Graduate School of Computational Neuroscience
University of Tübingen

T-SNE Visualisations With Adaptive Degrees of Freedom Parameter

Laboratory report

presented by

Artemii Shlychkov

The study was supervised by

Dmitry Kobak, PhD
Hertie Institute for AI in Brain Health

Duration of the lab rotation: 07.10.24 – 06.12.24
Deadline for submission: 03.01.25

Abstract T-distributed Stochastic Neighbor Embedding (t-SNE) is a widely used tool for dimensionality reduction and visualization of high-dimensional datasets. By replacing the Gaussian kernel in SNE with a Cauchy kernel (Student t-distribution with the degree-of-freedom parameter alpha set to 1), it alleviates the "crowding problem" in low-dimensional embeddings. This study explores how varying the degree-of-freedom parameter alpha affects t-SNE embeddings on both toy and real-world datasets (e.g., MNIST and single-cell RNA sequencing). We further propose a gradient-based method that treats alpha as a trainable parameter, allowing it to be adjusted during embedding optimization. Our experiments reveal that alpha values different from 1 can yield superior embeddings, reflected by reduced Kullback-Leibler (KL) divergence and higher k-Nearest Neighbors (kNN) recall scores, on some datasets. Overall, these results suggest that alpha optimization can lead to more faithful low-dimensional representations of high-dimensional data.

1. Introduction

Nowadays, datasets often span high-dimensional spaces, obscuring underlying structures and relationships. Dimensionality reduction and 2D visualizations address this challenge by transforming complex data into simpler, more interpretable forms also revealing hidden patterns or clusters. This approach facilitates exploratory data analysis and serves as a foundation for subsequent computational and analytical steps.

T-distributed Stochastic Neighbor Embedding (t-SNE) and related methods are widely used to visualise high-dimensional datasets in two dimensions, making them easier to perceive and analyse. Today, they are extensively applied in various scientific fields, proving suitable for uncovering underlying data structures, for example in single-cell transcriptomics (Kobak & Berens, 2019; Tasic et al., 2018) or digitized library collections (Schmidt, 2018).

T-SNE aims to capture the local structure of high-dimensional data while preserving global structure, such as the presence of clusters at different scales. This is achieved by iteratively minimising a Kullback-Leibler (KL) divergence loss function (1): a measure of how similarities (or affinities) in high-dimensional space diverge from similarities in low-dimensional space. Pairwise affinities are usually computed by passing squared Euclidean distances through some kernel k and then normalizing, so that their sum equals to one. In high dimensional space, for example, this leads to close neighbors having high affinities whereas distant samples having near-zero affinities.

$$\mathcal{L} = \sum_{i,j} p_{i,j} \log \frac{p_{i,j}}{q_{i,j}} \tag{1}$$

Compared to its predecessor, Stochastic Neighbour Embedding (SNE), t-SNE replaces the Gaussian kernel used for low-dimensional similarities between data points $q_{i,j}$ (2) with a Cauchy kernel – a special case of a general Student t-distribution kernel (3) with $\alpha = 1$. This important change mitigates the "crowding problem", where distinct high-dimensional clusters often overlap in low-dimensional embeddings. In addition, t-SNE introduces a symmetrized loss function that simplifies gradient computations (Van Der Maaten & Hinton, 2008).

$$q_{i,j} = \frac{k(d)}{Z} \tag{2}$$

$$k = \left(1 + \frac{d_{i,j}^2}{\alpha}\right)^{-\alpha} \quad (3)$$

While the degrees-of-freedom parameter (α) in modern t-SNE implementations is conventionally set to 1, it is not inherently constrained to this value and can take any real number. For large α, the t-distribution converges to a Gaussian, effectively resembling SNE-like behavior. It has been shown previously that using $\alpha < 1$ emphasizes finer structures, allowing the identification of smaller but meaningful clusters (Kobak et al., 2019). Conversely, embeddings with $\alpha > 1$ have been poorly explored, leaving their potential largely untapped.

Crucially, it can be argued that no single value of α universally optimizes embeddings across all datasets or objectives. The author of the original method contemplates that ideal α depends on the ratio of intrinsic dimensionality of the data to latent dimensionality and suggest to either set α as linearly dependant on the dimensionality d of the latent space ($\alpha = d - 1$, which basically results in $\alpha = 1$ for 2D latent space) or treat α as a free parameter, optimized alongside the main cost function (Van Der Maaten, 2009). More recently, Kitazono et al., proposed a method called inhomogeneous t-SNE, which optimizes the alpha parameter point-wise to capture the inhomogeneity in dataset, and achieved superior results for datasets with low and high-dimensional manifolds, compared to SNE or standard t-SNE (Kitazono et al., 2016).

However, to date, despite its apparent impact on visualization outcomes, α is usually selected arbitrarily, with $\alpha = 1$ being the most common default. In this work, we sought to treat α as a trainable parameter, optimizing it via gradient descent on the t-SNE loss function. With this approach, we aimed to identify data-specific α values that improve embedding quality. Specifically, we addressed two **core questions**: (1) how embeddings behave for $\alpha > 1$, and (2) whether adaptively optimizing α through gradient descent enhances t-SNE's performance.

2. Material & Methods

In our experiments, we used our own implementation of the openTSNE Python library (Poličar et al., 2024), the source code of which is available freely on GitHub. Our modifications enabled computing the alpha gradient and using the updated alpha value during the embedding process. These changes did not affect the normal workflow in any other way. In brief, the t-SNE embedding process can be divided into four steps:

1. Initialization
2. Affinity matrix computation
3. Early exaggeration
4. Optimization

Steps 1 and 2 are independent of the alpha parameter and run once for a given dataset. For all of our experiments, we used common settings for these steps:

1. **Initialization**: PCA with 2 components
2. **Affinity matrix**: Perplexity-based k-nearest neighbors (k set to `auto`), using a Euclidean distance metric and perplexity = 30
3. **Early exaggeration** step was performed in the conventional manner, using 250 iterations of optimization with a fixed alpha, an exaggeration parameter of 12, and a

learning rate set to $n\,samples\,/\,exagerration$, as suggested in (Linderman & Steinerberger, 2019)

4. In the **Optimization** step of the experiments with an adaptable alpha, we computed the alpha gradient and updated alpha accordingly at each iteration. We then performed a single embedding optimization step using this updated alpha. In contrast, in the experiments with a fixed alpha, the optimization step used the same alpha value for all iterations and otherwise identical parameters. All experiments ran for 500 optimization iterations in total. For the experiments with fixed alphas, we used the following grid of alpha values: {0.3, 0.6, 0.8, 1, 2, 4, 8, 16, 32, 64, 100, 128}

The exact formulas for the alpha gradient and the corresponding update rules will be discussed further. To formally evaluate the quality of the resulting embeddings, we used two metrics as suggested in (Gove et al., 2022):

1. KL-divergence. Measures the difference between the two probability distributions (original data neighbors vs. embedding neighbors). Because t-SNE explicitly minimizes KL-divergence, a lower value indicates a better embedding.
2. KNN recall. Defined as a fraction of the $k = 10$ nearest neighbors that are neighbors of p_i in both original space $n_k(i)$ and the embedding $v_k(i)$ for every point p_i. This value is averaged across all points (4). A higher value indicates better preservation of local structure.

$$\frac{1}{nk}\sum_{i=1}^{n}|n_k(i)\cap v_k(i)| \qquad (4)$$

KL-divergence and kNN recall averaged across several runs are expressed as mean ± standard error of mean.

For our experiments we used the following datasets:

1. Sci-kit Learn SwissRoll with varying number of samples.
2. MNIST, comprising n = 70,000 grayscale 28×28 images of handwritten digits.
3. Single cell RNA-sequencing data from (Tasic et al., 2018), further referred to as RNA-seq dataset, comprising n = 23,822 cells from adult mouse cortex (sequenced with Smart-seq2 protocol). Dimensions are genes, and the data are the integer counts of RNA transcripts of each gene in each cell.

3. Results

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

To begin with we explored different datasets, performing the standard t-SNE embedding workflow with an array of fixed alphas to see, how different alpha values affect the result. Swiss roll (Fig. 1, A) is represented by a rolled up 2D manifold in 3D space (Marsland, 2011). This dataset is known to be challenging for t-SNE because the algorithm often exhibits over-fragmentation: it is able to preserve the general structure of the data, but poorly represents the continuous nature of the original manifold. Instead, t-SNE tends to cluster nearby points together too tightly, producing an unnecessarily fragmented embedding (Fig. 1, B).

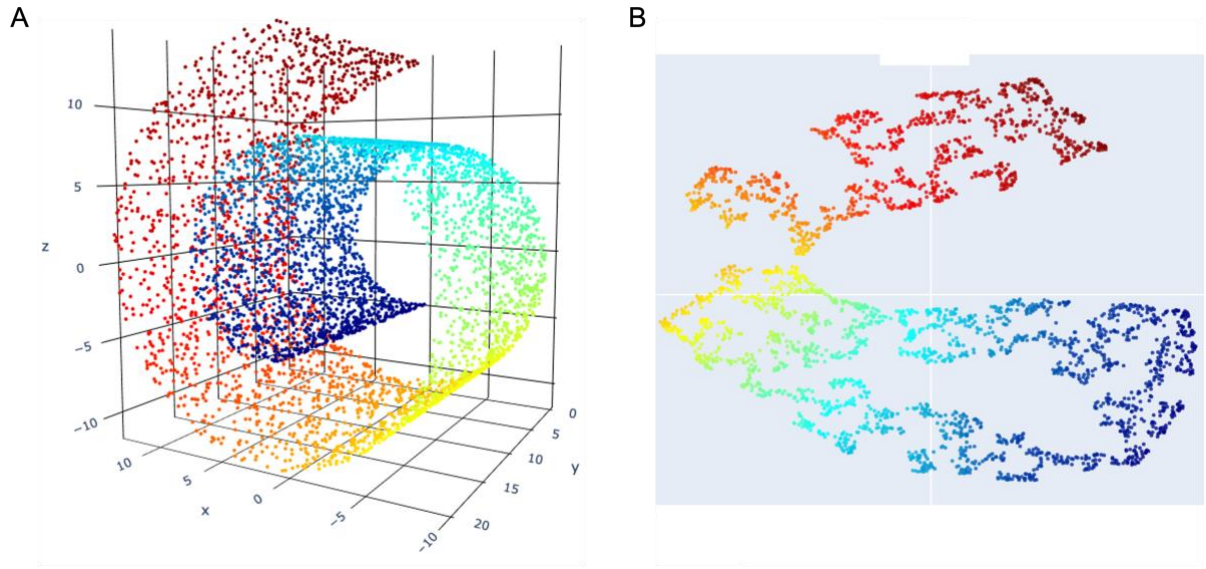

Figure 1. A. Swiss Roll with 3000 samples in the ambient space. B. Example *t-SNE embedding of the Swiss Roll from A with default parameters and* $\alpha = 1$.

