

Attraction-Repulsion Spectrum in Neighbor Embeddings

Jan Niklas Böhm

JAN-NIKLAS.BOEHM@UNI-TUEBINGEN.DE

Philipp Berens

PHILIPP.BERENS@UNI-TUEBINGEN.DE

Dmitry Kobak

DMITRY.KOBAK@UNI-TUEBINGEN.DE

*Institute for Ophthalmic Research, University of Tübingen
Otfried-Müller-Str. 25; 72076 Tübingen, Germany*

Editor: Samuel Kaski

Abstract

Neighbor embeddings are a family of methods for visualizing complex high-dimensional data sets using k NN graphs. To find the low-dimensional embedding, these algorithms combine an attractive force between neighboring pairs of points with a repulsive force between all points. One of the most popular examples of such algorithms is t-SNE. Here we empirically show that changing the balance between the attractive and the repulsive forces in t-SNE using the exaggeration parameter yields a spectrum of embeddings, which is characterized by a simple trade-off: stronger attraction can better represent continuous manifold structures, while stronger repulsion can better represent discrete cluster structures and yields higher k NN recall. We find that UMAP embeddings correspond to t-SNE with increased attraction; mathematical analysis shows that this is because the negative sampling optimization strategy employed by UMAP strongly lowers the effective repulsion. Likewise, ForceAtlas2, commonly used for visualizing developmental single-cell transcriptomic data, yields embeddings corresponding to t-SNE with the attraction increased even more. At the extreme of this spectrum lie Laplacian eigenmaps. Our results demonstrate that many prominent neighbor embedding algorithms can be placed onto the attraction-repulsion spectrum, and highlight the inherent trade-offs between them.

Keywords: dimensionality reduction, neighbor embedding, visualization

1. Introduction

T-distributed stochastic neighbor embedding (t-SNE) (van der Maaten and Hinton, 2008) is arguably among the most popular methods for low-dimensional visualization of complex high-dimensional data sets. It defines pairwise similarities called *affinities* between points in the high-dimensional space and aims to arrange the points in a low-dimensional space to match these affinities (Hinton and Roweis, 2003). Affinities decay exponentially with high-dimensional distance, making them infinitesimal for most pairs of points and making the $n \times n$ affinity matrix effectively sparse. Efficient implementations of t-SNE (van der Maaten, 2014; Linderman et al., 2019) explicitly truncate the affinities and use the k -nearest-neighbor (k NN) graph of the data with $k \ll n$ as the input.

We use the term *neighbor embedding* (NE) to refer to all dimensionality reduction methods that operate on the k NN graph of the data and aim to preserve neighborhood relationships (Yang et al., 2013, 2014). A prominent recent example of this class of algorithms is UMAP (McInnes et al., 2018), which has become popular in applied fields such as single-cell transcriptomics (Becht et al., 2019). It is based on stochastic optimization and typically produces more compact clusters than t-SNE.

