OPTIMAL T-KERNEL SHAPE IN T-SNE

Anonymous authors
Paper under double-blind review

ABSTRACT

Neighbor-embedding methods have become an established tool for visualizing complex high-dimensional datasets in various scientific disciplines. The celebrated t-SNE algorithm uses the Cauchy kernel to define embedding similarity between points, often strongly outperforming the original SNE that uses the Gaussian kernel. However, a theoretical justification and an empirical investigation of the optimal kernel shape have been lacking. Here we address this gap by studying the optimal kernel shape (t-distribution degree of freedom) in t-SNE. We use a simplified uniform ansatz to derive the optimal kernel shape and show that the tail thickness increases with intrinsic dimensionality. Implementing a fast kernel-shape optimization in both the Barnes–Hut and the FFT-based approximations of t-SNE, we qualitatively confirm theoretical predictions on simulated datasets. Finally, we conduct a large-scale empirical investigation of the learned kernel shapes across various datasets, and show that optimizing the kernel can reduce t-SNE's tendency to over-fragment unstructured, low-dimensional data and can lead to more accurate visualizations of single-cell RNA-seq data.

1 INTRODUCTION

Neighbor-embedding methods have enjoyed widespread adoption in recent years and are routinely used in various scientific fields for visualizing high-dimensional data (de Bodt et al., 2025). All of these methods can be seen as creating a low-dimensional layout of a weighted k-nearest neighbor (kNN) graph of the data. In the (t-)SNE family of algorithms (Hinton & Roweis, 2003; van der Maaten & Hinton, 2008), the embedding similarity between points is captured by a similarity kernel, chosen to be Gaussian in SNE and Cauchy in t-SNE.

Both choices are known to be suboptimal for some datasets. In particular, low-dimensional manifolds such as Swiss roll get severely over-fragmented in t-SNE (Figure 1b), while high-dimensional datasets such as MNIST tend to get overlapping clusters in SNE (Figure 1e). Nevertheless, the *de facto* standard in the field is to use the Cauchy similarity for all datasets, potentially creating

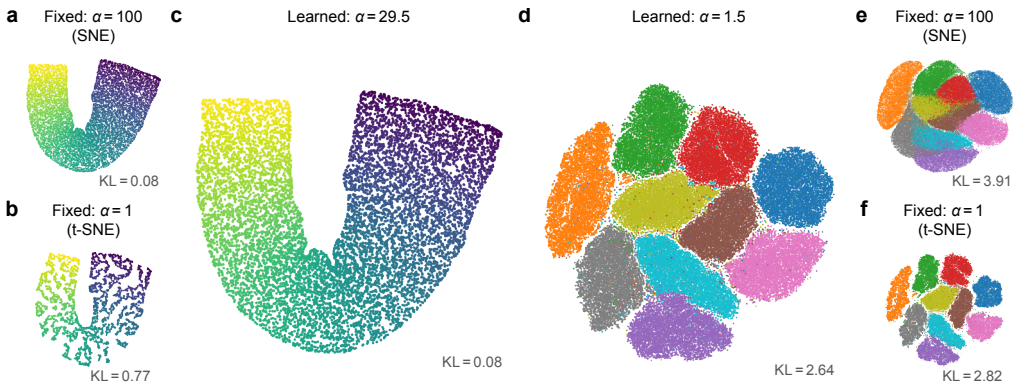

Figure 1: **Overview. (a–c)** Embedding of Swiss roll ($n = 10\,000$) using SNE, standard t-SNE, and our implementation with learned kernel shape (all with spectral initialization). The optimum is close to SNE. **(d–f)** The same for MNIST ($n = 70\,000$). Here the optimum is close to standard t-SNE.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

misleading embeddings if the intrinsic dimensionality of the data is low. Furthermore, the choice of the Cauchy similarity in t-SNE has been based on a heuristic argument, and there is no known principled reason for why it should be the optimal choice.

In this paper, we:

1. show that making the kernel shape (governed by the t-distribution degree of freedom in t-SNE) a learnable parameter yields adequate embeddings datasets with both low and high intrinsic dimensionality (Figure 1);
2. provide two fast implementations of kernel shape learning in t-SNE, based on the Barnes–Hut algorithm and on the FFT interpolation;
3. analyze the optimal degree of freedom theoretically based on a simplifying uniform ansatz;
4. empirically study the optimal degree of freedom across over 600 real-world datasets.

2 Related Work

In the original stochastic neighbor embedding (SNE) algorithm (Hinton & Roweis, 2003), the embedding is constructed by matching pairwise Gaussian affinities (similarities) in the embedding space to Gaussian affinities in the original high-dimensional space. Later, t-SNE (van der Maaten & Hinton, 2008) adopted Cauchy affinities in the embedding space, which are given by t-distribution kernel with $\nu = 1$ degree of freedom. The original t-SNE paper provided a heuristic motivation for this choice of kernel, but their motivation applies to a t-distribution kernel with any value of ν equally well. Subsequent work on parametric t-SNE (van der Maaten, 2009) suggested that the one should choose $\nu = d - 1$ where d is embedding dimensionality, but this again was only based on a heuristic argument. Note that the original SNE corresponds to $\nu \to \infty$.

Several works investigated t-SNE with $\nu \neq 1$ (Yang et al., 2009). In particular, Kobak et al. (2019) provided a fast approximate implementation for any ν and argued that using an even heavier-tailed kernel than $\nu = 1$ can reveal meaningful fine-scale cluster structure present in the data.

Some other neighbor-embedding algorithms use slightly different similarity kernels (McInnes et al., 2018; Wang et al., 2021; Islam & Fleischer, 2025), but they mostly stay close to the Cauchy shape.

Actually optimizing the value of ν for a given dataset has not attracted a lot of attention so far. Apart from van der Maaten (2009) who suggested optimizing ν in parametric t-SNE, we are only aware of the work by Kitazono et al. (2016) who suggested inhomogeneous t-SNE where a separate ν_i is optimized for each data point. However, we are not aware of either any existing fast implementations of learnable ν or of any theoretical or empirical analysis regarding what governs the optimal ν^*.