We applied t-SNE to this dataset with 3000 samples using a range of α values and, as expected, observed distinct embedding outcomes for each setting (further referred to as 'grid search'). Notably, higher α values (i.e., $\alpha > 1$) produced embeddings with less over-fragmentation. Data points that were close in the original high-dimensional space generally remained clustered in the lower-dimensional representation and the natural continuity of the data is better preserved. However, we still observed disruptions (e.g., at the transition between cyan and yellow regions on fig. 2) and distortions in the relative proportions of the Swiss Roll sections. Additionally, with larger alphas, at the transition between the yellow and red regions, the manifold appeared twisted along the Y-axis, suggesting that the "roll" partially unrolled along the X-axis before being squashed onto the XZ-plane (Fig. 2).

Despite these visible distortions, quantitative evaluation using KL-divergence and kNN recall suggest that the optimal α range lies between 2 and 8.

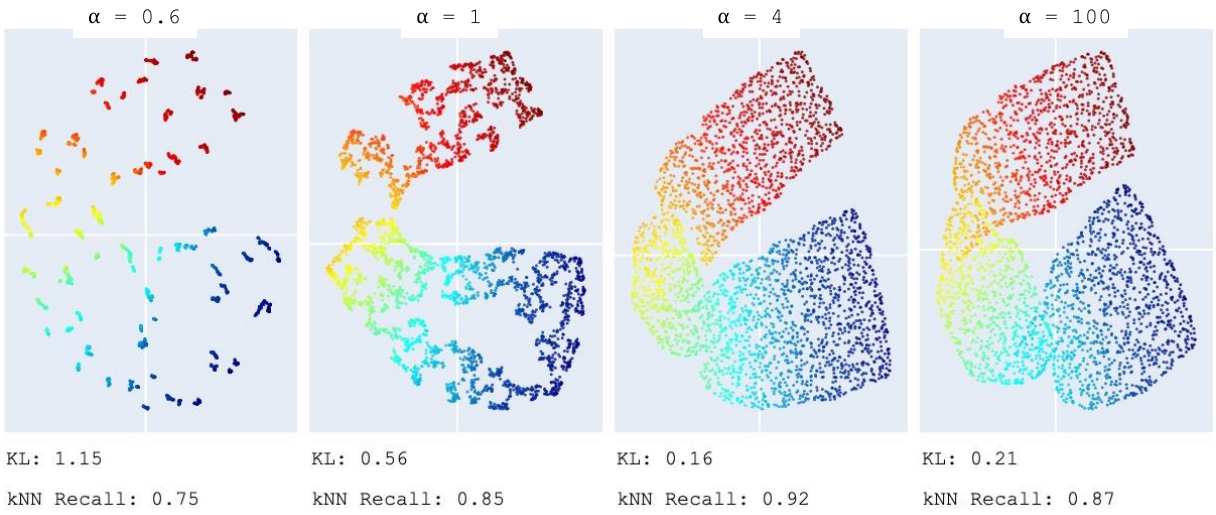

Figure 2. Example t-SNE embeddings of the Swiss Roll dataset (3,000 samples) with selected α *values. The conventional setting for t-SNE is* $\alpha = 1$. *Based on KL-divergence and kNN recall,* $\alpha = 4$ *produces the best embedding. At* $\alpha = 100$, *t-SNE essentially behaves like the original SNE.*

Next, we investigated whether different sample sizes of the Swiss Roll dataset would alter these observations. The results are represented on fig. 3.

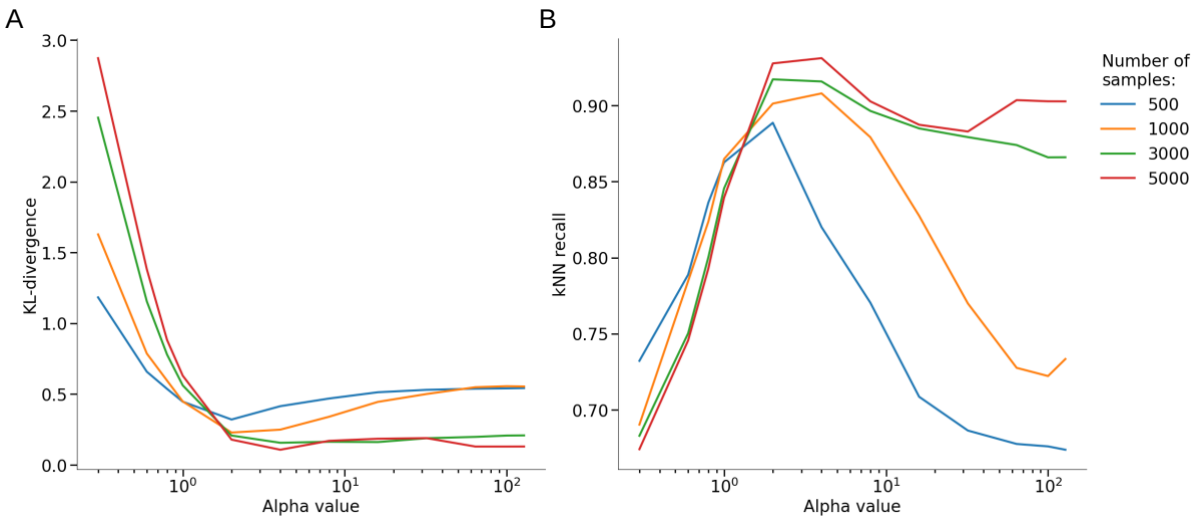

Figure 3. A. KL-divergence of t-SNE embeddings of the Swiss Roll dataset for different sample sizes, plotted against α on a logarithmic scale. B. kNN-recall for the same embeddings, again shown for different sample sizes and α values (x-axis in log-scale).

Varying the sample size indeed affected both KL-divergence and kNN recall, suggesting that the "optimal" α depends on how many points are included. Specifically, the best achievable KL-divergence decreased with larger sample sizes, whereas the optimal α exhibited a tendency to increase – though we did not see a clear relationship, presumably due to a coarse α grid.

Table 1. Optimal alpha and Minimal KL-divergence with respect to the number of samples in the Swiss Roll Dataset

Number of samples	Minimal KL-divergence	Optimal alpha
500	0.32	2
1000	0.22	2
3000	0.15	4
5000	0.10	4

Likewise, kNN recall in general improved with sample size, but without a consistent pattern for identifying the exact optimal α value that maximized this metric.

Table 2. Optimal alpha and Maximal kNN-divergence with respect to the number of samples in the Swiss Roll Dataset

Number of samples	Maximal kNN recall	Optimal Alpha
500	0.89	2
1000	0.91	4
3000	0.92	2
5000	0.93	4

Following the Swiss Roll experiments, we applied the same approach to two real-world datasets: a single-cell transcriptomics dataset from Tasic et al. (Fig.4) and MNIST (Fig.6). To reduce computational overhead for the single-cell data for this experiment, we used 1,000 selected genes and 5,000 cells.

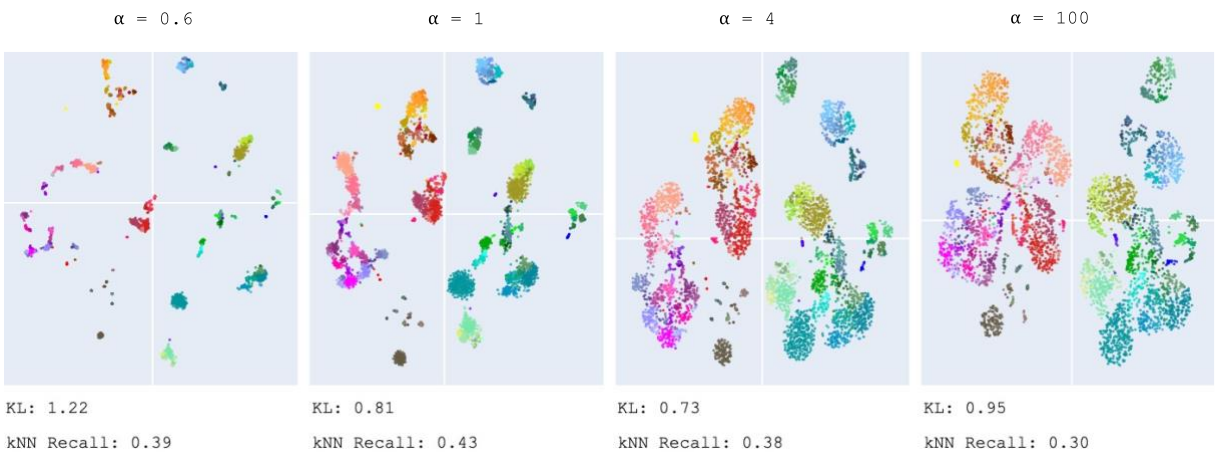

Figure 4. Example t-SNE embeddings of the RNA-seq dataset (5,000 samples) with selected α values. The conventional setting for t-SNE is $\alpha = 1$. Based on KL-divergence, $\alpha = 4$ produces the best embedding. At $\alpha = 100$, t-SNE essentially behaves like the original SNE.

Interestingly, for the single-cell dataset, KL-divergence suggested an optimal $\alpha = 2$, whereas kNN recall favored $\alpha = 1$ (Fig. 5).

