EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

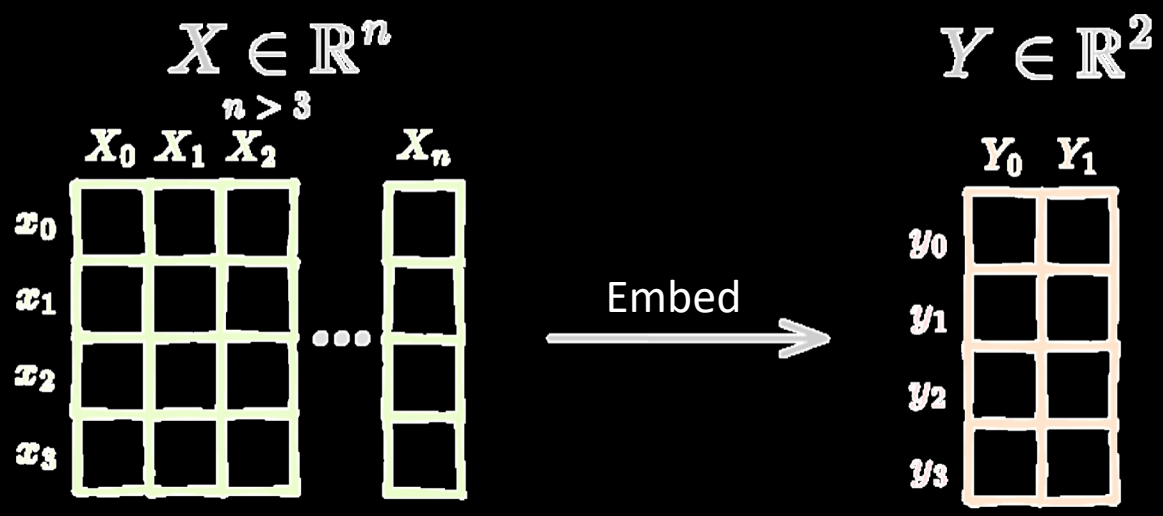
HERTIE INSTITUTE FOR
AI IN BRAIN HEALTH

t-SNE Visualisations With Adaptive Degrees of Freedom parameter

Artemii Shlychkov
Supervisor: D. Kobak

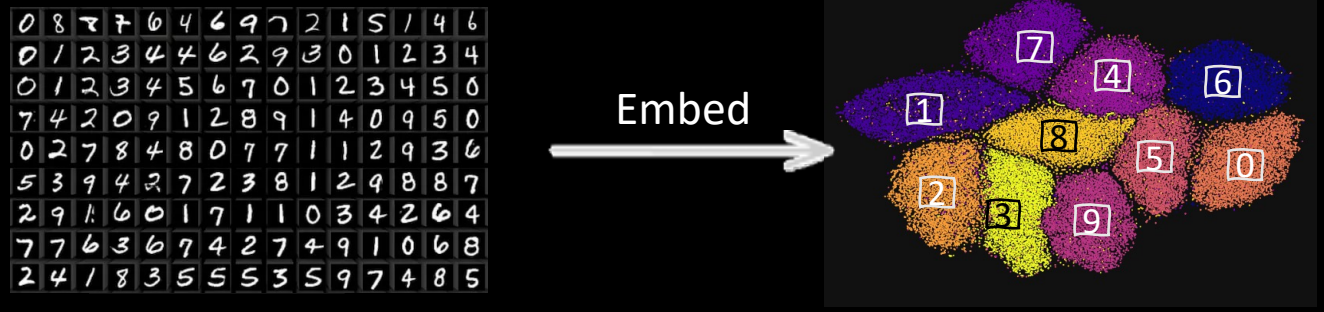
Tübingen, 2024

Core Idea of SNE



High-dimensional data

Two-dimensional representation



Loss function is the Kullback-Leibler divergence between pairwise similarities in the high-dimensional and in the low-dimensional spaces:

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

Where p_{ij} represent high dimensional similarities:

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

And q_{ij} represent low dimensional similarities.

Update rule for the points coordinates is:

$$y_i \leftarrow y_i - \eta \cdot \frac{\partial \mathcal{L}}{\partial y_i}$$

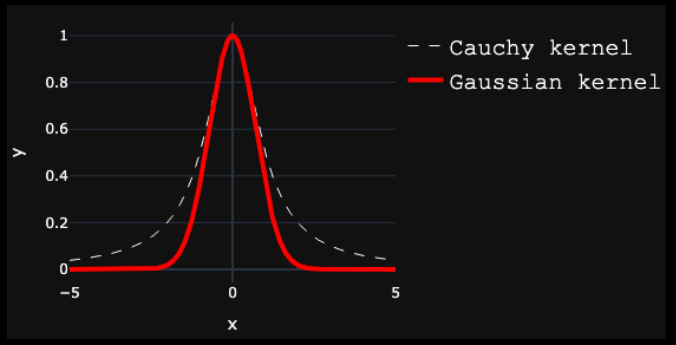
Low dimensional similarities $q_{i,j}$

$$q_{i,j} = \frac{k(d)}{Z} \quad \text{with } Z \text{ as a normalization term}$$

Gaussian kernel: $k = e^{(-d_{i,j}^2)}$

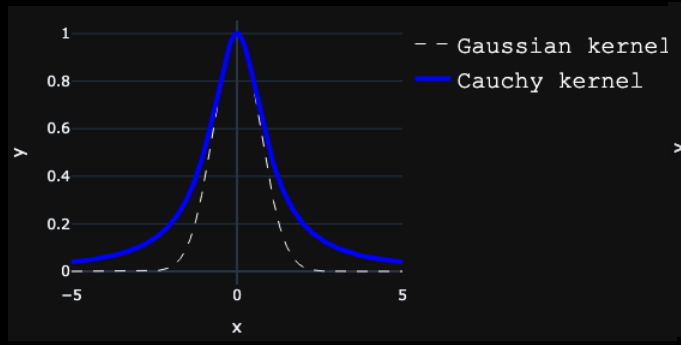
t-distribution kernel: $k = \left(1 + \frac{d_{i,j}^2}{\alpha}\right)^{-\alpha}$

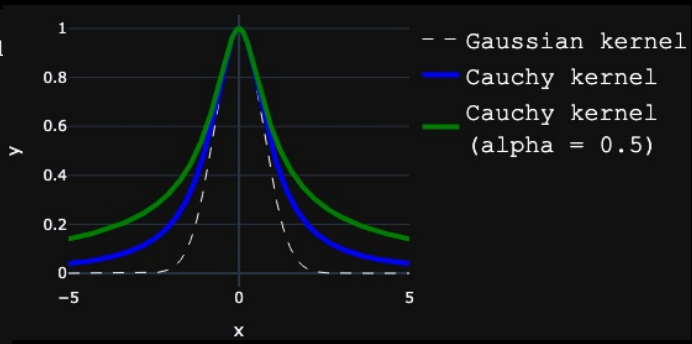
α => ∞ ? α = 1 α < 1 ✅



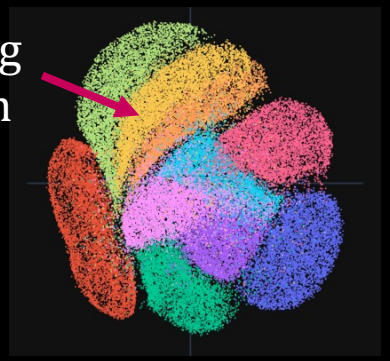
Optimize for alpha ?