3 Degree of freedom in t-SNE

3.1 Background

Given n high-dimensional data vectors $\mathbf{x}_i \in \mathbb{R}^D$, the goal of t-SNE is to obtain low-dimensional vectors $\mathbf{z}_i \in \mathbb{R}^d$ preserving neighborhood structure as much as possible. The loss is given by the Kullback–Leibler (KL) divergence between high-dimensional and low-dimensional affinities:

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

where both sets of affinities are normalized to sum to 1. High-dimensional affinities are supported on a k-NN graph of the data and traditionally have Gaussian weights:

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

where σ_i is chosen for each point i to make the entropy of $p_{j|i}$ equal to 2^P, where P is the desired perplexity. Most implementations use $P = 30$ as the default value. All $p_{j|i}$ values beyond the largest $K = 3P$ are set to zero in most modern implementations (van der Maaten, 2014), making

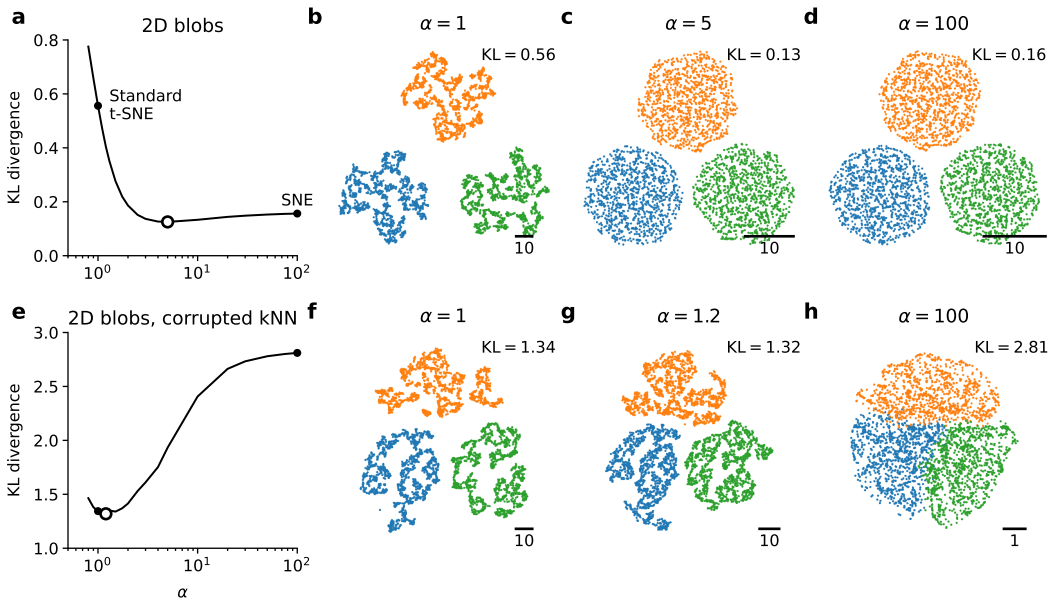

Figure 2: **Motivation. (a–d)** For t-SNE embedding of 2D Gaussian mixture blobs ($n = 3\,000$; perplexity 30), the optimal α is around 5. Standard t-SNE with $\alpha = 1$ leads to the formation of spurious mini-clusters. If perplexity calibration is turned off, the optimal α goes to infinity, resulting in SNE (Figure S2). **(e–h)** If the affinity matrix is corrupted by adding random connections between points ($0.37/n$ chance that a given pair of points gets connected), optimal α decreases to around 1.

the p_{ij} matrix sparse (modern implementations often use an approximate k-NN graph with $k = K$ to accelerate the computation of p_{ij}). Instead of Gaussian affinities, one can also use uniform affinities with identical $p_{j|i} = 1/K$ values (Artemenkov & Panov, 2020; Böhm et al., 2022). Either way, low-dimensional affinities are given by

$$q_{ij} = \frac{\xi(\|\mathbf{z}_i - \mathbf{z}_j\|)}{\sum_{k\neq l}\xi(\|\mathbf{z}_k - \mathbf{z}_l\|)} = \frac{\xi(d_{ij})}{\sum_{k\neq l}\xi(d_{kl})} = \frac{\xi(d_{ij})}{Z}, \tag{3}$$

where $\xi(\cdot)$ is some similarity kernel. For convenience, we will denote the denominator as Z.

SNE uses the Gaussian similarity kernel $\xi(d) = \exp(-d^2)$, while t-SNE uses the Cauchy similarity kernel $\xi(d) = 1/(1+d^2)$. Here we follow Kobak et al. (2019) and use a family of kernels given by

$$\xi(d) = \frac{1}{(1 + d^2/\alpha)^\alpha}, \tag{4}$$

which is a scaled t-distribution with $\nu = 2\alpha - 1$ degrees of freedom. The same family in a slightly different parameterization was also used by Yang et al. (2009). Note that $\alpha = 1$ gives standard t-SNE, while $\alpha \to \infty$ yields SNE; we will consider $\alpha = 100$ as an adequate approximation of SNE.

3.2 Optimizing the degree of freedom

To illustrate that the structure of the k-NN graph can strongly affect the optimal α, we consider a dataset of 2D Gaussian blobs (Figure 2a). The loss landscape $\mathcal{L}(\alpha)$ has its minimum at $\alpha^* \gg 1$ and is almost flat for $\alpha > \alpha^*$ (this holds for uniform affinities as well; Figure S1). If the perplexity calibration is switched off (i.e. $\sigma_i = \text{const}$), then $\alpha = 100$ is empirically the best (Figure S2). But if the k-NN graph is distorted by adding some random connections between points (Figure 2b), then the shape of $\mathcal{L}(\alpha)$ changes dramatically with the optimal $\alpha^* \approx 1$ and large α values yielding markedly worse KL values. These observations motivate the need to optimize α for a given dataset.

A straightforward calculation shows that the derivative of $\mathcal{L}$ with respect to α is given by

$$\frac{\partial \mathcal{L}}{\partial \alpha} = \sum_{ij}(p_{ij} - q_{ij})\left(\log\left(1 + \frac{d_{ij}^2}{\alpha}\right) - \frac{d_{ij}^2}{\alpha + d_{ij}^2}\right). \tag{5}$$

