EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Nuts and bolts of neighbour embeddings

Dmitry Kobak

28 November 2025

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Everything you always wanted to know about *t*-SNE but were afraid to ask

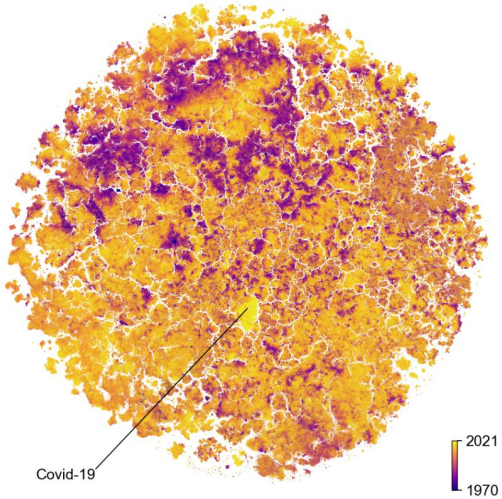
Dmitry Kobak

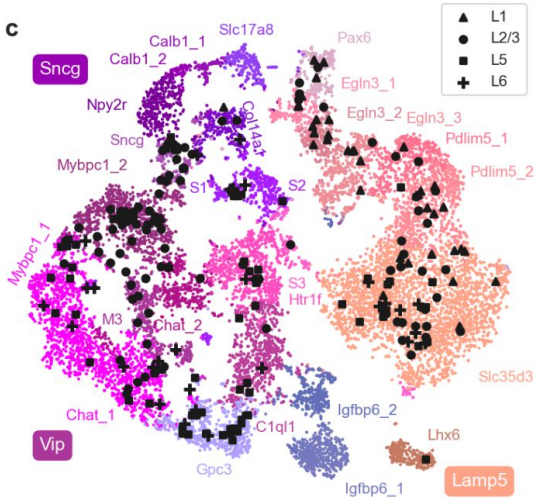
28 November 2025

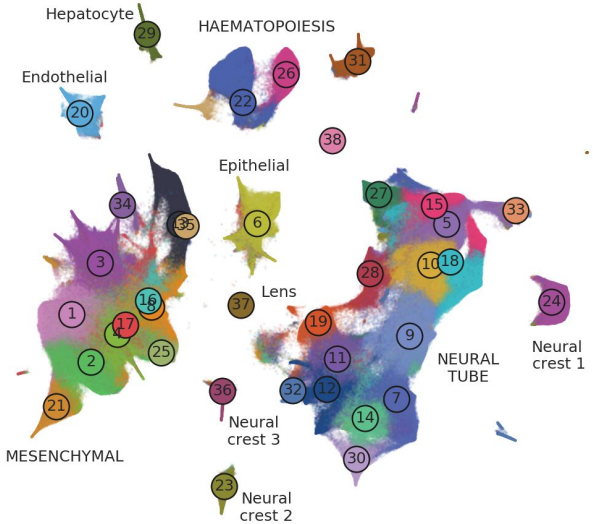
EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Much more than you ever
~~Everything you always~~ wanted to know about *t*-SNE but were afraid to ask

Dmitry Kobak
28 November 2025


Covid-19
2021
1970


c
L1
L2/3
L5
L6
Sncg
Calb1_1
Calb1_2
Slc17a8
Pax6
Npy2r
Egln3_1
Egln3_2
Egln3_3
Sncg
Pdlim5_1
Mybpc1_2
Pdlim5_2
S1
S2
Mybpc1_1
S3
Htr1f
M3
Chat_2
Slc35d3
Chat_1
Igfbp6_2
C1ql1
Lhx6
Vip
Gpc3
Lamp5
Igfbp6_1


Hepatocyte
HAEMATOPOIESIS
Endothelial
Epithelial
Lens
NEURAL TUBE
Neural crest 1
Neural crest 3
MESENCHYMAL
Neural crest 2

La Manno et al., Nature 2021

Neighbor embeddings

identify similar objects and embed them close to each other

Neighbor embeddings

identify similar objects and embed them close to each other

[PDF] **Stochastic neighbor embedding**

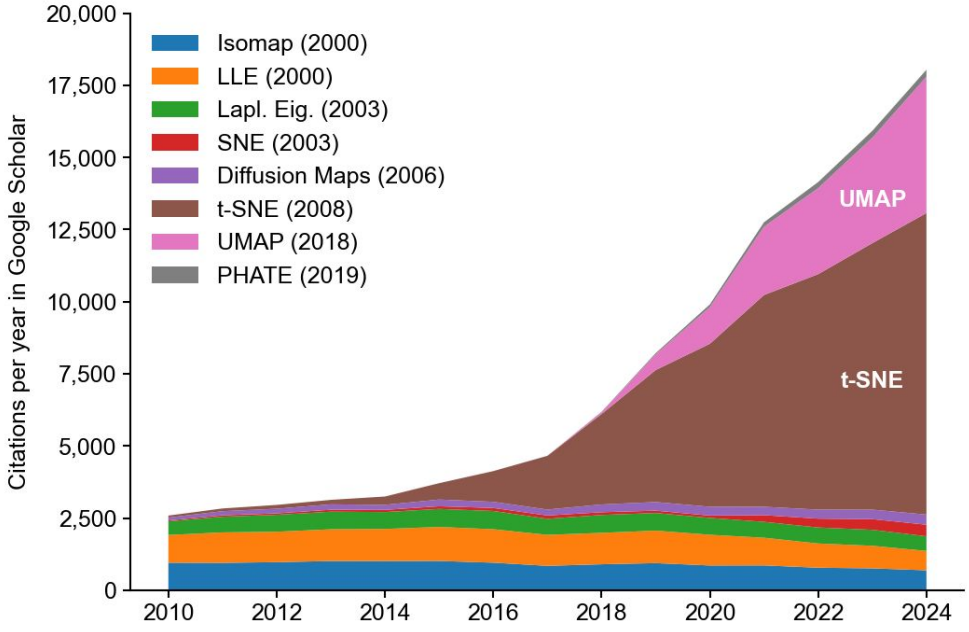
G Hinton, ST Roweis - NIPS, 2002 - Citeseer

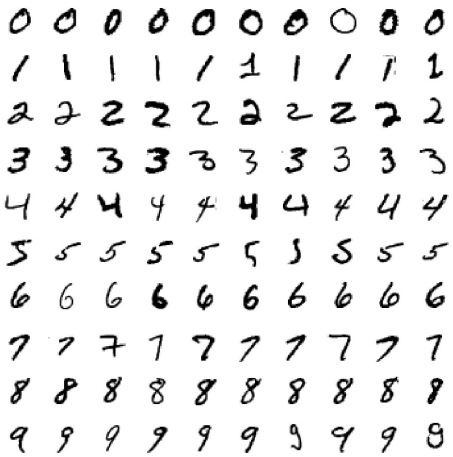
We describe a probabilistic approach to the task of placing objects, described by high-dimensional vectors or by pairwise dissimilarities, in a low-dimensional space in a way that preserves neighbor identities. A Gaussian is centered on each object in the high …

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

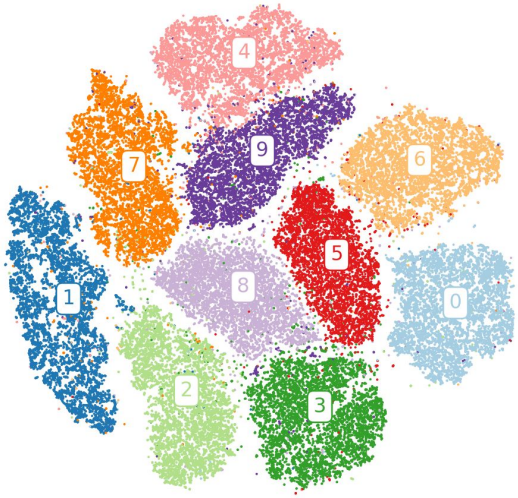
L Van der Maaten, G Hinton - Journal of machine learning research, 2008 - jmlr.org

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 …

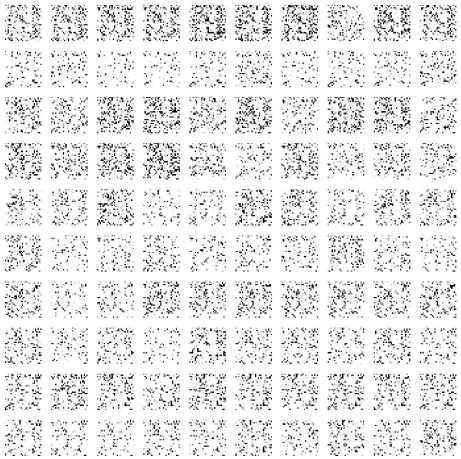




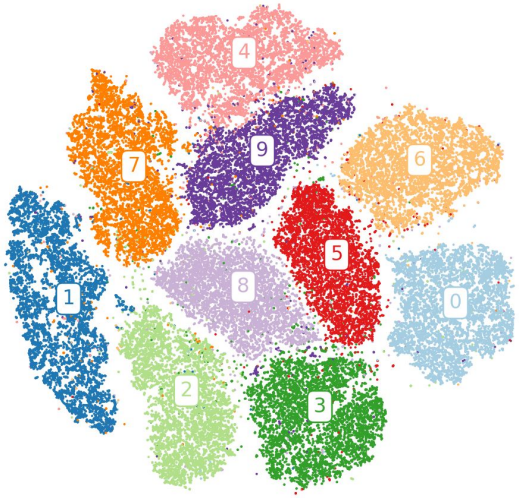
MNIST



t-SNE

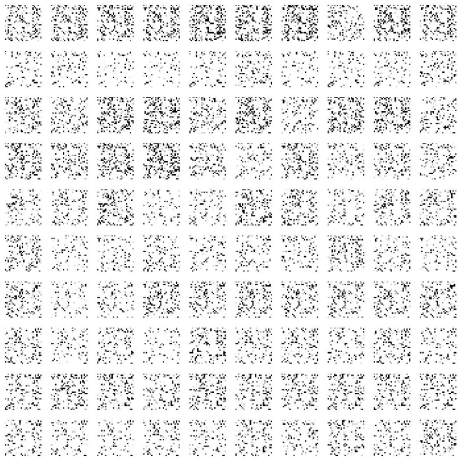


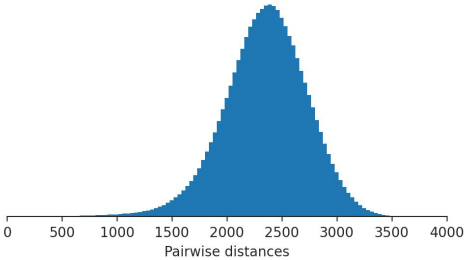
MNIST

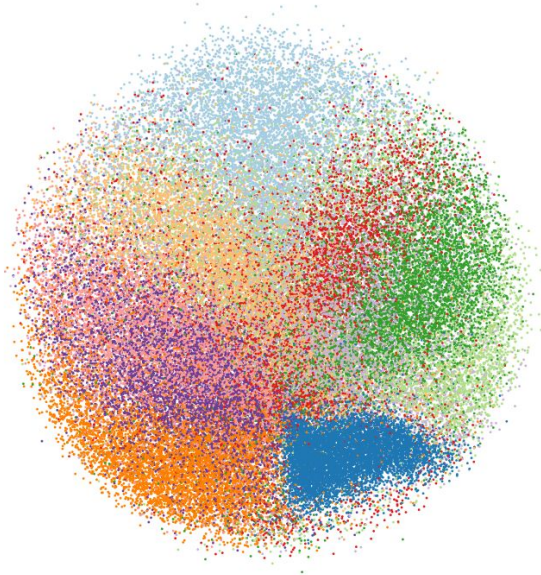


t-SNE

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

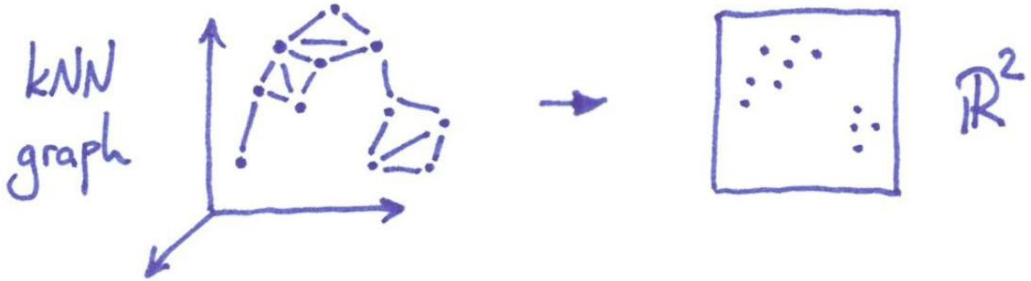




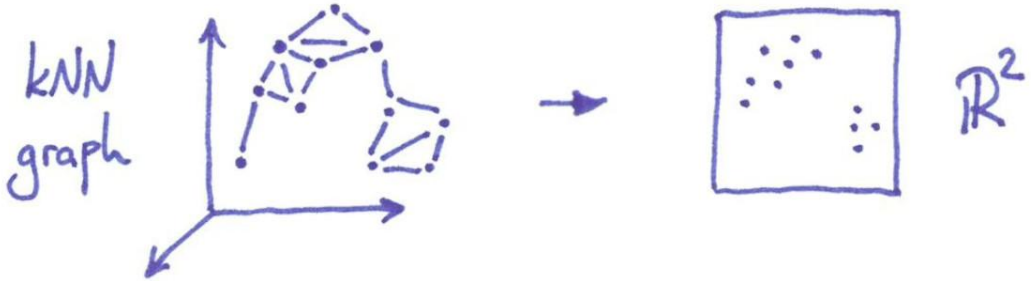


MDS

Neighbor embeddings



t-SNE

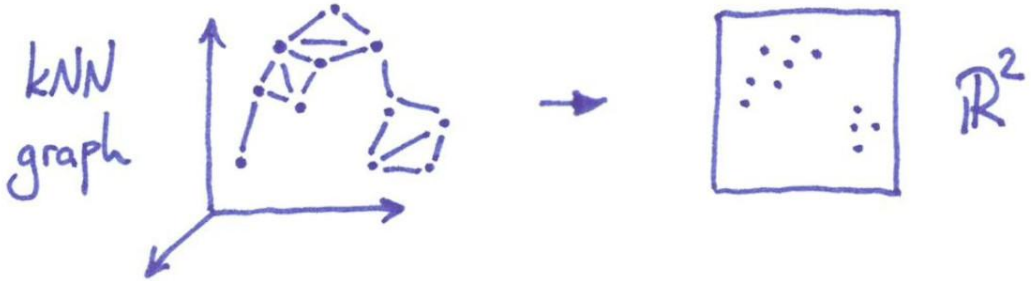


$$\mathcal{L}_{\text{t-SNE}} = \sum_{ij} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

p_{ij} can be normalized symmetrized adjacency matrix of the kNN graph.