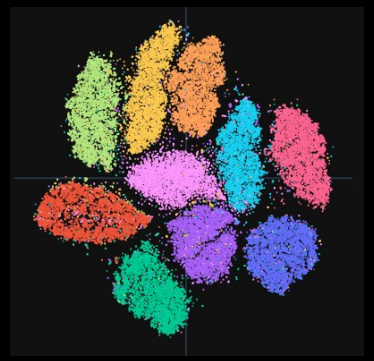
$$\frac{\partial \mathcal{L}}{\partial \alpha}$$

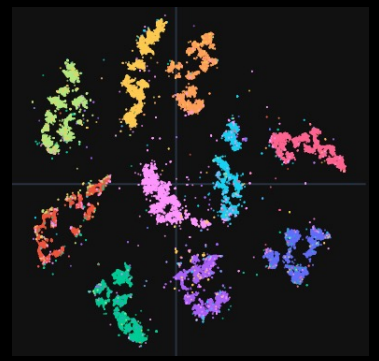




crowding problem





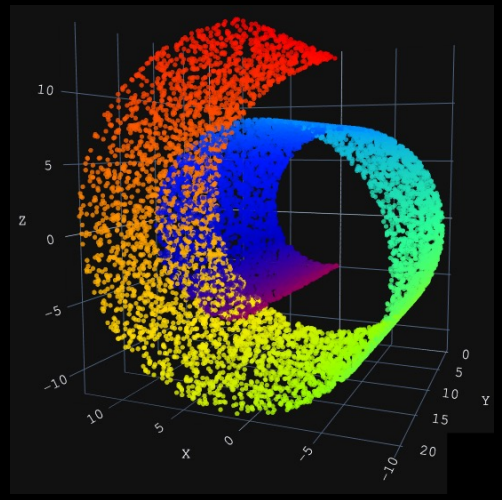


Kobak et al, 2020

Methods

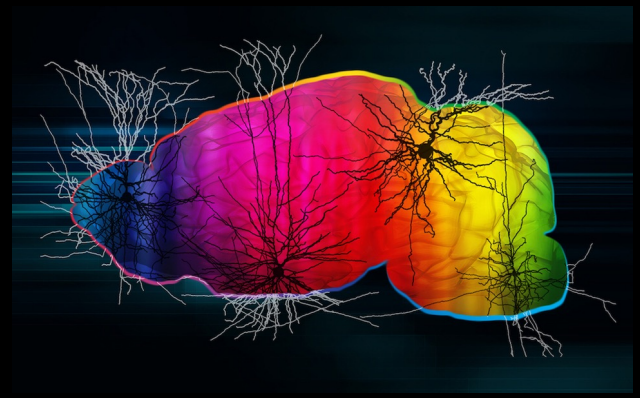
- openTSNE library:
  - pavlin-policar/openTSNE with standard t-SNE parameters
    - Custom implementation with adaptable alpha parameter
- Embedding quality metrics: KL divergence and kNN recall
- Explored datasets:

Swiss Roll



Digits / MNIST

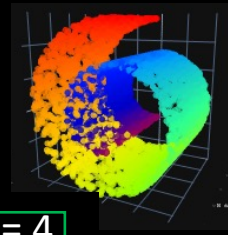
Single-cell transcriptomics
(Tasic et al dataset)*



*Tasic et al, 2018

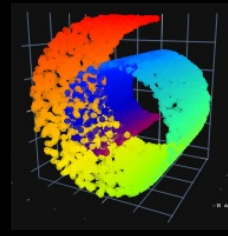
I. Exploring t-SNE embeddings of different datasets using an array of alphas

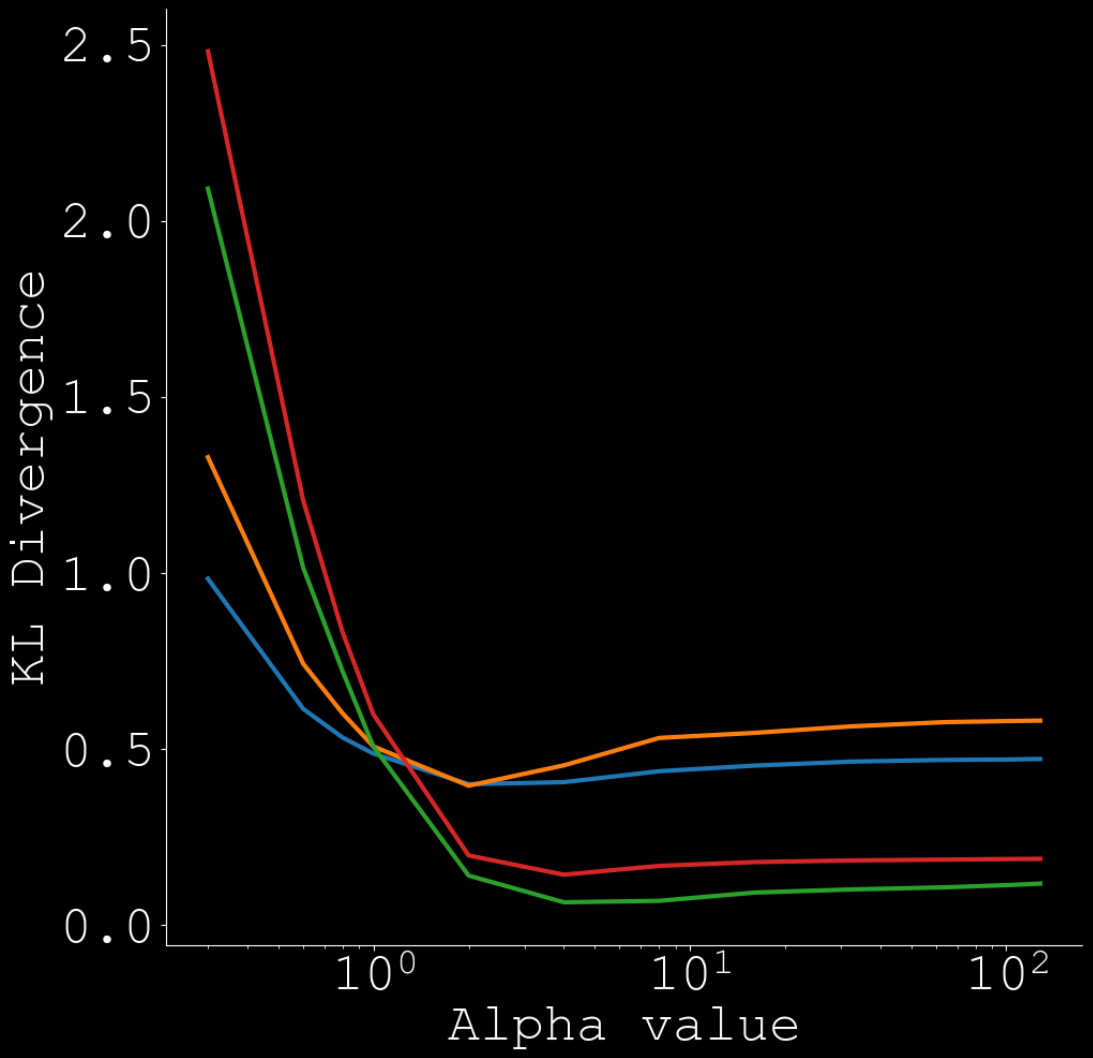
Swiss Roll (3000 samples)