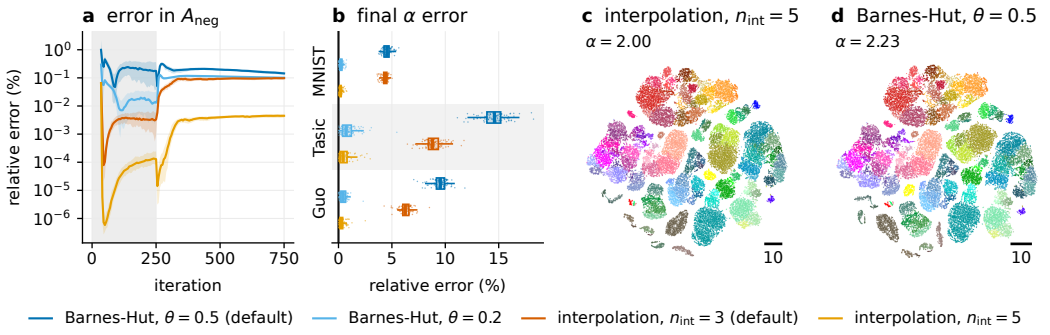

Figure 3: **Approximation accuracy.** **(a)** Relative error $|\tilde{A}_{\text{neg}} - A_{\text{neg}}|/A_{\text{neg}}$ on the Tasic dataset (Tasic et al., 2018) subsampled to $n = 10\,000$, where $\tilde{A}_{\text{neg}}$ is the approximation and A_{neg} the exact value. The optimization followed the exact gradient. Lines show the median and bands the 10–90% range over 100 random initializations. Early exaggeration iterations are shaded in gray. **(b)** The relative error $|\tilde{\alpha} - \alpha|/\alpha$ of the final learned α parameter over 100 random initializations. Here, each scheme runs its own optimization starting from the same initialization. Boxes show the quartiles and the median over 100 initializations. The datasets are MNIST, Tasic, and Guo, all subsampled to $n = 10\,000$. **(c, d)** Embeddings of the full Tasic dataset ($n = 23\,822$) with two approximation schemes starting from the same initialization. Although the learned α values differ by 12%, the visualizations look very similar.

A similar expression in terms of ν is given in van der Maaten (2009). This partial derivative can be can expanded into positive and negative terms A_{pos} and A_{neg}:

$$\frac{\partial \mathcal{L}}{\partial \alpha} = \underbrace{\sum_{ij} p_{ij} \left(\log\left(1 + \frac{d_{ij}^2}{\alpha}\right) - \frac{d_{ij}^2}{\alpha + d_{ij}^2} \right)}_{A_{\text{pos}}} - \underbrace{\sum_{ij} q_{ij} \left(\log\left(1 + \frac{d_{ij}^2}{\alpha}\right) - \frac{d_{ij}^2}{\alpha + d_{ij}^2} \right)}_{A_{\text{neg}}}. \tag{6}$$

The positive term is fast to compute because the p_{ij} matrix is sparse and the sum only goes over the edges of the k-NN graph. The negative term runs over all pairs has naïve complexity $\mathcal{O}(n^2)$ and requires approximations, similar to the repulsive forces arising in $\partial \mathcal{L}/\partial \mathbf{z}_i$.

We base our implementation on the `openTSNE` library (Poličar et al., 2024), which implements two approximations: Barnes–Hut (Yang et al., 2013; van der Maaten, 2014) and FFT-accelerated interpolation from FIt-SNE (Linderman et al., 2019). Both schemes approximate sums of the form

$$\sum_j K\big(\|\mathbf{z}_i - \mathbf{z}_j\|\big)\, w_j, \tag{7}$$

for every point i, with a kernel K and per-point weights w_j. A_{neg} can be written in this form:

$$A_{\text{neg}} = \frac{1}{Z} \sum_i \Big(\sum_j K(d_{ij}) \cdot 1 \Big), \quad K(d) = \xi(d) g(d), \quad g(d) = \log\left(1 + \frac{d^2}{\alpha}\right) - \frac{d^2}{\alpha + d^2}, \tag{8}$$

with weights $w_j = 1$. We implemented $\partial \mathcal{L}/\partial \alpha$ calculation in both schemes. Gradient descent was always initialized at $\alpha = 1$. Computational overhead was moderate, with runtime increase up to 25% compared to using fixed $\alpha = 1$, usually less.

Barnes–Hut The Barnes–Hut algorithm builds a space-partitioning tree on the embedding and evaluates Eq. 7 for point i by traversing from the root. When a cell is small and far enough from $\mathbf{z}_i$, i.e. $r_{\text{cell}}/\|\mathbf{z}_i - \mathbf{z}_{\text{cell}}\| < \theta$, where r_{cell} is the cell's width and $\mathbf{z}_{\text{cell}}$ the centroid of its N_{cell} points, the whole cell contributes $N_{\text{cell}} \cdot K(\|\mathbf{z}_i - \mathbf{z}_{\text{cell}}\|)$. Otherwise, the traversal descends into the cell's children. Larger θ corresponds to a coarser approximation. At $\theta = 0$ the sum is computed exactly.

FIt-SNE FIt-SNE evaluates Eq. 7 for a bounded, translation-invariant kernel K by spreading the weights onto a set of interpolation points, computing the interactions between the interpolation

points, and interpolating the result back onto the data points. The interactions of equispaced interpolation points form a convolution that can be accelerated by the fast Fourier transform. Computing $\partial\mathcal{L}/\partial\alpha$ thus adds one kernel sum, i.e. one extra FFT convolution, on each gradient descent iteration.

Approximation accuracy We profiled the approximation errors of both negative-gradient approaches. We obtained the exact negative gradient by setting $\theta = 0$ in the Barnes–Hut approximation. We found that both schemes approximated A_{neg} to within about 0.1% accuracy after the so-called early exaggeration phase (the first 250 gradient descent iterations when p_{ij} values are temporarily increased). FIt-SNE approach was more accurate than Barnes–Hut, especially with increased number of interpolation points (Figure 3a).

Although the errors at each individual iteration are small, they can accumulate over the course of the full optimization. We found that after an approximate optimization, the final learned alpha could be up to 15% off the value obtained using exact optimization (Figure 3b). However, the difference of 15%, e.g. between $\alpha = 2.0$ and $\alpha = 2.3$, barely impacted the resulting visualization (Figure 3c–d).

4 Theoretical analysis of the optimal degree of freedom

The road map of our analysis is as follows. First, we note that if the input data is two-dimensional, then the optimal α^* is, under some simplifying assumptions, infinite. Second, we consider a case of uniformly distributed high-dimensional data and show that the affinity profile over neighbor ranks converges to a power law for large input dimension D. Third, we show that under further simplifying assumptions, the optimal α is determined by the power law exponent in the limit of large D.

Theorem 4.1. *If high-dimensional data lies in a two-dimensional affine subspace, if perplexity calibration is replaced by fixed-width Gaussians $\sigma_i = \mathrm{const}$, and if p_{ij} affinities are obtained using global normalization so that $p_{ij} \propto \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2)$ and not truncated ($K = n - 1$), then the infimum of the KL divergence is zero and it is approached as $\alpha \to \infty$.*

Proof. The KL divergence converges to zero when the q_{ij} values approach the respective p_{ij}. This happens by arranging all embedding points in exactly the same configuration in $\mathbb{R}^2$ as their high-dimensional counterparts in the 2D affine subspace, scaling this configuration by $1/(\sqrt{2}\sigma)$, and letting $\alpha \to \infty$ which makes the output kernel Gaussian. □

This is confirmed in Figure S2. For actual t-SNE (with perplexity calibration and $K = 3P$) on 2D input data, the optimal α^* is finite, but remains $\gg 1$ close to SNE (Figure 2a).