$$q_{ij} = \frac{w_{ij}}{Z} = \frac{w_{ij}}{\sum w_{ij}} \text{ with } w_{ij} = \frac{1}{1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2}.$$

t-SNE



$$\mathcal{L}_{\text{t-SNE}} = \sum_{ij} p_{ij} \log \frac{p_{ij}}{q_{ij}} \sim -\sum_{ij} p_{ij} \log q_{ij} = -\sum_{ij} p_{ij} \log \frac{w_{ij}}{Z} = \underbrace{-\sum_{ij} p_{ij} \log w_{ij}}_{\text{attraction}} + \underbrace{\log \sum_{ij} w_{ij}}_{\text{repulsion}}.$$

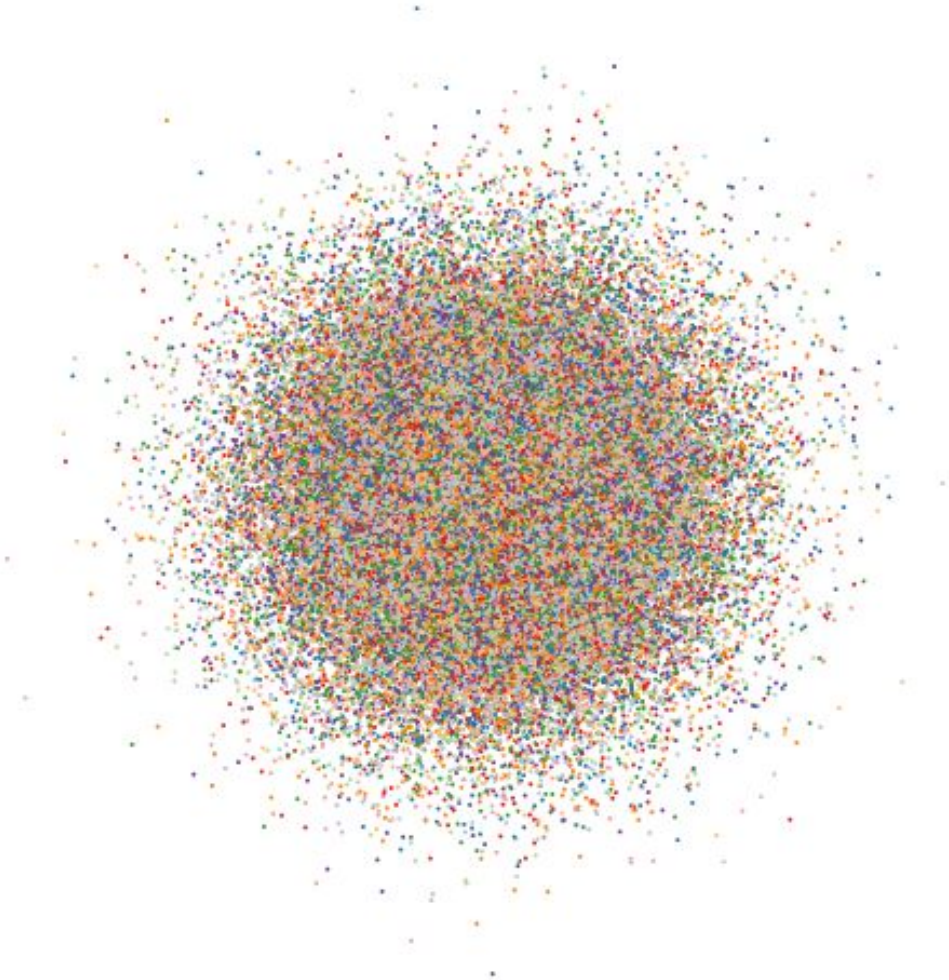
p_{ij} can be normalized symmetrized adjacency matrix of the kNN graph.

$$q_{ij} = \frac{w_{ij}}{Z} = \frac{w_{ij}}{\sum w_{ij}} \text{ with } w_{ij} = \frac{1}{1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2}.$$

Gradient descent

Gradient descent starting
from a random configuration.

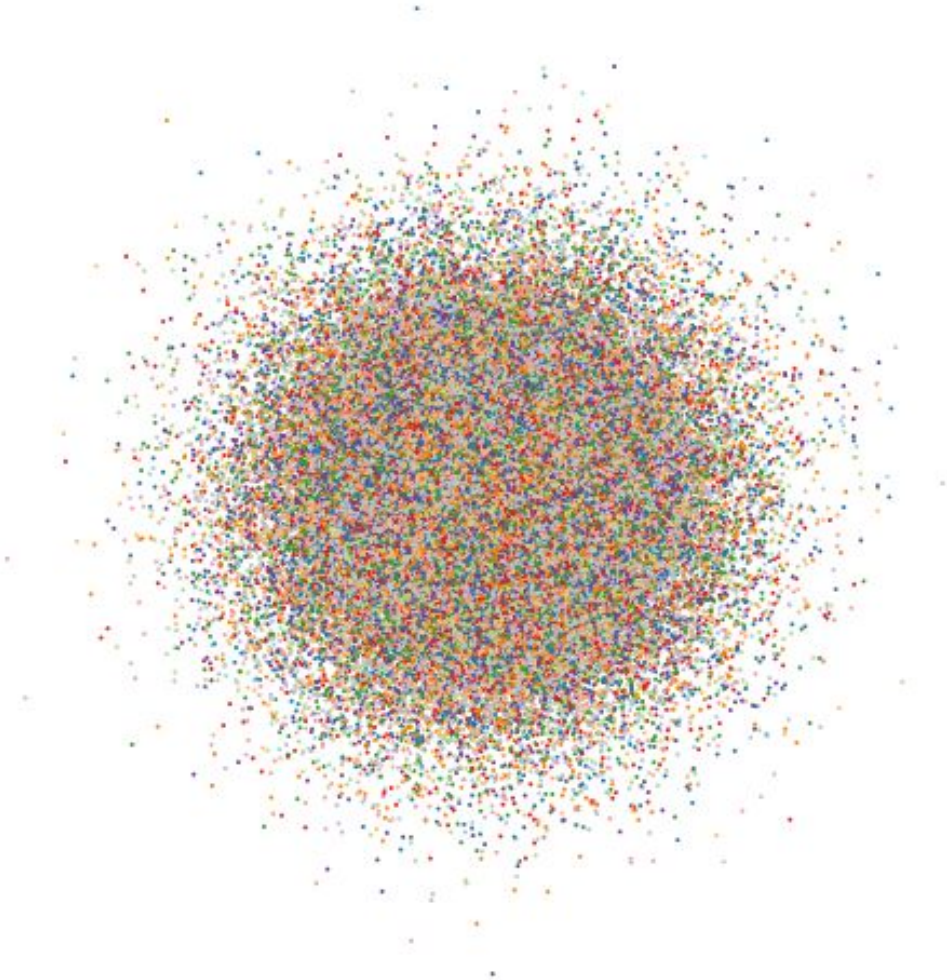
Each frame is scaled
(in reality embedding is
initialized small and slowly
grows in size).



Gradient descent

Optimization issues:

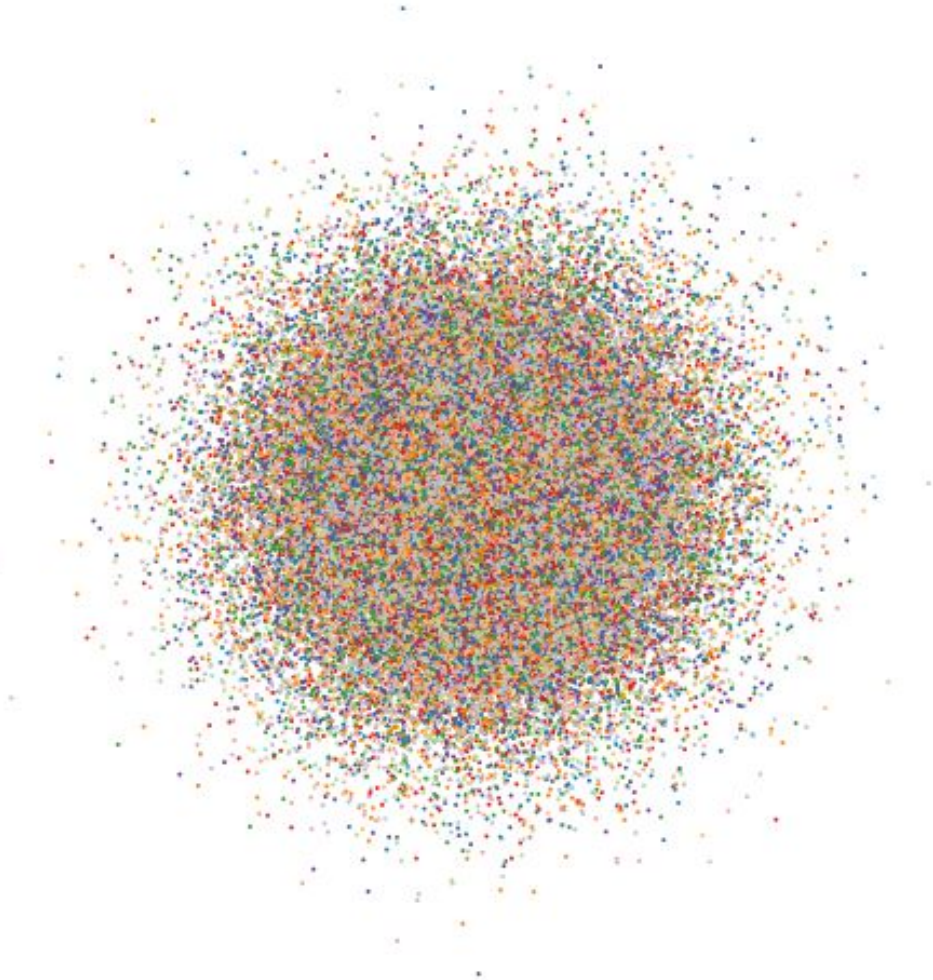
1) Initialization
2) Learning rate
3) Local minima
4) Runtime



Early exaggeration

Multiply all attractive forces by 12 for 250 iterations.

$$-\underbrace{\sum_{ij} p_{ij} \log w_{ij}}_{\text{attraction}} + \underbrace{\log \sum_{ij} w_{ij}}_{\text{repulsion}}$$



Early exaggeration

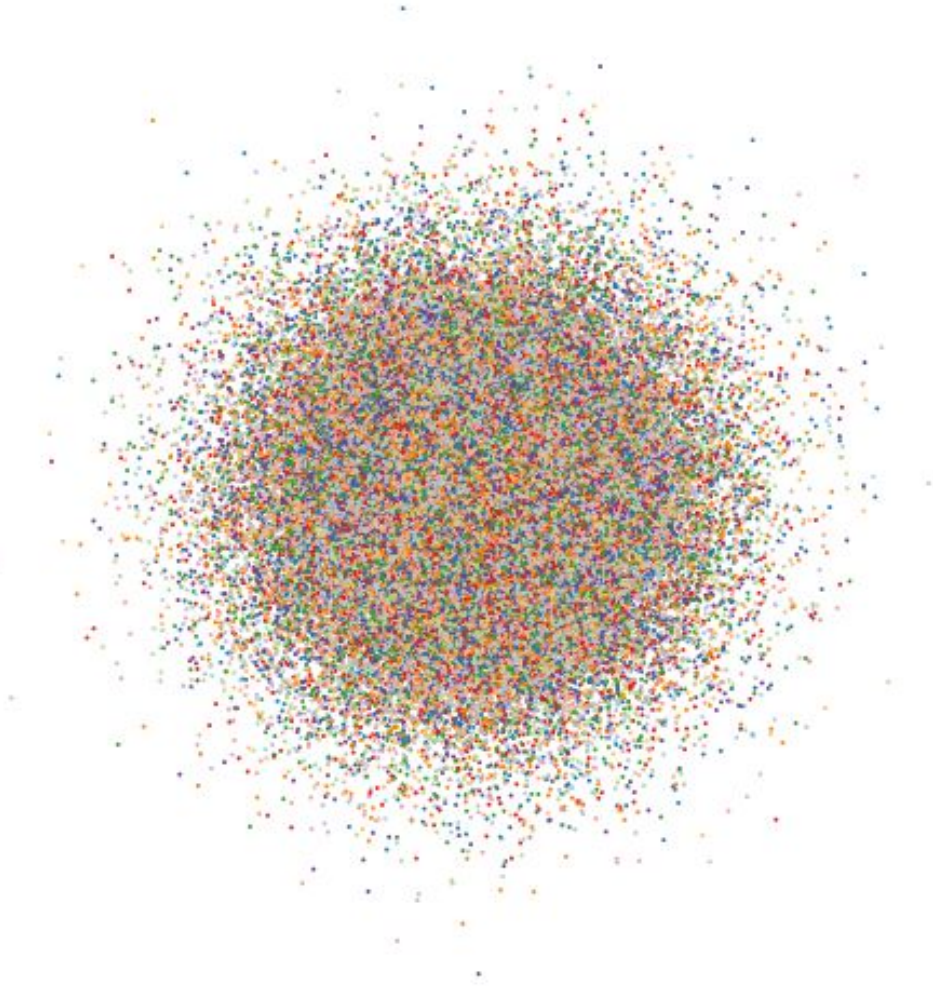
Multiply all attractive forces by 12 for 250 iterations.

$$\underbrace{-\sum_{ij} p_{ij} \log w_{ij}}_{\text{attraction}} + \underbrace{\log \sum_{ij} w_{ij}}_{\text{repulsion}}$$

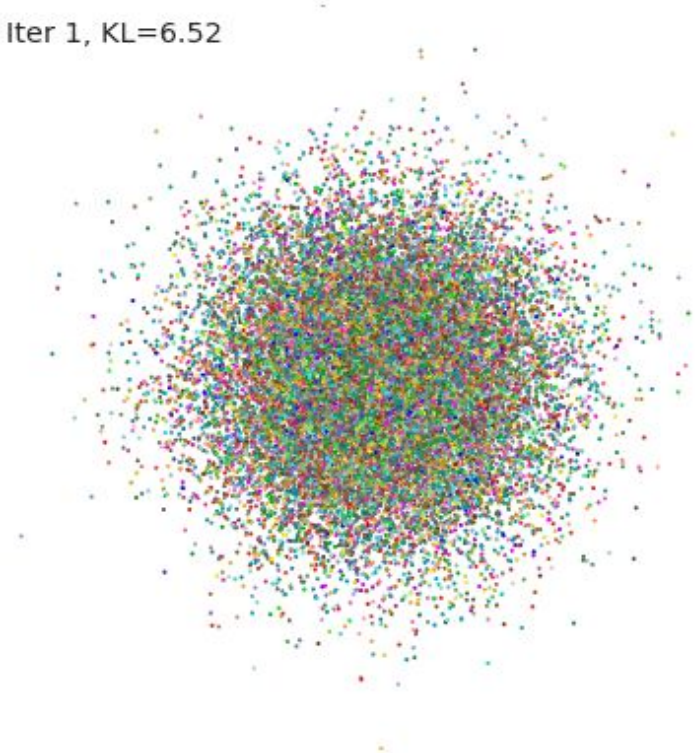
The learning rate should grow with sample size:

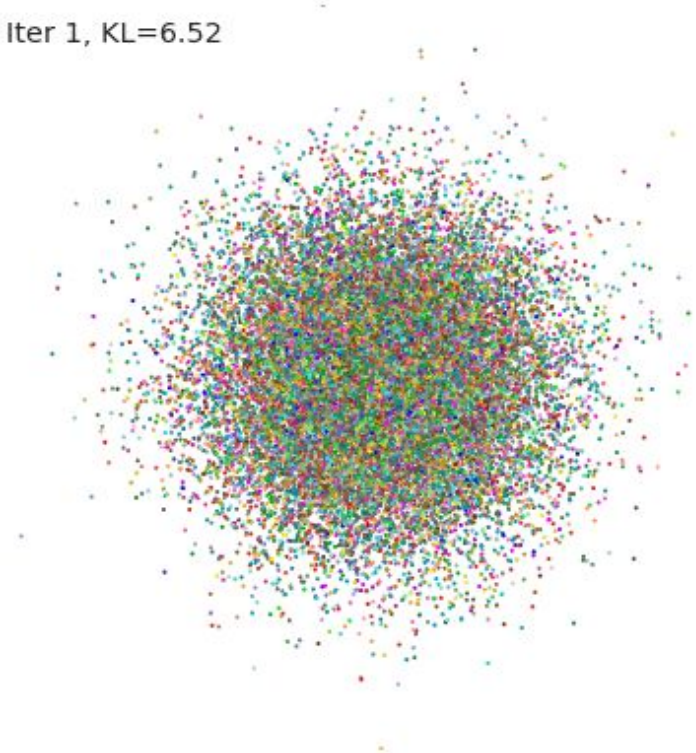
$$\eta = n \,/\, \text{exagg.}$$

Belkina et al. (2019) based on Linderman & Steinerberger (2019)



Early exaggeration: scRNA-seq data

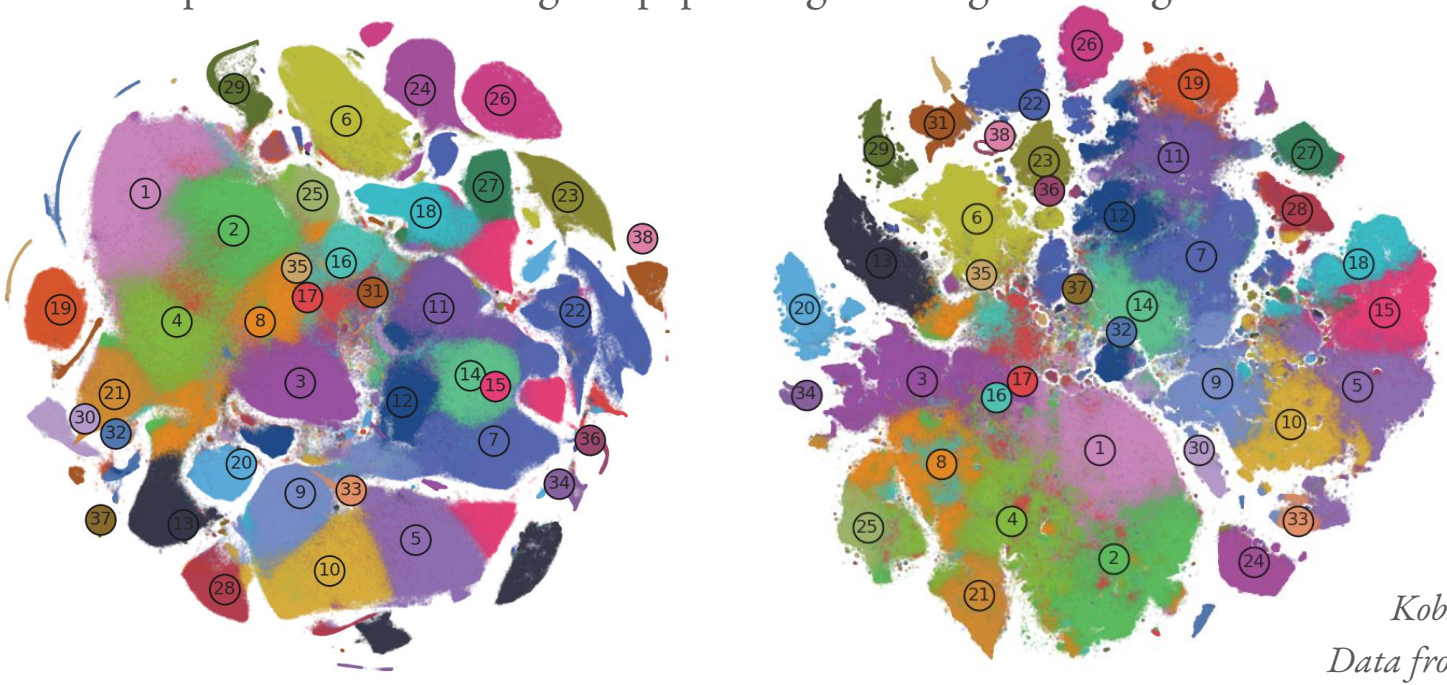




Data from Tasic et al. (2018)

Early exaggeration and the learning rate: real-world example

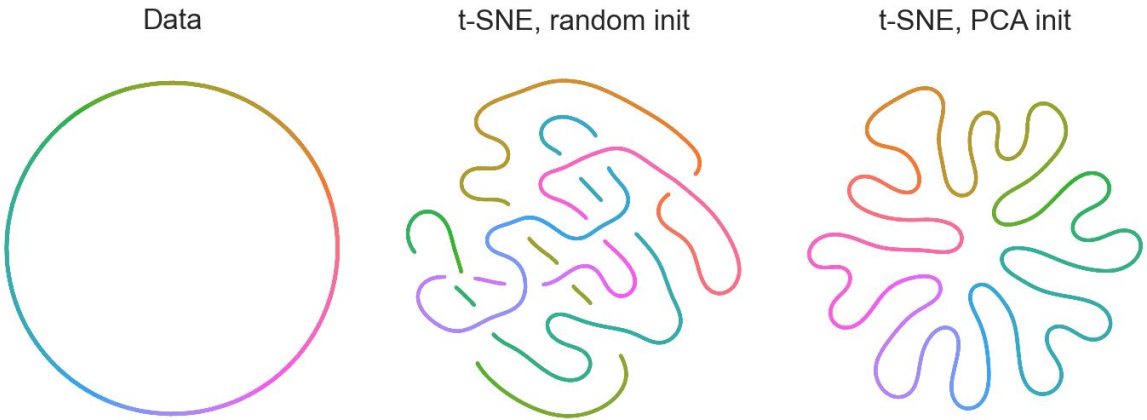
Single-cell transcriptomic study of mouse embryogenesis ($n \approx 2{,}000{,}000$).
Left — t-SNE published in the original paper. Right — high learning rate.



Kobak & Berens (2019)
Data from Cao et al. (2019)

Initialization

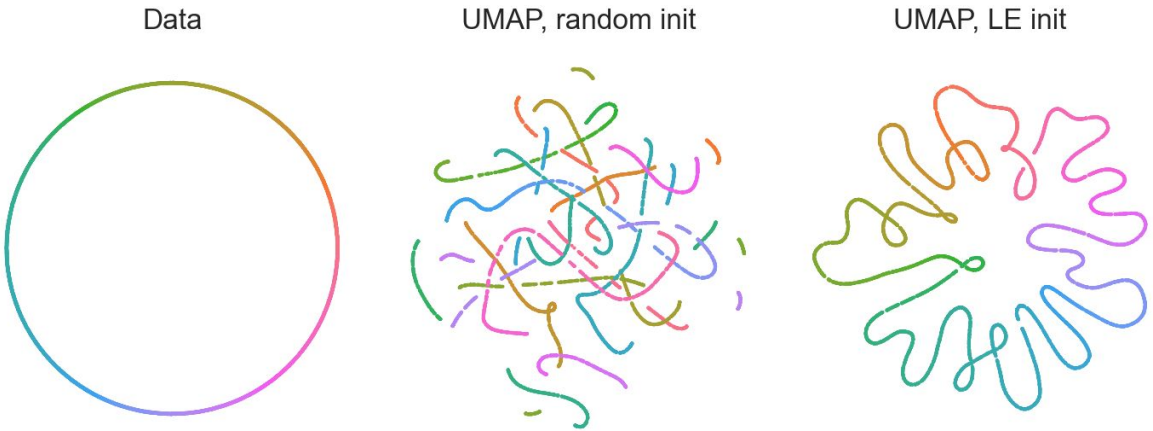
The loss function has many local minima. Even with early exaggeration, initialization can play a big role. With random initialization, t-SNE will often struggle to preserve global structure.



Kobak & Linderman, Nature Biotech 2021

Initialization

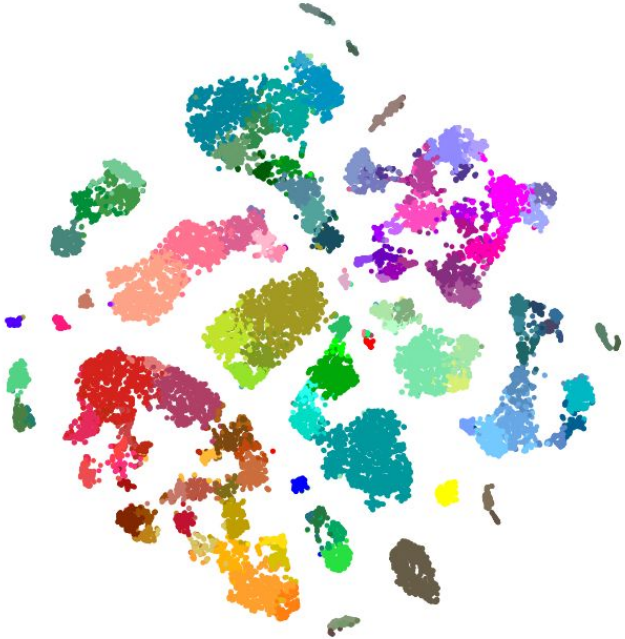
The loss function has many local minima. Even with early exaggeration, initialization can play a big role. With random initialization, t-SNE will often struggle to preserve global structure.



Kobak & Linderman, Nature Biotech 2021

Initialization: real-world example

t-SNE (random initialization)



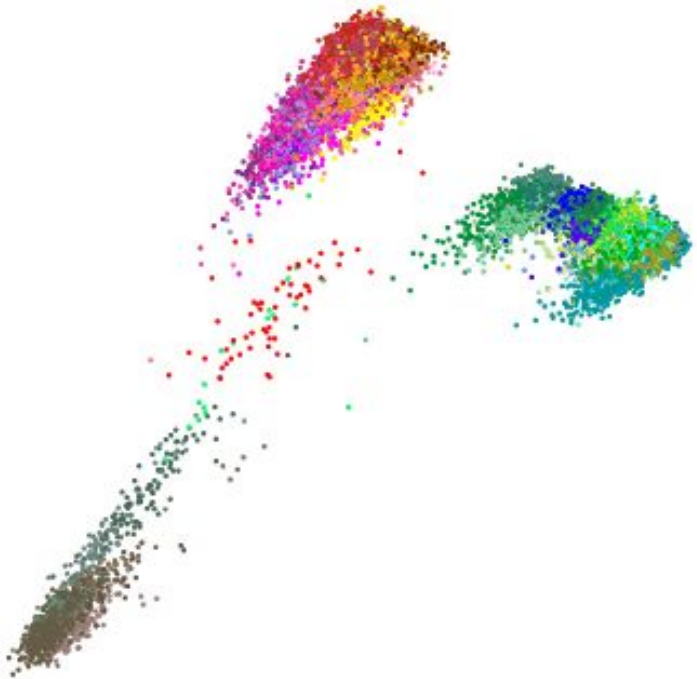
PCA

Single-cell RNA-seq data from mouse cortex

Kobak & Berens, Nature Comms 2019)

Data from Tasic et al. (2018)

Initialization:
real-world example



Single-cell RNA-seq data from mouse cortex

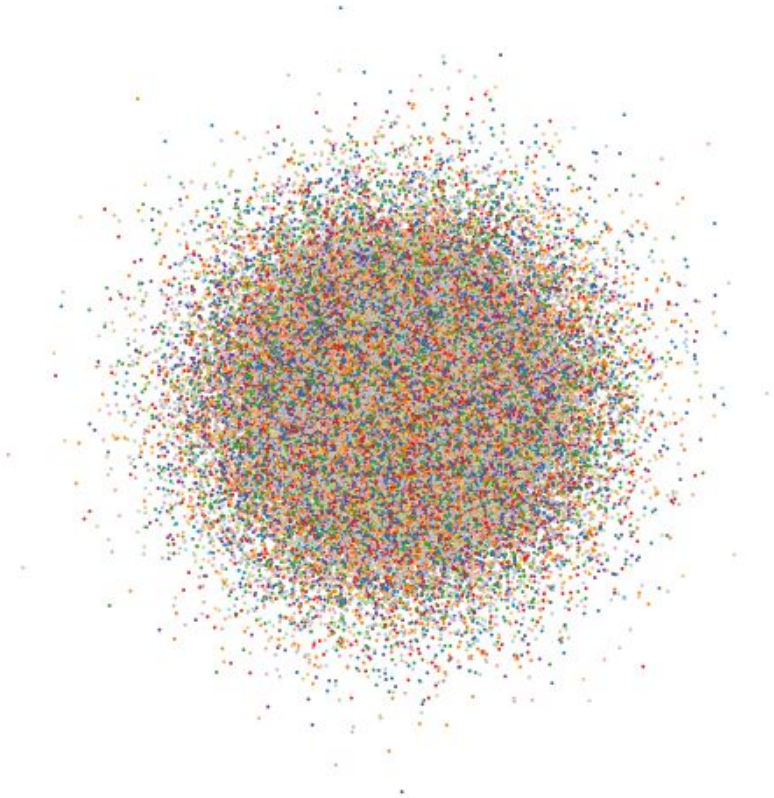
Kobak & Berens, Nature Comms 2019)
Data from Tasic et al. (2018)

Approximations

$$-\underbrace{\sum_{ij} p_{ij} \log w_{ij}}_{\text{attraction}} + \underbrace{\log \sum_{ij} w_{ij}}_{\text{repulsion}}$$

Approximate kNN graph with small k

Approximations:
* Barnes–Hut
* Fourier interpolation
* Sampling



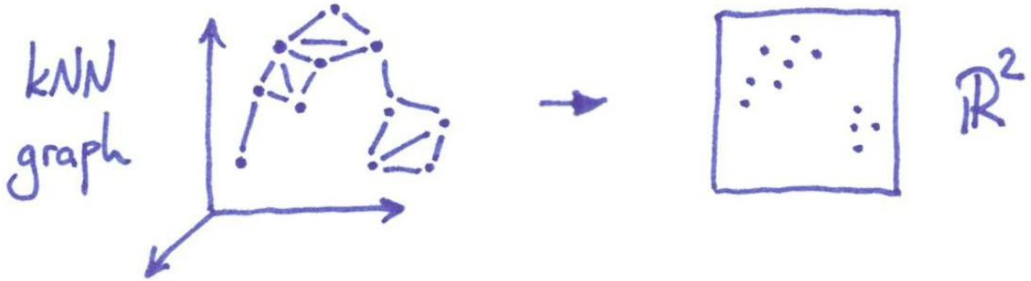
Optimization issues

A good t-SNE implementation should:

1. use informative initialization by default;
2. set appropriate optimization parameters (learning rate set to n);
3. support approximate nearest neighbors and $O(n)$ approximation of repulsive forces.