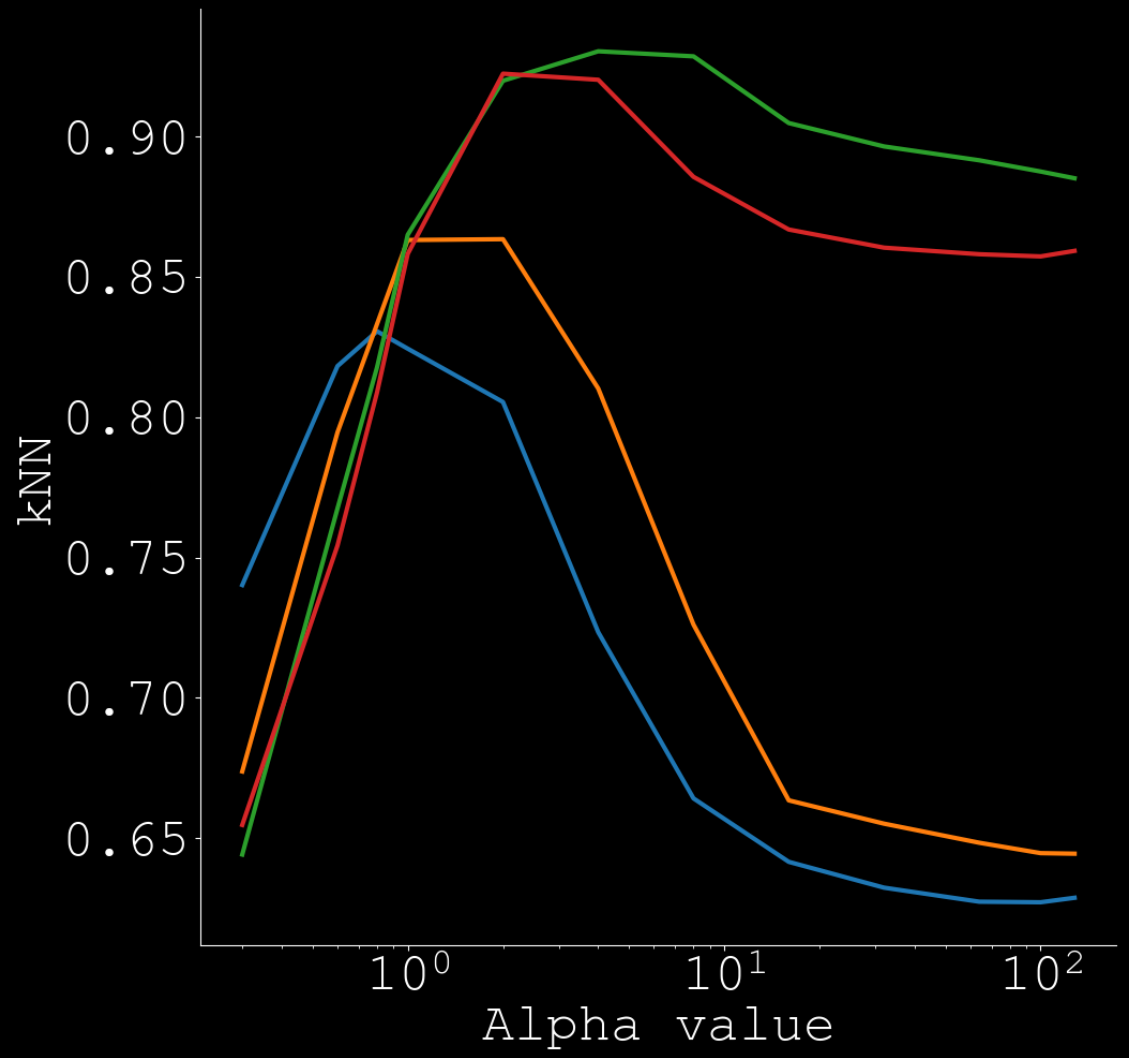


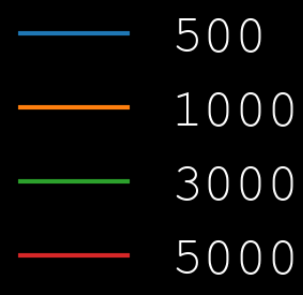
α = 0.3	α = 0.6	α = 0.8	α = 1	α = 2	α = 4
KL: 2.09 kNN Recall: 0.64	KL: 1.01 kNN Recall: 0.77	KL: 0.72 kNN Recall: 0.82	KL: 0.51 kNN Recall: 0.87	KL: 0.20 kNN Recall: 0.92	KL: 0.16 kNN Recall: 0.91
α = 8	**α = 16**	**α = 32**	**α = 64**	**α = 100**	**α = 128**
KL: 0.18 kNN Recall: 0.88	KL: 0.19 kNN Recall: 0.86	KL: 0.20 kNN Recall: 0.85	KL: 0.21 kNN Recall: 0.84	KL: 0.21 kNN Recall: 0.84	KL: 0.22 kNN Recall: 0.84

KL divergence and kNN recall for t-SNE embeddings of SwissRoll with different number of samples

