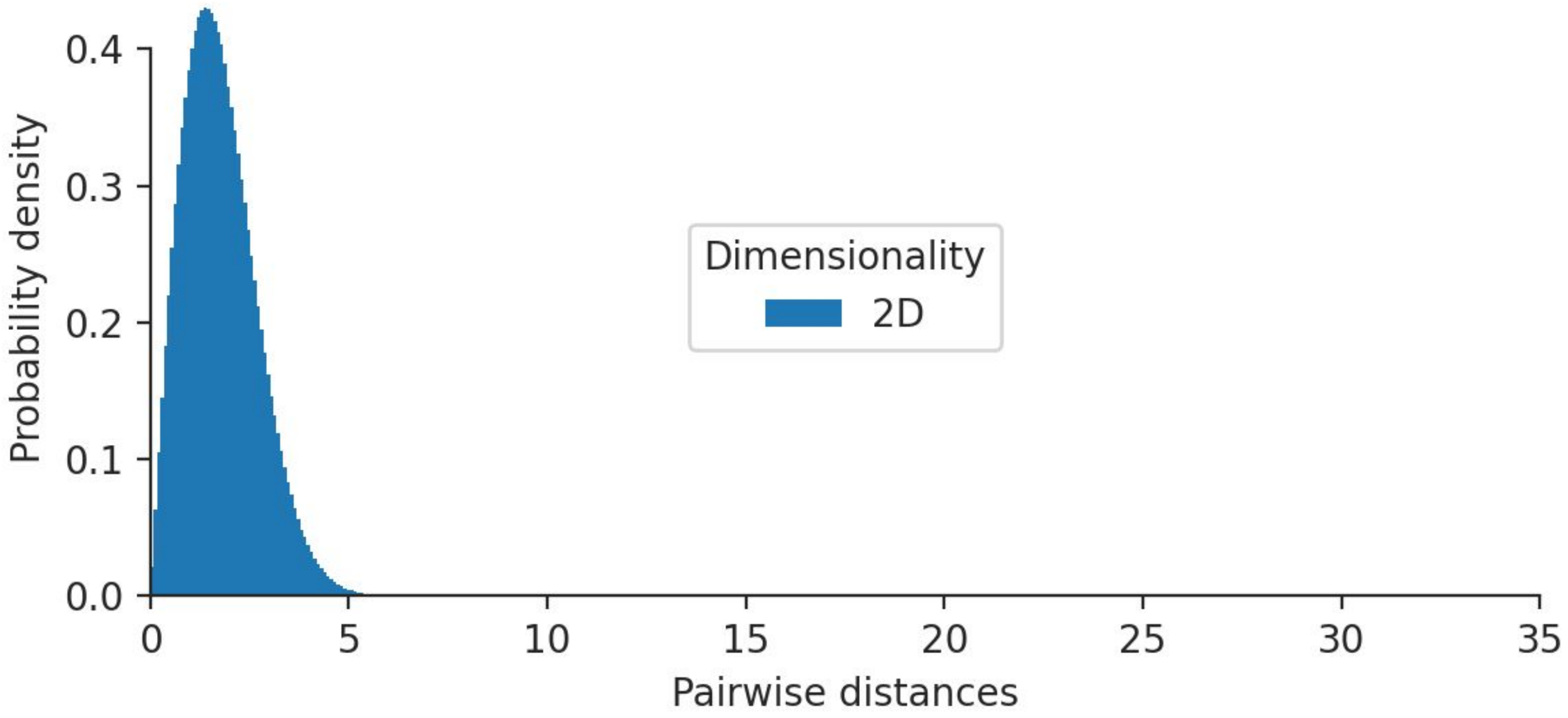
Preserving high-dimensional distances is usually a bad idea because it is not possible to preserve them *(curse of dimensionality)*.





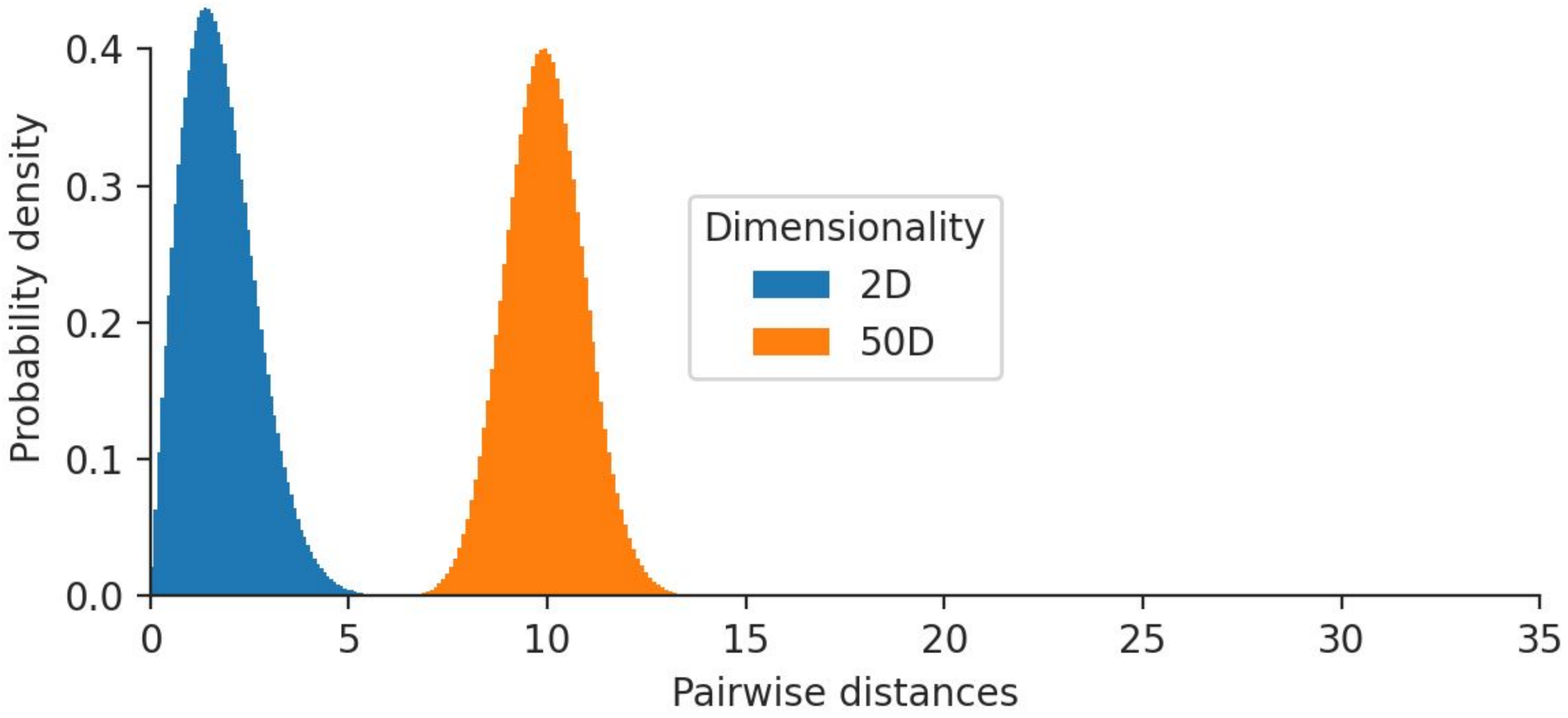
Why does MDS fail?

Pairwise distances between points in a standard Gaussian:



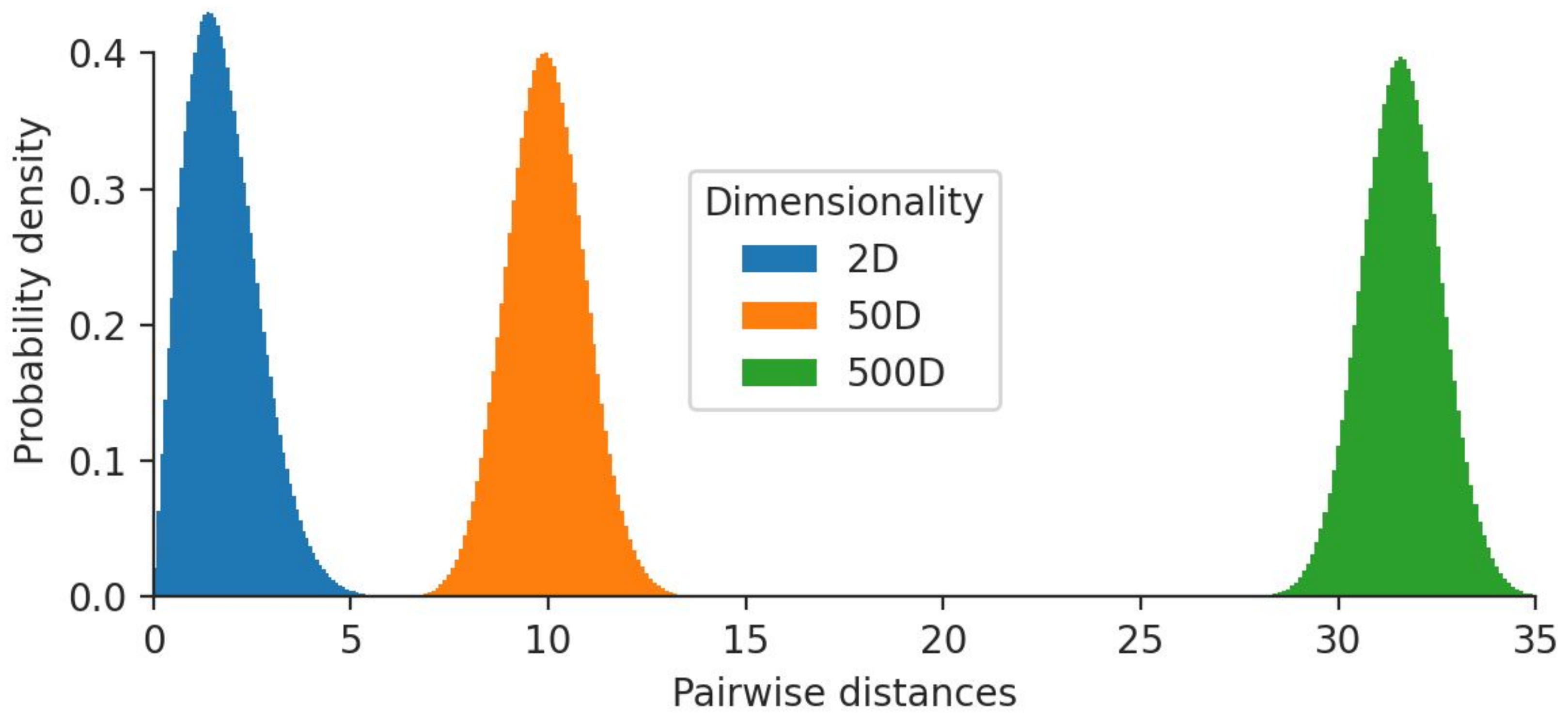
Why does MDS fail?

Pairwise distances between points in a standard Gaussian:



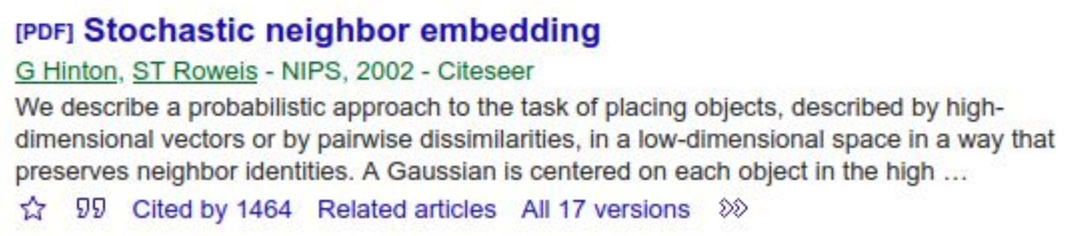
Why does MDS fail?

Pairwise distances between points in a standard Gaussian:



Neighbour embeddings

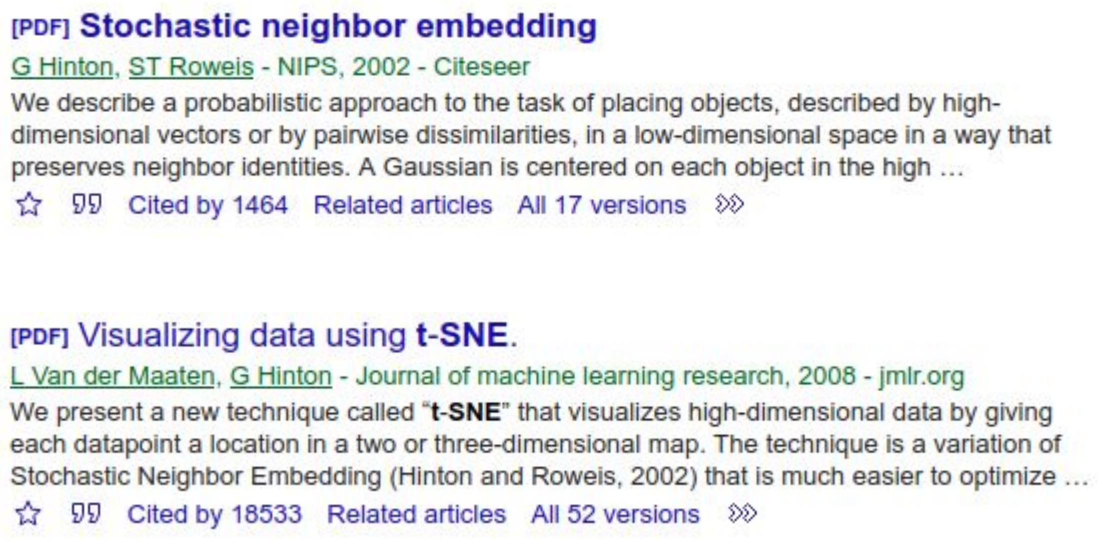
Idea: preserve *nearest neighbours* instead of preserving *distances*.