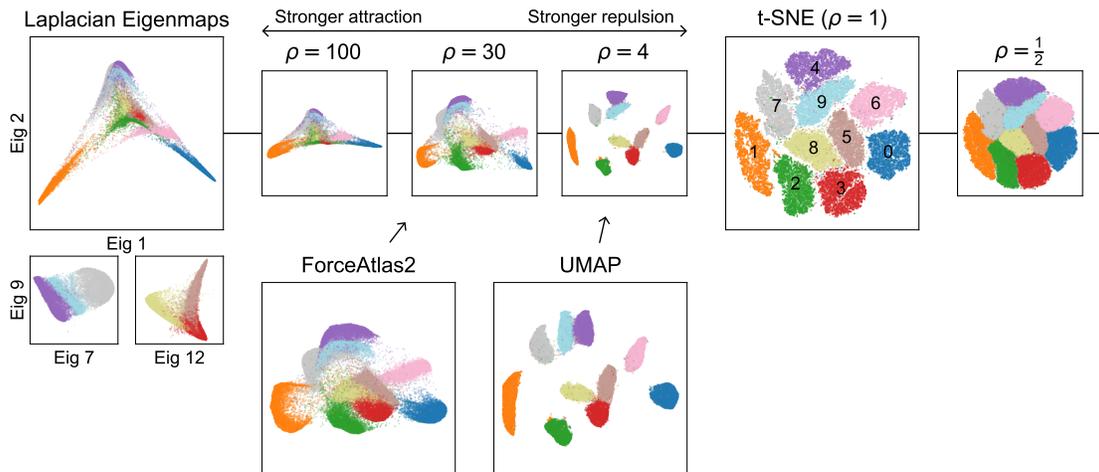


Figure 1: **Attraction-repulsion spectrum for the MNIST data.** Different embeddings of the MNIST data set of hand-written digits ($n = 70\,000$); colors denote digits as shown in the t-SNE panel. Multiplying all attractive forces by an exaggeration factor ρ yields a spectrum of embeddings. Values below 1 yield inflated clusters. Values above 1 yield more compact clusters. Higher values make multiple clusters merge, with $\rho \rightarrow \infty$ approximately corresponding to Laplacian eigenmaps (Linderman and Steinerberger, 2019). Insets show two subsets of digits separated in higher eigenvectors. UMAP is similar to $\rho \approx 4$. ForceAtlas2 is similar to $\rho \approx 30$.

Another example of neighbor embeddings are force-directed graph layouts (Noack, 2007, 2009), originally developed for graph drawing. One specific algorithm called ForceAtlas2 (Jacomy et al., 2014) has recently gained popularity in the single-cell transcriptomic community to visualize data sets capturing cells at different stages of development (Weinreb et al., 2018, 2020; Wagner et al., 2018a; Tusi et al., 2018; Kanton et al., 2019; Sharma et al., 2020).

In general, NE algorithms optimize the layout using attractive forces between all pairs of points connected by a k NN graph edge, thus placing them closer in the low-dimensional embedding. In addition, every point feels a repulsive force to every other point, which prevents trivial solutions, such as positioning all points on top of each other. While earlier algorithms took inspiration from physical systems (Fruchterman and Reingold, 1991), similar concepts arise naturally from the loss functions grounded in information theory (see below).

Here we provide a unifying account of NE algorithms. We study the spectrum of t-SNE embeddings that are obtained when increasing/decreasing the attractive forces between k NN graph neighbors, thereby changing the balance between attraction and repulsion. This leads to a trade-off between faithful representations of continuous and discrete structures. Remarkably, we discover that ForceAtlas2 and UMAP can both be accurately positioned on this spectrum (Figure 1). For UMAP, we use mathematical analysis and Barnes–Hut re-implementation to show that increased attraction is due to the negative sampling optimization strategy, and to derive the effective repulsion strength.

All our code is available at <https://github.com/berenslab/ne-spectrum>.

2. Related Work

Various trade-offs in SNE and t-SNE generalizations (Yang et al., 2009; Carreira-Perpiñán, 2010; Kobak et al., 2020; Venna et al., 2010; Amid et al., 2015; Amid and Warmuth, 2019; Narayan et al., 2015; Im et al., 2018) as well as in graph layout algorithms (Noack, 2007; Gansner et al., 2012) have been studied previously, but our work is the first to study the *exaggeration*-induced trade-off in t-SNE. Prior work used ‘early exaggeration’ only as an optimization trick (van der Maaten and Hinton, 2008) that allows to separate well-defined clusters (Linderman and Steinerberger, 2019; Arora et al., 2018).

Carreira-Perpiñán (2010) introduced the *elastic embedding* algorithm that has an explicit parameter λ controlling the attraction-repulsion balance. However, that paper suggests slowly increasing λ during optimization, as an optimization trick similar to the early exaggeration, and does not discuss trade-offs between high and low values of λ . The same holds for the graph layout model studied by Gansner et al. (2012), who similarly anneal the repulsion strength during optimization.

Our results on UMAP go against the common wisdom regarding what makes UMAP perform as it does (McInnes et al., 2018; Becht et al., 2019). No previous work suggested that negative sampling may have a drastic effect on the resulting embedding. In a follow-up paper, Damrich and Hamprecht (2021) analyzed the effective loss function of UMAP in more detail.

3. Neighbor Embeddings

The standard expositions of t-SNE, UMAP, and ForceAtlas2 (FA2) create the impression that these algorithms have little to do with each other. They use different affinities, different loss functions, different optimization strategies, and different large-sample approximations. They are introduced using different motivations. Importantly, the loss function in t-SNE includes a normalizing term which makes its optimization difficult, whereas the loss functions of UMAP and FA2 do not have such a term.