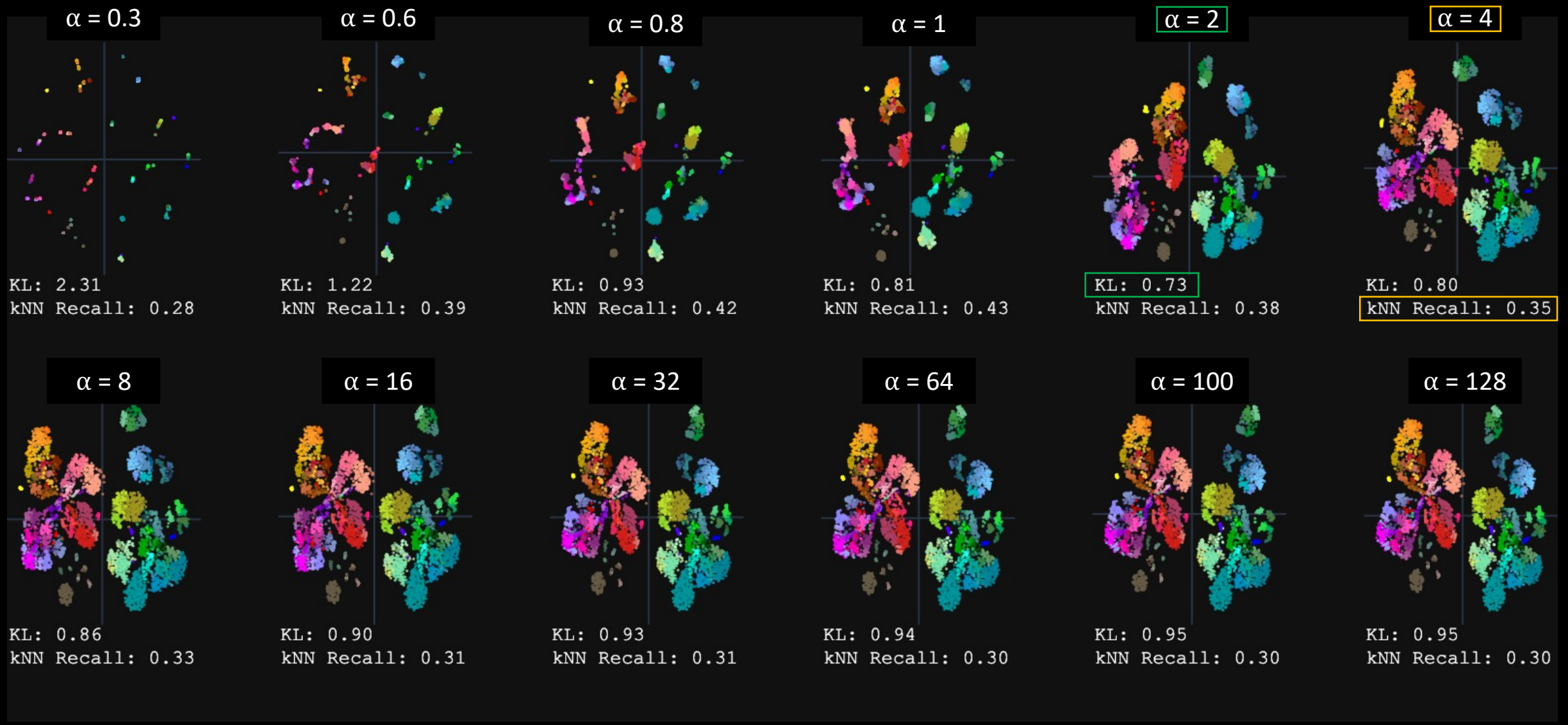




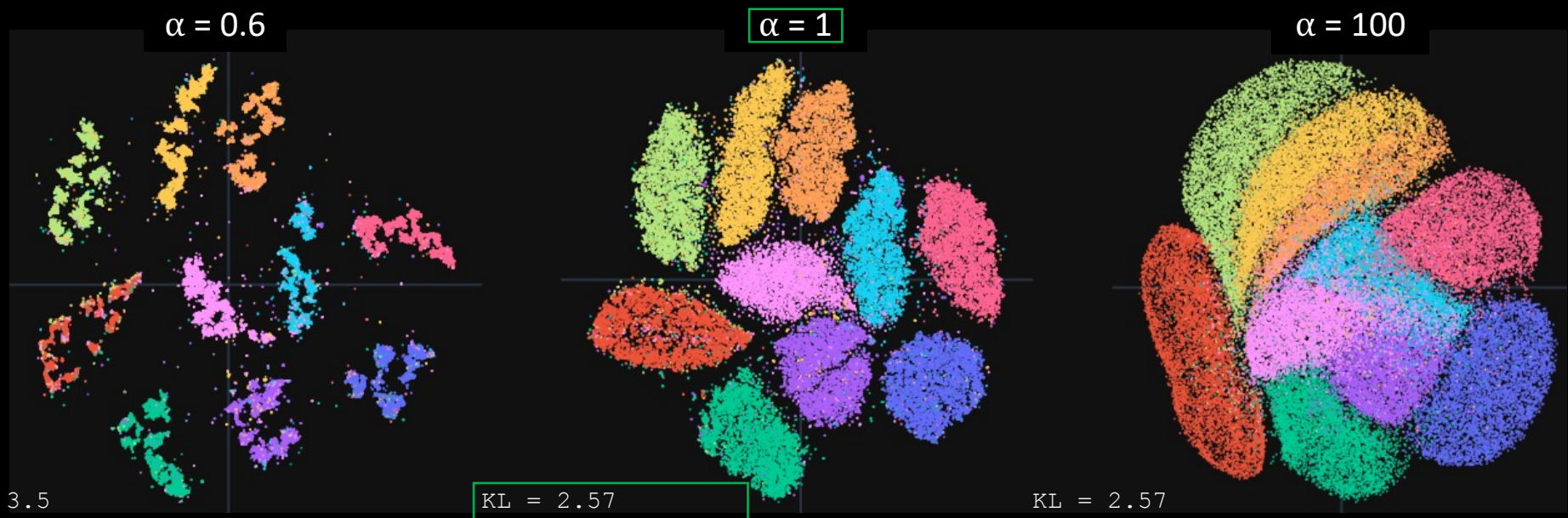


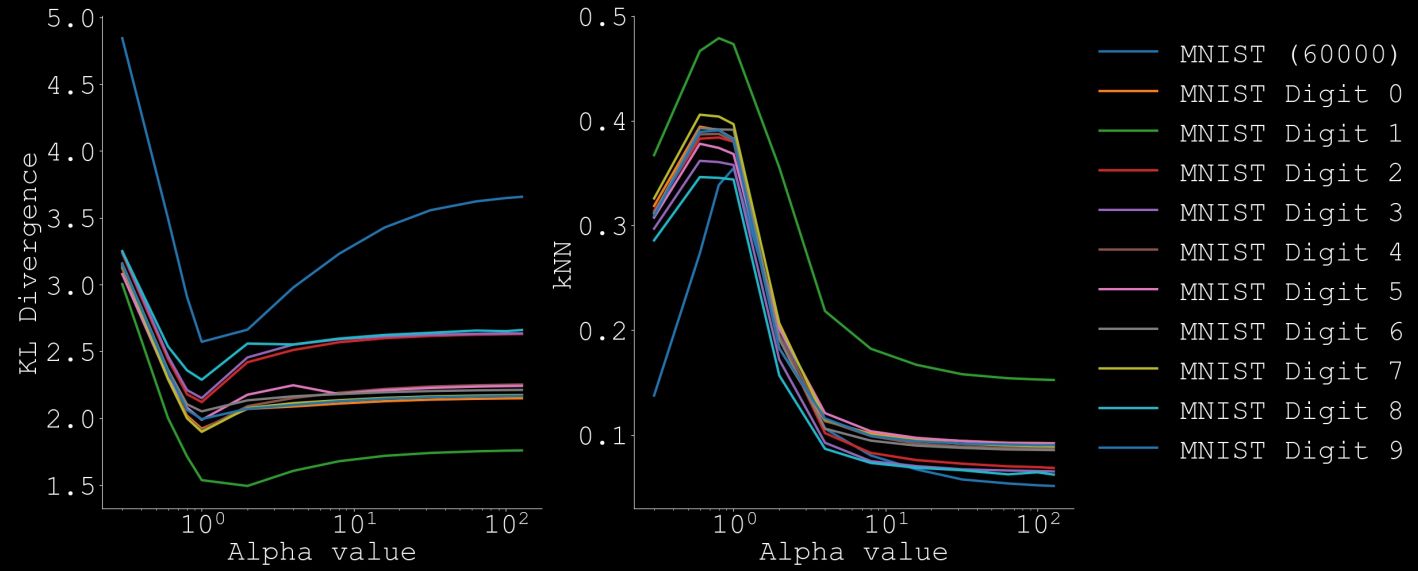


Single cell transcriptomics (5000 samples)



MNIST dataset (60000 samples)





II. Optimizing t-SNE embedding for the α parameter

Derivation of the gradient of alpha

Given the Loss function:

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

where

$$q_{i,j} = \frac{k}{Z}$$

$$\text{with kernel} \quad k = \left(1 + \frac{d_{i,j}^2}{\alpha}\right)^{-\alpha}$$

$$\text{and} \quad Z = \sum_{k \neq \ell} \left(1 + \frac{d_{k,\ell}^2}{\alpha}\right)^{-\alpha}$$