[PDF] Stochastic neighbor embedding
G Hinton, ST Roweis - NIPS, 2002 - Citeseer
We describe a probabilistic approach to the task of placing objects, described by high-dimensional vectors or by pairwise dissimilarities, in a low-dimensional space in a way that preserves neighbor identities. A Gaussian is centered on each object in the high …
Cited by 1464 Related articles All 17 versions

Neighbour embeddings

Idea: preserve *nearest neighbours* instead of preserving *distances*.

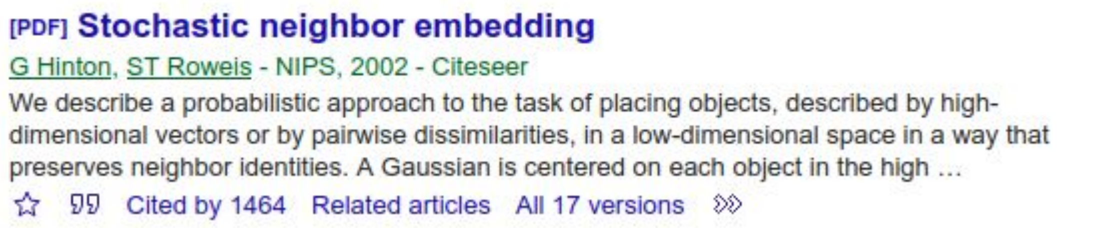


[PDF] **Stochastic neighbor embedding**
G Hinton, ST Roweis - NIPS, 2002 - Citeseer
We describe a probabilistic approach to the task of placing objects, described by high-dimensional vectors or by pairwise dissimilarities, in a low-dimensional space in a way that preserves neighbor identities. A Gaussian is centered on each object in the high …
☆ 99 Cited by 1464 Related articles All 17 versions »

[PDF] Visualizing data using **t-SNE**.
L Van der Maaten, G Hinton - Journal of machine learning research, 2008 - jmlr.org
We present a new technique called "**t-SNE**" that visualizes high-dimensional data by giving each datapoint a location in a two or three-dimensional map. The technique is a variation of Stochastic Neighbor Embedding (Hinton and Roweis, 2002) that is much easier to optimize …
☆ 99 Cited by 18533 Related articles All 52 versions »

Neighbour embeddings

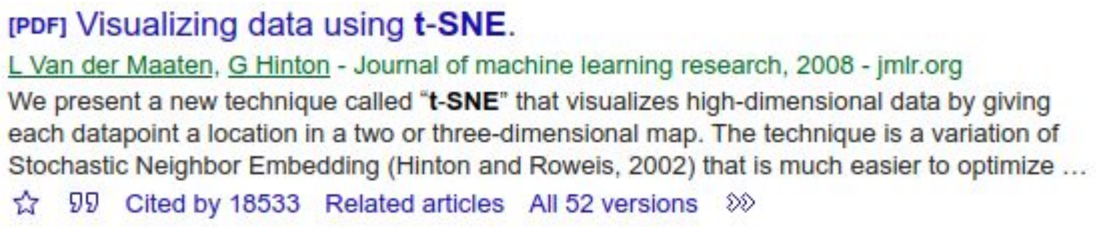
Idea: preserve *nearest neighbours* instead of preserving *distances*.



[PDF] **Stochastic neighbor embedding**

G Hinton, ST Roweis - NIPS, 2002 - Citeseer

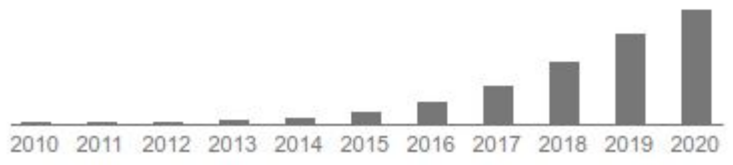
We describe a probabilistic approach to the task of placing objects, described by high-dimensional vectors or by pairwise dissimilarities, in a low-dimensional space in a way that preserves neighbor identities. A Gaussian is centered on each object in the high …



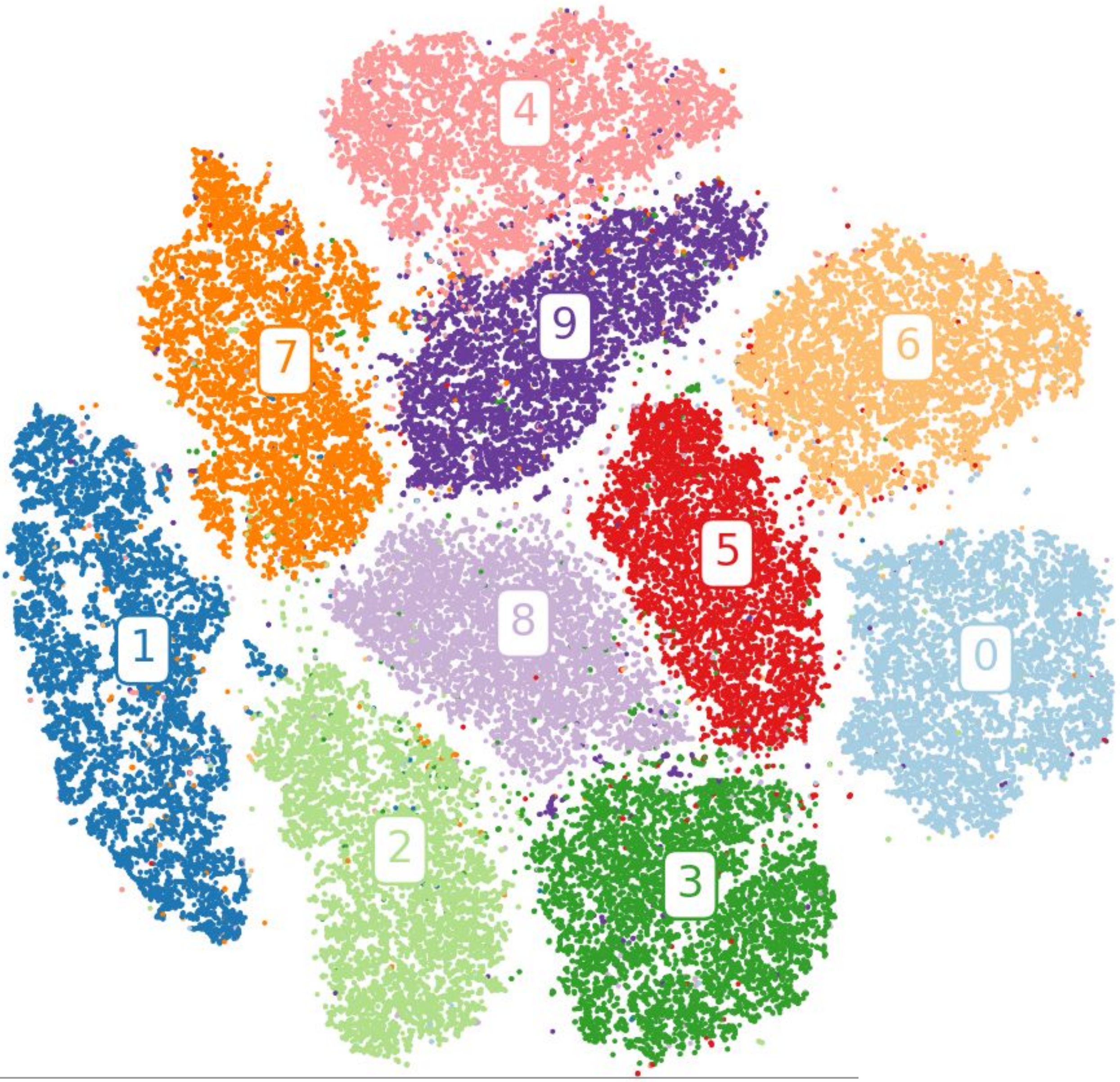
[PDF] Visualizing data using **t-SNE**.

L Van der Maaten, G Hinton - Journal of machine learning research, 2008 - jmlr.org

We present a new technique called "**t-SNE**" that visualizes high-dimensional data by giving each datapoint a location in a two or three-dimensional map. The technique is a variation of Stochastic Neighbor Embedding (Hinton and Roweis, 2002) that is much easier to optimize …

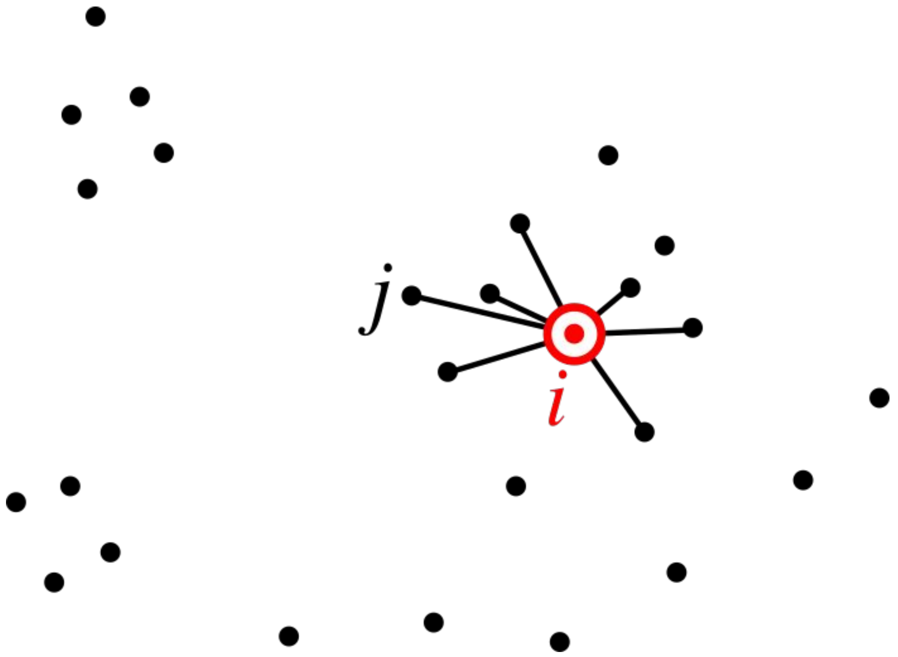


MNIST dataset: t-SNE



Stochastic neighbour embedding

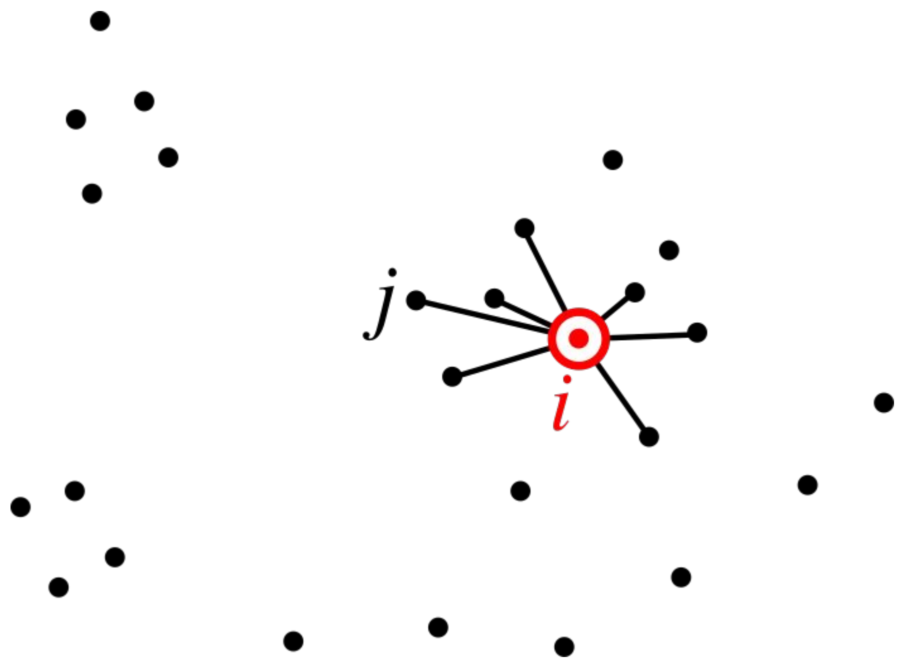
Loss function — Kullback-Leibler divergence between pairwise *similarities (affinities)* in the high-dimensional and in the low-dimensional spaces. Similarities are defined such that they sum to 1.



$$\mathcal{L} = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

Stochastic neighbour embedding

Loss function — Kullback-Leibler divergence between pairwise *similarities (affinities)* in the high-dimensional and in the low-dimensional spaces. Similarities are defined such that they sum to 1.



$$\mathcal{L} = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

High price for putting close neighbours far away.