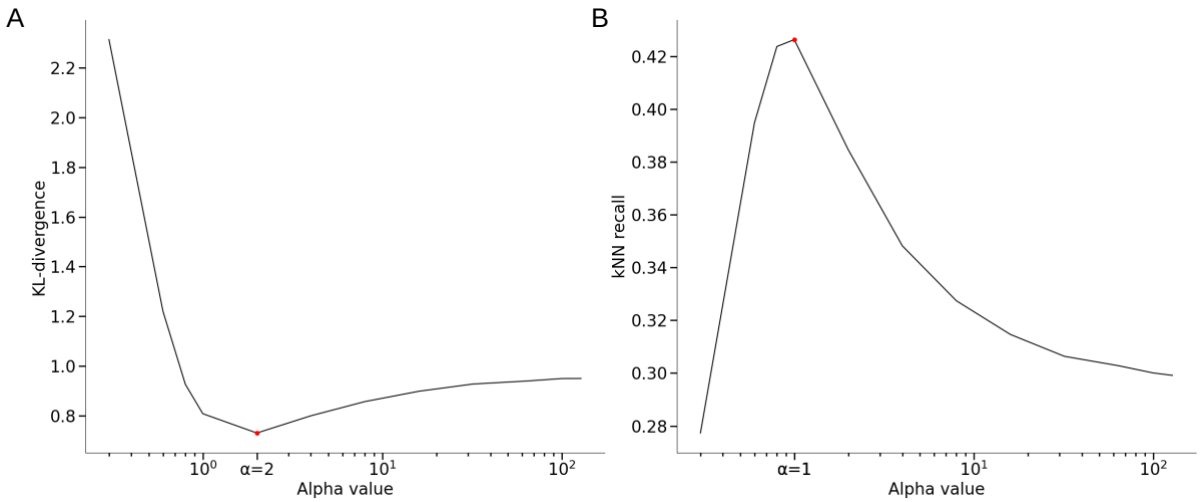

Figure 5. A. KL-divergence of t-SNE embeddings of the RNA-seq dataset, plotted against α on a logarithmic scale. B. kNN-recall for the same embeddings, again shown for different sample sizes and α values (x-axis in log-scale).

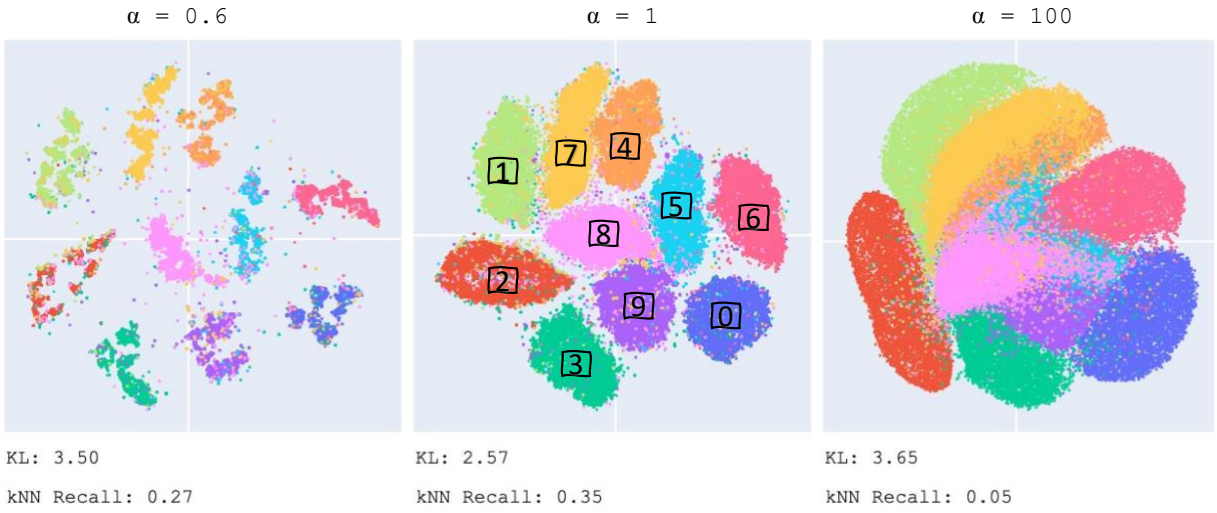

Figure 6. Example t-SNE embeddings of the MNIST dataset 7(0,000 samples) with selected α values. The conventional setting for t-SNE is $\alpha = 1$. Based on KL-divergence and kNN recall, $\alpha = 1$ also produces the best embedding. At $\alpha = 100$, t-SNE essentially behaves like the original SNE.

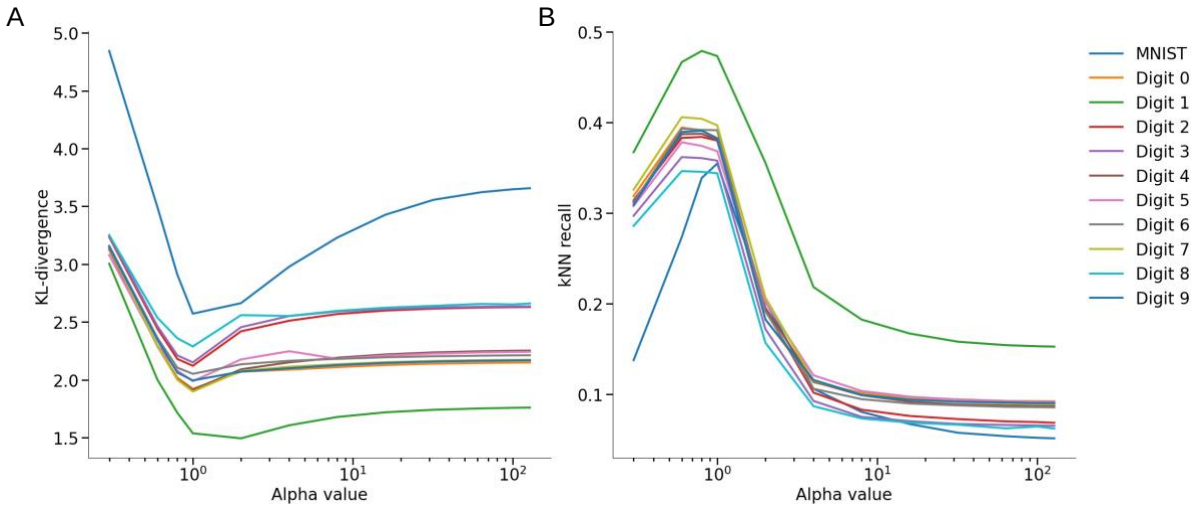

Figure 7. A. KL-divergence of t-SNE embeddings of the whole MNIST dataset as well as different digits separately, plotted against α on a logarithmic scale. B. kNN-recall for the same embeddings, again shown for different sample sizes and α values (x-axis in log-scale).

Turning to MNIST, both KL-divergence and kNN recall pointed to $\alpha = 1$ as optimal for the entire dataset. When analyzing each digit individually, KL-divergence also favored $\alpha = 1$, except for digit "1," which instead pointed to $\alpha = 2$.

These observations collectively support the notion that no single α universally optimizes t-SNE embeddings for all datasets or even different subsets within the same dataset. Consequently, incorporating α-tuning into one's t-SNE workflow might improve the algorithm in terms of achieving more meaningful embeddings. Hence, we proceeded to the second part of our work.

3.2 Optimizing t-SNE embedding for the α parameter

The degree-of-freedom (α) can be viewed as a free parameter to be optimized with respect to the cost function, thus fine-tuning the t-SNE embedding for a specific dataset. We performed this optimization via gradient descent by computing the partial derivative of the loss function with respect to α: $\frac{\partial \mathcal{L}}{\partial \alpha}$, and then applying the standard update rule:

$$\alpha \leftarrow \alpha - \eta \cdot \frac{\partial \mathcal{L}}{\partial \alpha} \tag{5}$$

where η is a learning rate for α. We first derived the analytical form of this gradient:

$$\frac{\partial L}{\partial \alpha} = \sum_{i,j} (p_{i,j} - q_{i,j}) \left(\log\left(1 + \frac{d_{i,j}^2}{\alpha}\right) - \frac{d_{i,j}^2}{\alpha + d_{i,j}^2} \right) \tag{6}$$

Subsequently, we integrated Equations 5 and 6 into the t-SNE workflow and tested the algorithm, which we refer to as the "exact" method on the Swiss Roll dataset with 3,000 data points (Fig. 8). Here and further on we fixed the alpha learning rate η to 0.5 as this value proved satisfactory for convergence dynamics in preliminary test runs.

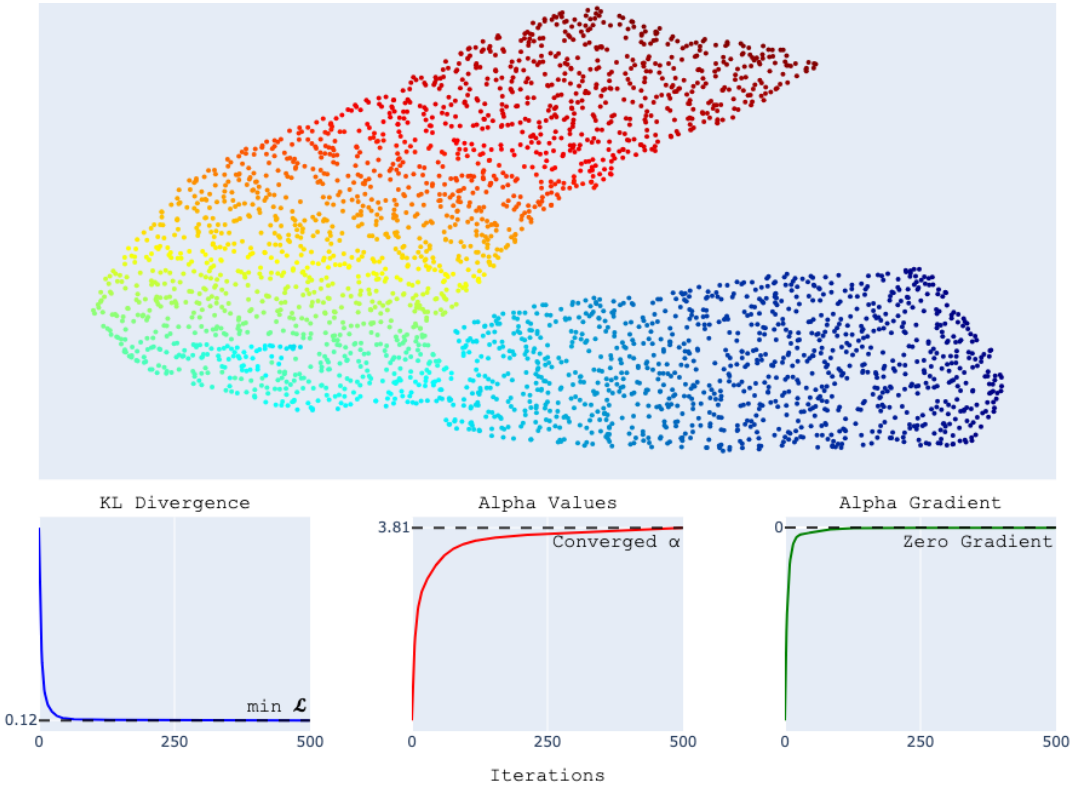

Figure 8. Top panel: example t-SNE embedding of the Swiss Roll with 3,000 samples with adaptive alpha parameter, optimized with the exact method. Lower panel: The progression of KL-divergence, alpha value and $\frac{\partial \mathcal{L}}{\partial \alpha}$ *over the course of the optimization. The initial* α *was set to 1, and the learning rate* η *for* α *was 0.5. After convergence, the final* $\alpha \approx 3.81$*, resulting in a KL-divergence of 0.12 and a kNN recall of 0.93.*