Taking partial derivative of the relevant term:

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

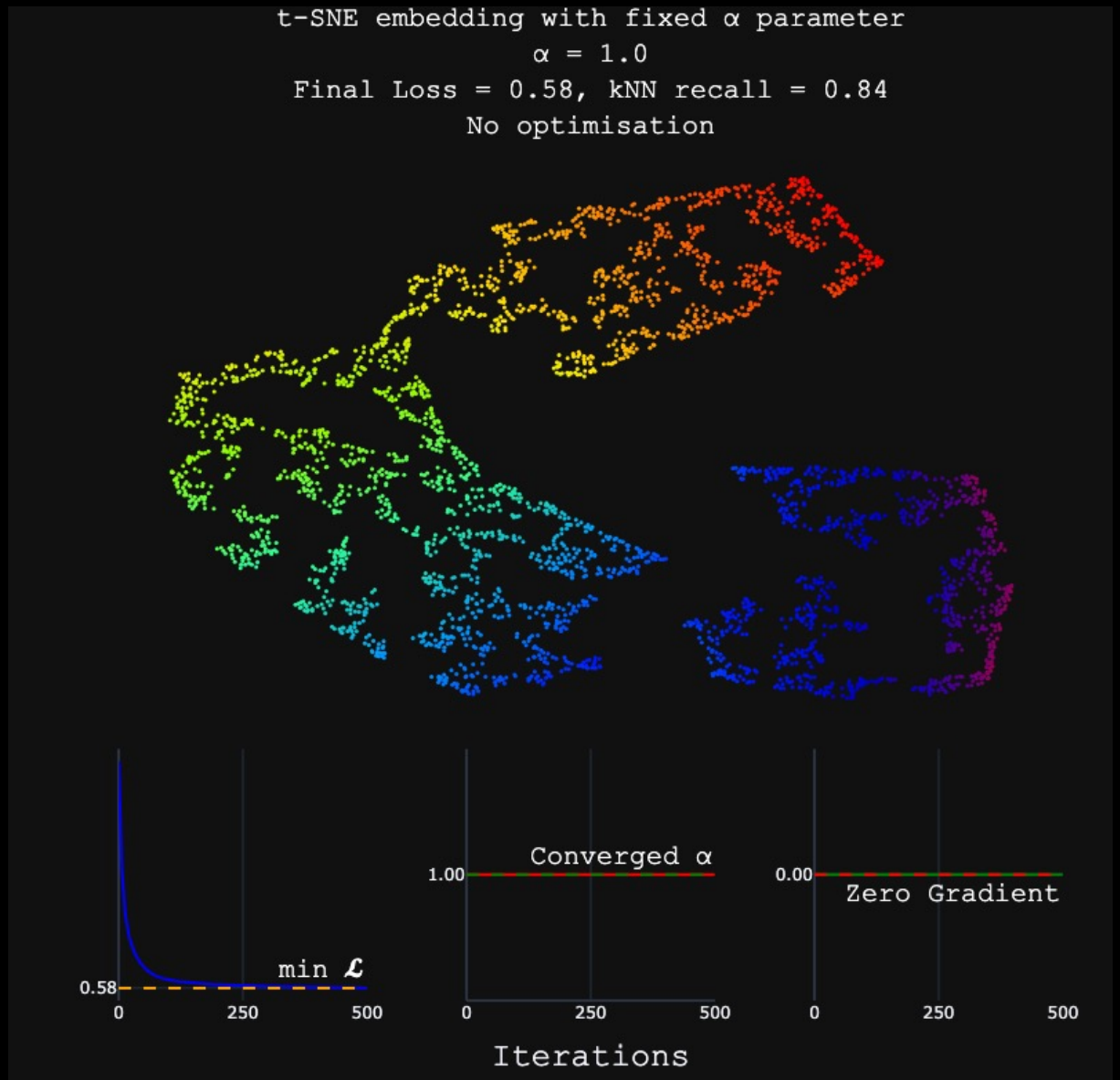
After several transformations finally obtain:

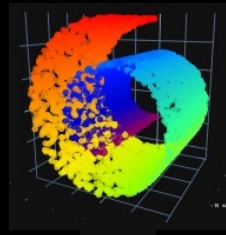
$$\boxed{\frac{\partial L}{\partial \alpha} = \sum_{i,j} (p_{i,j} - q_{i,j}) \left(\log \left(1 + \frac{d_{i,j}^2}{\alpha}\right) - \frac{d_{i,j}^2}{\alpha + d_{i,j}^2} \right).}$$

With the update rule:

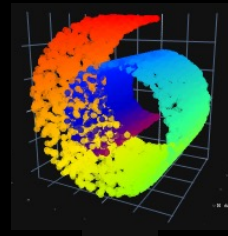
$$\boxed{\alpha \leftarrow \alpha - \eta \cdot \frac{\partial \mathcal{L}}{\partial \alpha}}$$