Stochastic neighbour embedding

High-dimensional similarities:

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2/2\sigma_i^2)}$$

Stochastic neighbour embedding

High-dimensional similarities:

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2/2\sigma_i^2)}$$

Kernel width is adaptively chosen to achieve the desired *perplexity* (default 30):

$$\mathcal{P} = 2^{\mathcal{H}}, \text{ where } \mathcal{H} = -\sum_{j\neq i} p_{j|i}\log_2 p_{j|i}$$

Stochastic neighbour embedding

High-dimensional similarities:

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2 / 2\sigma_i^2)}$$

Kernel width is adaptively chosen to achieve the desired *perplexity* (default 30):

$$\mathcal{P} = 2^{\mathcal{H}}, \text{ where } \mathcal{H} = -\sum_{j \neq i} p_{j|i} \log_2 p_{j|i}$$

Then symmetrize and normalize to sum to one:

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2n}$$

Stochastic neighbour embedding

High-dimensional similarities:

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2/2\sigma_i^2)}$$

Kernel width is adaptively chosen to achieve the desired *perplexity* (default 30):

$$\mathcal{P} = 2^{\mathcal{H}}, \text{ where } \mathcal{H} = -\sum_{j\neq i} p_{j|i}\log_2 p_{j|i}$$

Then symmetrize and normalize to sum to one:

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2n}$$

This defines all similarities as non-zero. But most will be ≈ 0, and can be set to 0 without affecting the result. Moreover, one can use uniform similarities:

$$p_{j|i} = 1/k \text{ for } k \text{ nearest neighbours}$$

Stochastic neighbour embedding

Low-dimensional similarities:

$$q_{ij} = \frac{w_{ij}}{Z}, \quad w_{ij} = k(\|\mathbf{y}_i - \mathbf{y}_j\|), \quad Z = \sum_{k \neq l} w_{kl}$$

Stochastic neighbour embedding

Low-dimensional similarities:

$$q_{ij} = \frac{w_{ij}}{Z}, \quad w_{ij} = k(\|\mathbf{y}_i - \mathbf{y}_j\|), \quad Z = \sum_{k \neq l} w_{kl}$$

Similarity kernel in SNE:

$$k(d) = \exp(-d^2)$$

Stochastic neighbour embedding

Low-dimensional similarities:

$$q_{ij} = \frac{w_{ij}}{Z}, \quad w_{ij} = k(\|\mathbf{y}_i - \mathbf{y}_j\|), \quad Z = \sum_{k \neq l} w_{kl}$$

Similarity kernel in SNE:

$$k(d) = \exp(-d^2)$$

Similarity kernel in t-SNE:

$$k(d) = 1/(1 + d^2)$$

Gradient descent

The loss is optimized via gradient descent (e.g. starting from a random configuration of points).

$$\begin{aligned}\mathcal{L} &= -\sum_{i,j} p_{ij} \log q_{ij} = -\sum_{i,j} p_{ij} \log \frac{w_{ij}}{Z} \\ &= -\sum_{i,j} p_{ij} \log w_{ij} + \log \sum_{i,j} w_{ij},\end{aligned}$$

Gradient descent

The loss is optimized via gradient descent (e.g. starting from a random configuration of points).

$$\mathcal{L} = -\sum_{i,j} p_{ij} \log q_{ij} = -\sum_{i,j} p_{ij} \log \frac{w_{ij}}{Z}$$
$$= -\sum_{i,j} p_{ij} \log w_{ij} + \log \sum_{i,j} w_{ij},$$

This works as a many-body simulation: close neighbours attract each other while all points repulse each other.

Gradient descent

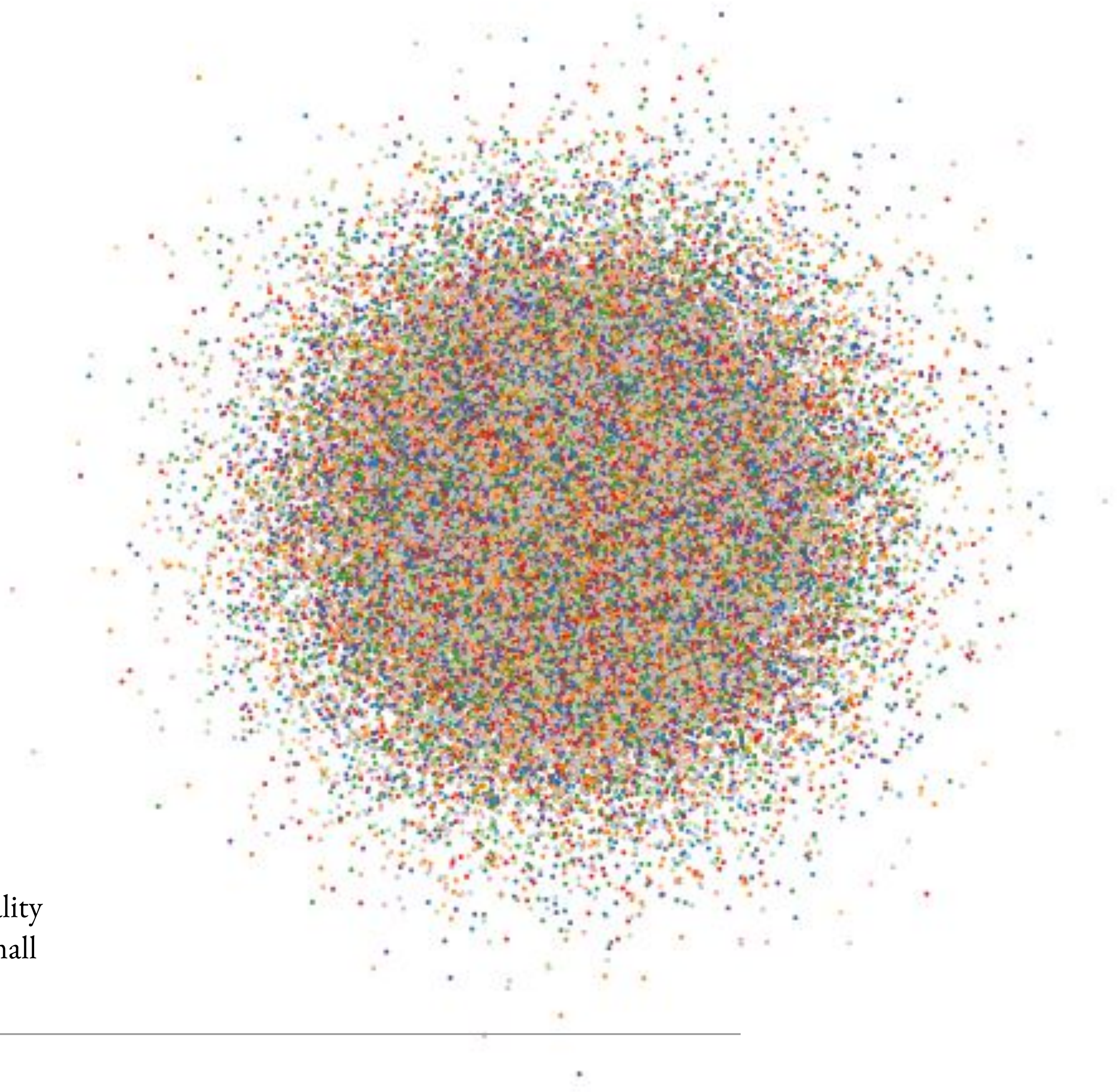
The loss is optimized via gradient descent (e.g. starting from a random configuration of points).

$$\begin{aligned}\mathcal{L} &= -\sum_{i,j} p_{ij} \log q_{ij} = -\sum_{i,j} p_{ij} \log \frac{w_{ij}}{Z} \\ &= -\sum_{i,j} p_{ij} \log w_{ij} + \log \sum_{i,j} w_{ij},\end{aligned}$$

This works as a many-body simulation: close neighbours attract each other while all points repulse each other.

$$\begin{aligned}\frac{\partial \mathcal{L}_{\text{t-SNE}}}{\partial \mathbf{y}_i} &= -2 \sum_j p_{ij} \frac{1}{w_{ij}} \frac{\partial w_{ij}}{\partial \mathbf{y}_i} + 2 \frac{1}{Z} \sum_j \frac{\partial w_{ij}}{\partial \mathbf{y}_i} \\ &\sim \sum_j p_{ij} w_{ij} (\mathbf{y}_i - \mathbf{y}_j) - \frac{1}{Z} \sum_j w_{ij}^2 (\mathbf{y}_i - \mathbf{y}_j)\end{aligned}$$

Gradient descent: MNIST



750 iterations.
Every 5th iteration shown.

Made with openTSNE.

Each frame is scaled (in reality embedding is initialized small and slowly grows in size).

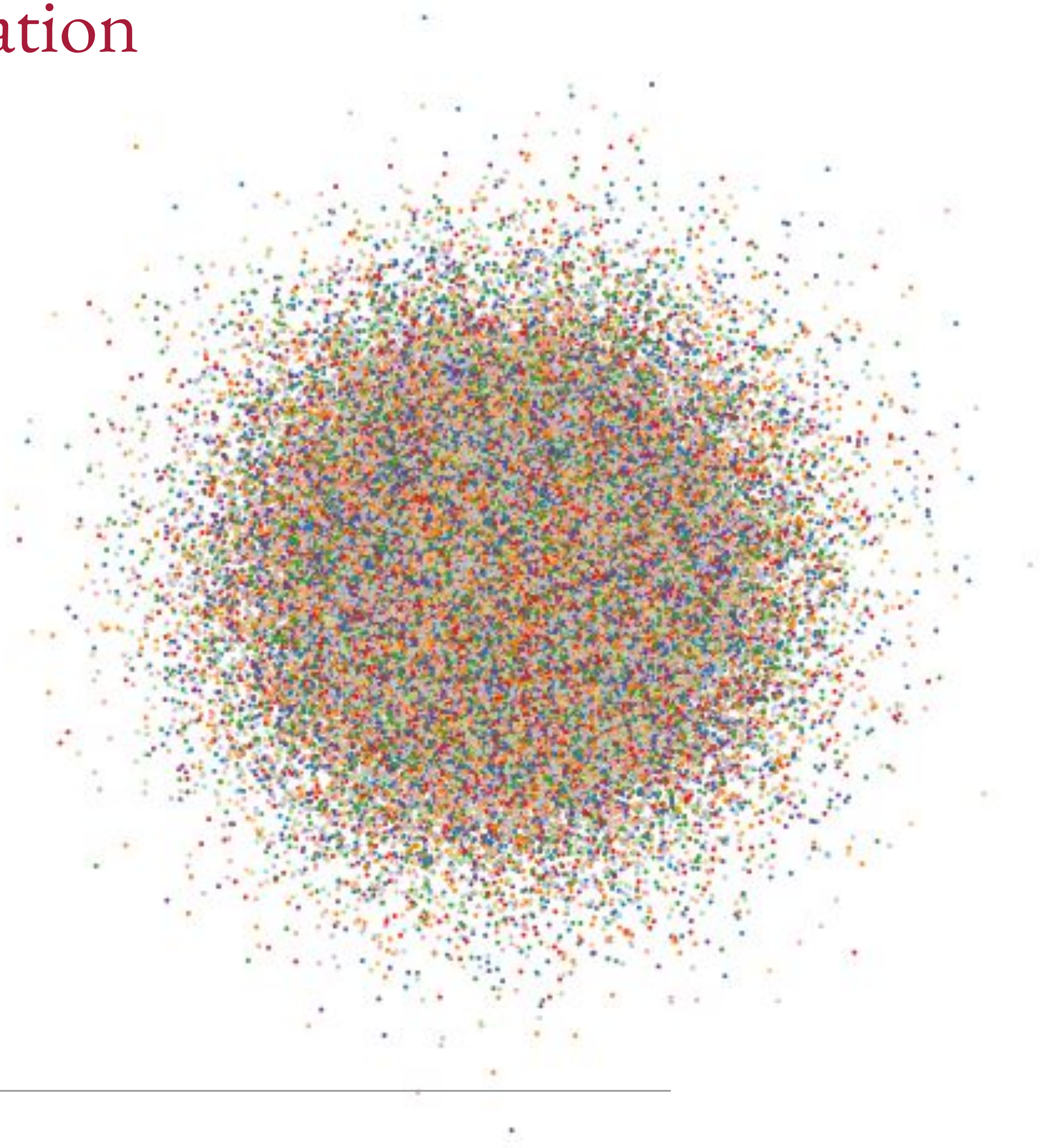
Early exaggeration

Multiply all attractive forces by 12 for 250 iterations.

Note that the learning rate should be high enough for this to work:

$$\eta = n/12$$

(Belkina et al., 2019)



Fast approximate implementations

Vanilla t-SNE has O(n^2) attractive and repulsive forces. To speed it up, we need to deal with both.

Fast approximate implementations

Vanilla t-SNE has $O(n^2)$ attractive and repulsive forces. To speed it up, we need to deal with both.

Attractive forces:

- Only use a small number of non-zero affinities, i.e. a sparse k-nearest-neighbour (kNN) graph. This reduces the number of forces. (Standard heuristic: $k = 3P$. For $P = 30$, this gives $k = 90$.)

Fast approximate implementations

Vanilla t-SNE has O(n^2) attractive and repulsive forces. To speed it up, we need to deal with both.

Attractive forces:

- Only use a small number of non-zero affinities, i.e. a sparse k-nearest-neighbour (kNN) graph. This reduces the number of forces. (Standard heuristic: $k = 3P$. For $P = 30$, this gives $k = 90$.)
- Use approximate kNN graphs. This speeds up graph construction.

Fast approximate implementations

Vanilla t-SNE has $O(n^2)$ attractive and repulsive forces. To speed it up, we need to deal with both.