Proposition 4.2. *Let p_m, $m \in \{1 \dots K\}$, be the average of $p_{i|j}$ values where i is the m-th closest neighbor of j. For n points from a locally uniform distribution with density f in D dimensions,*

$$p_m \approx \frac{e^{-Am^{2/D}}}{\sum_{l=1}^{K} e^{-Al^{2/D}}} \text{ with } A = \frac{(nfV_D)^{-2/D}}{2\sigma^2}; \qquad p_m \xrightarrow[D\to\infty]{} \frac{m^{-\beta}}{\sum_{l=1}^{K} l^{-\beta}}, \tag{9}$$

where V_D denotes the volume of unit ball in D dimensions, σ is adjusted to yield the desired perplexity $P \in (1, K)$, and β is the exponent for which the power law $m^{-\beta}$, normalized over $m \in \{1, \dots, K\}$, has perplexity P.

Proof. The expected number of points in a ball of radius r is $nfV_D r^D$. It follows that the radius with expected m points inside is $r_m = (nfV_D/m)^{-1/D}$. Plugging this distance into the Gaussian kernel and normalizing over the K retained neighbors gives the approximate result for finite D.

The profile is unchanged if the exponent is shifted by a constant, so we get

$$p_m \propto \exp\big(-A(m^{2/D} - 1)\big) = \exp\left(-\frac{2A}{D}\frac{m^{2/D} - 1}{2/D}\right).$$

It is a standard result that $\frac{m^{2/D}-1}{2/D} \to \log(m)$. Furthermore, perplexity calibration keeps $2A/D$ bounded: if it diverged, the profile would concentrate on $m = 1$, and if it vanished, the profile would become uniform (in neither case perplexity can be P given $1 < P < K$). For any D, the mapping $2A/D \mapsto \mathrm{Perp}(p_m)$ is strictly decreasing and thus injective, hence $2A/D$ converges. Therefore, the limit of p_m is the power law with exponent $-\beta$ needed to yield perplexity P. □

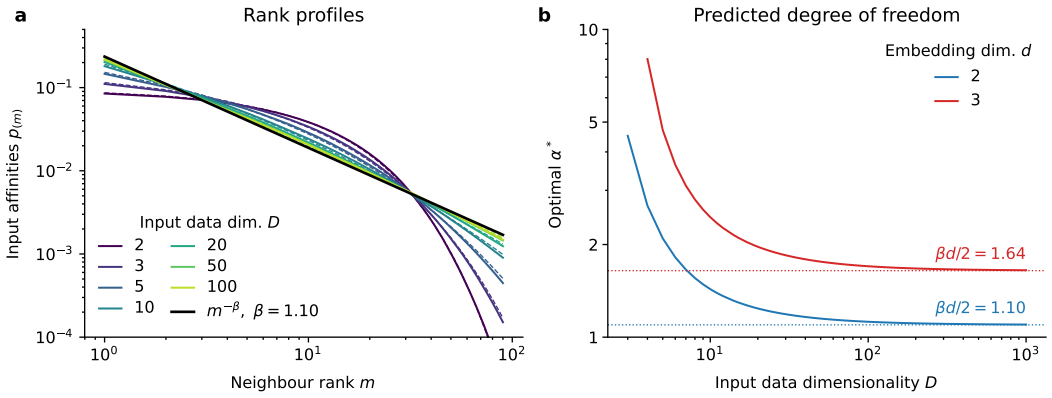

Figure 4: **Theoretical analysis. (a)** Rank profiles of the input affinities on uniformly sampled data. Solid lines: profiles for perplexity $P = 30$ with $K = 90$ retained neighbors measured on $n = 5\,000$ points sampled uniformly from the hypercube $[0, 1]^D$. Dashed lines: theoretical predictions (Proposition 4.2). Black line: the limiting power law $m^{-\beta}$ with $\beta = 1.10$. **(b)** Predicted optimal degree of freedom for embedding dimensions $d \in \{2, 3\}$, using assumptions behind Proposition 4.3. Solid lines are obtained using a numerical solver (see text). Dotted lines: predicted asymptotes $\beta d/2$. For $d = D$, the prediction is ∞.

We can find $\beta(P, K)$ by bisection search. For the default parameters we get $\beta(30, 90) = 1.10$. Enlarging K moves β upwards, but the dependence is weak: $\beta(30, 300) = 1.30$, $\beta(30, 3\,000) = 1.42$, and $\beta(30, 3\,000\,000) = 1.47$. We confirm the proposition by simulation (Figure 4a).

Proposition 4.3. *Assume that all K neighbors of each point are uniformly embedded around it in correct order, so that the number of input neighbors embedded within distance ρ scales as ρ^d and hence the distance to the m-th neighbor is $\rho_m^2 = \lambda m^{2/d}$ for some λ (no condition is imposed on how many non-neighbors lie between them). Consider a proxy objective $\mathrm{KL}(p. \,\|\, q.)$, for which we define the embedding profile*

$$q_m = \frac{(1 + \lambda m^{2/d}/\alpha)^{-\alpha}}{\sum_{l=1}^{K}(1 + \lambda l^{2/d}/\alpha)^{-\alpha}} \tag{10}$$

(unlike actual t-SNE, here the denominator goes only over K and not over $n(n-1)$ values). Then for $D \to \infty$, the infimum of the proxy objective over α and λ is zero, approached as $\lambda \to \infty$ and $\hat{\alpha}^* \to \beta d/2$ (which gives $\hat{\alpha}^* = \beta$ for $d = 2$).

Proof. For $D \to \infty$, the input profile becomes the power law. It can be matched by the output profile when $\lambda \to \infty$, giving $q_m \propto m^{-2\alpha/d}$. Matching the power law exponents yields $\hat{\alpha}^* = \beta d/2$. □

We can use two-parameter numerical optimization to compute the optimal α as a function of input dimensionality: $\hat{\alpha}^*(D) = \arg\min_\alpha \min_\lambda \mathrm{KL}(p. \,\|\, q.)$. The resulting predicted values are shown in Figure 4b for $d = 2$ and $d = 3$. The predicted $\hat{\alpha}^*$ decreases monotonically with D, but slowly: for $d = 2$ it is still $\hat{\alpha}^* \approx 1.4$ at $D = 10$, well above its asymptote $\beta = 1.1$. At $D = d$ the prediction diverges, as in Theorem 4.1. For actual t-SNE on two-dimensional data, α^* is finite but large (Figure 2a), reflecting boundary effects, density variations, and the resulting point-to-point variation in σ_i, all of which the uniform model ignores.

5 Experimental study of the effect of input dimensionality

Our theoretical analysis found that the optimal α depends on the intrinsic dimensionality D of the data. It diverges for small $D \approx d$ and decreases monotonically towards the asymptote $\beta d/2$.