Encouragingly, the average final converged α (3.42 ± 0.11 across 5 runs) aligns with the optimal range determined by our earlier grid search. Meanwhile, the mean KL divergence (0.14 ± 0.009) is even lower than before, and the mean kNN recall (0.93 ± 0.0014) remains essentially unchanged. The embedding preserves the overall continuity of the Swiss Roll in two dimensions, with only minor disruptions (e.g., in the transition from orange to red on fig. 9). Notably, in certain runs, we observed an even more "unrolled" manifold, achieving a KL divergence of 0.05 (see Appendix). As a sanity check, we applied a standard t-SNE algorithm with a fixed $\alpha = 3.42$, previously identified via adaptive optimization (Figure 9).

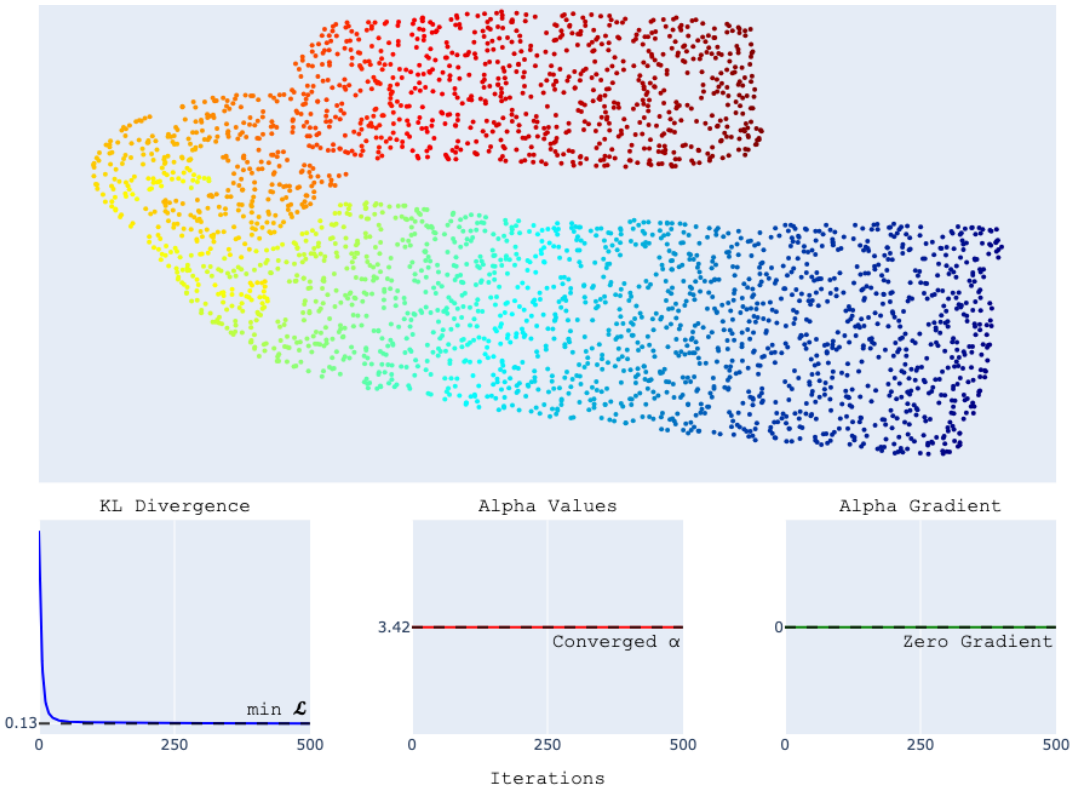

Figure 9. Top panel: example t-SNE embedding of the Swiss Roll with 3,000 samples with fixed alpha parameter. Lower panel: The progression of KL-divergence, alpha value and $\frac{\partial \mathcal{L}}{\partial \alpha}$ *over the course of the optimization. The* α *was fixed to 3.42, so the gradient remains constant all the way. The embedding process resulted in a KL-divergence of 0.17 and a kNN recall of 0.91.*

The resulting embedding closely matched the adaptive run, yielding nearly identical mean KL divergence (0.14 ± 0.009) and kNN recall (0.92 ± 0.002) metrics. Notably, these results also align well with those obtained through the earlier grid search.

Next, to further check alpha gradient behavior, we repeated the adaptive optimization but started from an initial $\alpha = 5$ (Figure 10).

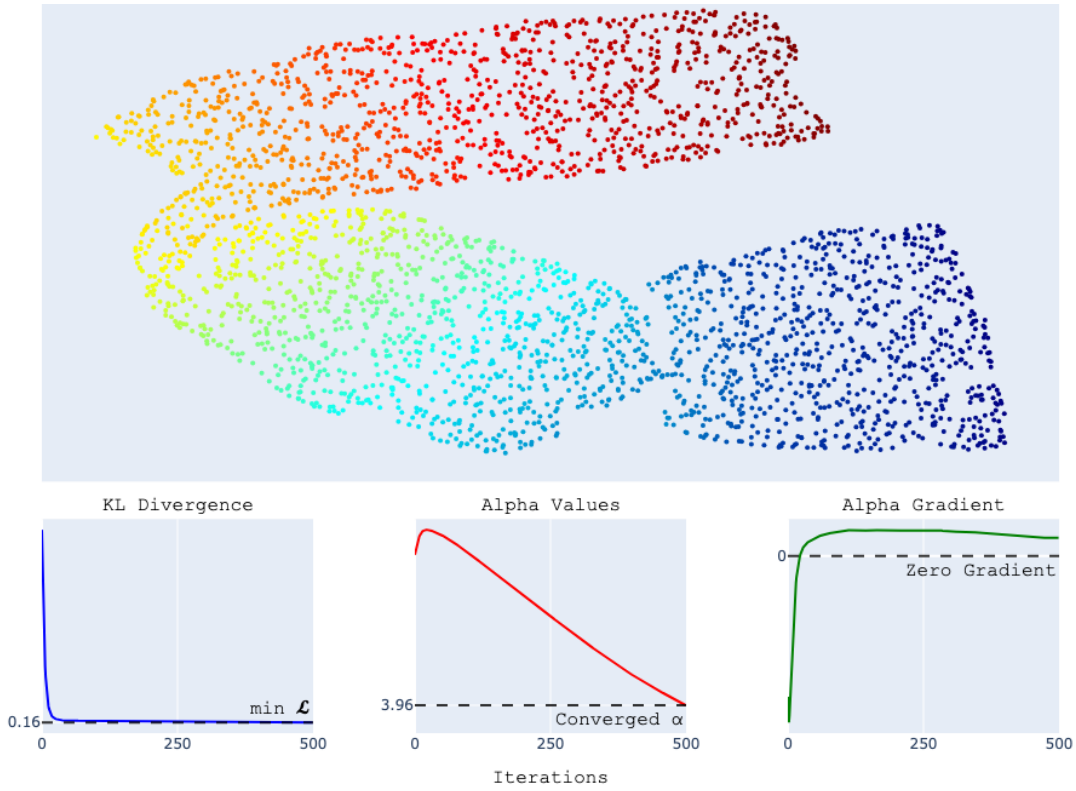

Figure 10. Top panel: example t-SNE embedding of the Swiss Roll with 3,000 samples with adaptive alpha parameter, optimized with the exact method. Lower panel: The progression of KL-divergence, alpha value and $\frac{\partial \mathcal{L}}{\partial \alpha}$ *over the course of the optimization. The initial* α *was set to 5, and the learning rate* η *for* α *was 0.5. After convergence, the final* $\alpha \approx 3.96$*, resulting in a KL-divergence of 0.16 and a kNN recall of 0.91.*

Although this run showed slightly higher KL divergence (0.16 ± 0.004) – reflected in the overall appearance of the embedding – the α parameter evolved as anticipated: it gradually decreased toward a similar final value (3.91 ± 0.1) observed in the run initialized at $\alpha = 1$. However, 500 iterations did not fully suffice for α to converge; we still noted a steady trend in the gradient approaching zero as α drifted downward. This pattern reinforces the notion that, given more iterations, α would likely converge to the same optimum.

Exact computations of alpha gradient work reasonably well for datasets smaller than 10000 datapoints. However, as sample size increases, the computational complexity grows on the order of $\mathcal{O}(n^2)$. To address this challenge, the original t-SNE algorithm uses approximations – most notably the Barnes-Hut method – to compute low-dimensional affinities more efficiently. By leveraging quad-tree-based space partitioning, Barnes-Hut accelerates gradient computations for large datasets and reduces the complexity to approximately $\mathcal{O}(n \log n)$. In this approach, distant points are grouped into summary nodes within the quad-tree, decreasing the number of pairwise distance calculations.

In addition, crucial computational components – including loss-function gradient estimation and quad-tree operations – are implemented in Cython for performance gains, with parallelization through `prange` enabling multi-threaded execution. This combined approach efficiently handles large datasets.