Despite all these differences, we claim that these algorithms are intimately related (Figure 1). In this section, we cast t-SNE, UMAP, FA2, and Laplacian eigenmaps (LE) in a common mathematical framework, using consistent notation and highlighting the similarities between them. The empirical results will be presented in the following sections. We denote the original high-dimensional points as \mathbf{x}_i and their low-dimensional positions as \mathbf{y}_i .

3.1 T-SNE

T-SNE measures similarities between \mathbf{x}_i by *affinities* v_{ij} and *normalized affinities* p_{ij}

$$p_{ij} = \frac{v_{ij}}{n}, \quad v_{ij} = \frac{p_{ij} + p_{ji}}{2}, \quad p_{ji} = \frac{v_{ji}}{\sum_{k \neq i} v_{k|i}}, \quad v_{ji} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma_i^2}\right).$$

For fixed i , p_{ji} is a probability distribution over all points $j \neq i$ (all p_{ii} are set to zero), and the variance of the Gaussian kernel σ_i^2 is chosen to yield a pre-specified value of the *perplexity* of this probability distribution, $\mathcal{P} = 2^{\mathcal{H}}$, where $\mathcal{H} = -\sum_{j \neq i} p_{ji} \log_2 p_{ji}$ is the entropy. The symmetrized affinities v_{ij} are then normalized by n for p_{ij} to form a probability distribution over the set of all pairs of points (i, j) . Modern implementations (van der Maaten, 2014; Linderman et al., 2019) construct a k NN graph with $k = 3\mathcal{P}$ neighbors and only consider affinities between connected nodes as non-zero. The default perplexity value in most implementations is $\mathcal{P} = 30$.

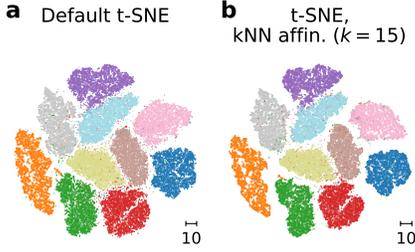


Figure 2: **The role of affinities in t-SNE.** MNIST data set. **(a)** Default t-SNE, Gaussian affinities, perplexity 30. **(b)** t-SNE with binary k NN affinities: all nonzero p_{ij} are the same, and $p_{ij} > 0$ iff point i is among 15 nearest neighbors of point j , or vice versa.

While t-SNE traditionally uses Gaussian affinities, the affinity matrix can be simplified without having a large impact on the resulting layout. In particular, one can use the k NN ($k = 15$) adjacency matrix $\mathbf{A} = [a_{ij}]$ to construct symmetric binary affinities $v_{ij} = a_{ij} \vee a_{ji}$, and then obtain p_{ij} by normalizing the entire matrix to sum to 1. The resulting ‘ k NN affinities’ typically yield t-SNE embeddings that are almost identical to the default ones (Figure 2).

Similarities in the low-dimensional space are defined as

$$q_{ij} = \frac{w_{ij}}{Z}, \quad w_{ij} = \frac{1}{1 + d_{ij}^2}, \quad d_{ij} = \|\mathbf{y}_i - \mathbf{y}_j\|, \quad Z = \sum_{k \neq l} w_{kl},$$

with all q_{ii} set to 0. The points \mathbf{y}_i are then rearranged in order to minimize the Kullback–Leibler (KL) divergence $\mathcal{D}_{\text{KL}}(\{p_{ij}\} \parallel \{q_{ij}\}) = \sum_{i,j} p_{ij} \log(p_{ij}/q_{ij})$ between p_{ij} and q_{ij}

$$\mathcal{L}_{\text{t-SNE}} \sim - \sum_{i,j} p_{ij} \log \frac{w_{ij}}{Z} = - \sum_{i,j} p_{ij} \log w_{ij} + \log \sum_{i,j} w_{ij}, \quad (1)$$

where we drop constant terms and take into account that $\sum p_{ij} = 1$. The first term contributes attractive forces to the gradient while the second term yields repulsive forces. Indeed, using $\partial w_{ij} / \partial \mathbf{y}_i = -2w_{ij}^2(\mathbf{y}_i - \mathbf{y}_j)$, the gradient, up to a constant factor, can be written as

$$\frac{\partial \mathcal{L}_{\text{t-SNE}}}{\partial \mathbf{y}_i} \sim \sum_j v_{ij} w_{ij} (\mathbf{y}_i - \mathbf{y}_j) - \frac{n}{Z} \sum_j w_{ij}^2 (\mathbf{y}_i - \mathbf{y}_j). \quad (2)$$