To test these predictions, we ran t-SNE optimization on $n = 50\,000$ points sampled uniformly from the hypercube $[0, 1]^D$ for $D \in [2, 64]$. We found that the learned α followed the predicted shape (Figure 5, black lines). At $D = d$ the learned α^* was large and kept growing further if the number

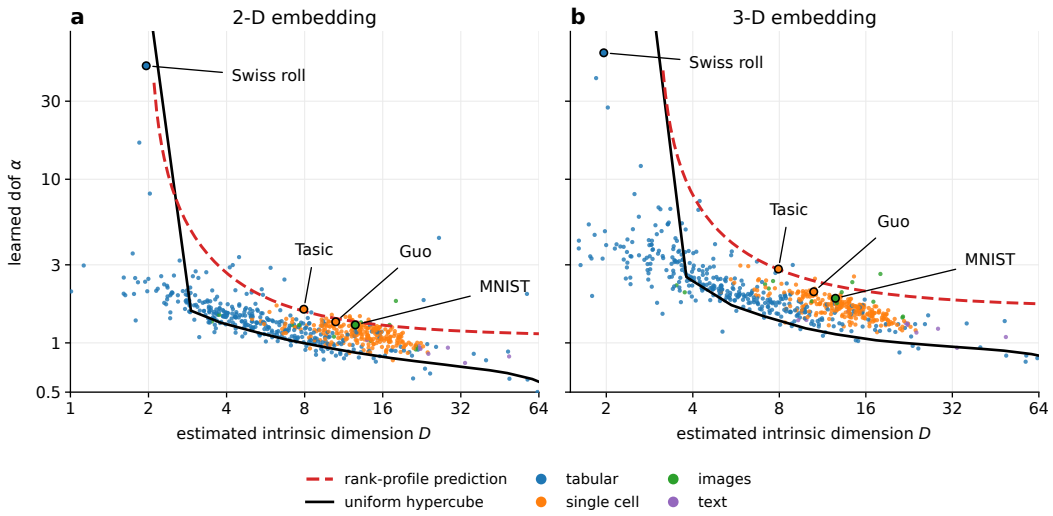

Figure 5: **Empirical study of the optimal α^*.** Red line: theoretical predictions. Black line: empirical results using $n = 50\,000$ points uniformly sampled from a hypercube $[0, 1]^D$. Colored points: 633 data sets downsampled to 5 000 points each. Here we used the Levina–Bickel estimator of intrinsic dimensionality. We highlight datasets used in our qualitative figures.

of gradient descent iterations was increased. For $D > d$ the learned α^* first dropped sharply and then slowly decreased with D. Even though the specific numerical values of α^* that we obtained here differed from the theoretical predictions (Figure 5, red dashed lines) due to all simplifying assumptions made by the theoretical analysis, we consider them in a good qualitative agreement.

To investigate whether the same relation holds on real-world non-uniform data, we collected 633 datasets from tabular, single-cell, image, and text-embedding domains. We considered all datasets with at least 2 000 samples and at least 8 numeric features from the OpenML (Vanschoren et al., 2014) and the UCI repositoritories (Dua & Graff, 2017), all experiments with 2 000 to 150 000 cells from the Single Cell Expression Atlas (Papatheodorou et al., 2020), and a number of widely used benchmarks. We excluded simulated datasets and near-duplicates. In the end we had 366 tabular, 242 single-cell, 15 image, and 10 text datasets.

Each domain was preprocessed following a single standard recipe. For single-cell data, we used the standard Scanpy pipeline (Wolf et al., 2018), for text data we used TF-IDF representation over the 2 000 most common terms, and for images we used pixel intensities rescaled to $[0, 1]$. Tabular features were z-scored. We subsampled each dataset to 5 000 points (or kept all points, if fewer) and embedded it in 2D and 3D with learnable α. We used the default parameters of `openTSNE` including perplexity 30 and PCA initialization, but used 2 000 instead of 750 iterations in order to ensure better convergence of α parameter.

For each dataset, we also estimated its intrinsic dimensionality using Levina & Bickel (2004)'s estimator with 90 nearest neighbors. We found that the real-world datasets also followed our theoretical predictions (Figure 5), with the optimal α^* inversely related to the intrinsic dimensionality.

In most cases the learned α^* was above the t-SNE default of 1, but the embeddings did not exhibit large differences. However, for some datasets, the learned α showed meaningful structure that was distorted in standard t-SNE (Figure 6). In human testis, germ cells develop from spermatogonia through early and pachytene spermatocytes into spermatids, and this process should ideally appear continuous in the embedding (Green et al., 2018; Shami et al., 2020). But in the human testis dataset (Guo et al., 2020) (subsampled to 20 000 cells), standard t-SNE breaks the spermatogonia and the spermatids into separate clusters obscuring the continuous biological process. Furthermore, the pink cluster (transitional spermatids) was split off from the main arc, and most of these cell types appeared fragmented into smaller clusters. In contrast, with the learned $\alpha^* = 1.68$, the germ cells formed one continuous arc in correct developmental order (Figure 6a) and the transitional spermatids were correctly absorbed into the transitionary position between round and elongating spermatids.