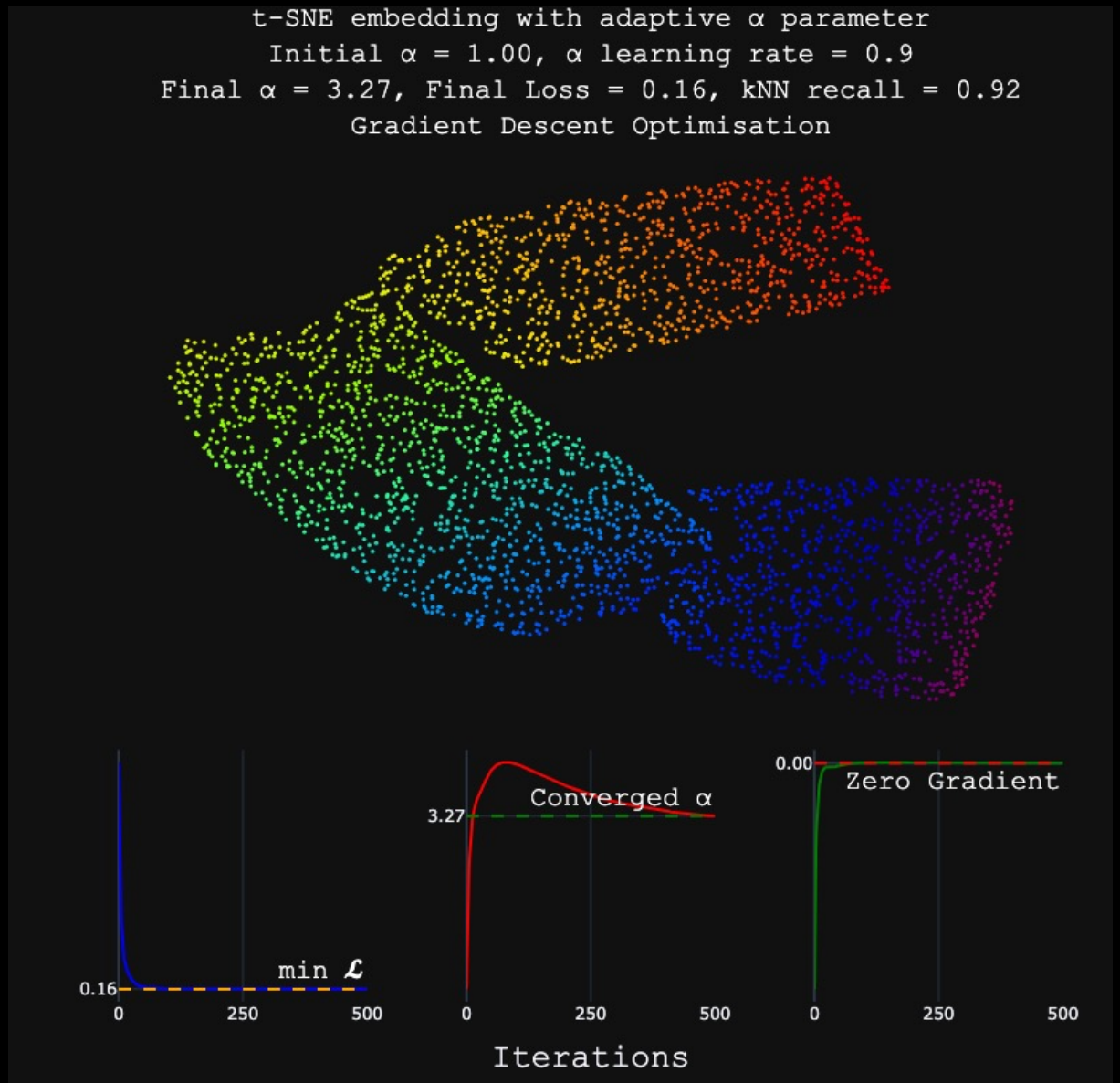
Testing alpha optimization on toy data



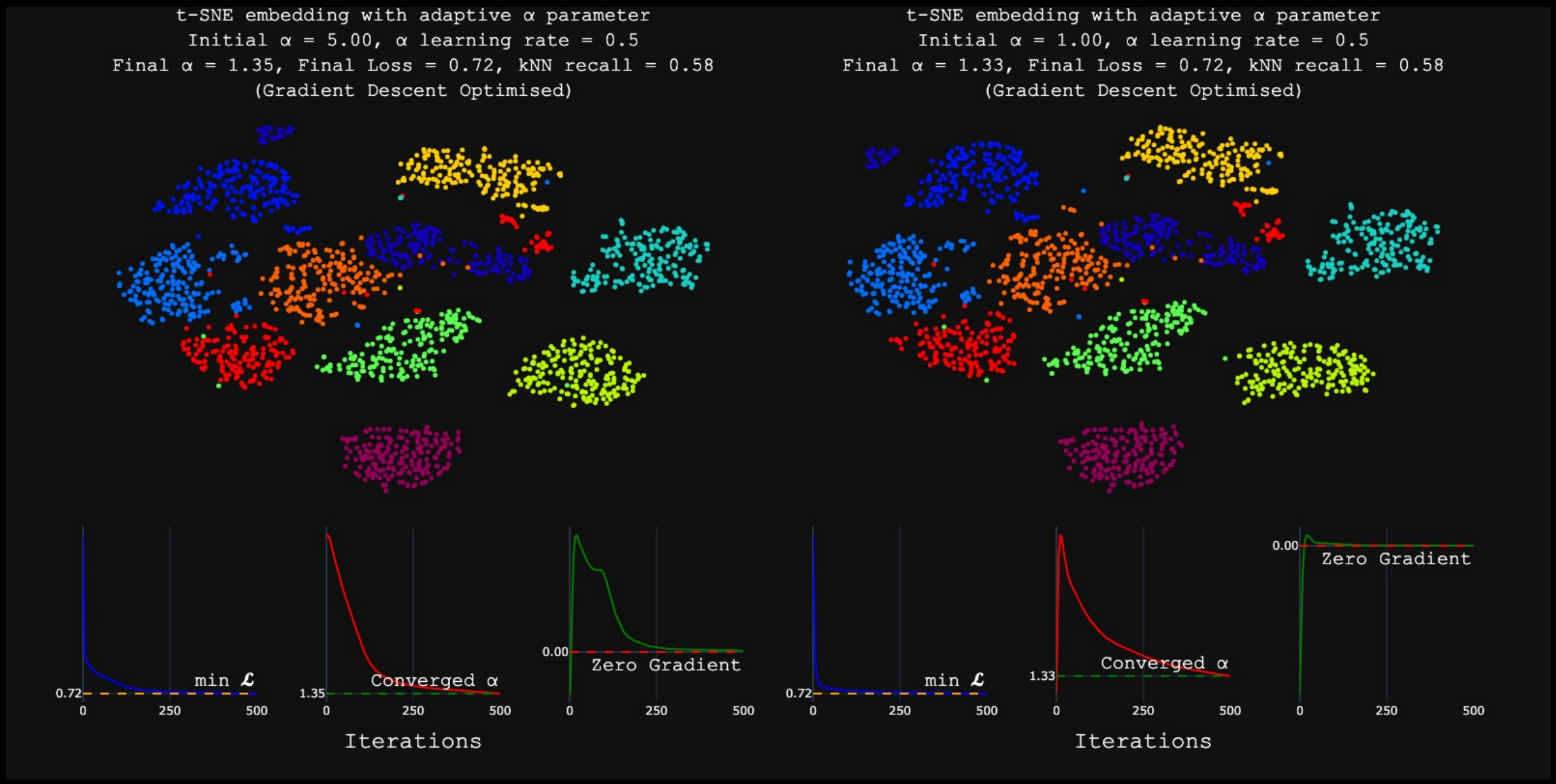


Testing alpha optimization on toy data

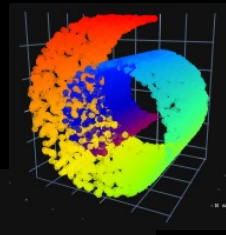




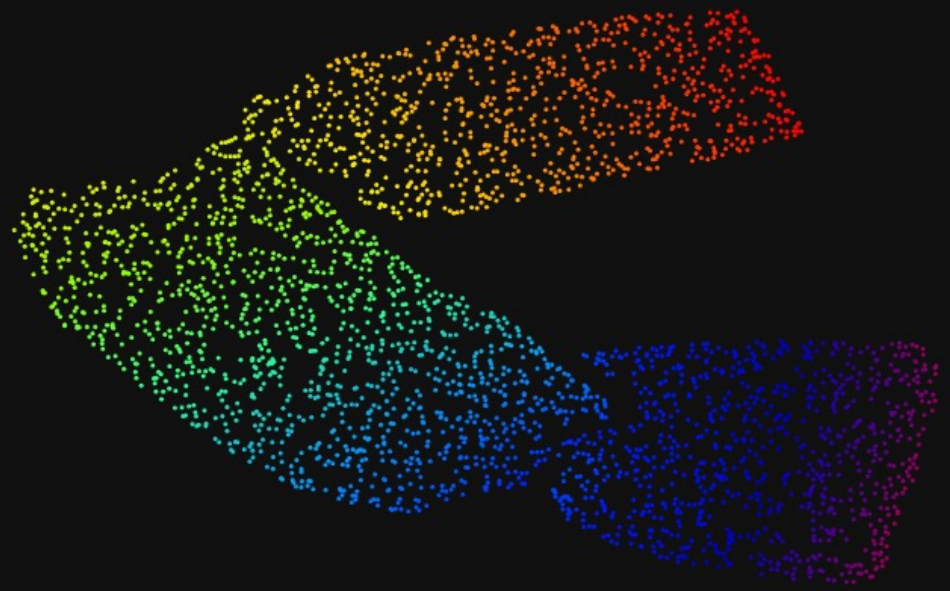
Testing alpha optimization on toy data