3.2 Exaggeration in t-SNE

A standard optimization trick for t-SNE called *early exaggeration* (van der Maaten and Hinton, 2008; van der Maaten, 2014) is to multiply the first sum in the gradient by a factor $\rho > 1$ during the initial iterations of gradient descent. This increases the attractive forces and allows similar points to gather into clusters more effectively. Modern implementations use $\rho = 12$ for the initial 250 iterations (van der Maaten, 2014) by default. The gradient of t-SNE with exaggeration can be written as

$$\frac{\partial \mathcal{L}_{\text{t-SNE}}(\rho)}{\partial \mathbf{y}_i} \sim \sum_j v_{ij} w_{ij} (\mathbf{y}_i - \mathbf{y}_j) - \frac{n}{\rho Z} \sum_j w_{ij}^2 (\mathbf{y}_i - \mathbf{y}_j) \quad (3)$$

and the corresponding loss function can be written in a functional form akin to the KL divergence

$$\mathcal{L}_{\text{t-SNE}}(\rho) = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{w_{ij}/Z^{\frac{1}{\rho}}}. \quad (4)$$

However, for $\rho \neq 1$ the values $w_{ij}/Z^{\frac{1}{\rho}}$ in the denominator do not sum to 1 so Eq. (4) is not a KL divergence between probability distributions.

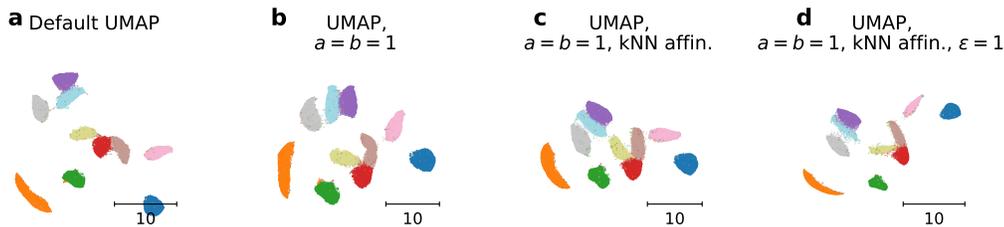


Figure 3: **UMAP with various simplifications.** MNIST data set. **(a)** Default UMAP with $a \approx 1.6$ and $b \approx 0.9$ and LE initialization. **(b)** UMAP with $a = b = 1$ and PCA initialization, the default choice for our experiments. **(c)** The same as in (b), but using binary k NN affinities ($v_{ij} = 1$ iff point i is among 15 nearest neighbors of point j , or vice versa). **(d)** The same as in (c), but with $\epsilon = 1$.

3.3 UMAP

Using the same notation as above, UMAP aims to optimize the cross-entropy loss between v_{ij} and w_{ij} , without normalizing them into probabilities:

$$\mathcal{L}_{\text{UMAP}} = \sum_{i,j} \left[v_{ij} \log \frac{v_{ij}}{w_{ij}} + (1 - v_{ij}) \log \frac{1 - v_{ij}}{1 - w_{ij}} \right], \quad (5)$$

where the $1 - v_{ij}$ term is approximated by 1 as most v_{ij} are 0, as done in the original implementation of UMAP (see Sainburg et al., 2021, Section 2.3.2). Note that UMAP differs from t-SNE in how exactly it defines v_{ij} (it uses adaptive Laplacian kernel with $k = 15$ by default), but its result does not change much when using the same binary affinities v_{ij} we introduced above for t-SNE (Figure 3c). Therefore, we believe that the difference in affinities is not what drives the difference in layout between t-SNE and UMAP in practice; see below for the experimental evidence.