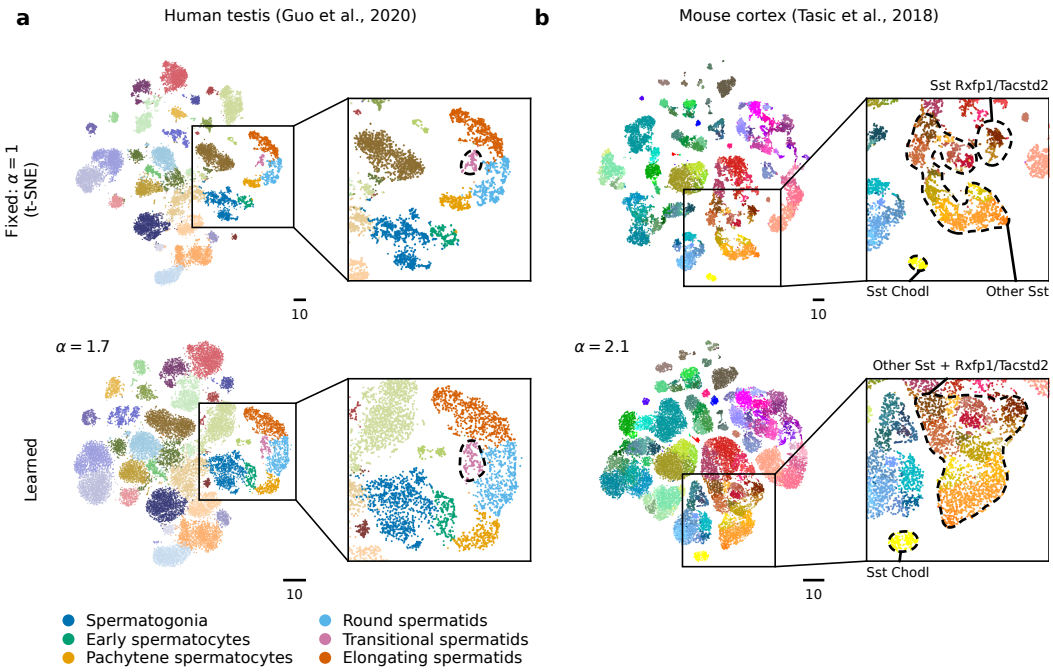

Figure 6: **Learnable kernel shape better reflects known biology.** Embeddings with $\alpha = 1$ (top) and learned α^* (bottom), using identical affinities and initialization. **(a)** Human testis dataset (Guo et al., 2020), subsampled to 20 000 cells, with germ cells colored by developmental stage. **(b)** Mouse cortex dataset (Tasic et al., 2018), colored by cell type. In both cases standard t-SNE exhibits over-fragmentation, see text.

In the mouse cortex dataset (Tasic et al., 2018), standard t-SNE splits the *Sst* interneurons into multiple fragmented clusters, separating *Sst Chodl* and three types we label *Sst Rxfp1/Tacstd2* from the other *Sst* types (Figure 6b). However, *Sst* interneurons form a continuous family with only *Sst Chodl* being a truly distinct population (Tasic et al., 2018; Yao et al., 2021; Scala et al., 2021). Although other *Sst* interneurons do exhibit variability, their transcriptional and electrophysiological properties vary continuously from type to type (Tasic et al., 2018; Scala et al., 2021). With the learned $\alpha^* = 2.11$, the embedding combined togther all *Sst* types apart from *Sst Chodl*, better reflecting the interneuron organization established in existing literature.

6 The effect of increasing kNN graph noise

For data lying on a two-dimensional manifold, our analysis predicts a large optimal α, i.e. a kernel close to SNE (Section 4). Real kNN graphs, however, are rarely this clean, and Figure 2e–h suggested that even a small number of spurious edges can pull the optimal α towards standard t-SNE.

To study this more systematically, we uniformly sampled $n = 10\,000$ points from the Swiss roll manifold and introduced varying amounts of noise directly into the input affinity matrix. In order to do that, we found the largest existing p_{ij} value and assigned the same value to random pairs (i, j), normalizing the entire matrix to sum to 1 in the end. This essentially adds random connections to the kNN graph while keeping all existing edges intact. We added such connections until they carried a desired fraction L of the total affinity mass. On the unperturbed graph, the learned α was very large ($\alpha \approx 76$) and would grow even further if the optimization ran for more iterations, as expected for two-dimensional data (Figure 7a). Introducing noise made the learned α drop sharply: already at $L = 0.001$, α fell to about 7, and it then decreased steadily towards 1, reaching $\alpha \approx 1.7$ at $L = 0.05$ (Figure 7a). The resulting embedding barely changed until around $L = 0.0375$ (Figure 7c).

While holding α fixed at the SNE level worked well on the unperturbed graph, it yielded a much worse embedding already with $L = 0.0125$ (Figure 7d). Intuitively, the added random edges pull

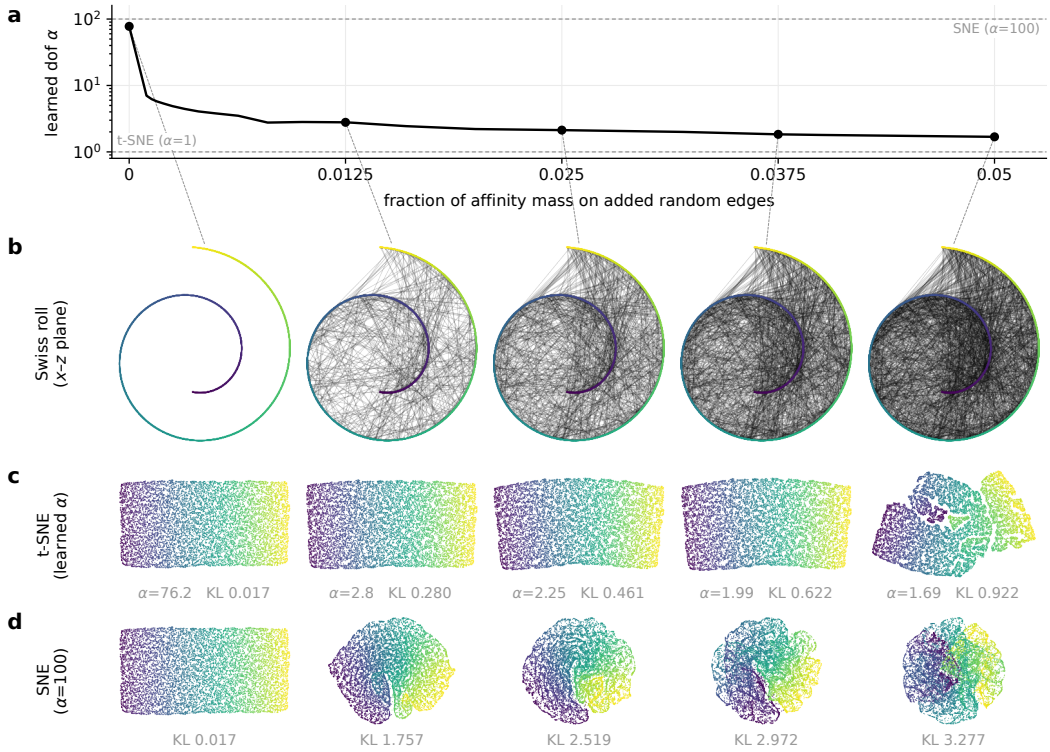

Figure 7: **Heavier tails accommodate more noise.** We added connections between random pairs of points to the affinity matrix of a Swiss roll dataset ($n = 10\,000$). **(a)** Learned α (median over 5 random seeds) falls from the SNE regime with no noise to $\sim$2 at $L = 0.025$. **(b)** The Swiss roll along its height axis, with every added edge. **(c–d)** Embeddings using t-SNE with learned α and using SNE ($\alpha = 100$) using identical affinities and identical spectral initialization.

distant parts of the manifold together, first folding it into a disk and then mixing it together. Heavier kernel tails thus allow the embedding to accommodate not only kNN graphs coming from higher-dimensional data, but also kNN graphs with higher levels of noise. A heavy-tailed kernel can place spurious neighbors far away in the embedding while still assigning them non-negligible similarity, whereas a Gaussian kernel can only do so by pulling them very close.