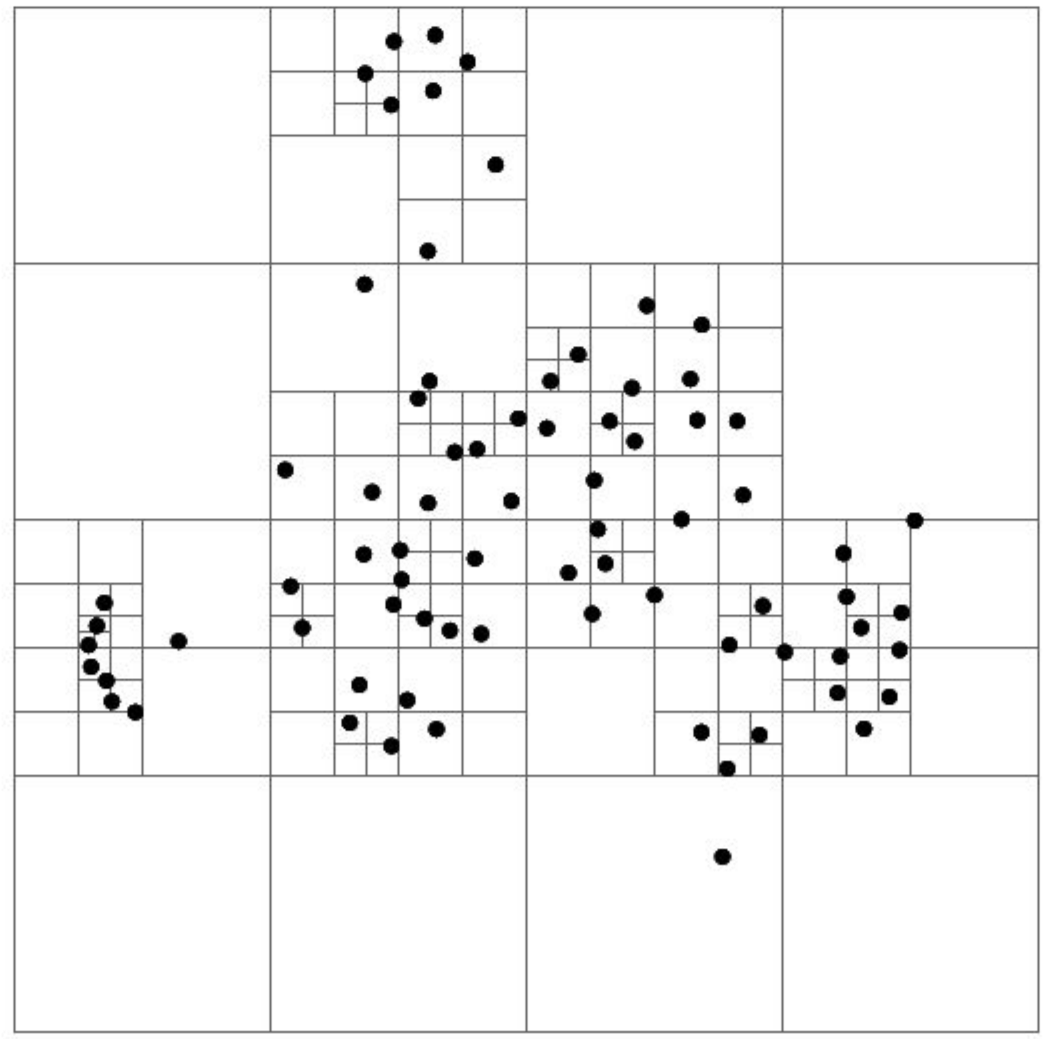
Attractive forces:

- Only use a small number of non-zero affinities, i.e. a sparse k-nearest-neighbour (kNN) graph. This reduces the number of forces. (Standard heuristic: $k = 3P$. For $P = 30$, this gives $k = 90$.)
- Use approximate kNN graphs. This speeds up graph construction.

Repulsive forces:

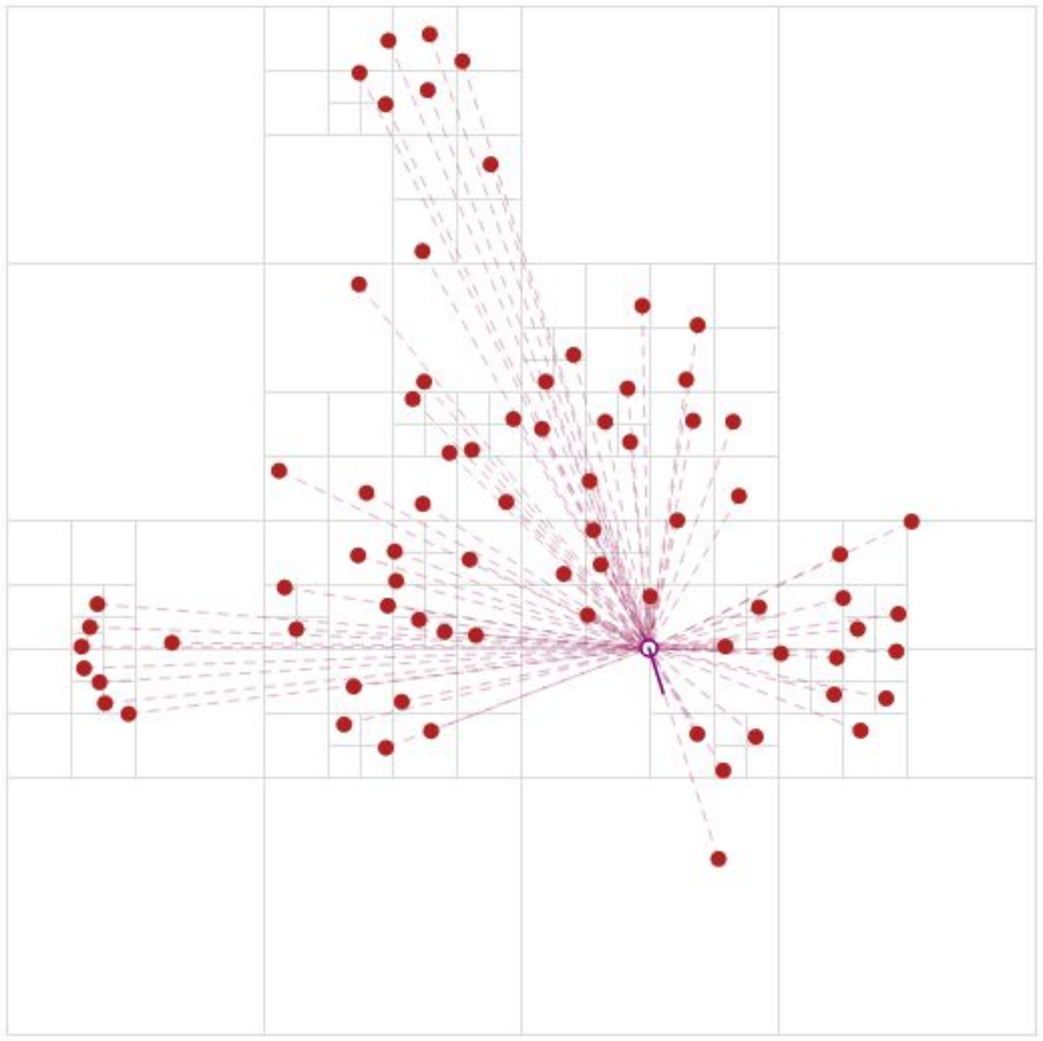
- Barnes-Hut t-SNE (BH t-SNE, 2013): $O(n \log(n))$
- FFT-accelerated interpolation-based t-SNE (FIt-SNE, 2019): $O(n)$
- Noise contrastive estimation / negative sampling (NCVis, 2020): $O(n)$

Barnes-Hut approximation



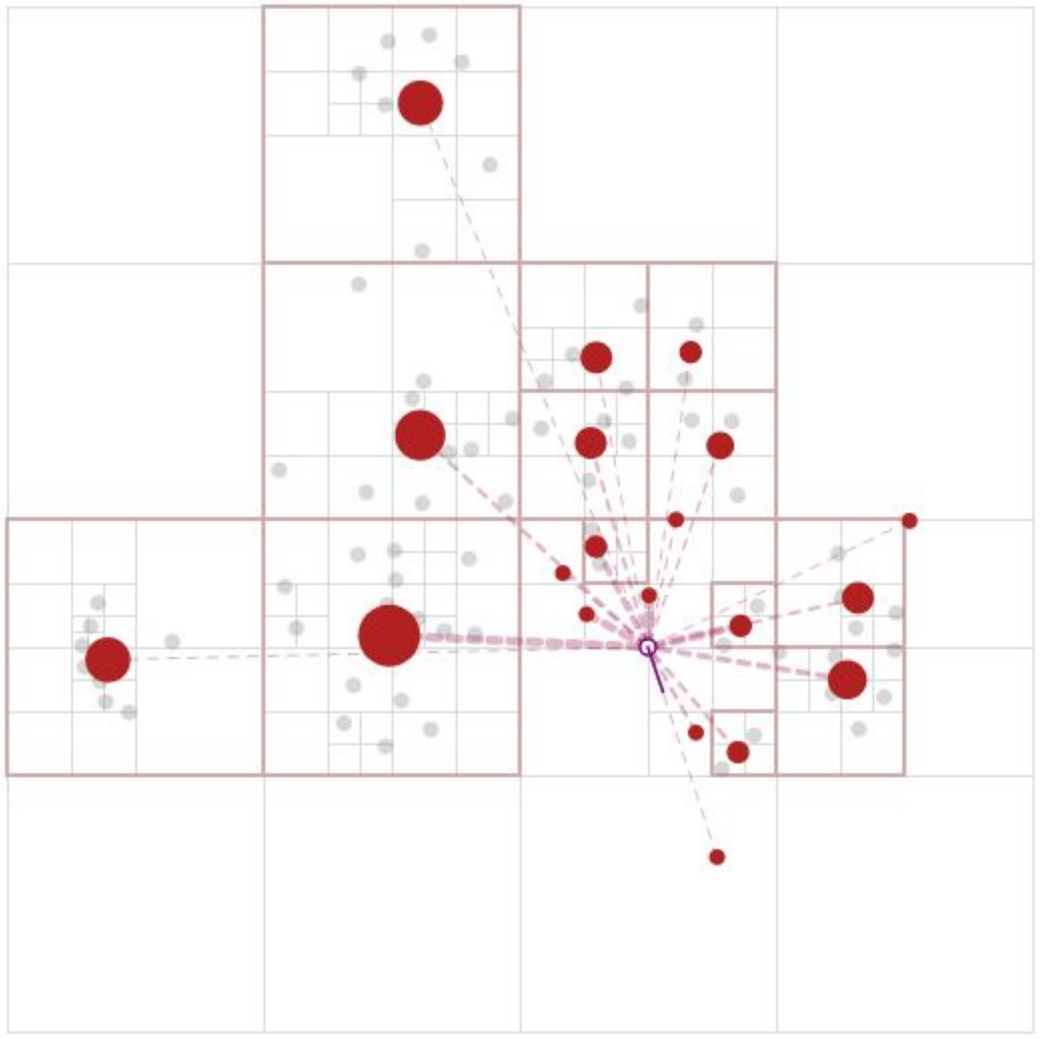
https://jheer.github.io/barnes-hut

Barnes-Hut approximation



https://jheer.github.io/barnes-hut

Barnes-Hut approximation

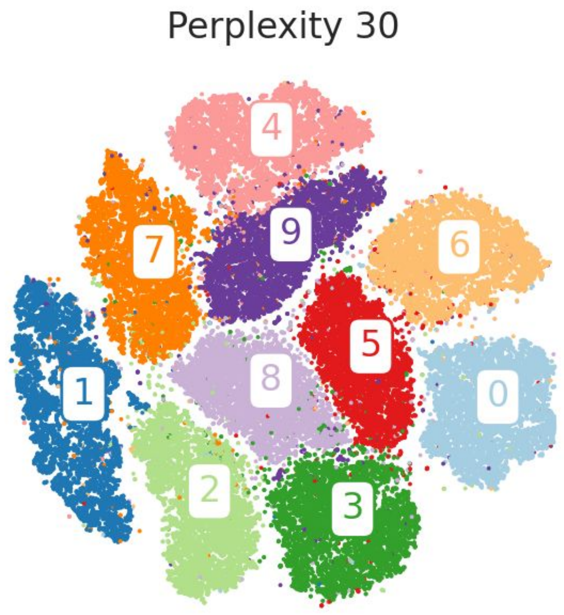


https://jheer.github.io/barnes-hut

Kumar Bipin

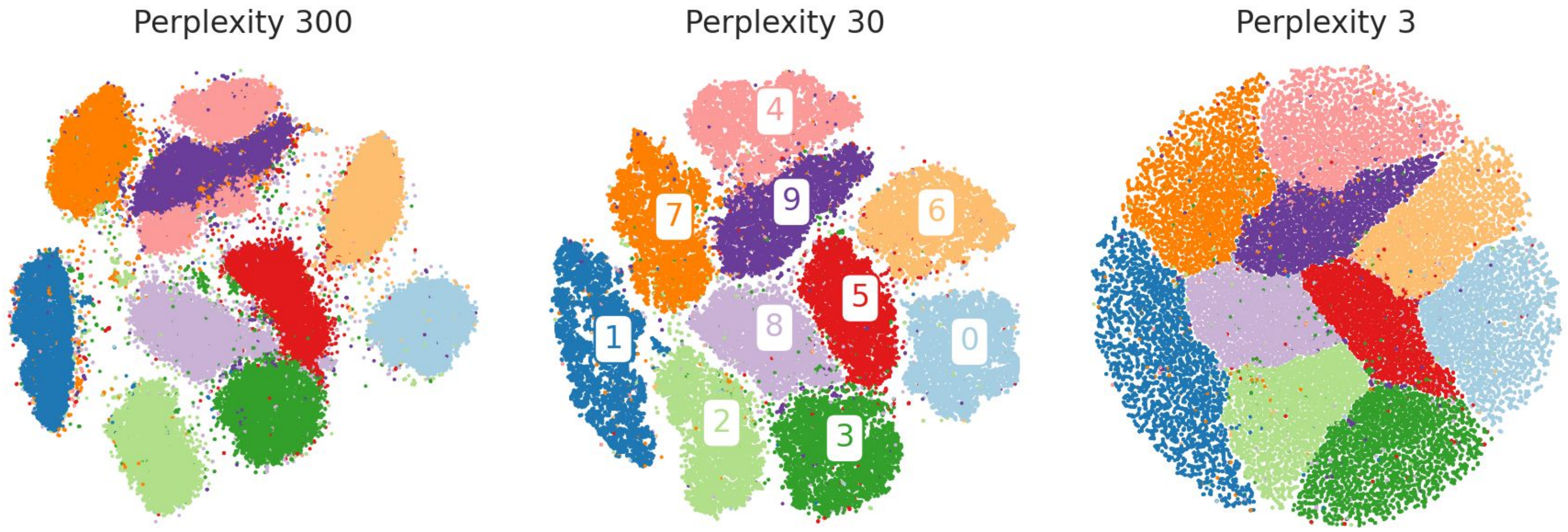
Perplexity and the number of neighbours

Perplexity can be seen as the 'effective' number of neighbours that enter the loss function. Default perplexity is 30.