=> In Python, use `openTSNE` *(Poličar et al., 2024)*.

t-SNE

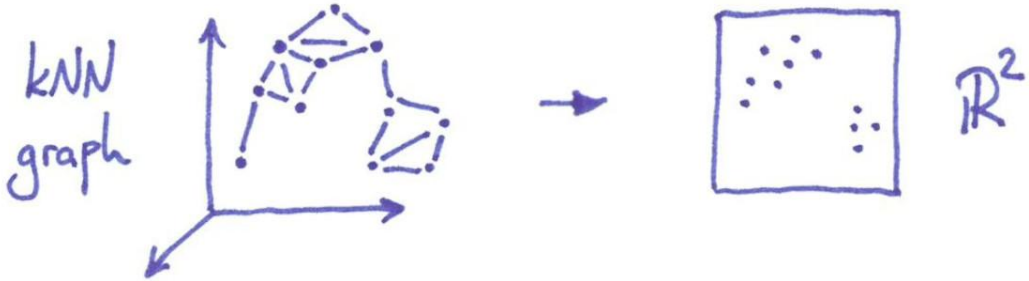


$$\mathcal{L}_{\text{t-SNE}} = \sum_{ij} p_{ij} \log \frac{p_{ij}}{q_{ij}} \sim -\sum_{ij} p_{ij} \log q_{ij} = -\sum_{ij} p_{ij} \log \frac{w_{ij}}{Z} = \underbrace{-\sum_{ij} p_{ij} \log w_{ij}}_{\text{attraction}} + \underbrace{\log \sum_{ij} w_{ij}}_{\text{repulsion}}.$$

p_{ij} can be normalized symmetrized adjacency matrix of the kNN graph.

$$q_{ij} = \frac{w_{ij}}{Z} = \frac{w_{ij}}{\sum w_{ij}} \text{ with } w_{ij} = \frac{1}{1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2}.$$

t-SNE



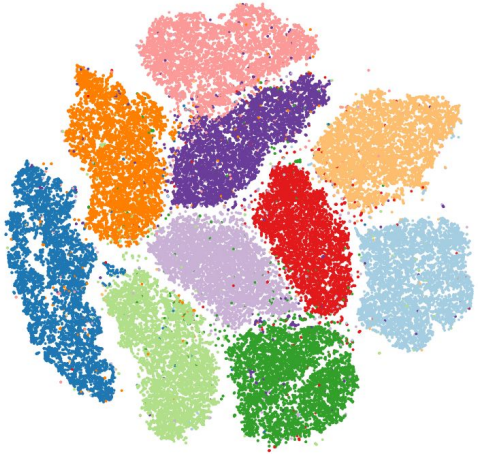
$$\mathcal{L}_{\text{t-SNE}} = \sum_{ij} p_{ij} \log \frac{p_{ij}}{q_{ij}} \sim -\sum_{ij} p_{ij} \log q_{ij} = -\sum_{ij} p_{ij} \log \frac{w_{ij}}{Z} = \underbrace{-\sum_{ij} p_{ij} \log w_{ij}}_{\text{attraction}} + \underbrace{\log \sum_{ij} w_{ij}}_{\text{repulsion}}.$$

p_{ij} can be normalized symmetrized adjacency matrix of the kNN graph.

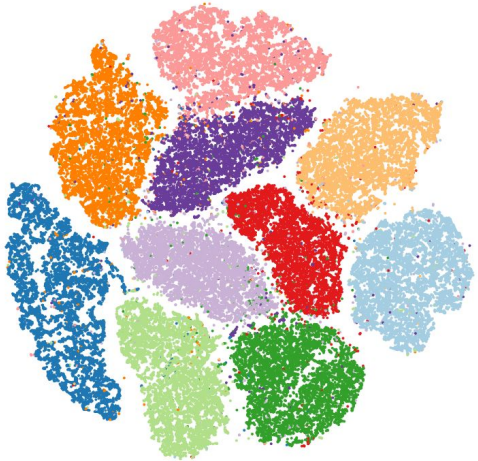
Usually high-dim affinities are obtained with a Gaussian kernel via *perplexity* parameter

$$q_{ij} = \frac{w_{ij}}{Z} = \frac{w_{ij}}{\sum w_{ij}} \text{ with } w_{ij} = \frac{1}{1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2}.$$

t-SNE with perplexity 30

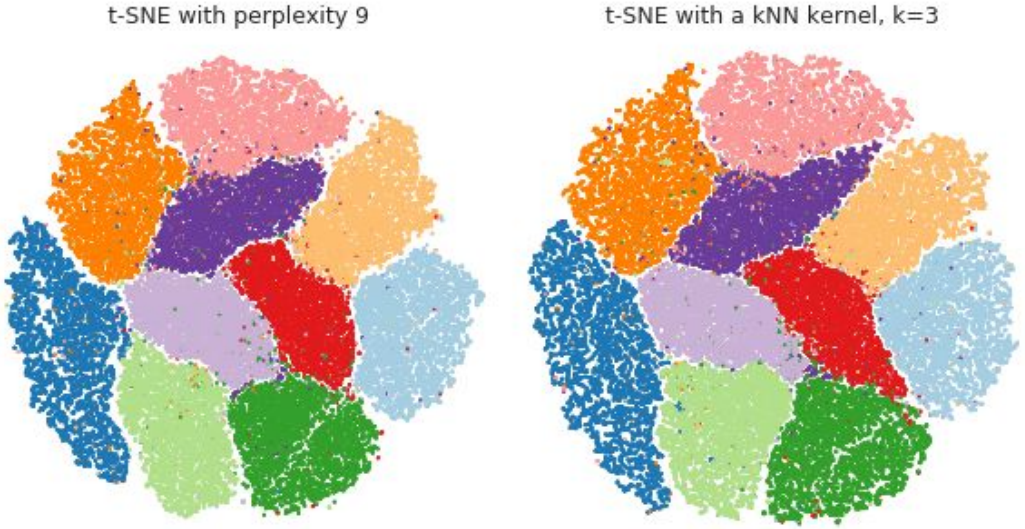


t-SNE with a kNN kernel (k=10)



p_{ij} can be normalized symmetrized adjacency matrix of the kNN graph.

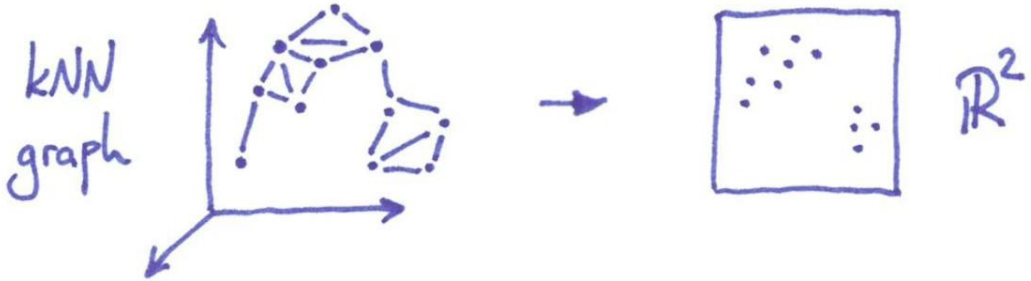
Usually high-dim affinities are obtained with a Gaussian kernel via *perplexity* parameter



p_{ij} can be normalized symmetrized adjacency matrix of the kNN graph.

Usually high-dim affinities are obtained with a Gaussian kernel via *perplexity* parameter

t-SNE

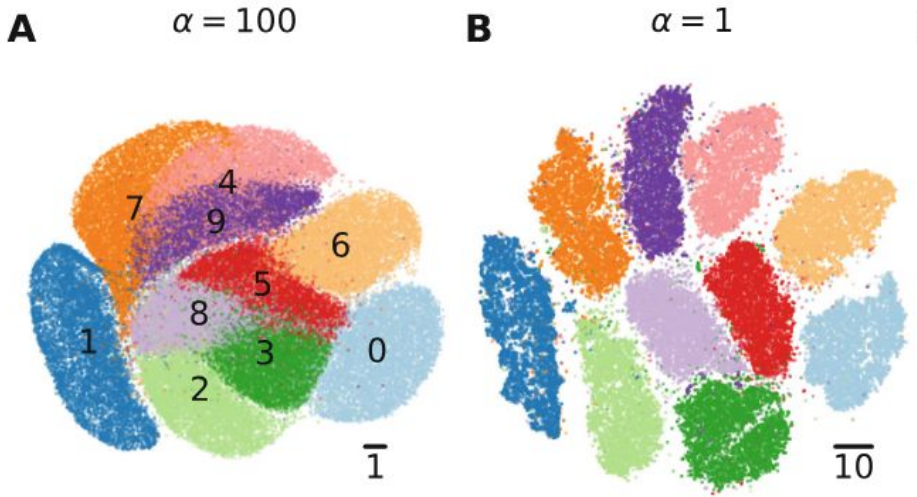


$$\mathcal{L}_{\text{t-SNE}} = \sum_{ij} p_{ij} \log \frac{p_{ij}}{q_{ij}} \sim -\sum_{ij} p_{ij} \log q_{ij} = -\sum_{ij} p_{ij} \log \frac{w_{ij}}{Z} = \underbrace{-\sum_{ij} p_{ij} \log w_{ij}}_{\text{attraction}} + \underbrace{\log \sum_{ij} w_{ij}}_{\text{repulsion}}.$$

p_{ij} can be normalized symmetrized adjacency matrix of the kNN graph.

$$q_{ij} = \frac{w_{ij}}{Z} = \frac{w_{ij}}{\sum w_{ij}} \text{ with } w_{ij} = \frac{1}{1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2}.$$

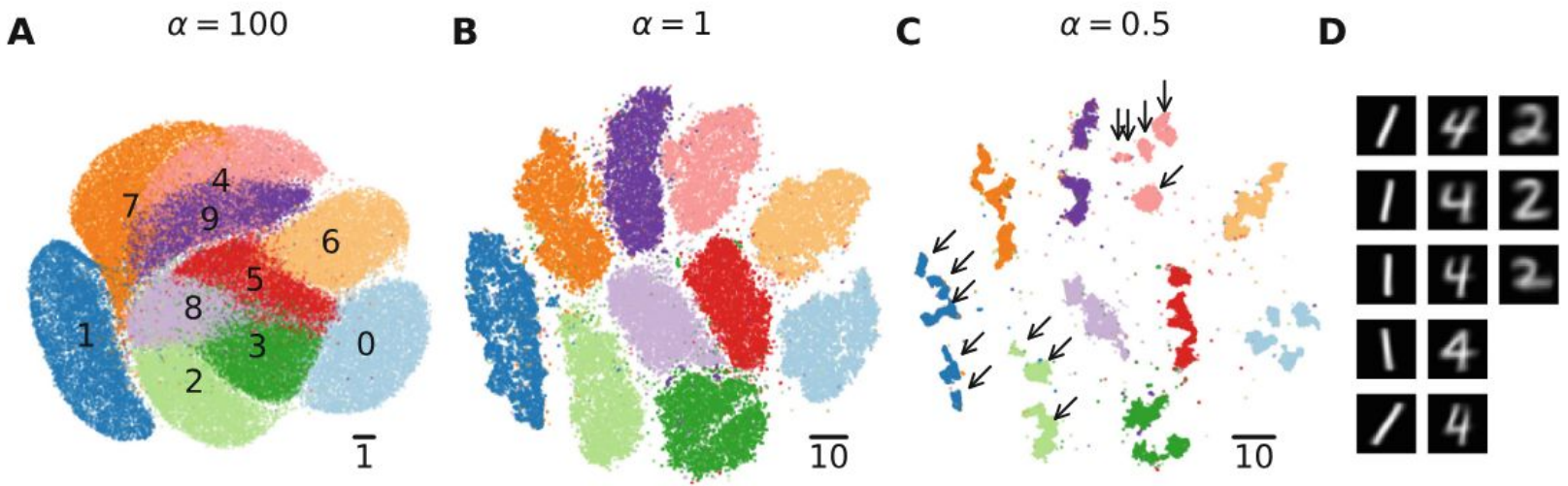
The *t*-distribution kernel: from SNE to *t*-SNE



$$k(d) = \frac{1}{(1 + d^2/\alpha)^\alpha}$$

Kobak et al., ECML 2019

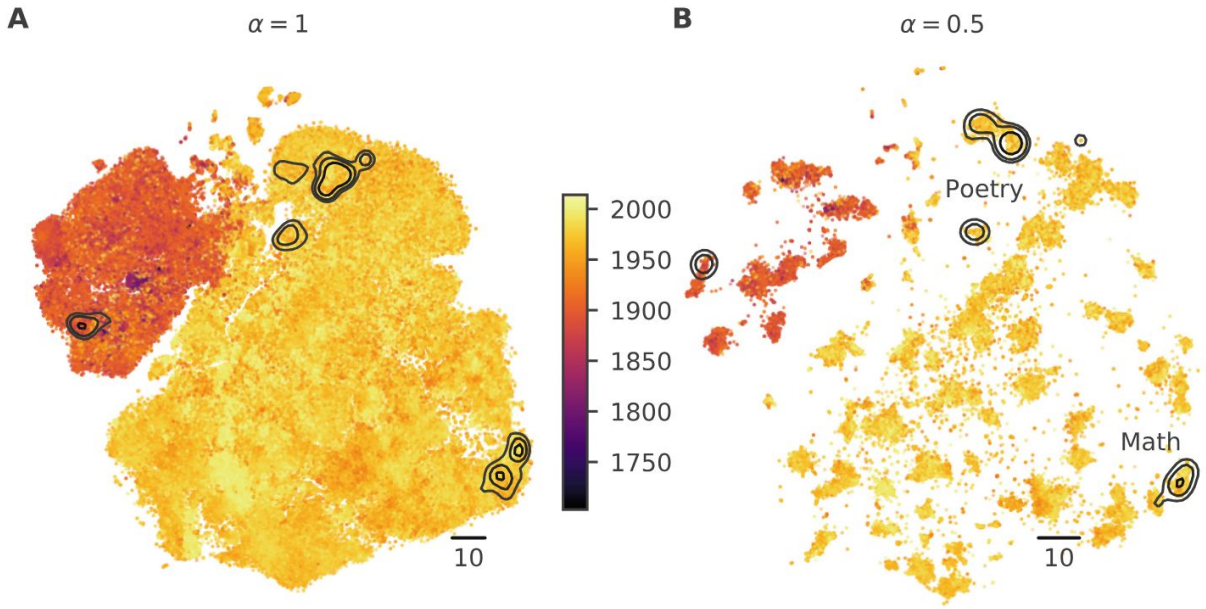
The *t*-distribution kernel: beyond *t*-SNE