Dropping terms that do not depend on \mathbf{y}_i , we obtain

$$\mathcal{L}_{\text{UMAP}} \sim - \sum_{i,j} v_{ij} \log w_{ij} - \sum_{i,j} \log(1 - w_{ij}), \quad (6)$$

which is the same loss function as the one introduced earlier in LargeVis (Tang et al., 2016). The first term, corresponding to attractive forces, is the same as in t-SNE, but the second, repulsive, term is different. Taking $w_{ij} = 1/(1 + d_{ij}^2)$ as in t-SNE, the UMAP gradient is given by

$$\frac{\partial \mathcal{L}_{\text{UMAP}}}{\partial \mathbf{y}_i} \sim \sum_j v_{ij} w_{ij} (\mathbf{y}_i - \mathbf{y}_j) - \sum_j \frac{1}{d_{ij}^2 + \epsilon} w_{ij} (\mathbf{y}_i - \mathbf{y}_j), \quad (7)$$

where $\epsilon = 0.001$ is added to the denominator to prevent numerical problems for $d_{ij} \approx 0$. Note that UMAP uses $w_{ij} = 1/(1 + ad_{ij}^{2b})$ as an output kernel with $a \approx 1.6$ and $b \approx 0.9$ by default. However, setting $a = b = 1$ does not strongly affect the result (Figure 3b). Moreover, when we modified the UMAP implementation to set $\epsilon = 1$, the resulting embeddings also stayed qualitatively similar (Figure 3d). So here again, we believe that these details are not what drives the difference in layout between t-SNE and UMAP in practice; see below for the experimental evidence.

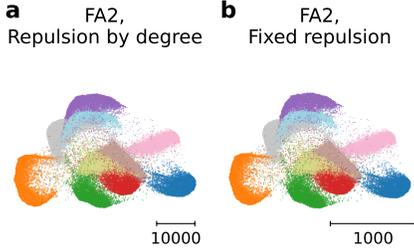


Figure 4: **The effect of edge repulsion in ForceAtlas2.** MNIST data set. (a) FA2 with repulsion by degree. (b) FA2 without repulsion by degree. Note the difference in scale.

If $\epsilon = 1$, the gradient becomes identical to the t-SNE gradient, up to the n/Z factor in front of the repulsive forces. Moreover, UMAP implementation allows to use an arbitrary γ factor in front of the repulsive forces, which makes it easier to compare the loss functions

$$\frac{\partial \mathcal{L}_{\text{UMAP}}(\gamma)}{\partial \mathbf{y}_i} \sim \sum_j v_{ij} w_{ij} (\mathbf{y}_i - \mathbf{y}_j) - \gamma \sum_j \frac{1}{d_{ij}^2 + \epsilon} w_{ij} (\mathbf{y}_i - \mathbf{y}_j). \quad (8)$$

Note that LargeVis used $\gamma = 7$ by default but UMAP sets $\gamma = 1$, as follows from its cross-entropy loss function.

Whereas it is possible to approximate the full repulsive term with the same techniques as used in t-SNE (van der Maaten, 2014; Linderman et al., 2019), UMAP takes a different approach and follows LargeVis in using *negative sampling* (Mikolov et al., 2013) of repulsive forces: on each gradient descent iteration, only a small number m of randomly picked repulsive forces are applied to each point for each of the $\sim k$ attractive forces that it feels. Other repulsive terms are ignored. The default value is $m = 5$. The effect of negative sampling on the resulting embedding has not been studied before.

3.4 ForceAtlas2

Force-directed graph layouts are usually introduced directly via attractive and repulsive forces, even though it is easy to write down a suitable loss function (Noack, 2007). ForceAtlas2 (FA2) has attractive forces proportional to d_{ij} and repulsive forces proportional to $1/d_{ij}$ (Jacomy et al., 2014):

$$\frac{\partial \mathcal{L}_{\text{FA2}}}{\partial \mathbf{y}_i} = \sum_j v_{ij} (\mathbf{y}_i - \mathbf{y}_j) - \sum_j \frac{(h_i + 1)(h_j + 1)}{d_{ij}^2} (\mathbf{y}_i - \mathbf{y}_j), \quad (9)$$

where h_i denotes the degree of node i in the input graph. This is known as *edge repulsion* in the graph layout literature (Noack, 2007, 2009) and is important for embedding graphs that have nodes of very different degrees. However, for symmetrized k NN graphs, assuming that they do not have too many ‘hubs’ (Radovanovic et al., 2010), $h_i \approx k$, so $(h_i + 1)(h_j + 1)$ term contributes a roughly constant $\sim k^2$ factor to the repulsive forces, and can be compensated by decreasing all distances by a factor of k . Indeed, for the MNIST data set, removing the edge repulsion factor led to a ~ 15 times decrease in scale but otherwise almost the same embedding (Figure 4).