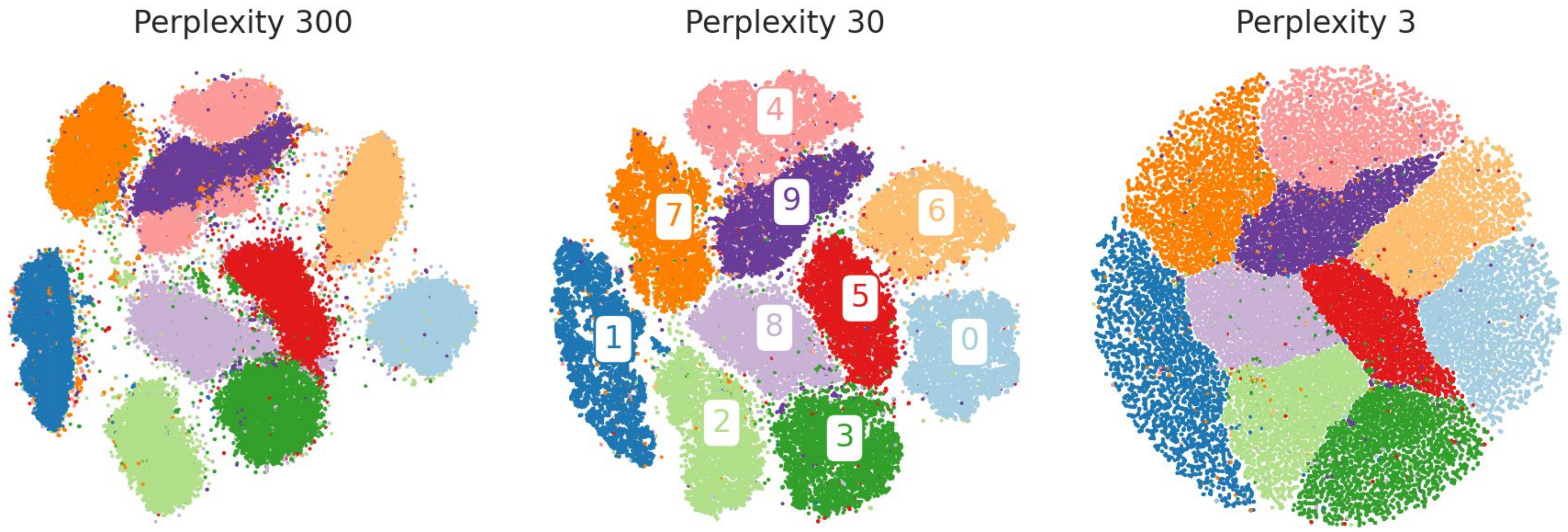
Perplexity and the number of neighbours

Perplexity can be seen as the 'effective' number of neighbours that enter the loss function. Default perplexity is 30.



Perplexity and the number of neighbours

Perplexity can be seen as the ‘effective’ number of neighbours that enter the loss function. Default perplexity is 30.

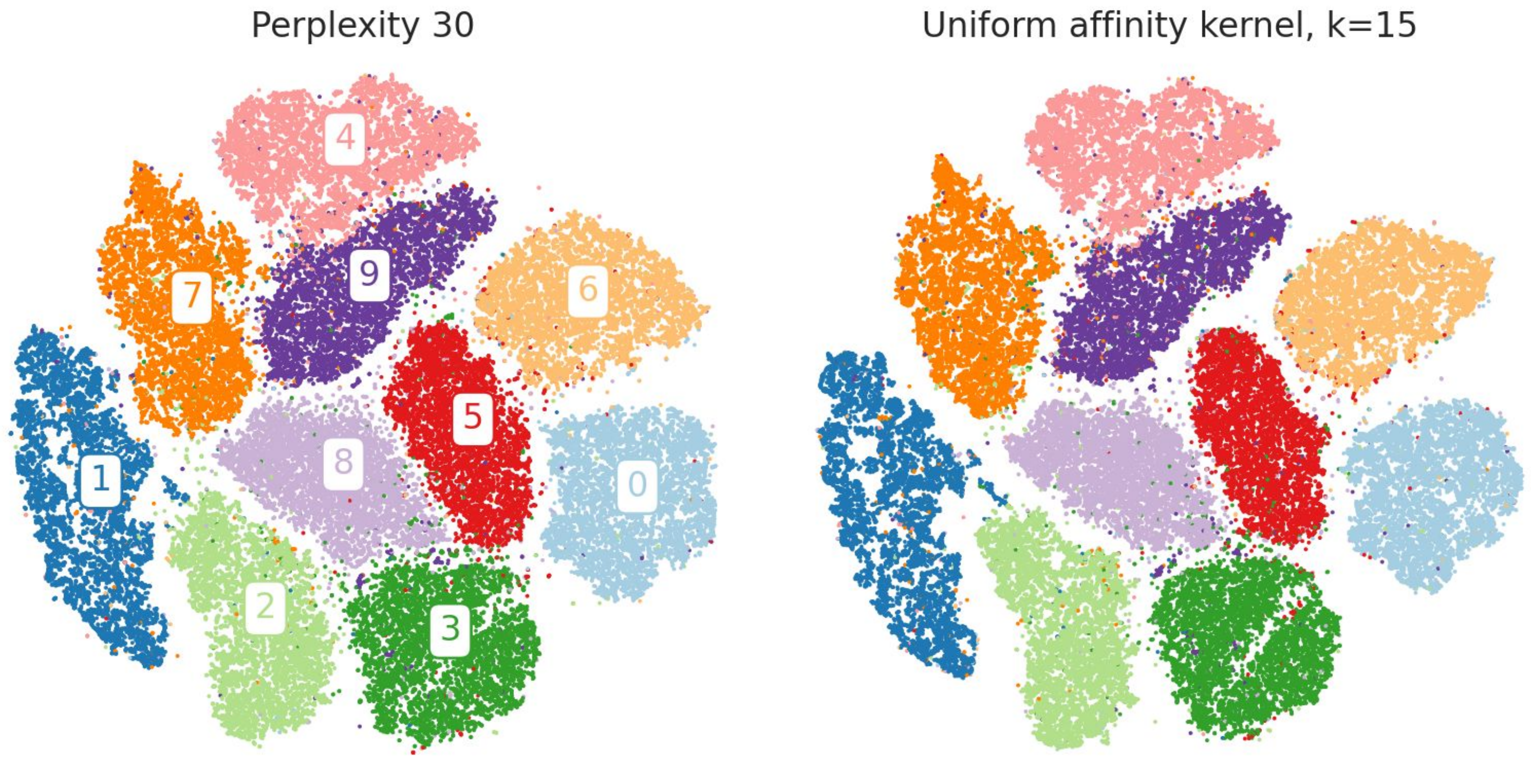


Much smaller values are rarely useful.

Much larger values are impractical or even computationally prohibitive.

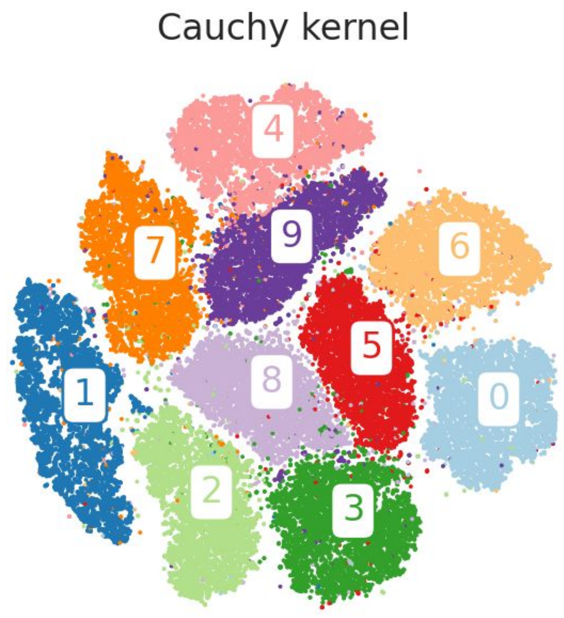
Uniform affinity

Gaussian affinities with perplexity P can usually be replaced by the uniform affinities with $k \approx P/2$.



Low-dimensional similarity kernel

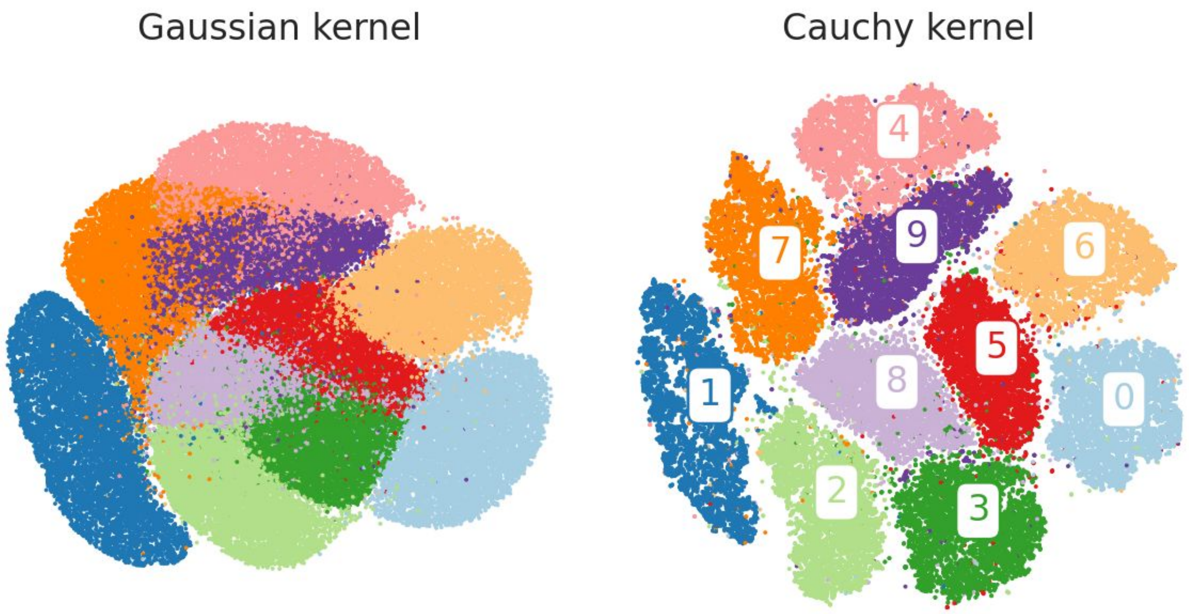
The main innovation of t-SNE compared to SNE was the Cauchy kernel, addressing the ‘crowding problem’ of SNE.



(Kobak et al., 2020)

Low-dimensional similarity kernel

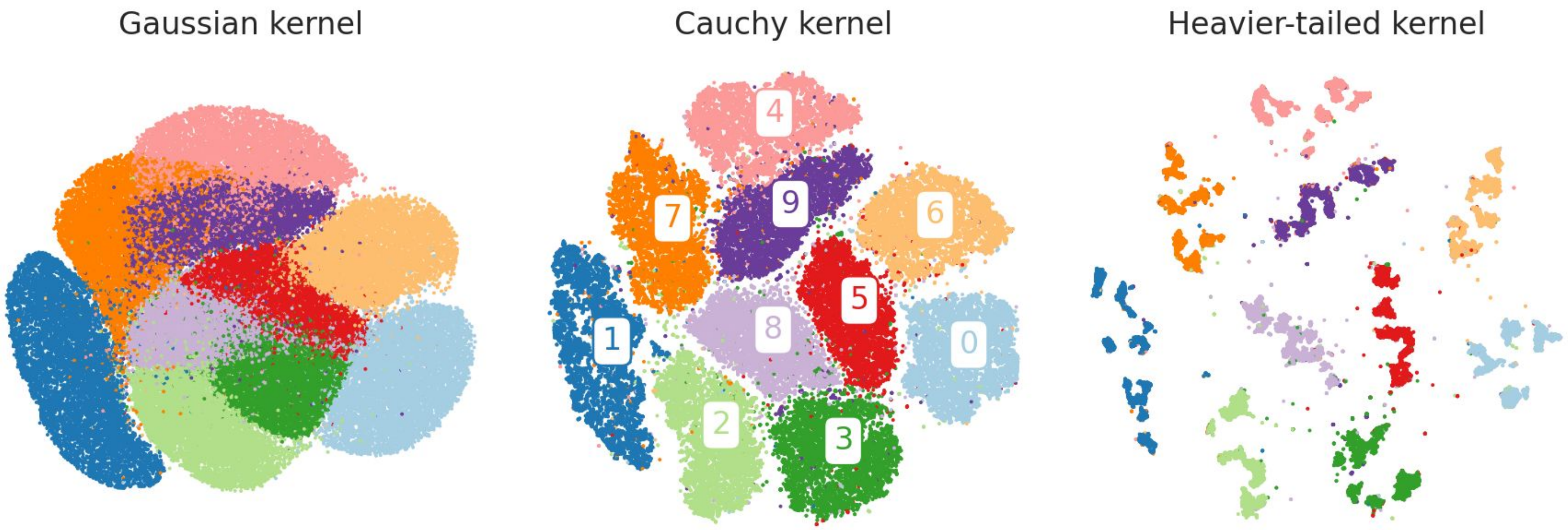
The main innovation of t-SNE compared to SNE was the Cauchy kernel, addressing the ‘crowding problem’ of SNE.



(Kobak et al., 2020)

Low-dimensional similarity kernel

The main innovation of t-SNE compared to SNE was the Cauchy kernel, addressing the 'crowding problem' of SNE.