Comparing BH-approximation with exact gradient

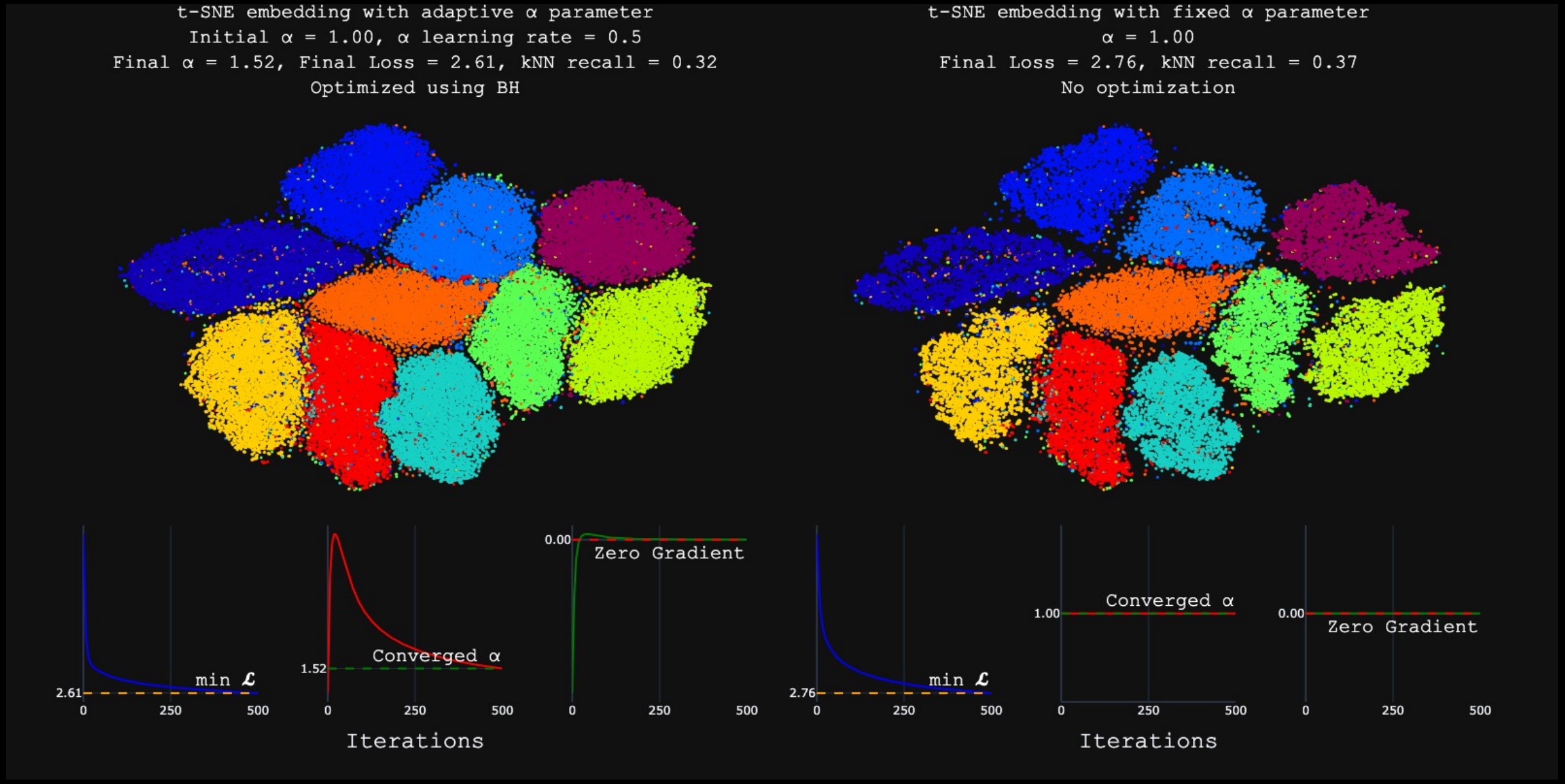


t-SNE embedding with adaptive α parameter
Initial α = 1.00, α learning rate = 0.9
Final α = 3.27, Final Loss = 0.16, kNN recall = 0.92
Gradient Descent Optimization