Notably, the terms required to compute low-dimensional affinities also appear in the derived alpha-gradient expression. It is therefore intuitive to apply the Barnes-Hut approximation to the alpha-gradient computations, which we implemented. As an initial demonstration, we evaluated this Barnes-Hut–based approach ("BH method") on the Swiss Roll dataset with 3,000 points (Fig. 11).

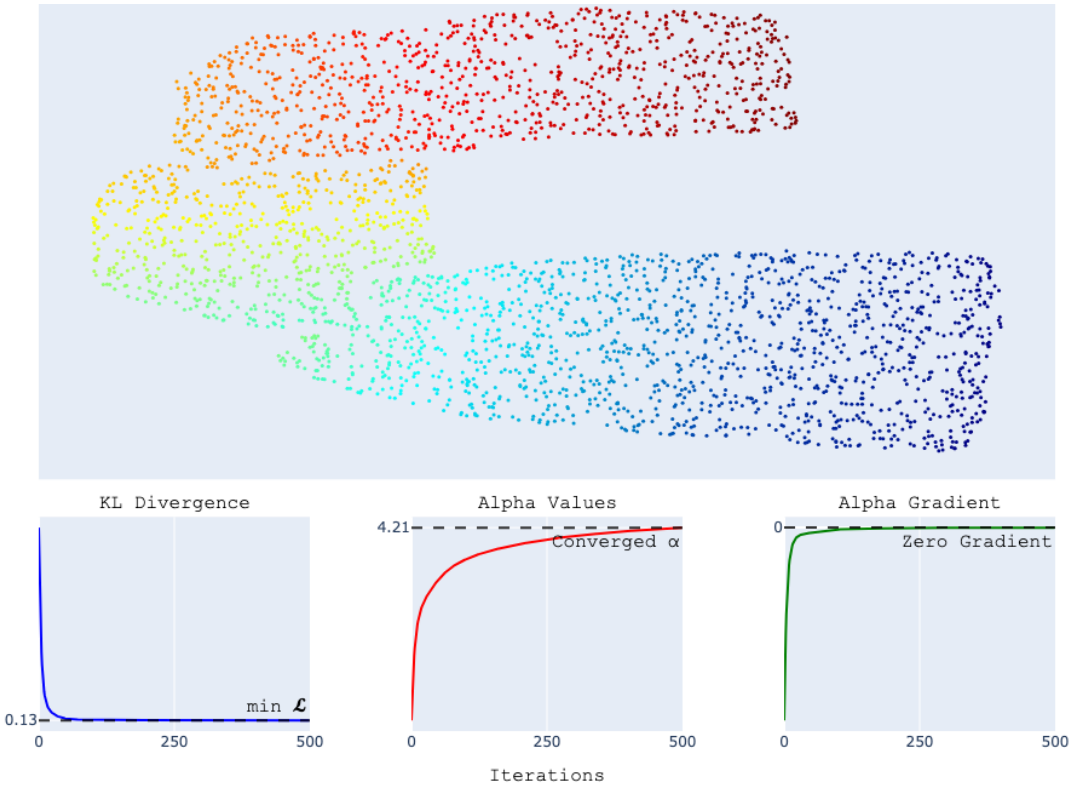

Figure 11. Top panel: example t-SNE embedding of the Swiss Roll with 3,000 samples with adaptive alpha parameter, optimized with the BH method. Lower panel: The progression of KL-divergence, alpha value and $\frac{\partial \mathcal{L}}{\partial \alpha}$ *over the course of the optimization. The initial* α *was set to 1, and the learning rate* η *for* α *was 0.5. After convergence, the final* $\alpha \approx 4.21$*, resulting in a KL-divergence of 0.13 and a kNN recall of 0.93.*

The alpha optimization proceeds as expected: the alpha gradient converges to zero, and alpha itself settles at approximately 4.41 ± 0.32 (averaged across 5 runs). Compared to the exact computation, the resulting embedding appears similar and achieves comparable performance metrics, with a mean KL divergence of 0.12 ± 0.011 and a mean kkNN recall of 0.93 ± 0.001. However, runtime did not improve at this dataset size (3,000 samples), likely because the overhead of building quad-trees outweighs any acceleration from Barnes-Hut approximations. To further examine runtime performance, we tested a larger Swiss Roll dataset (10,000 points) under the same conditions (Figure 12). As anticipated, the Barnes-Hut method offered a notable speedup, taking roughly 2 seconds per iteration versus 5 seconds for the exact method on the same hardware.

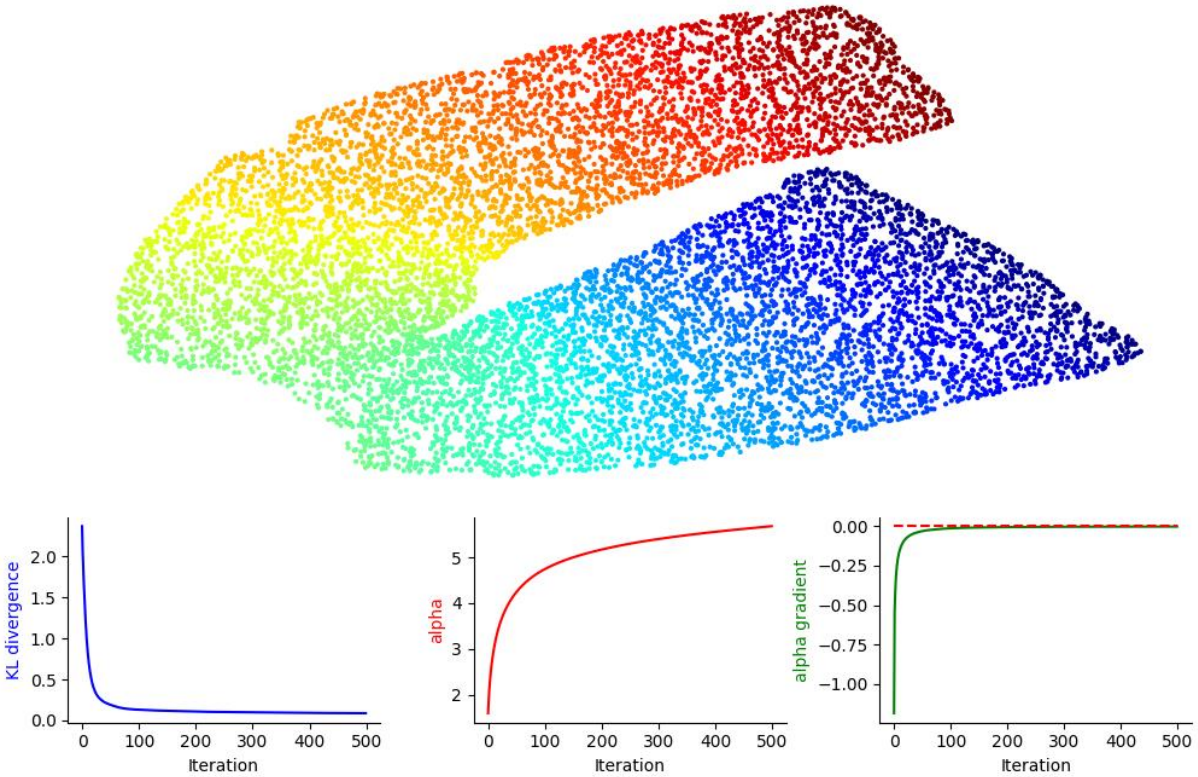

Figure 12. Top panel: example t-SNE embedding of the Swiss Roll with 10,000 samples with adaptive alpha parameter, optimized with the BH method. Lower panel: The progression of KL-divergence, alpha value and $\frac{\partial \mathcal{L}}{\partial \alpha}$ *over the course of the optimization. The initial* α *was set to 1, and the learning rate* η *for* α *was 0.5. After convergence, the final* $\alpha \approx 5.68$*, resulting in a KL-divergence of 0.09 and a kNN recall of 0.91.*

We can now examine in greater detail how the optimal α (in terms of KL divergence) changes with the sample size of the Swiss Roll dataset. Interestingly, contrary to our earlier findings starting at 3,000 samples, for this particular dataset the optimal α does not consistently increase with growing sample size.

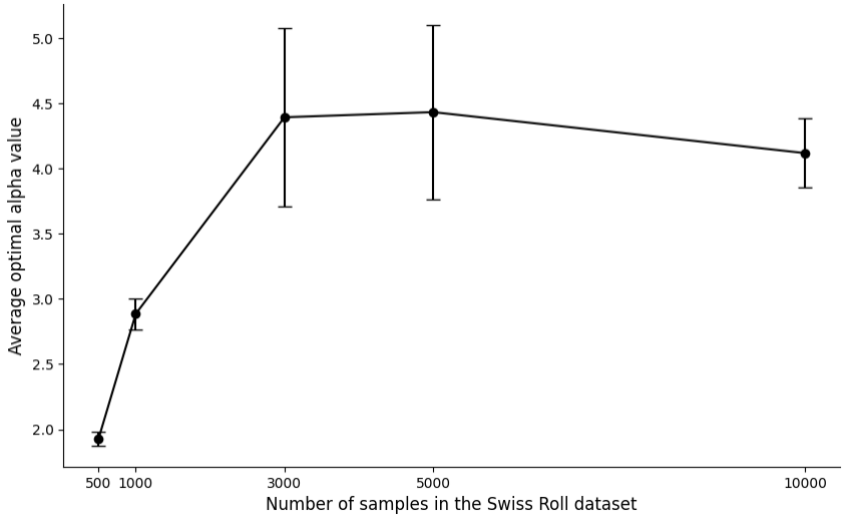

Figure 13. Optimal α (determined by the KL-divergence of t-SNE with alpha-optimization) plotted against the sample size for the Swiss Roll dataset.

Having confirmed satisfactory results on this toy dataset, we proceeded to assess our approach on real-world datasets with much larger sample size, specifically MNIST and an RNA-seq dataset. For the MNIST dataset, alpha optimization provided a modest improvement in KL divergence over standard t-SNE, reducing it from 2.76 to 2.61 and converging to a final $\alpha \approx 1.52$ (slightly above the default value of 1). A similar outcome was observed for the full RNA-seq dataset, where alpha optimization lowered the KL divergence from 1.59 to 1.35, resulting in $\alpha \approx 2.25$.