7 Discussion and Limitations

Here we investigated the role of the degree of freedom in t-SNE's t-distribution both theoretically and empirically. We found that datasets with low intrinsic dimensionality benefited from a higher degree of freedom, while datasets with high-dimensional noise benefited from a lower degree of freedom. To automatically select the most appropriate parameter, we implemented learnable degree of freedom in the Barnes–Hut and FFT-based approximations of t-SNE, with minimal computational overhead. This allows t-SNE to work well on a wide range of datasets. In a large-scale benchmark, we found that for many real-world datasets the optimal degree of freedom was close to the t-SNE's default. But we also identified some cases where the learned degree of freedom improved the embedding by preventing over-fragmentation and better representing continuous variation. We also developed a theoretical analysis of the optimal degree of freedom. While reliant on several simplifying assumptions, the qualitative predictions were corroborated experimentally. Future work should aim to make the theoretical assumptions more realistic.

AI USE STATEMENT

AI systems were not involved in planning the study.

We used Claude Code to write code, generate figures, and run most experiments. This took place in long interactive coding sessions with extensive human guidance. Claude Code was instrumental in running the large-scale empirical study, where it automatically identified, downloaded, preprocessed, and analyzed 633 publicly available datasets.

When extending the `openTSNE` code, we implemented the Barnes–Hut approach by hand and used Claude Code to assist with the FIt-SNE implementation. Both implementations were thoroughly reviewed by the authors. We validated the resulting code against the existing `openTSNE` unit tests, and carefully reviewed all the newly generated unit tests.

The theoretical analysis was developed in a long interactive session with Claude (up to Opus 5.5 and Fable 5). Most of the manuscript text (including the theory section) was written by the authors. We used Claude to edit parts of the text for readability.

ETHICS STATEMENT

We do not see any ethical issues with this work.

REPRODUCIBILITY STATEMENT

We are preparing a pull request to `openTSNE` with our implementation of learnable α. The code for all experiments in this paper will be separately hosted on GitHub. We also included our code as supplementary.

REFERENCES

Aleksandr Artemenkov and Maxim Panov. NCVis: noise contrastive approach for scalable visualization. In *Proceedings of The Web Conference 2020*, pp. 2941–2947, 2020.

Jan Niklas Böhm, Philipp Berens, and Dmitry Kobak. Attraction-repulsion spectrum in neighbor embeddings. *Journal of Machine Learning Research*, 23(95):1–32, 2022.

Cyril de Bodt, Alex Diaz-Papkovich, Michael Bleher, Kerstin Bunte, Corinna Coupette, Sebastian Damrich, Enrique Fita Sanmartin, Fred A Hamprecht, Emőke-Ágnes Horvát, Dhruv Kohli, Smita Krishnaswamy, John A. Lee, Boudewijn P. F. Lelieveldt, Leland McInnes, Ian T. Nabney, Maximilian Noichl, Pavlin G. Poličar, Bastian Rieck, Guy Wolf, Gal Mishne, and Dmitry Kobak. Low-dimensional embeddings of high-dimensional data. *arXiv preprint arXiv:2508.15929*, 2025.

Dheeru Dua and Casey Graff. UCI Machine Learning Repository, 2017. URL `https://archive.ics.uci.edu`.

Christopher Daniel Green, Qianyi Ma, Gabriel L. Manske, Adrienne Niederriter Shami, Xianing Zheng, Simone Marini, Lindsay Moritz, Caleb Sultan, Stephen J. Gurczynski, Bethany B. Moore, Michelle D. Tallquist, Jun Z. Li, and Saher Sue Hammoud. A comprehensive roadmap of murine spermatogenesis defined by single-cell rna-seq. *Developmental Cell*, 46(5):651–667.e10, 2018. ISSN 1534-5807.

Jingtao Guo, Xichen Nie, Maria Giebler, Hana Mlcochova, Yueqi Wang, Edward J. Grow, Robin Kim, Melissa Tharmalingam, Gabriele Matilionyte, Cecilia Lindskog, Douglas T. Carrell, Rod T. Mitchell, Anne Goriely, James M. Hotaling, and Bradley R. Cairns. The dynamic transcriptional cell atlas of testis development during human puberty. *Cell Stem Cell*, 26(2):262–276.e4, Feb 2020. ISSN 1934-5909. doi: 10.1016/j.stem.2019.12.005.

Geoffrey Hinton and Sam Roweis. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems*, pp. 857–864, 2003.

Mohammad Tariqul Islam and Jason W Fleischer. The shape of attraction in umap: Exploring the embedding forces in dimensionality reduction. *arXiv preprint arXiv:2503.09101*, 2025.

Jun Kitazono, Nistor Grozavu, Nicoleta Rogovschi, Toshiaki Omori, and Seiichi Ozawa. t-distributed stochastic neighbor embedding with inhomogeneous degrees of freedom. In *International Conference on Neural Information Processing*, pp. 119–128. Springer, 2016.

Dmitry Kobak, George Linderman, Stefan Steinerberger, Yuval Kluger, and Philipp Berens. Heavy-tailed kernels reveal a finer cluster structure in t-SNE visualisations. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 124–139. Springer, 2019.

Elizaveta Levina and Peter Bickel. Maximum likelihood estimation of intrinsic dimension. In L. Saul, Y. Weiss, and L. Bottou (eds.), *Advances in Neural Information Processing Systems*, pp. 777–784. MIT Press, 2004.

George C Linderman, Manas Rachh, Jeremy G Hoskins, Stefan Steinerberger, and Yuval Kluger. Fast interpolation-based t-SNE for improved visualization of single-cell RNA-seq data. *Nature Methods*, 16:243–245, 2019.

Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Irene Papatheodorou, Pablo Moreno, Jonathan Manning, Alfonso Muñoz-Pomer Fuentes, Nancy George, Silvie Fexova, Nuno A Fonseca, Anja Füllgrabe, Matthew Green, Ni Huang, Laura Huerta, Haider Iqbal, Monica Jianu, Suhaib Mohammed, Lingyun Zhao, Andrew F Jarnuczak, Simon Jupp, John Marioni, Kerstin Meyer, Robert Petryszak, Cesar Augusto Prada Medina, Carlos Talavera-López, Sarah Teichmann, Juan Antonio Vizcaino, and Alvis Brazma. Expression atlas update: from tissues to single cells. *Nucleic Acids Research*, 48(D1):D77–D83, 01 2020. ISSN 0305-1048.

Pavlin G Poličar, Martin Stražar, and Blaž Zupan. openTSNE: a modular python library for t-SNE dimensionality reduction and embedding. *Journal of Statistical Software*, 109:1–30, 2024.