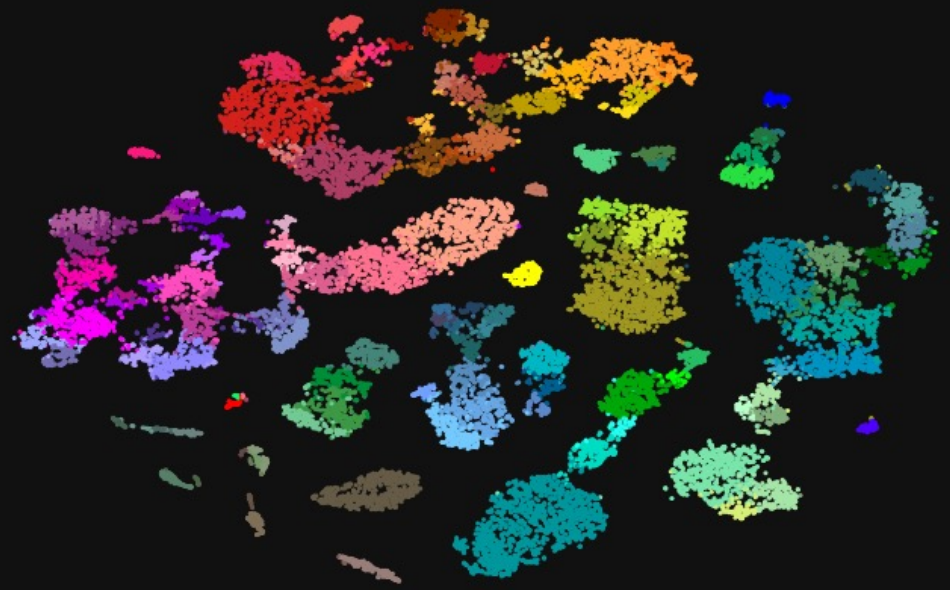
t-SNE embedding with adaptive α parameter
Initial α = 1.00, α learning rate = 0.9
Final α = 3.93, Final Loss = 0.16, kNN recall = 0.92
Barnes-Hut Optimization

Optimizing alpha on MNIST dataset

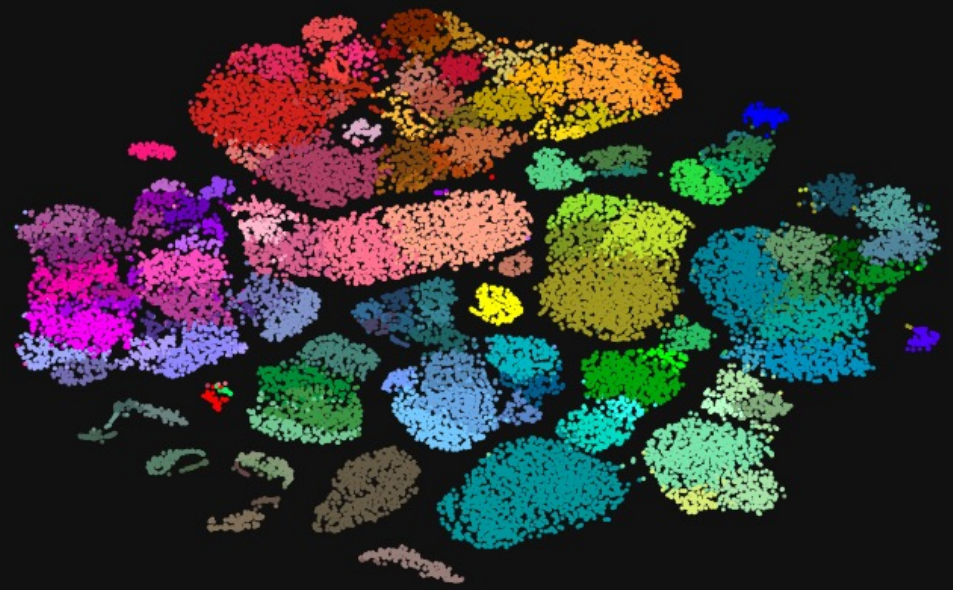


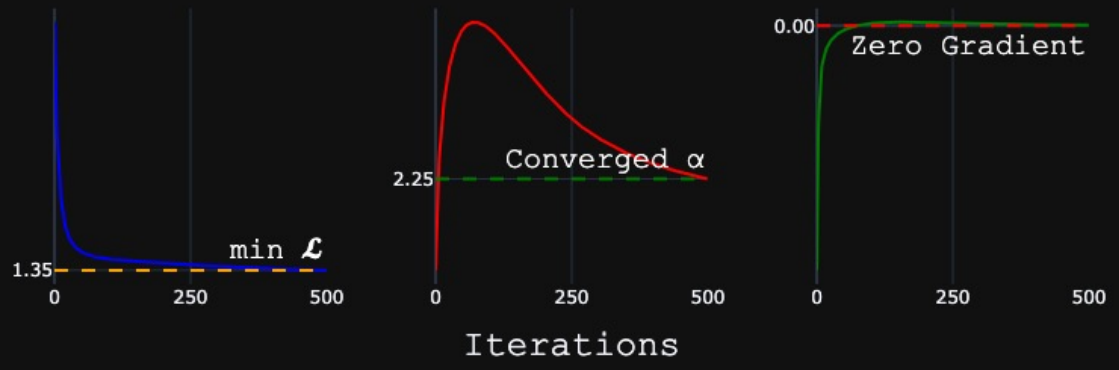
Optimizing alpha on transcriptomics dataset

t-SNE embedding with fixed α parameter
α = 1.00
Final Loss = 1.59, kNN recall = 0.45
No optimization



t-SNE embedding with adaptive α parameter
Initial α = 1.00, α learning rate = 0.5
Final α = 2.25, Final Loss = 1.35, kNN recall = 0.38
Optimized with Barnes-Hutt





Optimizing alpha on transcriptomics dataset

Further work

- Exploring larger variety of real-world datasets, employing more objective metrics for the embedding quality
- Exploring interactions of the alpha parameter with other t-SNE parameters (e.g. perplexity)
- Further integration of alpha gradient computation into existing openTSNE framework to achieve faster results at less computational costs

Thank you for your attention!

‘Drift’ optimization for alpha