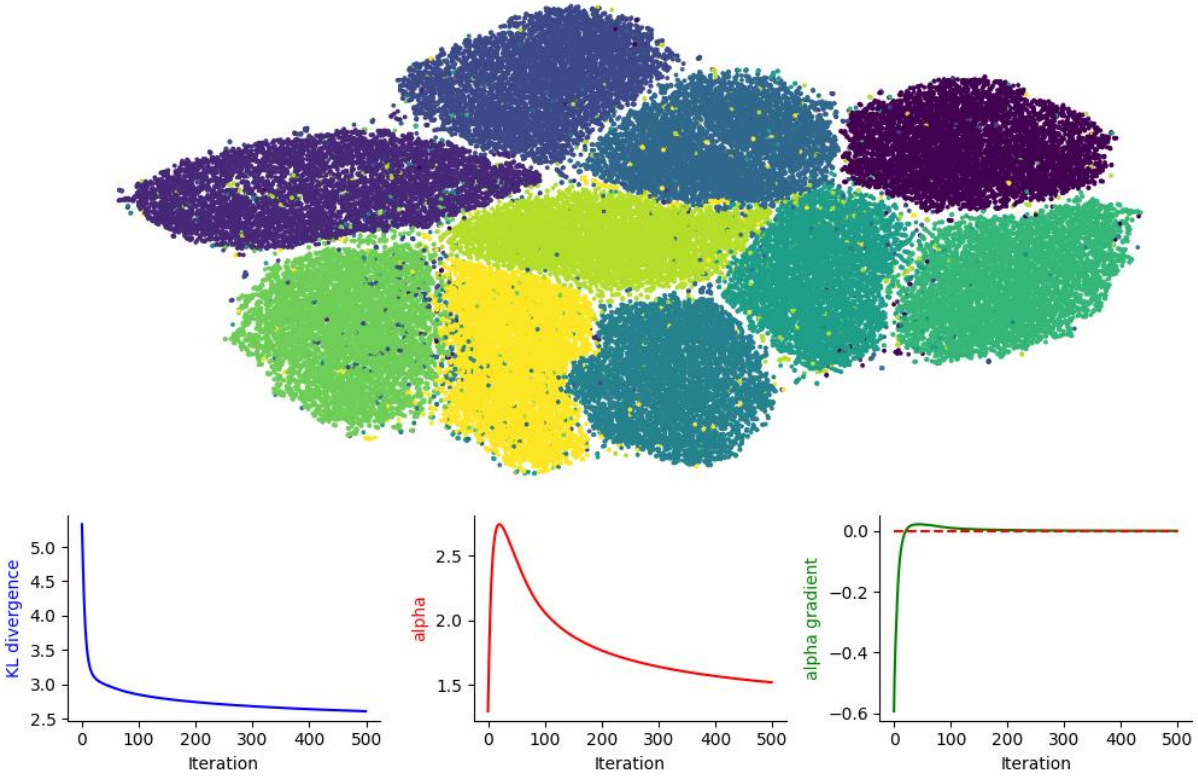

Figure 14. Top panel: t-SNE embedding of the MNIST with 70,000 samples with adaptive alpha parameter, optimized with the BH method. Lower panel: The progression of KL-divergence, alpha value and $\frac{\partial \mathcal{L}}{\partial \alpha}$ *over the course of the optimization. The initial α was set to 1, and the learning rate η for α was 0.5. After convergence, the final* $\alpha \approx 1.52$*, resulting in a KL-divergence of 2.61 and a kNN recall of 0.32.*

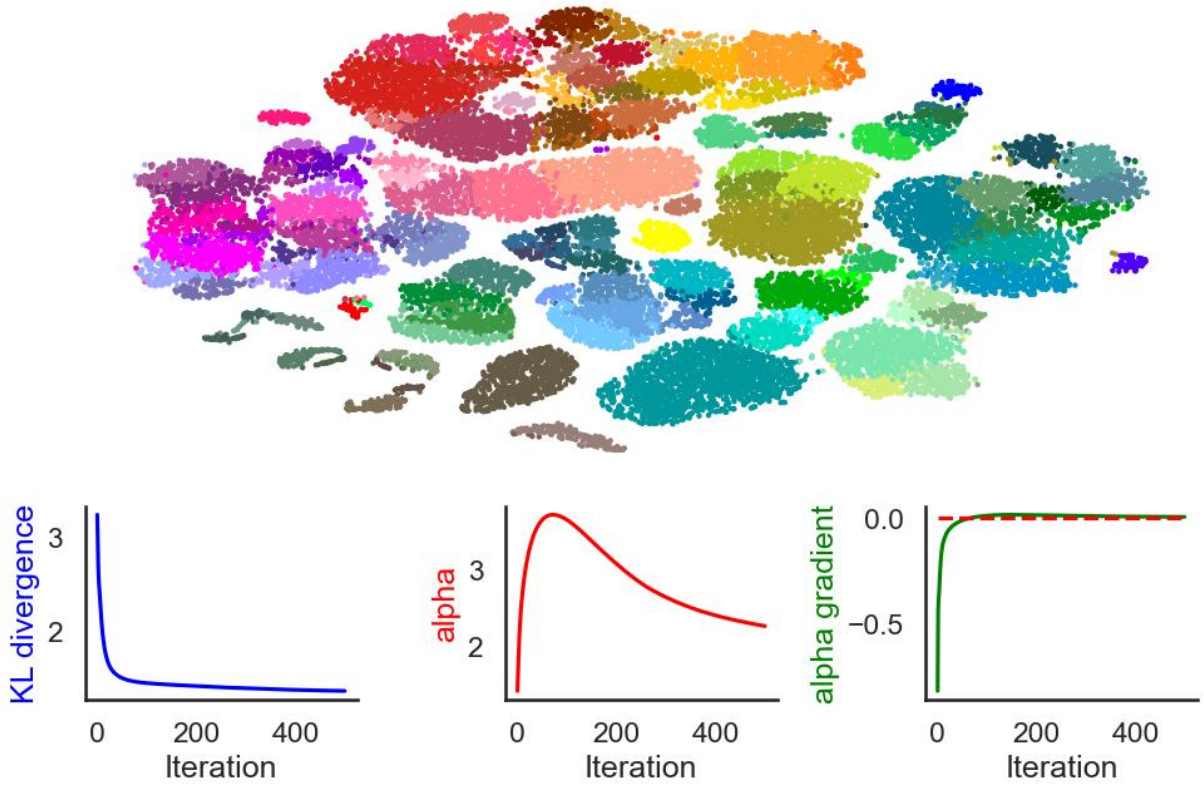

Figure 15. Top panel: t-SNE embedding of the RNA-seq dataset with 25,000 samples with adaptive alpha parameter, optimized with the BH method. Lower panel: The progression of KL-divergence, alpha value and $\frac{\partial \mathcal{L}}{\partial \alpha}$ *over the course of the optimization. The initial α was set to 1, and the learning rate η for α was 0.5. After convergence, the final* $\alpha \approx 2.25$*, resulting in a KL-divergence of 1.35 and a kNN recall of 0.38.*

5. Discussion

In this study, we investigated how the t-SNE method behaves under variations of the degree-of-freedom parameter α, and demonstrated a gradient-based approach for α-optimization. In attempt to quantitively estimate the quality of resulting t-SNE embeddings and compare one embedding with another objectively we employed two metrics: KL-divergence (the loss function of the t-SNE algorithm, which is directly minimized during the embedding process) and kNN recall reflecting the preservation of local structure.

Our findings across both toy and real-world datasets reveal that no single α value produces universally optimal embeddings. Rather, the best α depends on both the dataset (e.g., Swiss Roll vs. MNIST vs. RNA-seq) and even on subsets of data within the same dataset.

Results on the Swiss Roll dataset highlight that $\alpha > 1$ can to some extent mitigate the well-known "over-fragmentation" effect, leading to more continuous embeddings. A grid-based search revealed that α in the range of approximately 2–8 better preserve local relationships and unroll the manifold.

Similarly, the single-cell RNA-seq data often achieves lower KL-divergence with α larger than 1 or even close to 4.

Conversely, the best embeddings for the whole MNIST dataset (as well as for all digits individually except for 1) in these experiments were achieved with alpha equal to 1.

Additionally, our experiments showed that the "optimal" α may change with sample size, suggesting that the distribution of points strongly influences how the t-SNE kernel behaves.

It is worth mentioning here, that if kNN recall is used as an objective metrics, the optimal α can be different from those based on KL-divergence. However, we have to stress that no metric is inherently superior than another, as they reflect different aspects of the embedding: KL-divergence focuses on aligning the probability distributions between original and embedded spaces, whereas kNN recall shows the preservation of local neighborhoods. The choice of metric depends on the specific goals of the analysis and the structures one aims to capture.

Building on obtained results, we developed a gradient-based method to adaptively optimize α by taking the derivative of the t-SNE loss function (KL-divergence) with respect to α. Applying this approach to our selected datasets yielded α values that matched the best fixed α from our grid searches (e.g., $\alpha \approx 3.3 - 3.8$ for the Swiss Roll). We further integrated the Barnes-Hut approximation for computing the α-gradient, which lowered computational overhead for larger datasets (10,000 or more samples). Notably, for MNIST and the full RNA-seq dataset, the adaptively optimized α ultimately converged to values between 1.5 and 2.3, offering modest but measurable gains in KL-divergence relative to standard t-SNE ($\alpha = 1$).

It should be noted, that this work has certain limitations to consider in follow up research. Beyond the datasets and metrics considered here, investigating a wider spectrum of real-world data would help clarify the applicability of α-optimization. This could be complemented by employing multiple different objective metrics to better assess embedding quality.

Another direction of further improving t-SNE algorithm would involve exploring the interplay between α and other t-SNE parameters, particularly perplexity, which is sadly beyond the scope of this research. Nevertheless, some of the preliminary experiments on the Swiss Roll with higher perplexity values (e.g., 50) and fixed $\alpha >> 1$ resulted in much better embeddings, with lower KL-divergency metrics (see Appendix).

Finally, to take full advantage of α-optimization, it should be seamlessly integrated into openTSNE or other comparable frameworks. We expect that with proper implementation, the added optimization step would not substantially increase computational or time requirements, unlike the more manual approach presented in this work. By making α-optimization a routine

part of generating t-SNE embeddings, such frameworks could broaden the applicability and impact of this dimensionality reduction technique in research.