$$k(d) = \frac{1}{(1 + d^2/\alpha)^\alpha}$$

Kobak et al., ECML 2019

The *t*-distribution kernel: beyond *t*-SNE

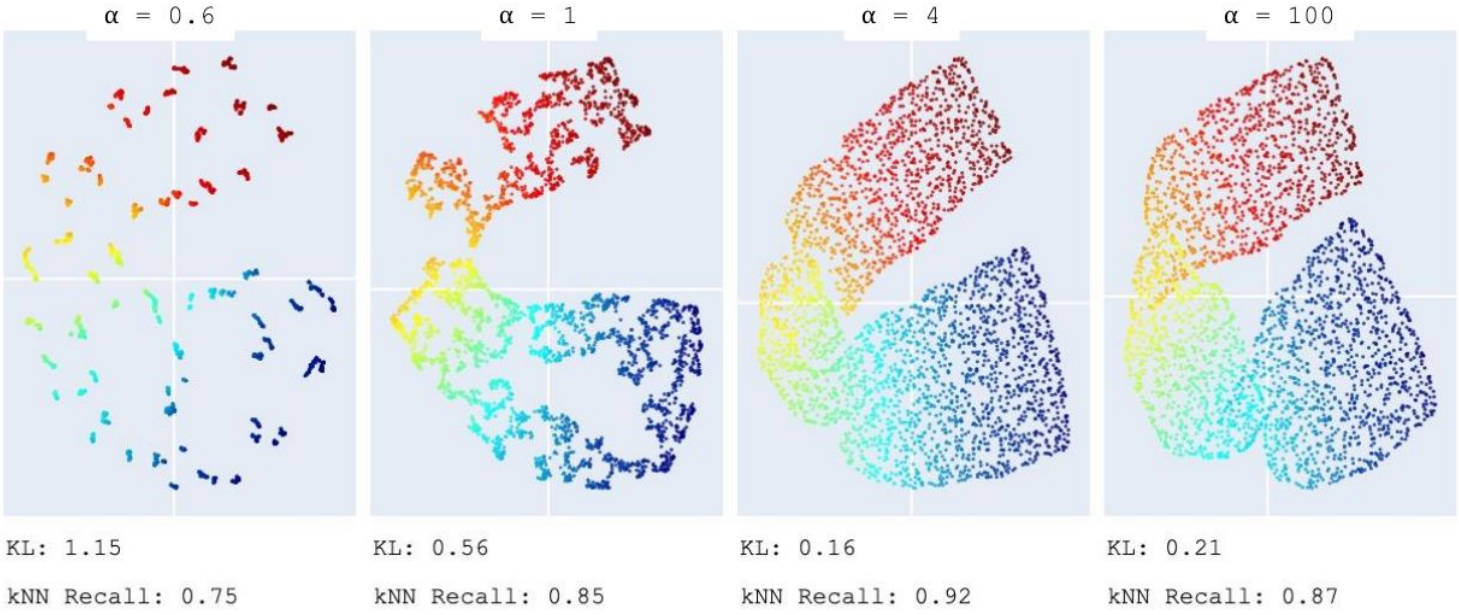


Russian language part of the HathiTrust library ($n = 408\,291$).

Kobak et al., ECML 2019

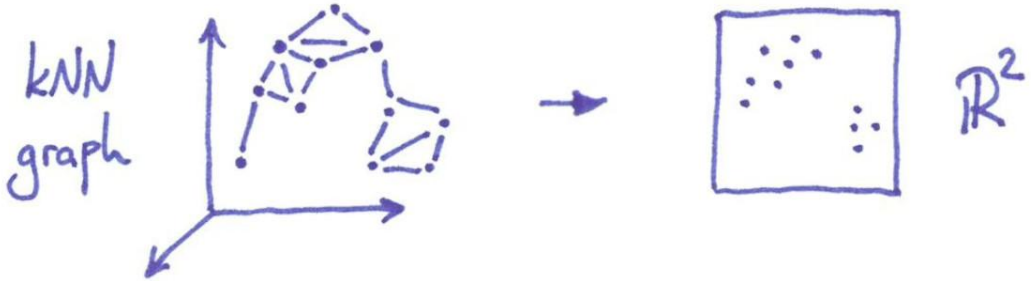
The *t*-distribution kernel: optimizing the α

$$\frac{\partial L}{\partial \alpha} = \sum_{i,j} (p_{i,j} - q_{i,j}) \left(\log \left(1 + \frac{d_{i,j}^2}{\alpha} \right) - \frac{d_{i,j}^2}{\alpha + d_{i,j}^2} \right)$$



Artemii Schlychkov, work in progress

t-SNE



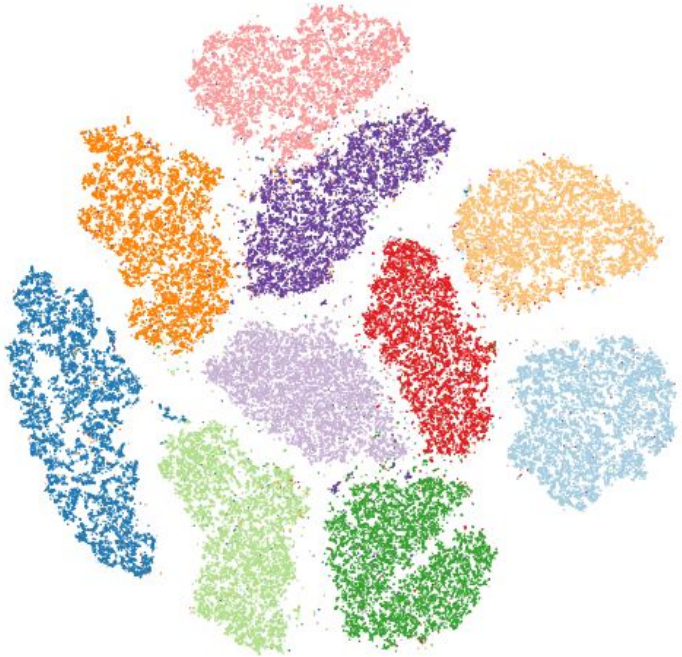
$$\mathcal{L}_{\text{t-SNE}} = \sum_{ij} p_{ij} \log \frac{p_{ij}}{q_{ij}} \sim -\sum_{ij} p_{ij} \log q_{ij} = -\sum_{ij} p_{ij} \log \frac{w_{ij}}{Z} = \underbrace{-\sum_{ij} p_{ij} \log w_{ij}}_{\text{attraction}} + \underbrace{\log \sum_{ij} w_{ij}}_{\text{repulsion}}.$$

p_{ij} can be normalized symmetrized adjacency matrix of the kNN graph.

$$q_{ij} = \frac{w_{ij}}{Z} = \frac{w_{ij}}{\sum w_{ij}} \text{ with } w_{ij} = \frac{1}{1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2}.$$

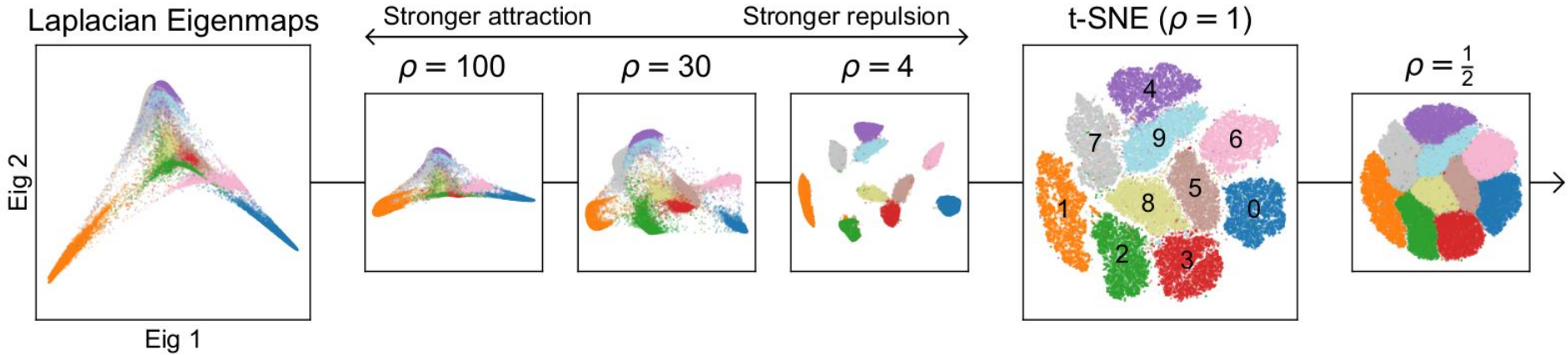
Attraction-repulsion spectrum

Niklas Böhm



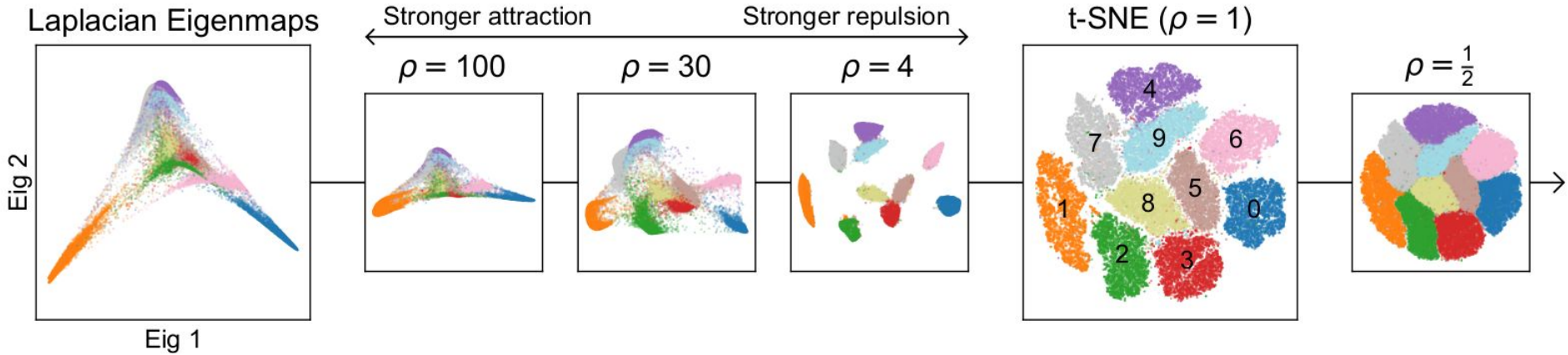
Böhm et al., JMLR 2022

Attraction-repulsion spectrum



Böhm et al., JMLR 2022

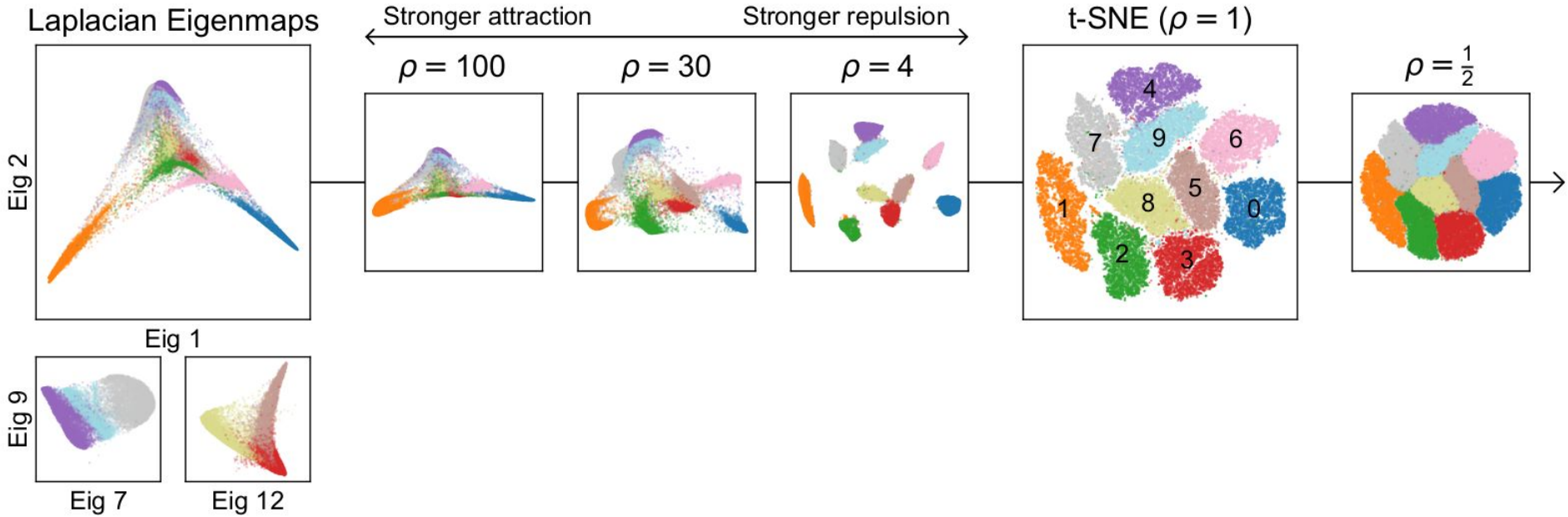
Attraction-repulsion spectrum



For very strong exaggeration, t-SNE approximates Laplacian Eigenmaps of the affinity matrix *(Linderman & Steinerberger, 2020).*

Böhm et al., JMLR 2022

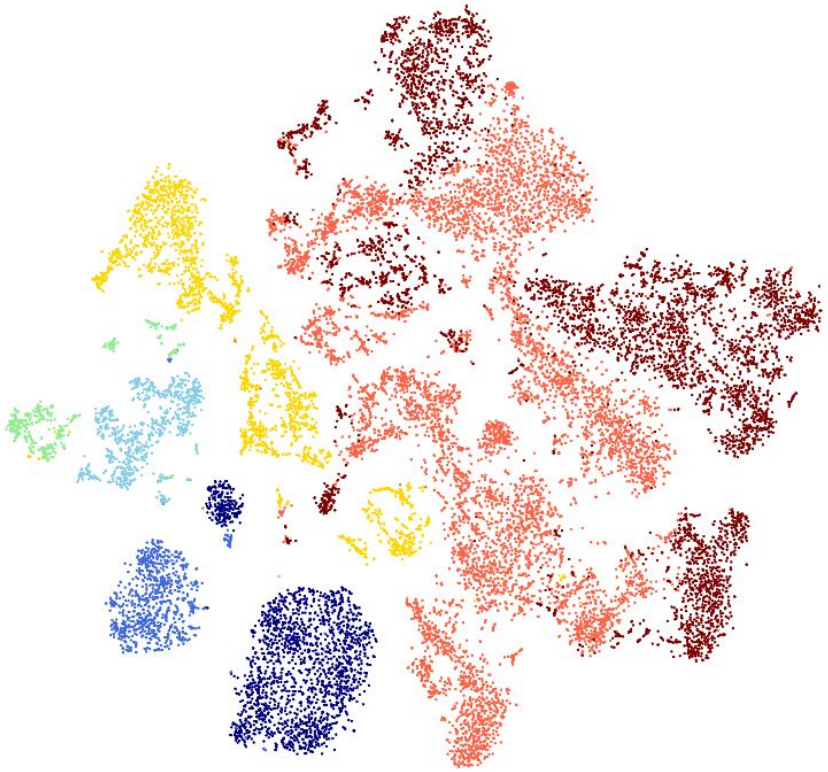
Attraction-repulsion spectrum



The ‘data-agnostic’ repulsion term brings our information from higher Laplacian eigenvectors.