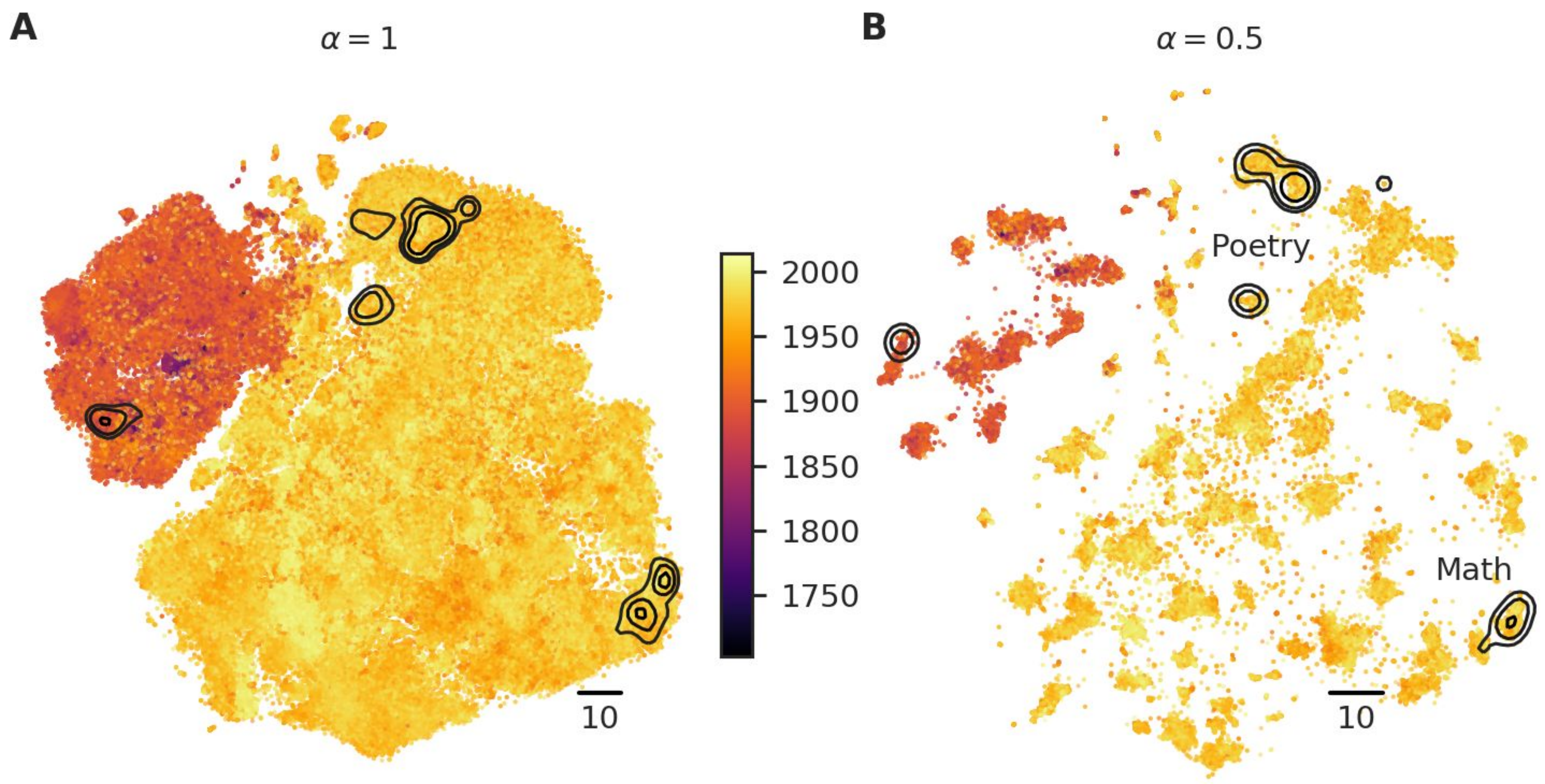


Even heavier-tailed kernels can bring out even finer cluster structure.

(Kobak et al., 2020)

Low-dimensional similarity kernel

HathiTrust library, Russian language ($n \approx 400{,}000$):



(Kobak et al., 2020)

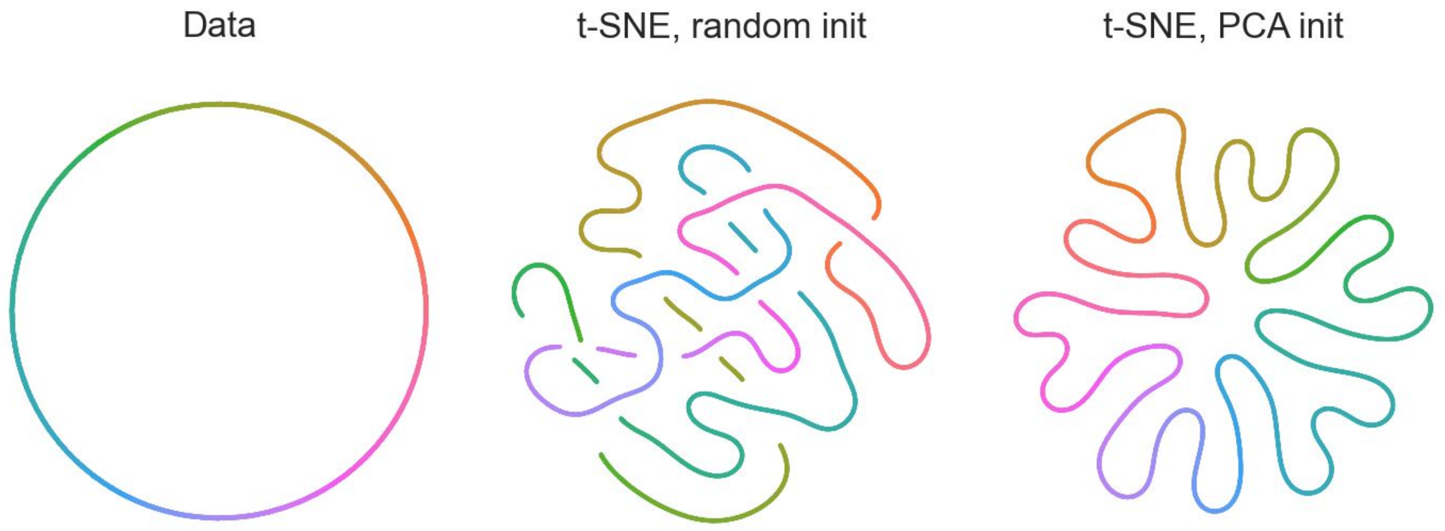
The role of initialization

t-SNE preserves local structure (neighbours) but often struggles to preserve global structure. The loss function has many local minima and initialization can play a large role.

(Kobak and Linderman, 2021)

The role of initialization

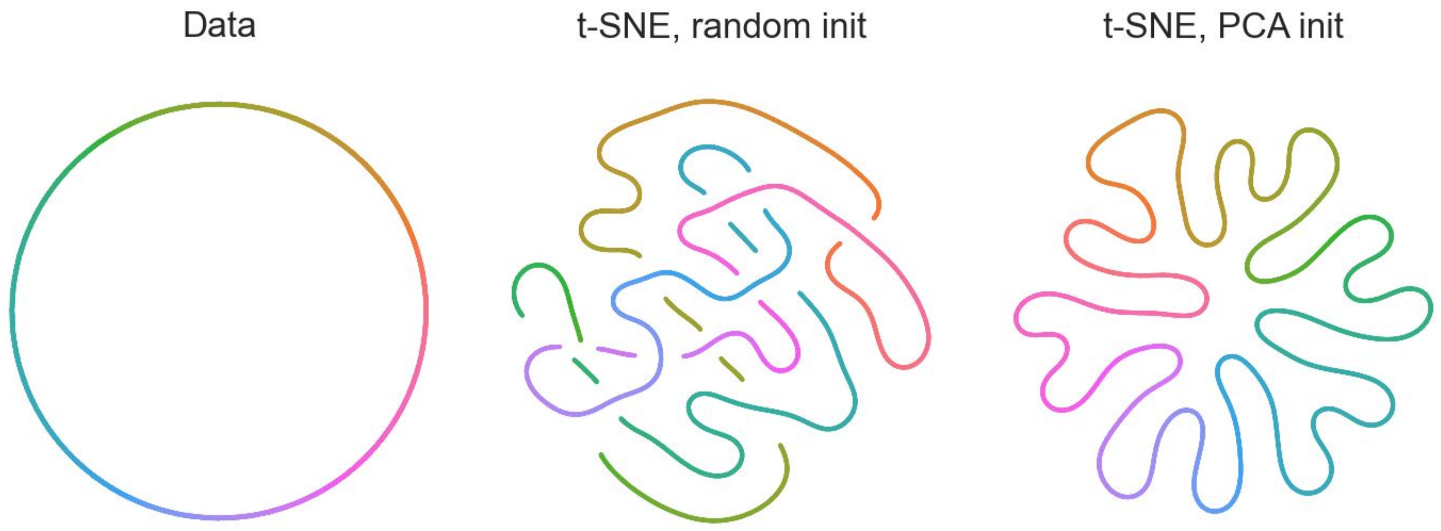
t-SNE preserves local structure (neighbours) but often struggles to preserve global structure. The loss function has many local minima and initialization can play a large role.



(Kobak and Linderman, 2021)

The role of initialization

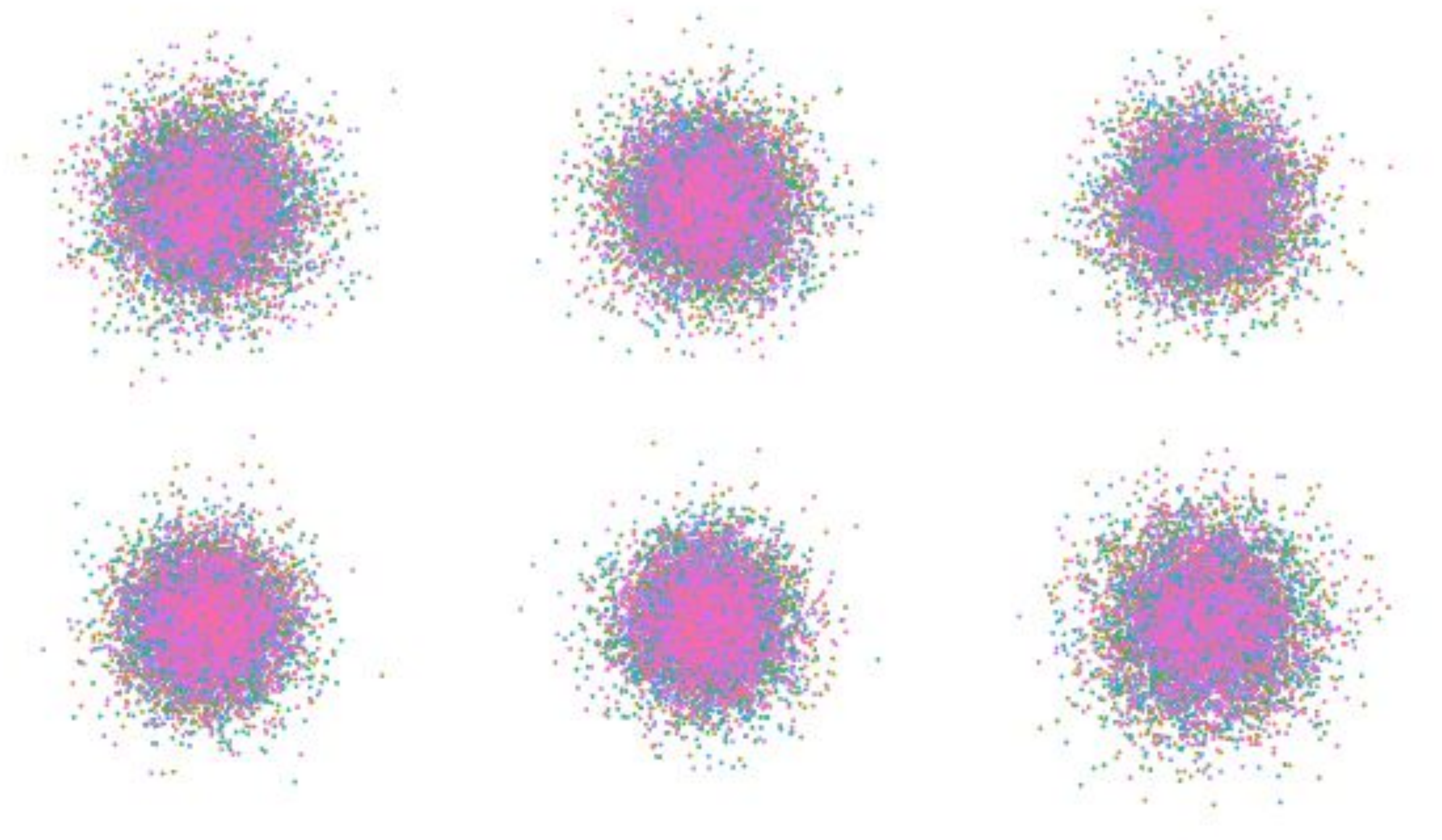
t-SNE preserves local structure (neighbours) but often struggles to preserve global structure. The loss function has many local minima and initialization can play a large role.



Always use informative initialization, e.g. PCA.