6. Appendix

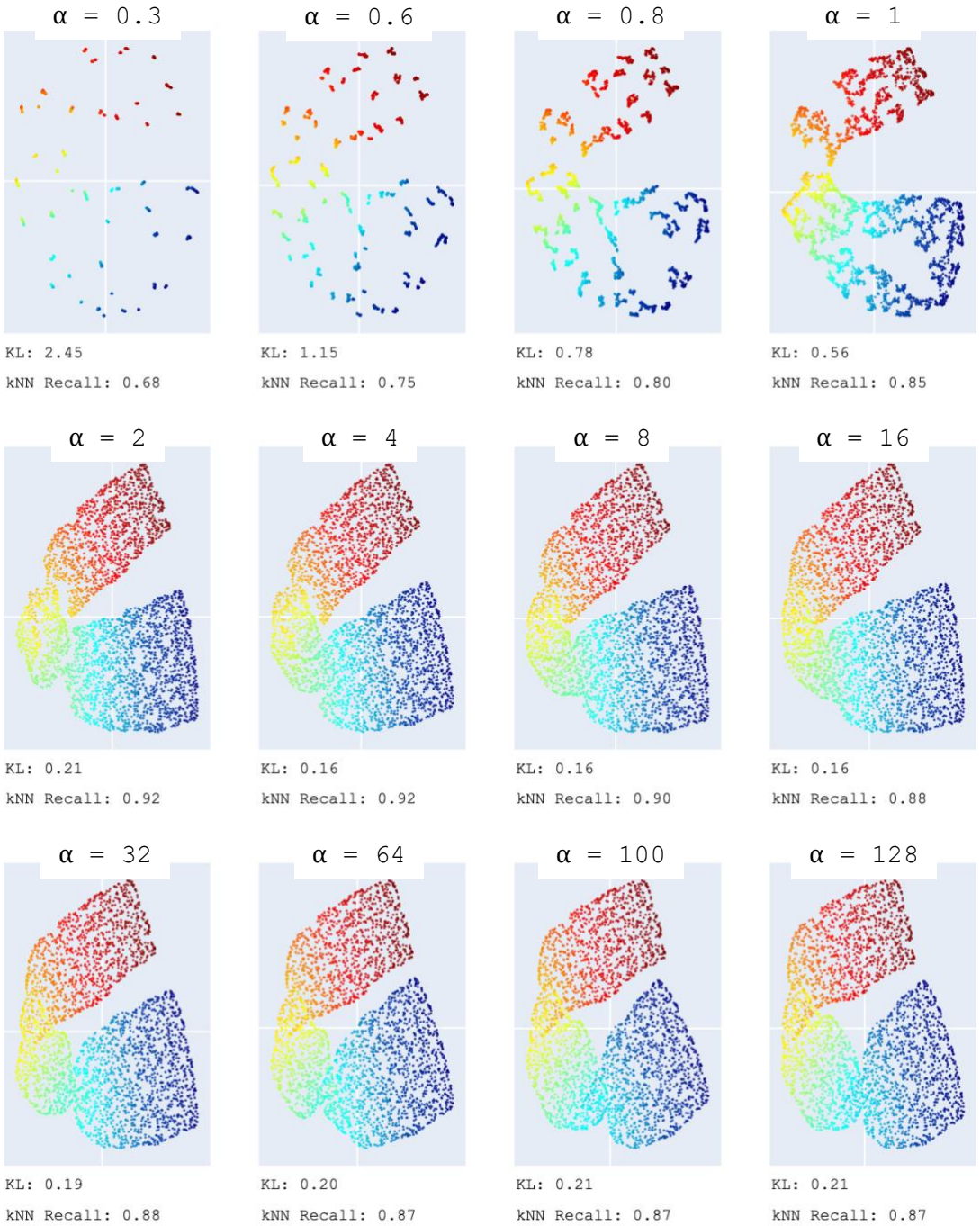

Figure 16. Example t-SNE embeddings of the Swiss Roll dataset (3000 samples) with the whole array of α values

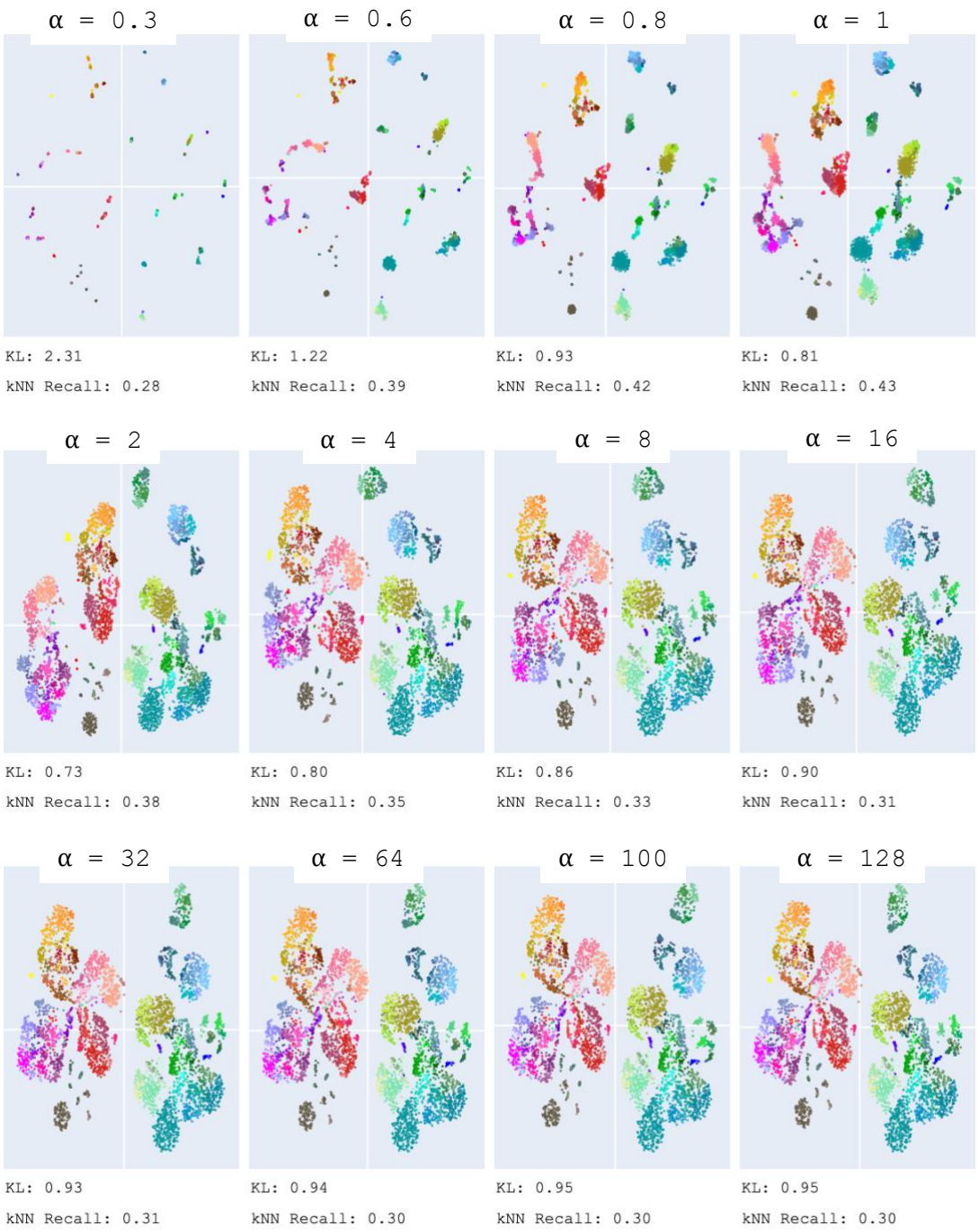

Figure 17. Example t-SNE embeddings of the RNA-seq dataset (5000 samples) with the whole array of α values