Federico Scala, Dmitry Kobak, Matteo Bernabucci, Yves Bernaerts, Cathryn René Cadwell, Jesus Ramon Castro, Leonard Hartmanis, Xiaolong Jiang, Sophie Laturnus, Elanine Miranda, Shalaka Mulherkar, Zheng Huan Tan, Zizhen Yao, Hongkui Zeng, Rickard Sandberg, Philipp Berens, and Andreas S. Tolias. Phenotypic variation of transcriptomic cell types in mouse motor cortex. *Nature*, 598(7879):144–150, Oct 2021. ISSN 1476-4687.

Adrienne Niederriter Shami, Xianing Zheng, Sarah K. Munyoki, Qianyi Ma, Gabriel L. Manske, Christopher D. Green, Meena Sukhwani, Kyle E. Orwig, Jun Z. Li, and Saher Sue Hammoud. Single-cell rna sequencing of human, macaque, and mouse testes uncovers conserved and divergent features of mammalian spermatogenesis. *Developmental Cell*, 54(4):529–547.e12, 2020. ISSN 1534-5807.

Bosiljka Tasic, Zizhen Yao, Lucas T Graybuck, Kimberly A Smith, Thuc Nghi Nguyen, Darren Bertagnolli, Jeff Goldy, Emma Garren, Michael N Economo, Sarada Viswanathan, et al. Shared and distinct transcriptomic cell types across neocortical areas. *Nature*, 563(7729):72, 2018.

Laurens van der Maaten. Learning a parametric embedding by preserving local structure. In *International Conference on Artificial Intelligence and Statistics*, pp. 384–391, 2009.

Laurens van der Maaten. Accelerating t-SNE using tree-based algorithms. *Journal of Machine Learning Research*, 15(1):3221–3245, 2014.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11), 2008.

Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. OpenML: networked science in machine learning. *SIGKDD Explor. Newsl.*, 15(2):49–60, June 2014. ISSN 1931-0145.

Yingfan Wang, Haiyang Huang, Cynthia Rudin, and Yaron Shaposhnik. Understanding how dimension reduction tools work: an empirical approach to deciphering t-SNE, UMAP, TriMAP, and PaCMAP for data visualization. *Journal of Machine Learning Research*, 22(201):1–73, 2021.

F. Alexander Wolf, Philipp Angerer, and Fabian J. Theis. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biology*, 19(1):15, Feb 2018. ISSN 1474-760X.

Zhirong Yang, Irwin King, Zenglin Xu, and Erkki Oja. Heavy-tailed symmetric stochastic neighbor embedding. In *Advances in Neural Information Processing Systems*, pp. 2169–2177, 2009.

Zhirong Yang, Jaakko Peltonen, and Samuel Kaski. Scalable optimization of neighbor embedding for visualization. In *International Conference on Machine Learning*, pp. 127–135. PMLR, 2013.

Zizhen Yao, Cindy T.J. van Velthoven, Thuc Nghi Nguyen, Jeff Goldy, Adriana E. Sedeno-Cortes, Fahimeh Baftizadeh, Darren Bertagnolli, Tamara Casper, Megan Chiang, Kirsten Crichton, Song-Lin Ding, Olivia Fong, Emma Garren, Alexandra Glandon, Nathan W. Gouwens, James Gray, Lucas T. Graybuck, Michael J. Hawrylycz, Daniel Hirschstein, Matthew Kroll, Kanan Lathia, Changkyu Lee, Boaz Levi, Delissa McMillen, Stephanie Mok, Thanh Pham, Qingzhong Ren, Christine Rimorin, Nadiya Shapovalova, Josef Sulc, Susan M. Sunkin, Michael Tieu, Amy Torkelson, Herman Tung, Katelyn Ward, Nick Dee, Kimberly A. Smith, Bosiljka Tasic, and Hongkui Zeng. A taxonomy of transcriptomic cell types across the isocortex and hippocampal formation. *Cell*, 184(12):3222–3241.e26, Jun 2021.

A Appendix

A.1 Supplementary Figures

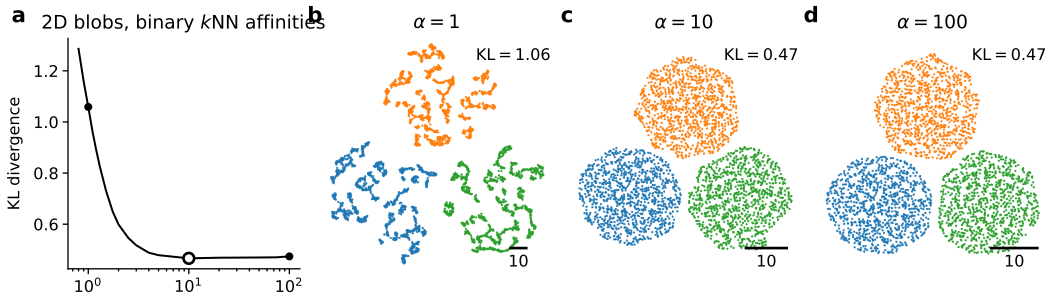

Figure S1: The same as Figure 2a–d, but using uniform affinities with equal weights on the k-NN graph with $k = 10$. The optimal α is about 10.

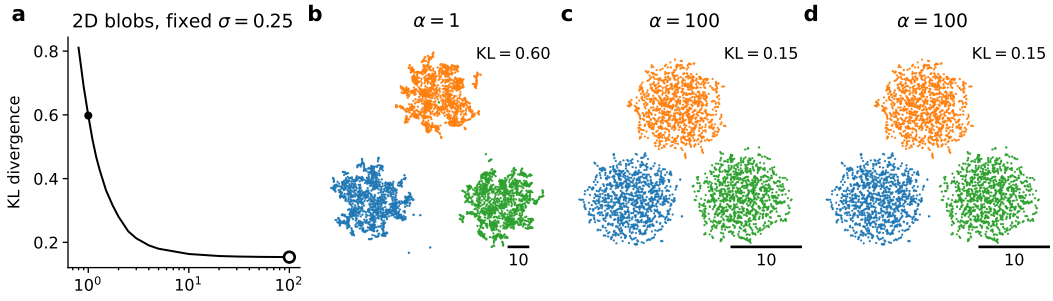

Figure S2: The same as Figure 2a–d, but using affinities with fixed width of Gaussian kernel, instead of relying on perplexity calibration. Note that if additionally p_{ij} values are computed using global normalization (see Section 4), then $\alpha \to \infty$ is provably optimal and yields zero KL divergence. Here we use standard normalization and symmetrization, and we still find that $\alpha = 100$ yields the lowest loss.