(Kobak and Linderman, 2021)

What happens with random initialization

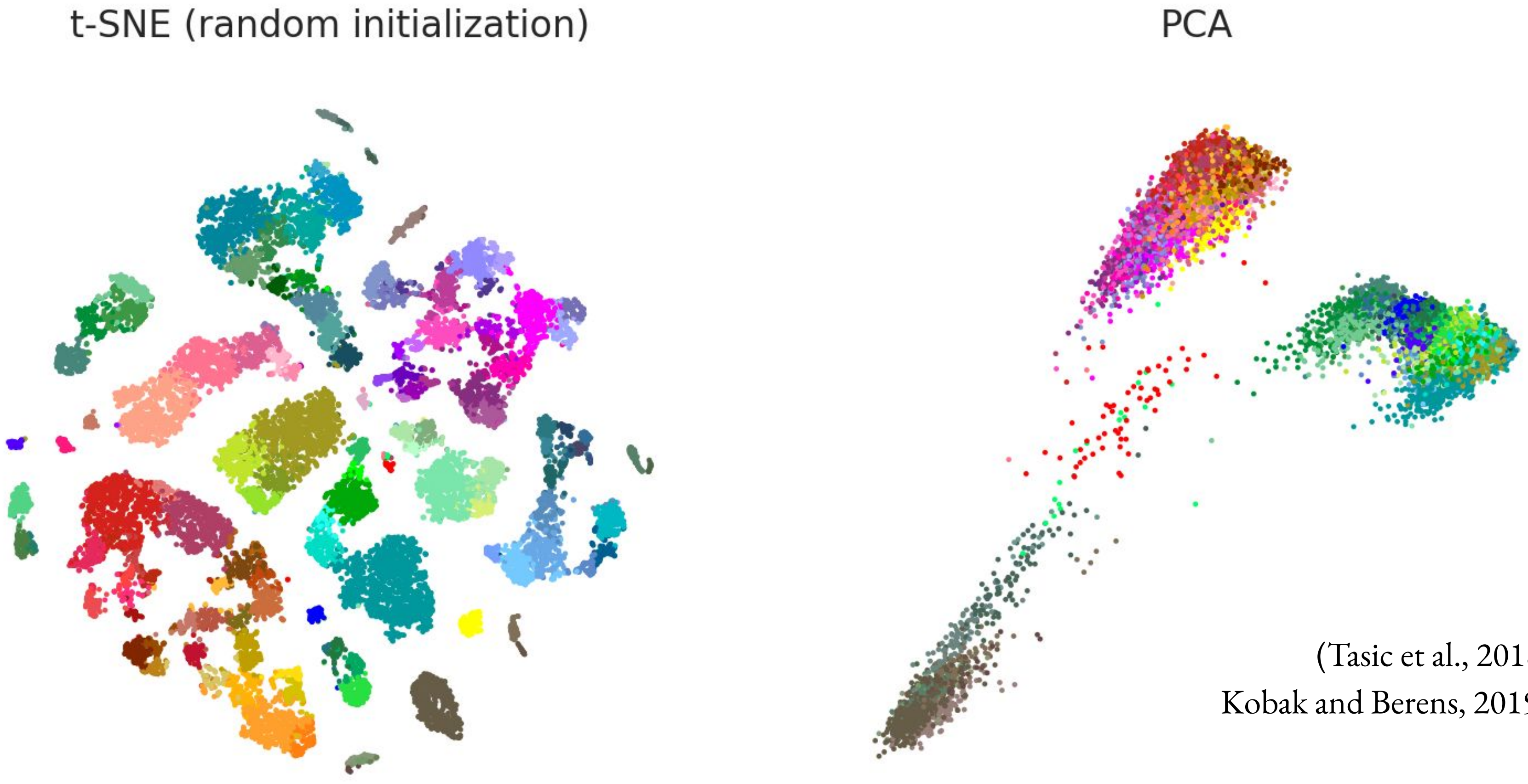


Note: strong exaggeration approximates Laplacian Eigenmaps.

(Linderman and Steinerberger, 2019)

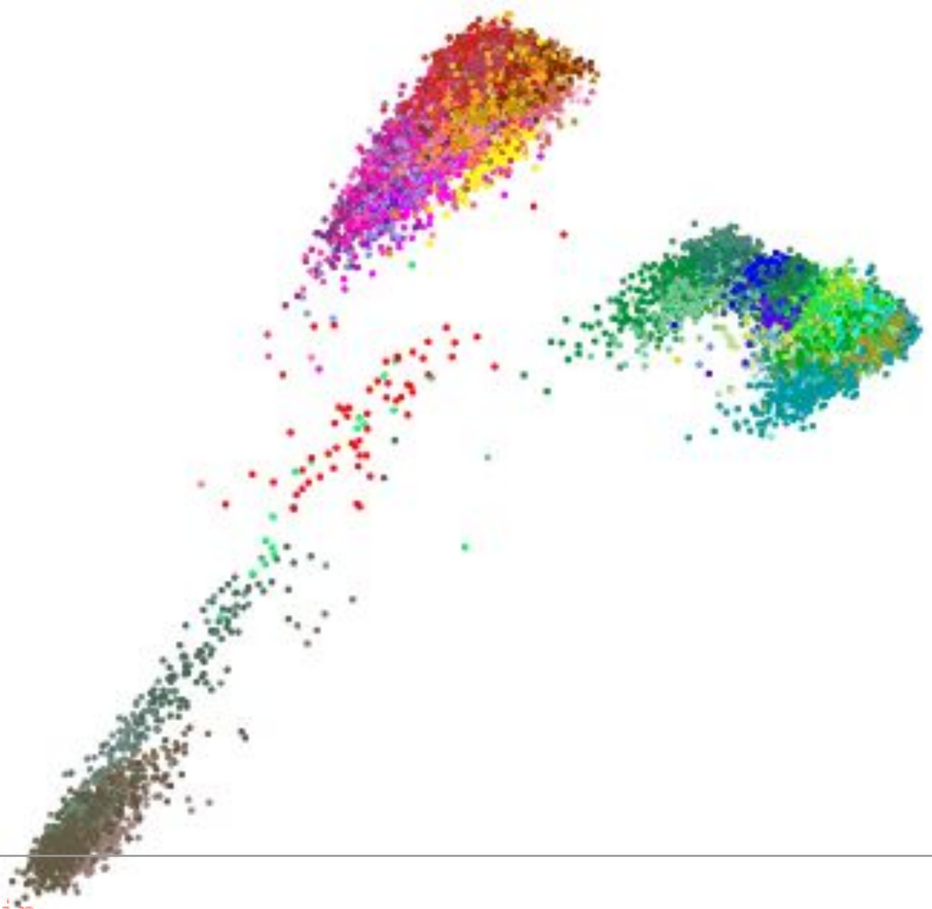
Global structure

t-SNE preserves local structure (neighbours) but often struggles to preserve global structure: real-life example from single-cell transcriptomics.



(Tasic et al., 2018;
Kobak and Berens, 2019)

PCA initialization



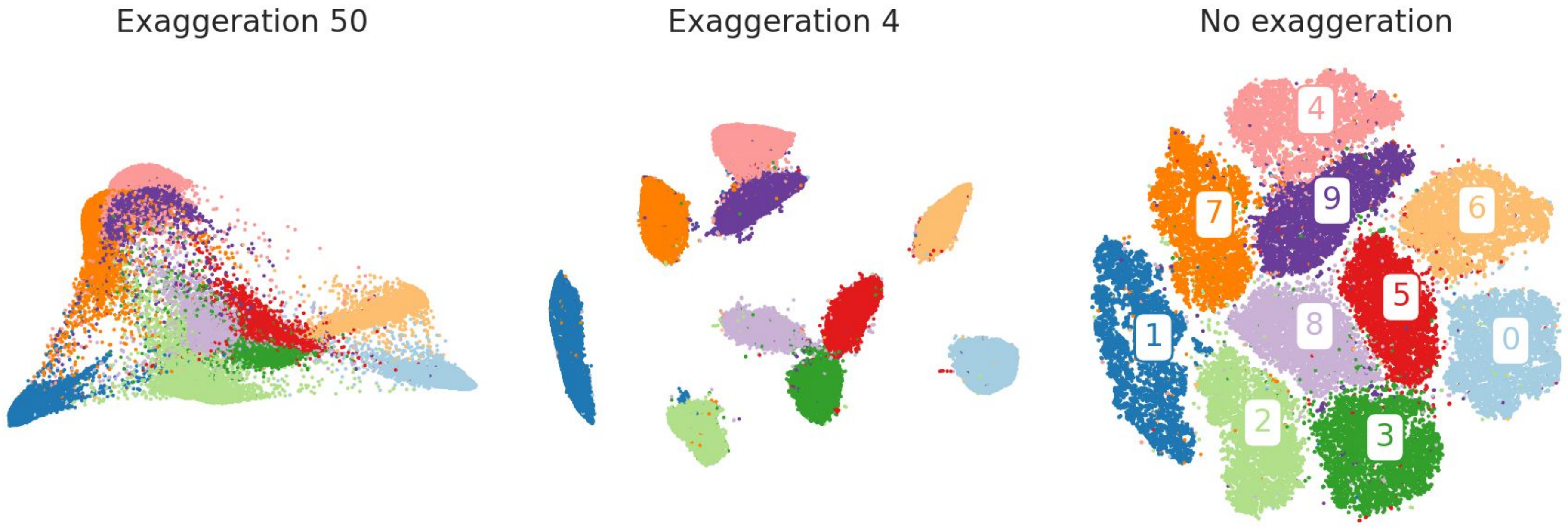
Attraction-repulsion spectrum

Early exaggeration multiplies attractive forces by 12 for 250 iterations. What happens if we keep the exaggeration on throughout the optimization?

(Böhm et al., 2020)

Attraction-repulsion spectrum

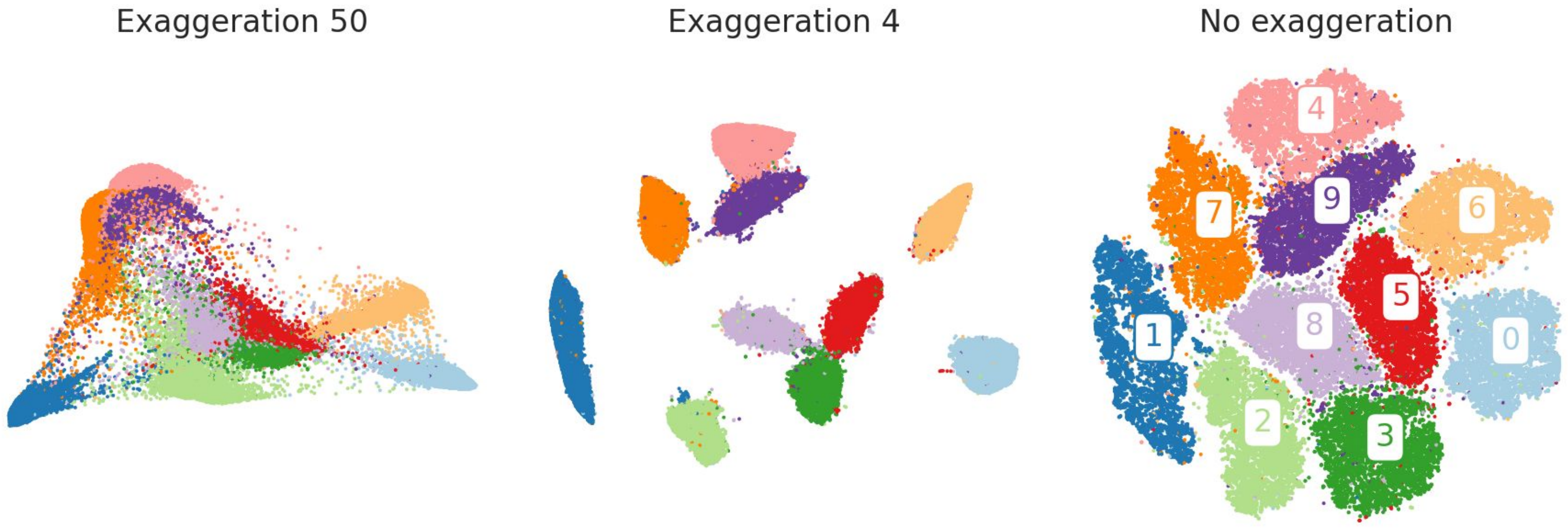
Early exaggeration multiplies attractive forces by 12 for 250 iterations. What happens if we keep the exaggeration on throughout the optimization?



(Böhm et al., 2020)

Attraction-repulsion spectrum

Early exaggeration multiplies attractive forces by 12 for 250 iterations. What happens if we keep the exaggeration on throughout the optimization?

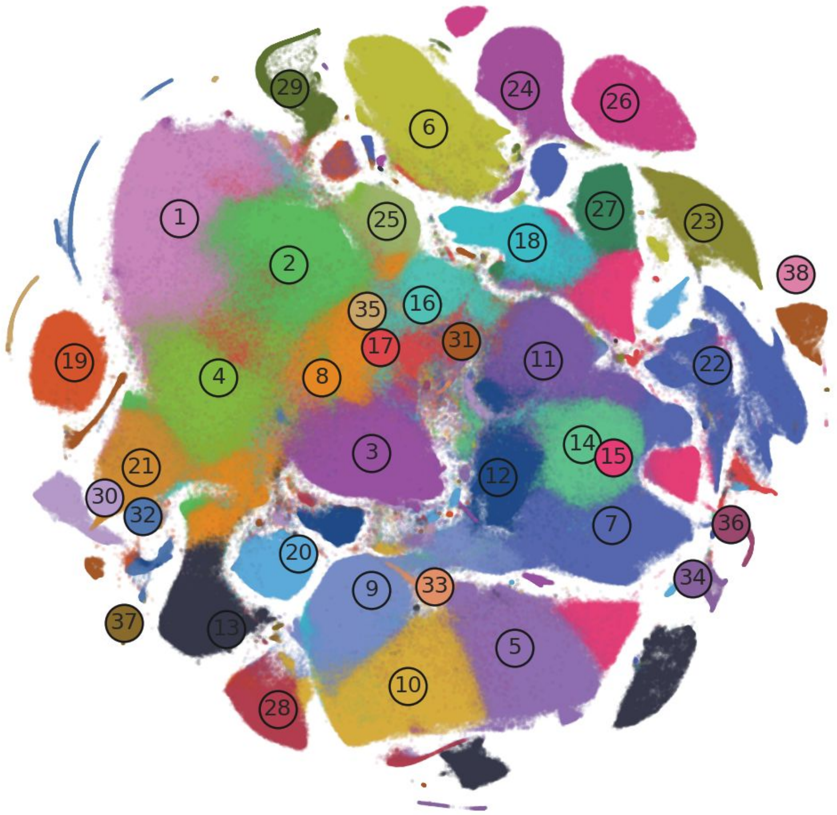


Many other methods, e.g. UMAP, produce embeddings that approximately fall on this spectrum.