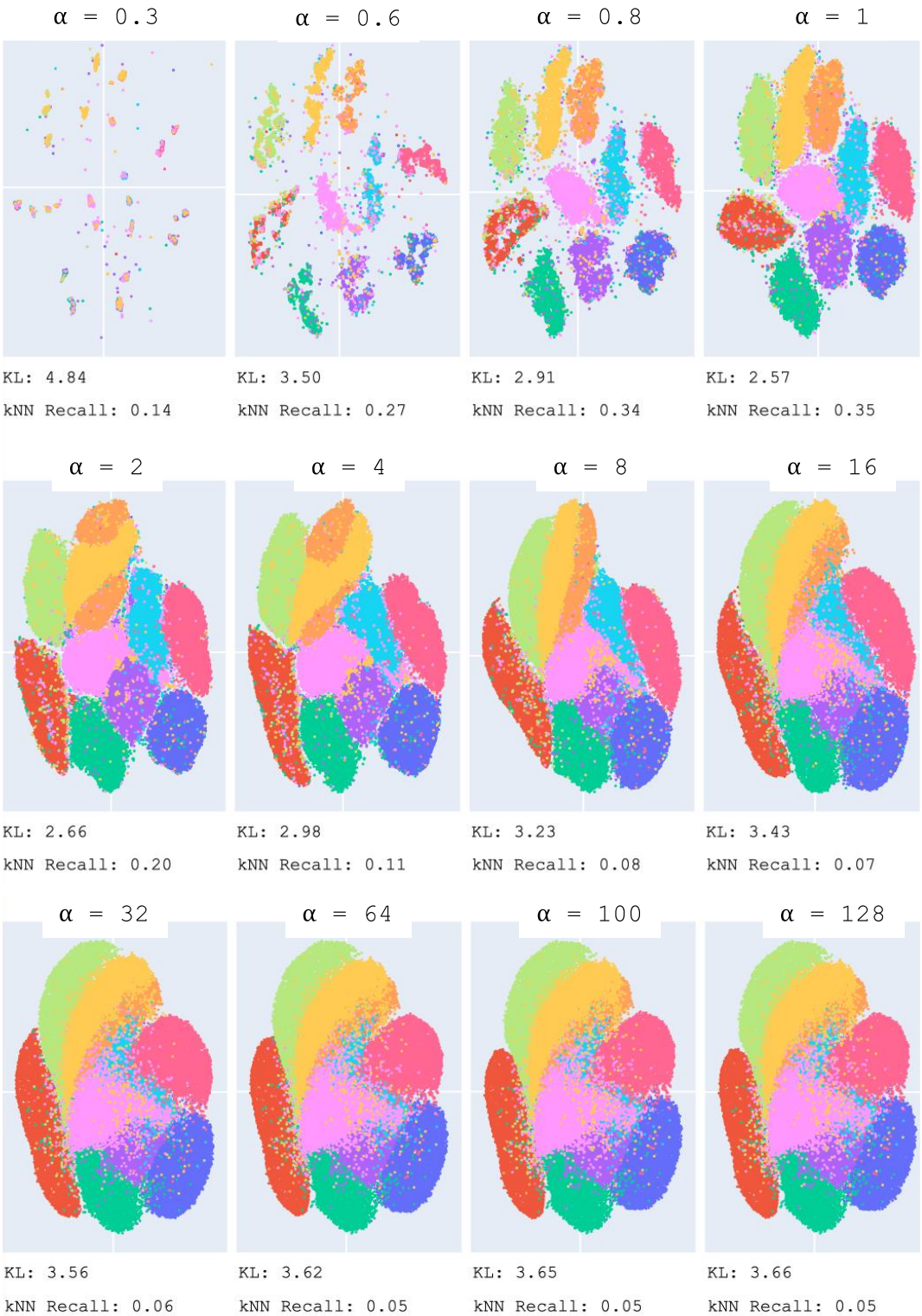

Figure 18. Example t-SNE embeddings of the MNIST dataset (70000 samples) with the whole array of α values

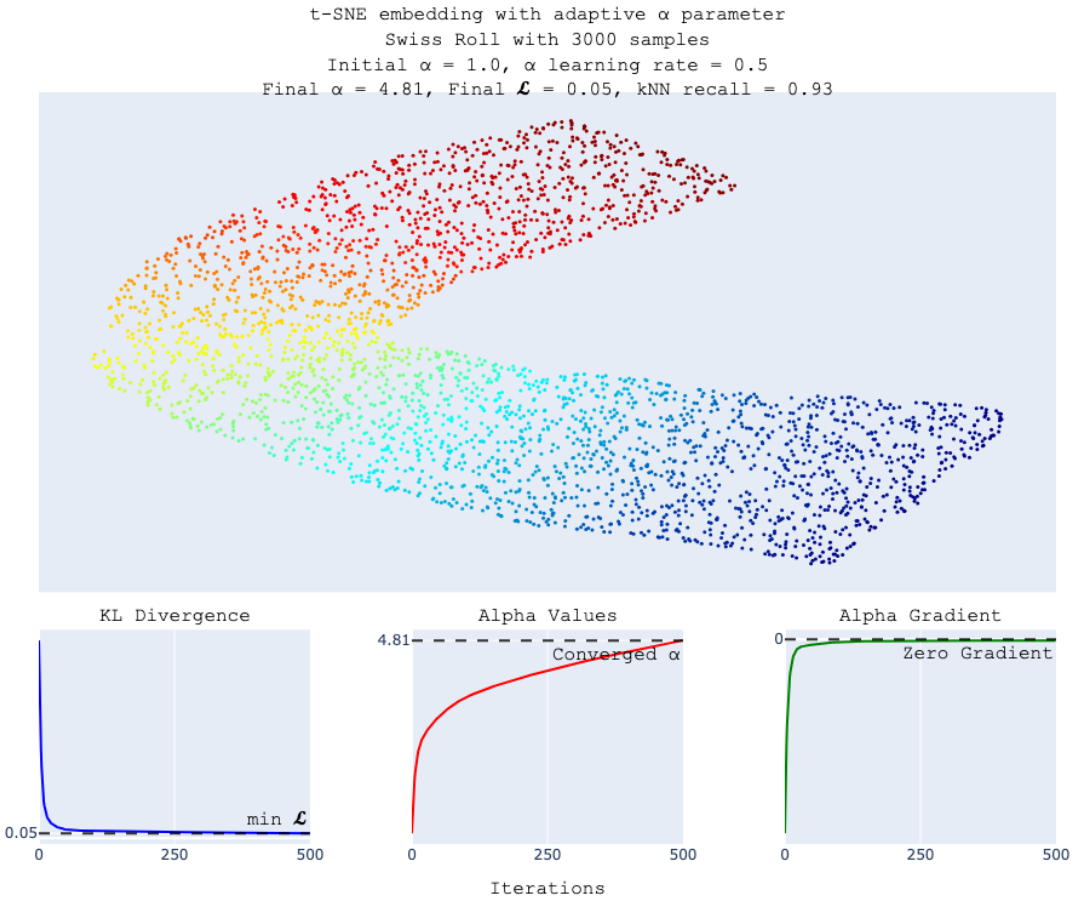

Figure 19. Top panel: example t-SNE embedding of the Swiss Roll with 3,000 samples with adaptive alpha parameter, optimized with the exact method. Lower panel: The progression of KL-divergence, alpha value and $\frac{\partial \mathcal{L}}{\partial \alpha}$ *over the course of the optimization. The initial* α *was set to 1, and the learning rate* η *for* α *was 0.5. After convergence, the final* $\alpha \approx 4.81$*, resulting in a KL-divergence of 0.05 and a kNN recall of 0.93.*

Derivation of the alpha gradient:

The loss function, up to a constant term $\sum_{i,j} p_{i,j} \log p_{i,j}$, can be rewritten as follows:

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

Let:

$$s_{i,j} = 1 + \frac{d_{i,j}^2}{\alpha}$$

$$y_{i,j} = s_{i,j}^{-\alpha}$$

$$Z = \sum_{k \neq \ell} y_{k,\ell}$$

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

Then, the loss function becomes:

$$\mathcal{L} = -\sum_{i,j} p_{i,j} \left(\log y_{i,j} - \log Z \right) = -\sum_{i,j} p_{i,j} \log y_{i,j} + \log Z.$$

The gradient is:

$$\frac{\partial \mathcal{L}}{\partial \alpha} = -\sum_{i,j} p_{i,j} \frac{\partial \log y_{i,j}}{\partial \alpha} + \frac{\partial \log Z}{\partial \alpha}.$$

Since:

$$\log y_{i,j} = -\alpha \log s_{i,j}$$

we have:

$$\frac{\partial \log y_{i,j}}{\partial \alpha} = -\log s_{i,j} - \alpha \frac{\partial \log s_{i,j}}{\partial \alpha}.$$

But:

$$\frac{\partial \log s_{i,j}}{\partial \alpha} = \frac{\partial}{\partial \alpha}\left(\log\left(1 + \frac{d_{i,j}^2}{\alpha}\right)\right) = -\frac{d_{i,j}^2}{\alpha^2 s_{i,j}}.$$

Therefore:

$$\frac{\partial \log y_{i,j}}{\partial \alpha} = -\log s_{i,j} + \frac{\alpha d_{i,j}^2}{\alpha^2 s_{i,j}} = -\log s_{i,j} + \frac{d_{i,j}^2}{\alpha s_{i,j}}$$

Now we compute $\frac{\partial \log Z}{\partial \alpha}$:

$$\frac{\partial \log Z}{\partial \alpha} = \frac{1}{Z}\frac{\partial Z}{\partial \alpha} = \frac{1}{Z}\sum_{k \neq \ell} \frac{\partial y_{k,\ell}}{\partial \alpha}.$$

But:

$$\frac{\partial y_{k,\ell}}{\partial \alpha} = y_{k,\ell}\left(-\log s_{k,\ell} + \frac{d_{k,\ell}^2}{\alpha + d_{k,\ell}^2}\right).$$

So:

$$\frac{\partial \log Z}{\partial \alpha} = \sum_{k \neq \ell} q_{k,\ell}\left(-\log s_{k,\ell} + \frac{d_{k,\ell}^2}{\alpha + d_{k,\ell}^2}\right).$$

Putting it all together:

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

The final expression therefore:

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

7. References

1. Gove, R., Cadalzo, L., Leiby, N., Singer, J. M., & Zaitzeff, A. (2022). New guidance for using t-SNE: Alternative defaults, hyperparameter selection automation, and comparative evaluation. *Visual Informatics*, *6*(2), 87–97.
2. Kitazono, J., Grozavu, N., Rogovschi, N., Omori, T., & Ozawa, S. (2016). t-Distributed Stochastic Neighbor Embedding with Inhomogeneous Degrees of Freedom. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9949 LNCS*, 119–128.
3. Kobak, D., & Berens, P. (2019). The art of using t-SNE for single-cell transcriptomics. *Nature Communications*, *10*(1).
4. Kobak, D., Linderman, G., Steinerberger, S., Kluger, Y., & Berens, P. (2019). Heavy-Tailed Kernels Reveal a Finer Cluster Structure in t-SNE Visualisations. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Cham: Springer International Publishing*.
5. Linderman, G. C., & Steinerberger, S. (2019). Clustering with t-SNE, Provably. *Https://Doi.Org/10.1137/18M1216134*, *1*(2), 313–332.
6. Marsland, S. (2011). Machine Learning. In *Machine Learning* (2nd ed.). Chapman and Hall/CRC.
7. Schmidt, B. (2018). Stable random projection: lightweight, general-purpose dimensionality reduction for digitized libraries. *Journal of Cultural Analytics*.
8. Tasic, B., Yao, Z., Graybuck, L. T., Smith, K. A., Nguyen, T. N., Bertagnolli, D., Goldy, J., Garren, E., Economo, M. N., Viswanathan, S., Penn, O., Bakken, T., Menon, V., Miller, J., Fong, O., Hirokawa, K. E., Lathia, K., Rimorin, C., Tieu, M., … Zeng, H. (2018). Shared and distinct transcriptomic cell types across neocortical areas. *Nature 2018 563:7729*, *563*(7729), 72–78.
9. Van Der Maaten, L. (2009). Learning a Parametric Embedding by Preserving Local Structure. *Artificial Intelligence and Statistics*, 384–391.
10. Van Der Maaten, L., & Hinton, G. (2008). Visualizing Data using t-SNE. In *Journal of Machine Learning Research* (Vol. 9).