Attraction-repulsion spectrum

Niklas Böhm

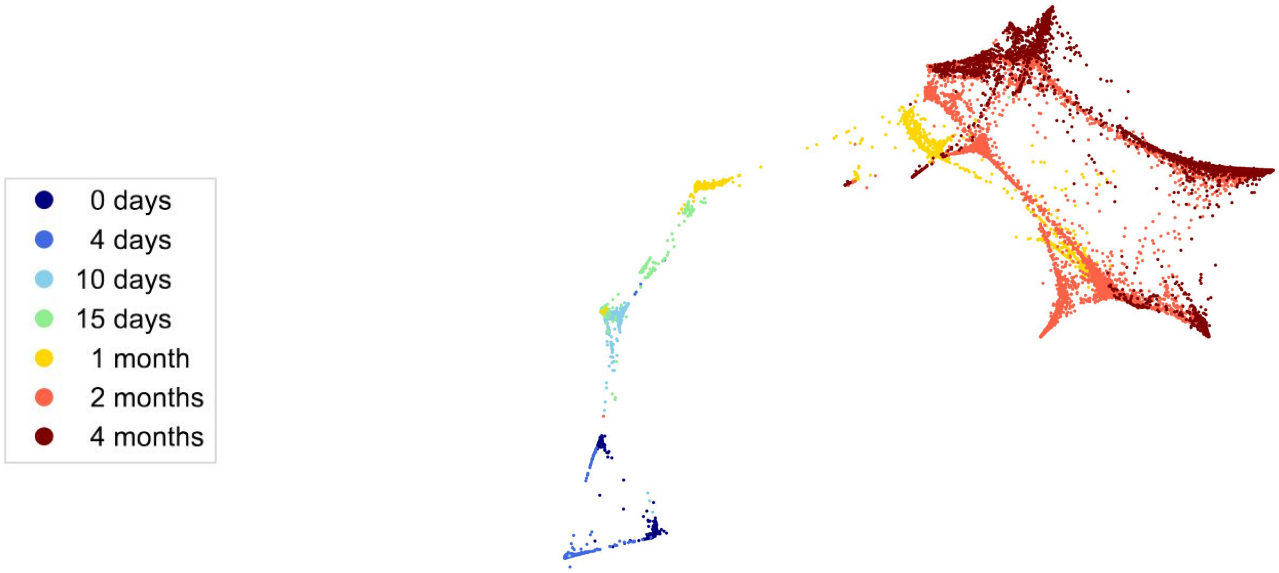


Brain organoid scRNA-seq data
from *Kanton et al., Nature 2019*

Böhm et al., JMLR 2022

Attraction-repulsion spectrum

Niklas Böhm

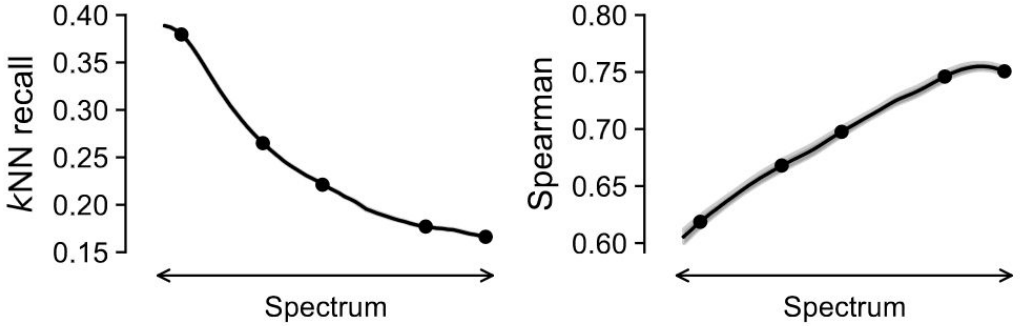


Brain organoid scRNA-seq data
from *Kanton et al., Nature 2019*

Böhm et al., JMLR 2022

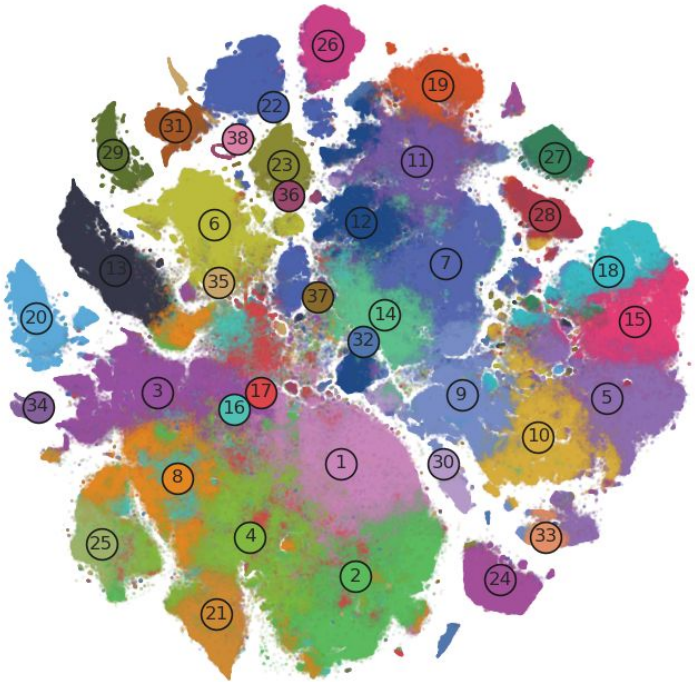
Local-global trade-off

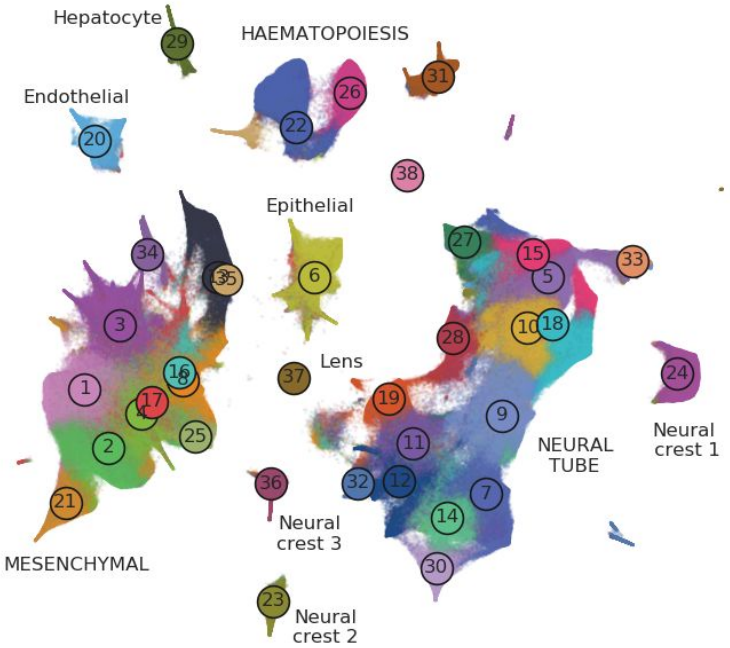
Sebastian Damrich



Damrich et al., bioRxiv 2023

Local-global trade-off (n = 2 mln)



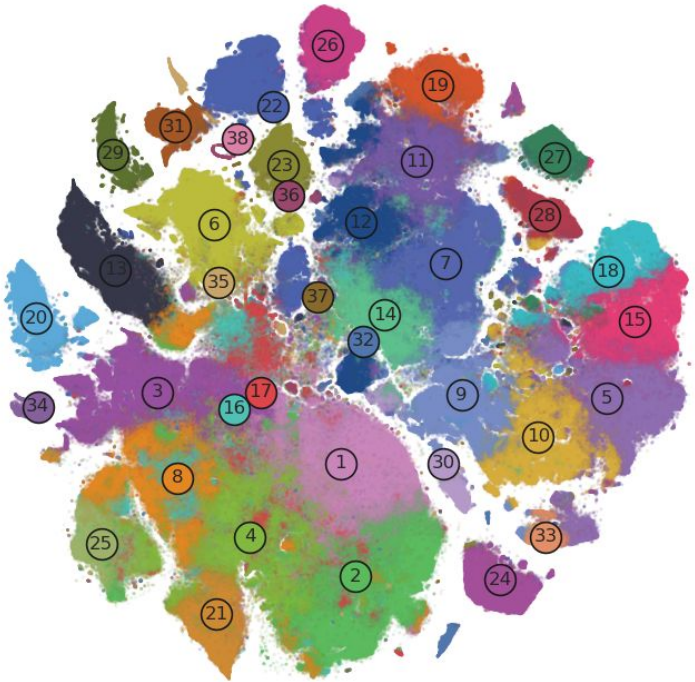


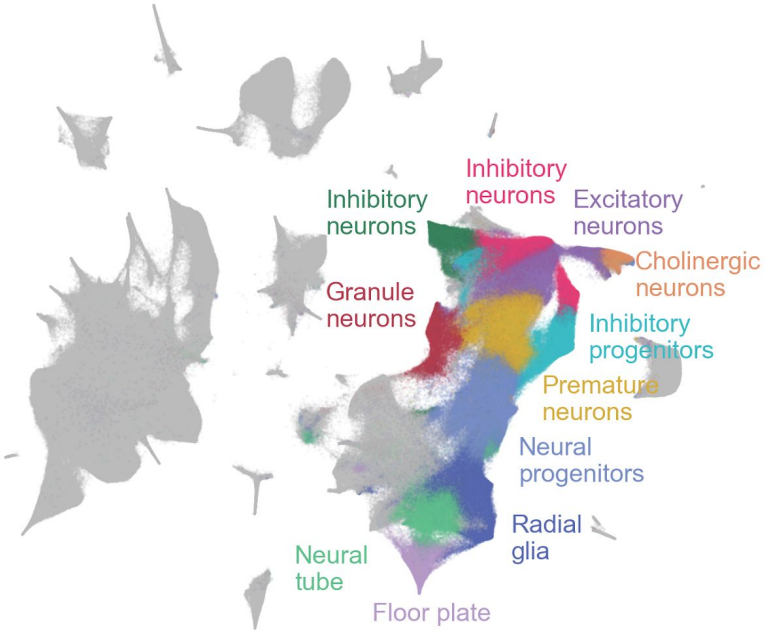
Data from

Cao et al., Nature 2019

Kobak & Berens, Nature Comms 2019

Local-global trade-off (n = 2 mln)





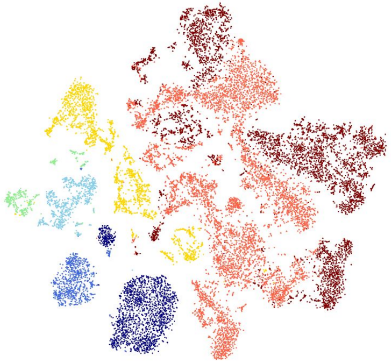
Data from

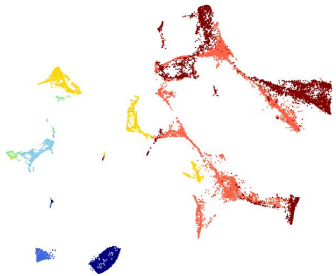
Cao et al., Nature 2019

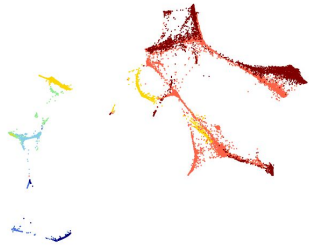
Kobak & Berens, Nature Comms 2019

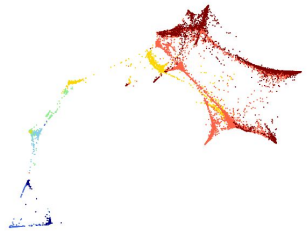
UMAP is equivalent to *t*-SNE with increased attraction