(Böhm et al., 2020)

Continuity-discreteness tradeoff

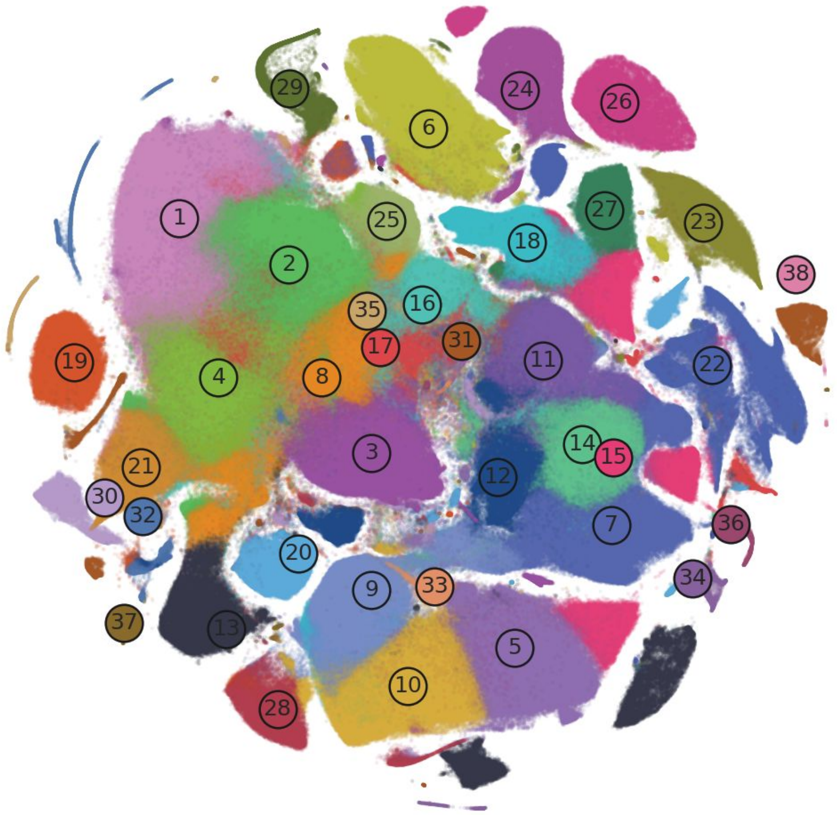
Single-cell transcriptomic study of mouse embryogenesis ($n \approx 2{,}000{,}000$).
Left — original t-SNE. Right — high learning rate, PCA initialization.

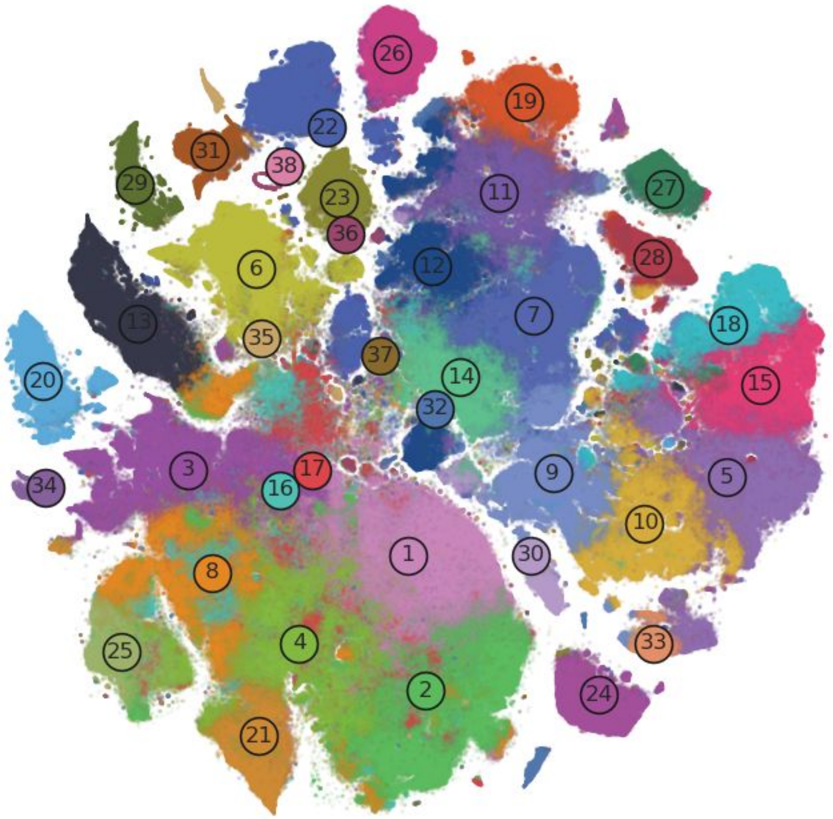


(Cao et al., 2019; Kobak and Berens, 2019)

Continuity-discreteness tradeoff

Single-cell transcriptomic study of mouse embryogenesis ($n \approx 2{,}000{,}000$).
Left — original t-SNE. Right — high learning rate, PCA initialization.

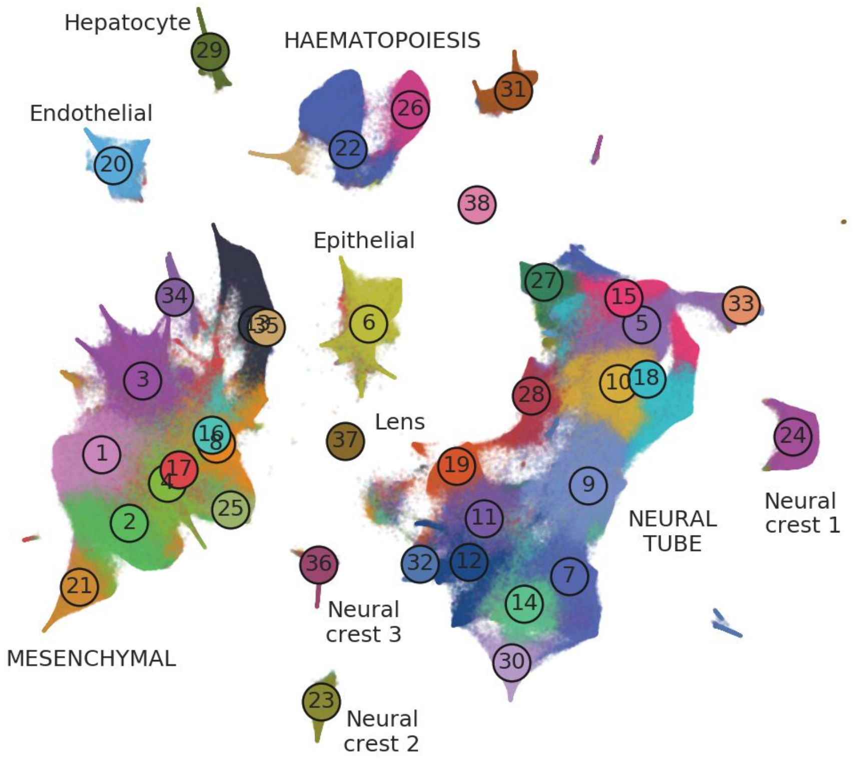


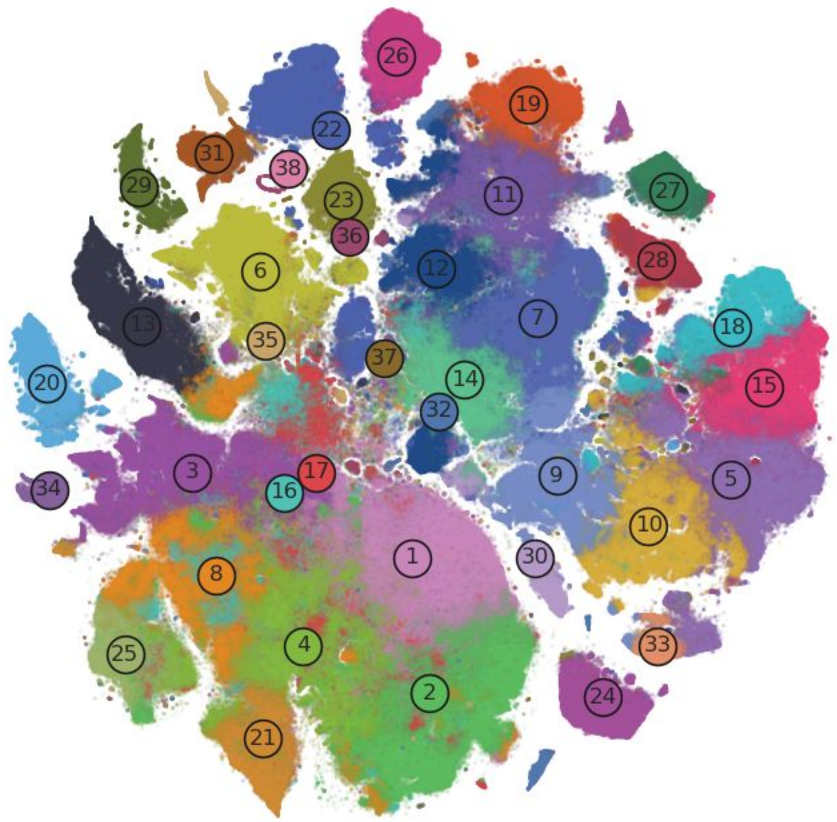


(Cao et al., 2019; Kobak and Berens, 2019)

Continuity-discreteness tradeoff

Single-cell transcriptomic study of mouse embryogenesis ($n \approx 2{,}000{,}000$).
Left — with exaggeration. Right — without exaggeration.

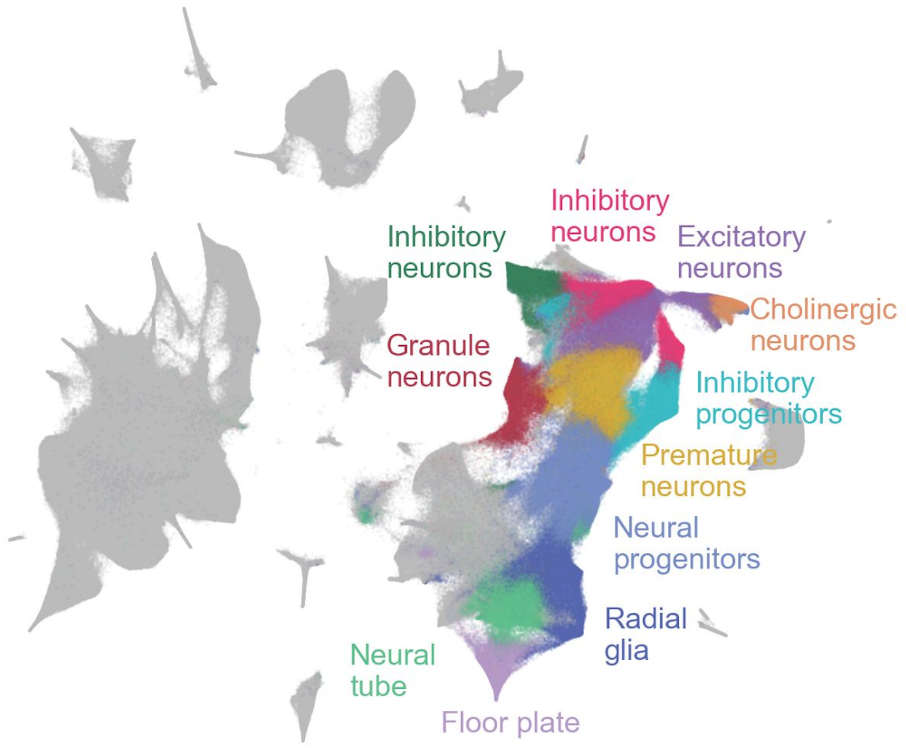


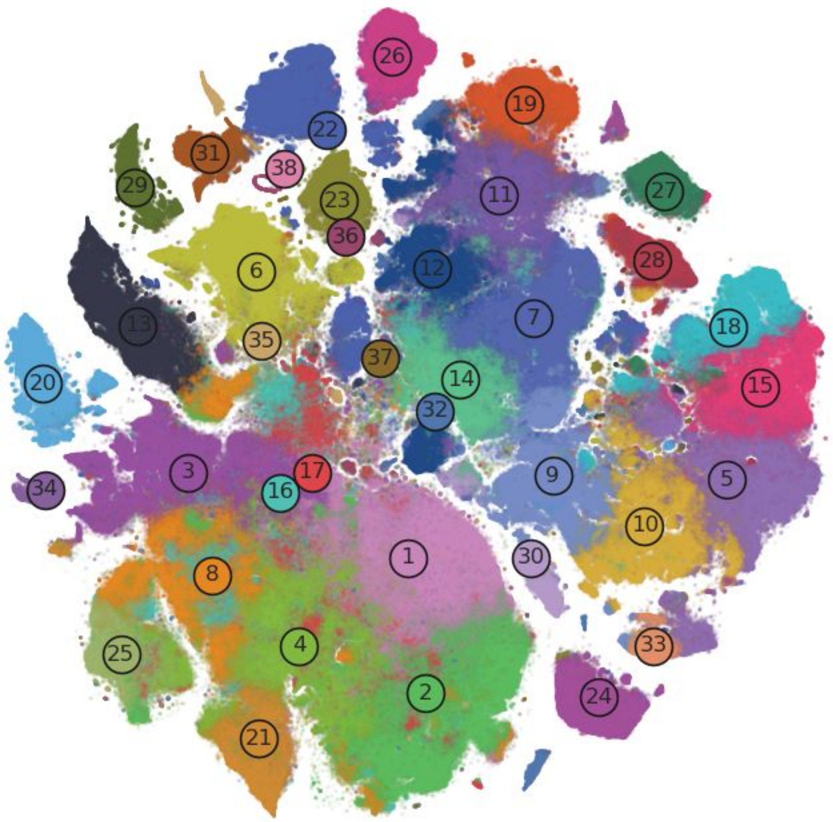


(Cao et al., 2019; Kobak and Berens, 2019)

Continuity-discreteness tradeoff

Single-cell transcriptomic study of mouse embryogenesis ($n \approx 2{,}000{,}000$).
Left — with exaggeration. Right — without exaggeration.





(Cao et al., 2019; Kobak and Berens, 2019)