Sebastian Damrich







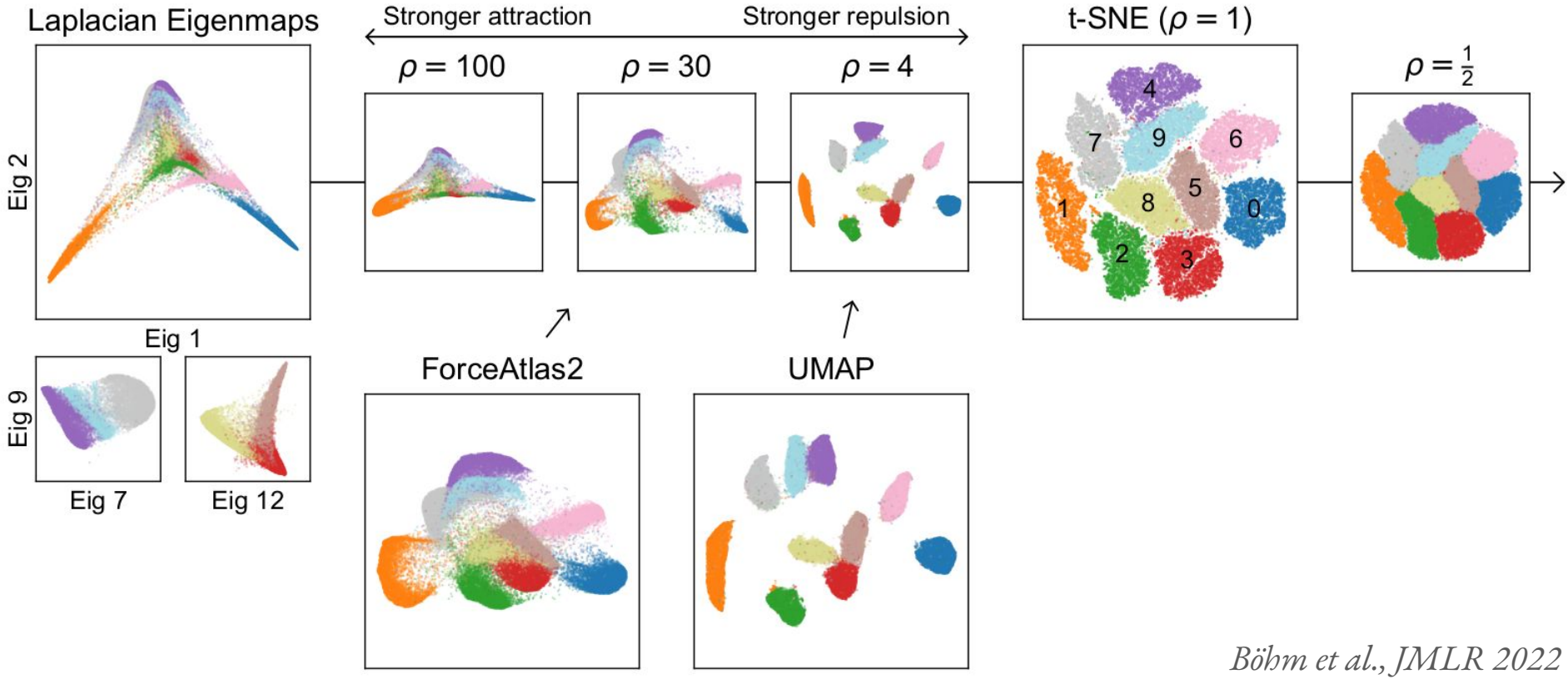


UMAP

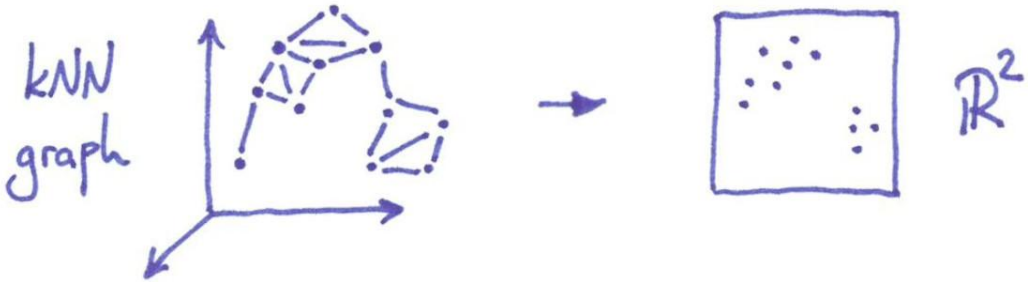
ForceAtlas2

Damrich et al., ICLR 2023

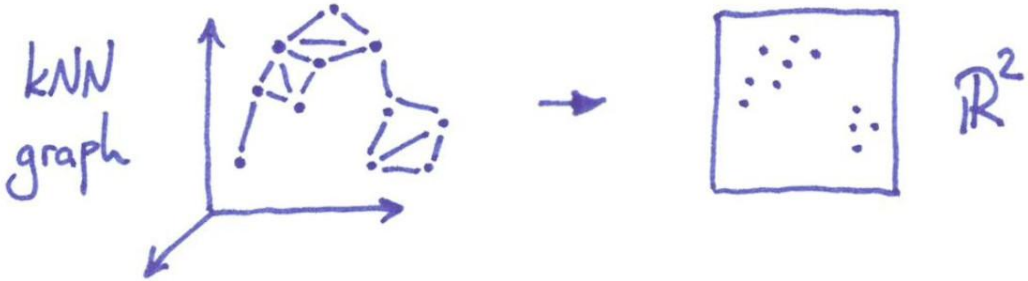
Other neighbour embedding algorithms



Böhm et al., JMLR 2022



1. What if my graph is not a *k*NN graph?
2. What if I want higher embedding dimensionality?



1. What if my graph is not a *k*NN graph?
2. What if I want higher embedding dimensionality?

$$-\sum_{ij} p_{ij} \log \frac{w_{ij}}{Z} \qquad -\sum_{ij} p_{ij} \log \left(\frac{w_{ij}}{w_{ij} + \sum_{k \in \mathcal{I}_m} w_{ik}} \right)$$

Sebastian Damrich

Damrich et al., ICLR 2023

InfoNC-t-SNE: sampling-based approximation

Can be optimized by SGD in batches for any output dimensionality.

$$\mathcal{L}_{\text{InfoNCE}} = -\sum_{ij} p_{ij} \log \left(\frac{w_{ij}}{w_{ij} + \sum_{k \in \mathcal{I}_m} w_{ik}} \right).$$

Damrich et al., ICLR 2023

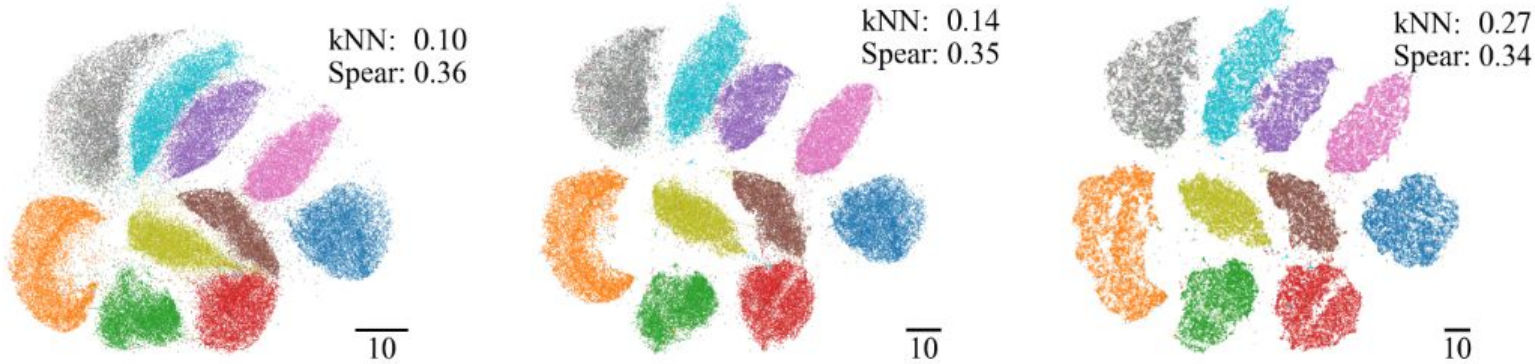
InfoNC-t-SNE

$$\mathcal{L}_{\text{InfoNCE}} = -\sum_{ij} p_{ij} \log \left(\frac{w_{ij}}{w_{ij} + \sum_{k \in \mathcal{I}_m} w_{ik}} \right).$$

Damrich et al., ICLR 2023

InfoNC-t-SNE

$$\mathcal{L}_{\text{InfoNCE}} = -\sum_{ij} p_{ij} \log \left(\frac{w_{ij}}{w_{ij} + \sum_{k \in \mathcal{I}_m} w_{ik}} \right).$$



(i) $m = 5$ (j) $m = 50$ (k) $m = 500$

Damrich et al., ICLR 2023

Parametric InfoNC-t-SNE

$$\mathcal{L}_{\text{InfoNCE}} = -\sum_{ij} p_{ij} \log \left(\frac{w_{ij}}{w_{ij} + \sum_{k \in \mathcal{I}_m} w_{ik}} \right).$$

Can optimize a parametric mapping (e.g. MLP) with the same loss!

(c) InfoNC-t-SNE

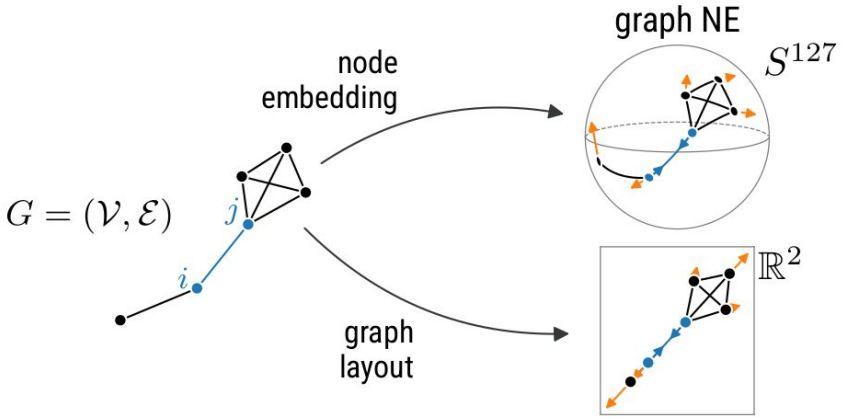
(f) Param. InfoNC-t-SNE

Damrich et al., ICLR 2023

Graph layouts & node embeddings via NE algorithms

Niklas Böhm

Marius Keute



InfoNCE

$$\ell_{ij} = -\log \frac{\exp(\mathbf{y}_i^\top \mathbf{y}_j/\tau)}{\sum_k \exp(\mathbf{y}_i^\top \mathbf{y}_k/\tau)}$$

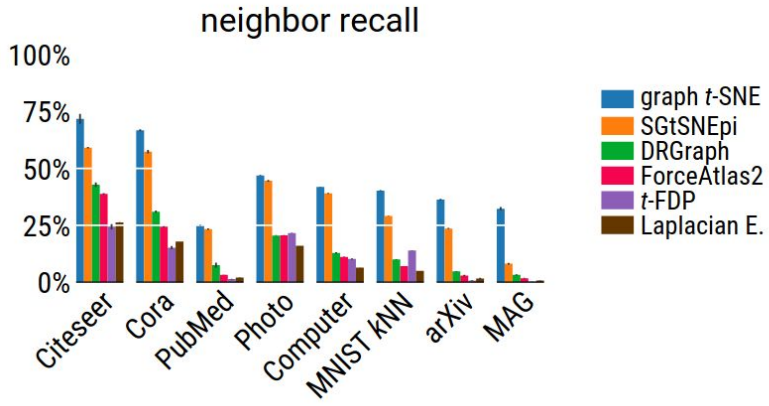
KL divergence

$$\ell_{ij} = -\log \frac{(1+\|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_{kl}(1+\|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}}$$

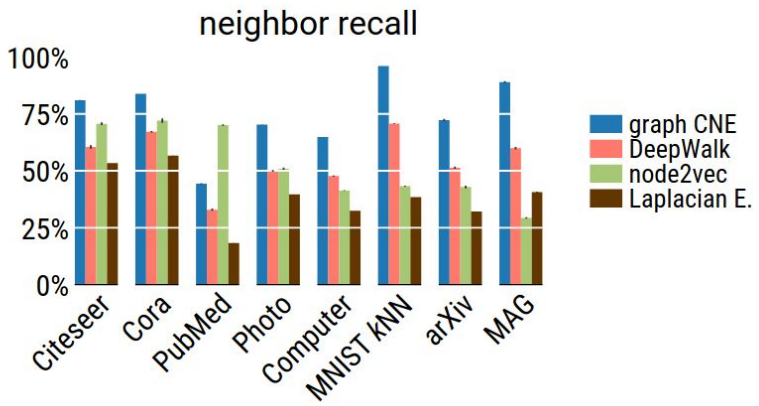
Böhm & Keute et al., TMLR 2025

Graph layouts & node embeddings via NE algorithms

2D embeddings



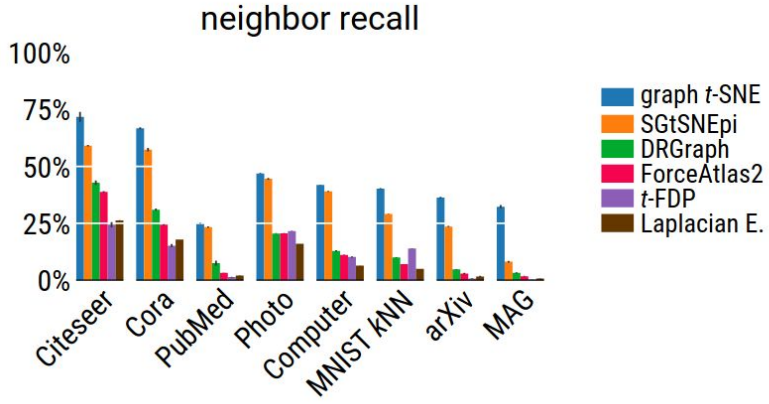
128D embeddings

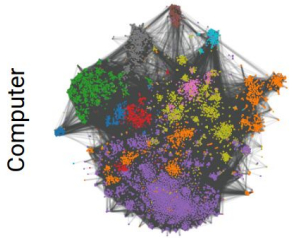


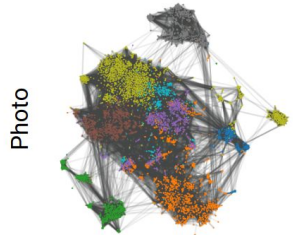
Böhm & Keute et al., TMLR 2025

Graph layouts & node embeddings via NE algorithms

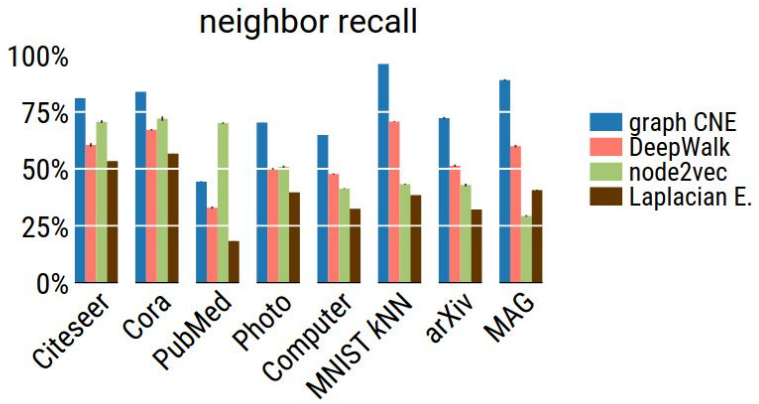
2D embeddings





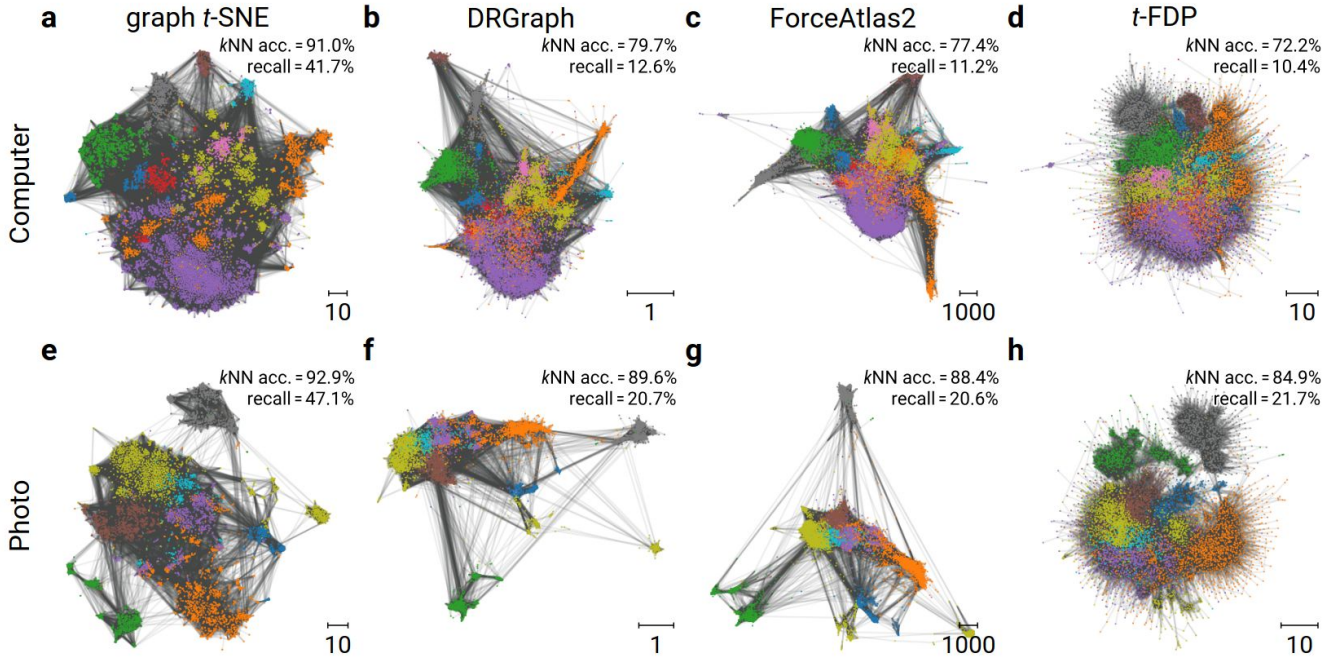


128D embeddings



Böhm & Keute et al., TMLR 2025

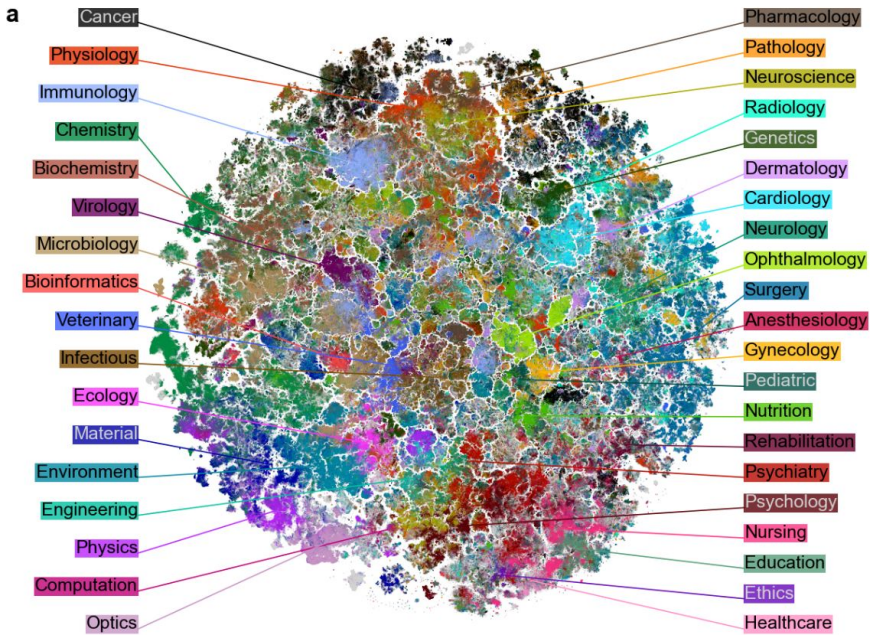
Graph layouts & node embeddings via NE algorithms

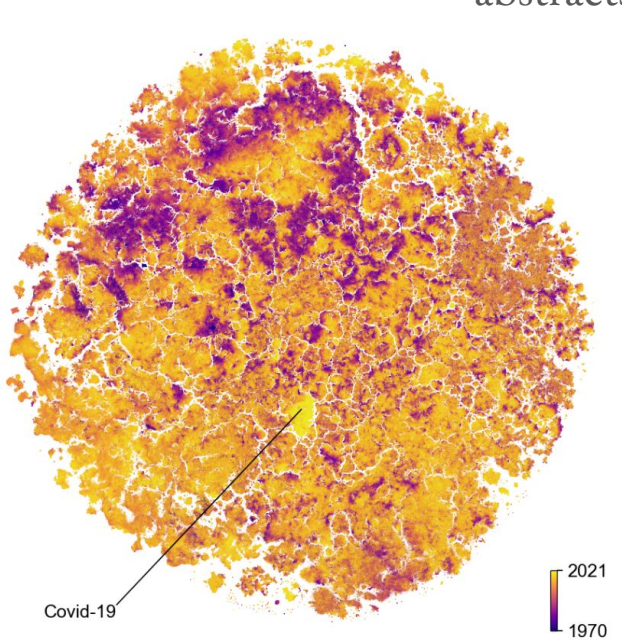


Böhm & Keute et al., TMLR 2025

LLM-based embeddings of text data

All 21 million English abstracts from PubMed



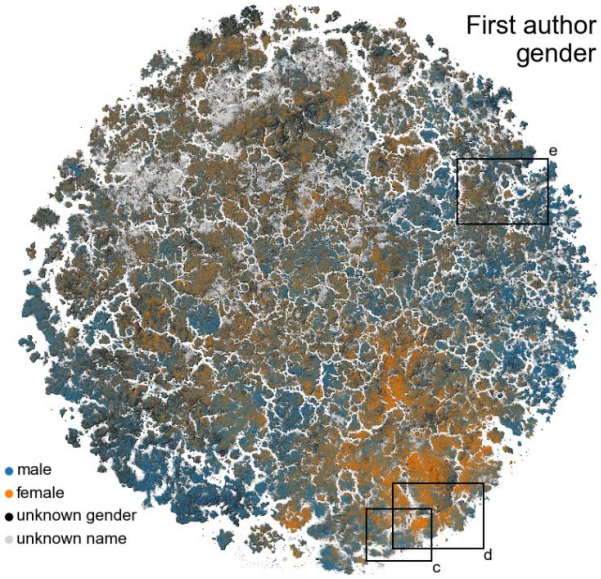


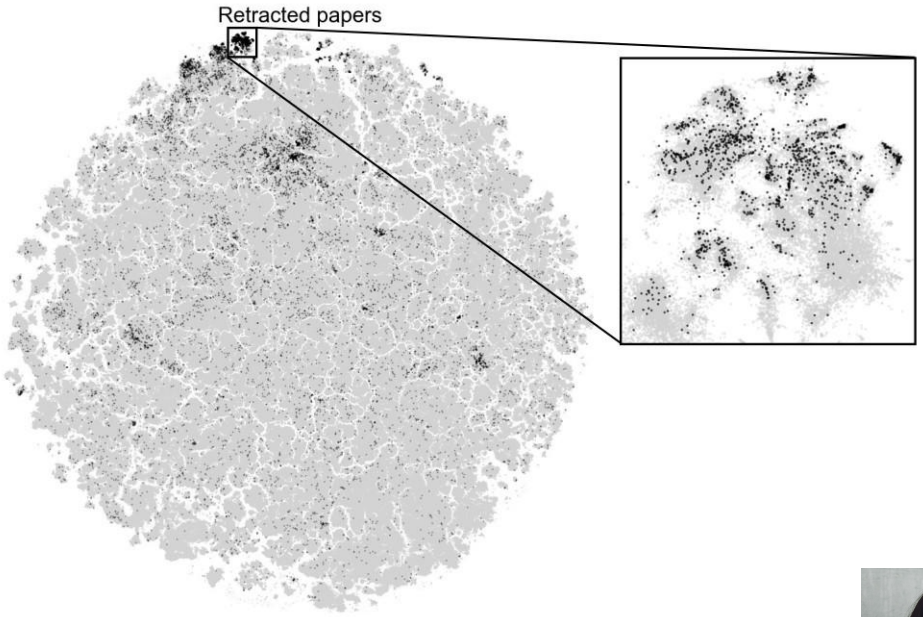
https://static.nomic.ai/pubmed.html

González-Márquez et al., Patterns 2024

Rita González Márquez

LLM-based embeddings of text data

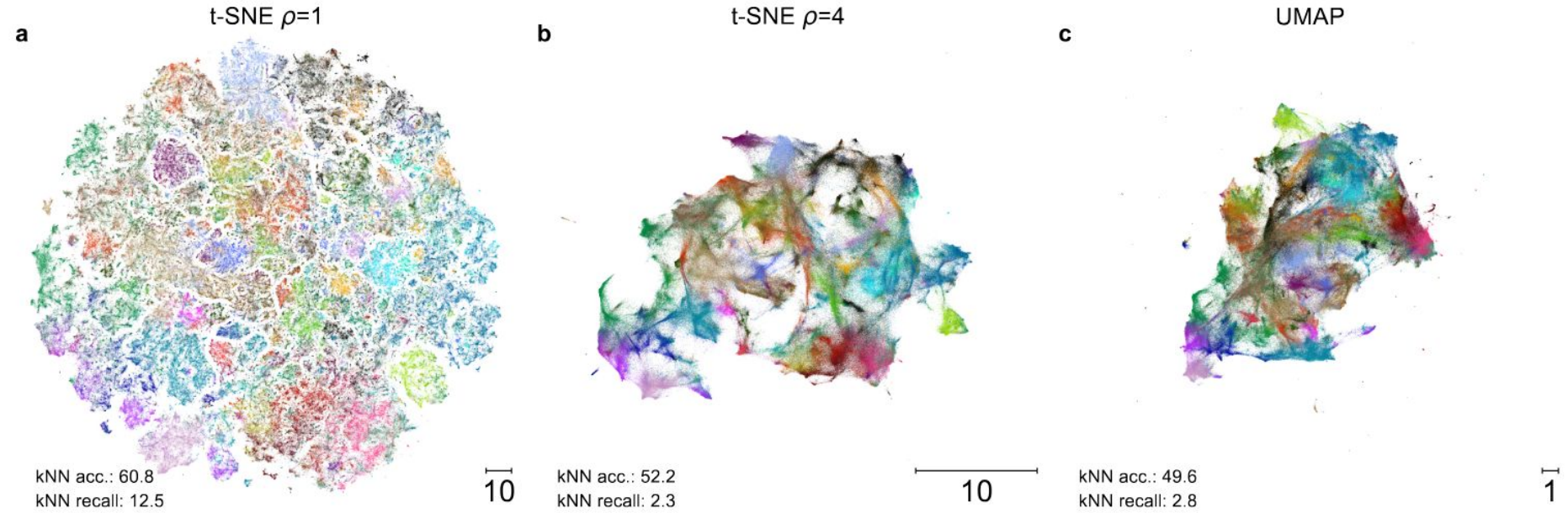




González-Márquez et al., Patterns 2024

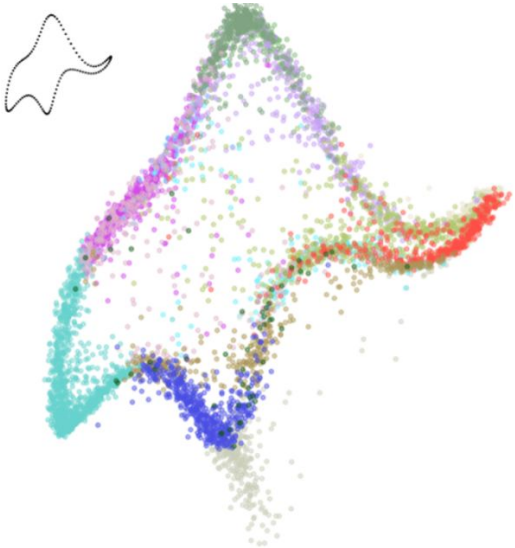
Rita González Márquez

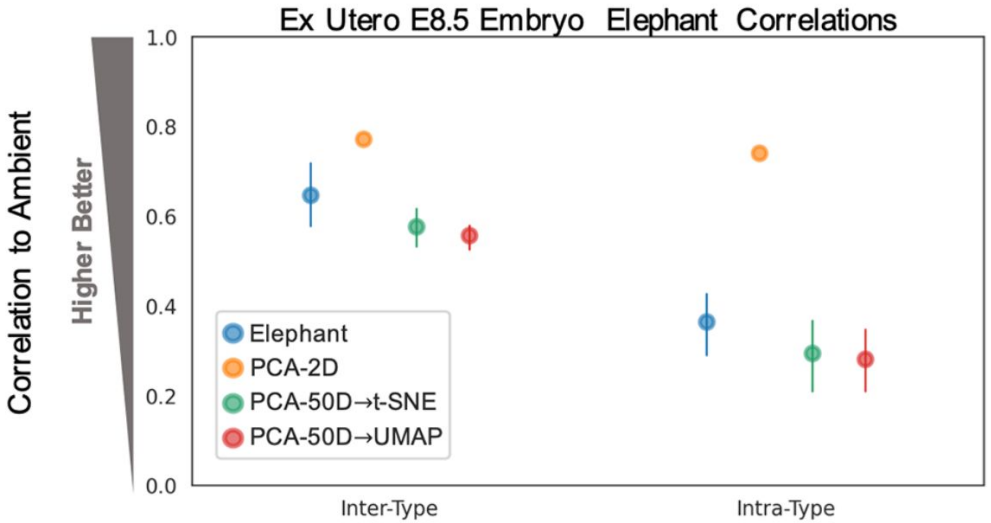
Attraction-repulsion spectrum (*t*-SNE/UMAP) for PubMed



Embeddings of a subset of the PubMed dataset using different neighbor embedding methods. Subset size: 1,000,000 labeled papers.

“All embeddings are wrong …”

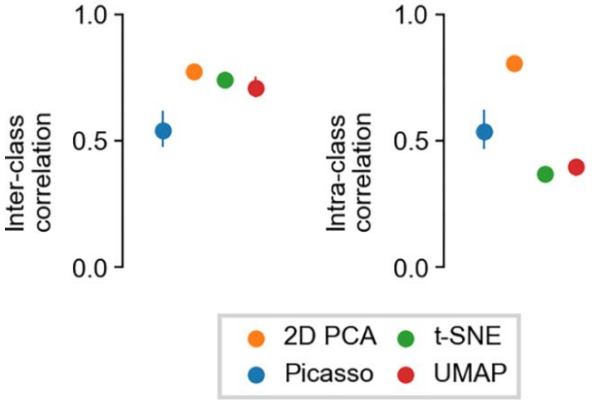




Chari & Pachter, PLoS Comp Bio 2023

“... but some are useful”

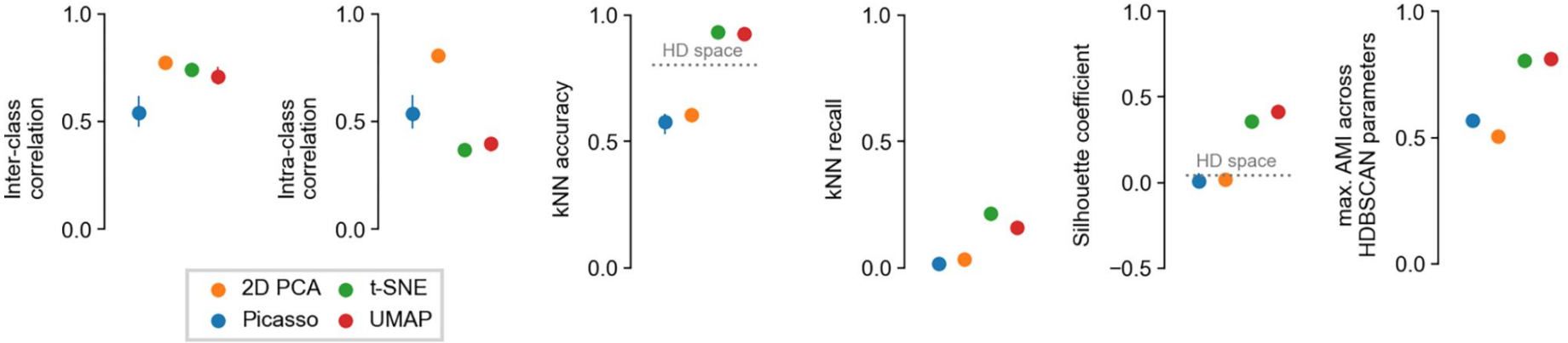
Jan Lause



Lause et al., PLoS Comp Bio 2024

“... but some are useful”

Jan Lause



Lause et al., PLoS Comp Bio 2024