3.5 Laplacian Eigenmaps

Laplacian eigenmaps (Belkin and Niyogi, 2002; Coifman and Lafon, 2006) is a method for dimensionality reduction that leverages spectral graph theory. Its loss function can be written with a quadratic constraint

$$\mathcal{L}_{LE} = \sum_{ij} v_{ij} \|\mathbf{y}_i - \mathbf{y}_j\|^2 \quad \text{s. t. } \mathbf{Y}^\top \mathbf{D} \mathbf{Y} = \mathbf{I}, \quad (10)$$

where \mathbf{D} is a diagonal matrix with $D_{ii} = \sum_j V_{ij}$ for affinity matrix $\mathbf{V} = [v_{ij}]$, \mathbf{I} is the identity matrix, and \mathbf{Y} is the embedding matrix having \mathbf{y}_i as rows. This loss function can be minimized by solving a generalized eigenvalue problem (Appendix A). The quadratic constraint in some sense serves the role of repulsive forces, preventing collapse of the embedding to a single point.

Carreira-Perpiñán (2010) and Linderman and Steinerberger (2019) noticed that the attractive term in the t-SNE loss function on its own reduces to the unconstrained loss function of Laplacian eigenmaps. Indeed, if $\rho \rightarrow \infty$, the relative repulsion strength becomes infinitesimal and the embedding shrinks to a point with all $w_{ij} \rightarrow 1$. This means that the gradient from Equation 2 reduces to $\sum_j v_{ij}(\mathbf{y}_i - \mathbf{y}_j)$, which coincides with the gradient of Laplacian eigenmaps (apart from the quadratic constraint). A more detailed analysis in Appendix A shows that when $\rho \rightarrow \infty$, the entire embedding shrinks to a single point, but the leading eigenvectors of the graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{V}$ shrink the slowest. This makes t-SNE with large values of ρ produce embeddings very similar to LE, which computes the leading eigenvectors of the normalized Laplacian (Appendix A).

This theoretical finding immediately suggests that it might be interesting to study t-SNE with exaggeration $\rho > 1$ not only as an optimization trick, but in itself, as an intermediate method between LE and standard t-SNE.

3.6 Implementation

All experiments were performed in Python. We ran all packages with default parameters, unless specified otherwise. We used openTSNE 0.6.0 (Poličar et al., 2019), a Python reimplementaion of FIt-SNE (Linderman et al., 2019). When using $\rho < 12$, we used the default early exaggeration with $\rho_{\text{early}} = 12$, and exaggeration ρ for all subsequent iterations. For $\rho \geq 12$ no early exaggeration was used and exaggeration ρ was applied throughout. The learning rate was set to $\eta = n / \max(\rho, \rho_{\text{early}})$ as suggested by Belkina et al. (2019). Note that we used default Gaussian affinities for all experiments.

We used UMAP 0.5.1 with Cauchy similarity kernel (by setting $a = b = 1$). We used default UMAP affinities for all experiments. The Barnes–Hut implementation of UMAP was developed in Cython (Behnel et al., 2011), on top of the openTSNE package. We extended the package to leave out the calculation of the normalization constant Z , take into account the ϵ and γ parameters from Equation 8, and load the default UMAP affinities as computed by UMAP itself. For these experiments we also set $a = b = 1$.

For FA2 we used the fa2 package (Chippada, 2017), which employs a Barnes–Hut approximation to speed up computation of the repulsive forces. We developed a patch that makes it possible to disable the repulsion by degree and applied it on top of the current version 0.3.5. The input to FA2 was the unweighted symmetrized approximate k NN graph $\mathbf{A} \vee \mathbf{A}^\top$, where \mathbf{A} is the k NN adjacency matrix constructed with Annoy (Bernhardsson, 2013) with $k = 15$. By default, all algorithms were optimized for 750 iterations.

Unless stated otherwise, we used principal component analysis (PCA) initialization to remove any differences between algorithms due to initialization strategies (Kobak and Linderman, 2021) and

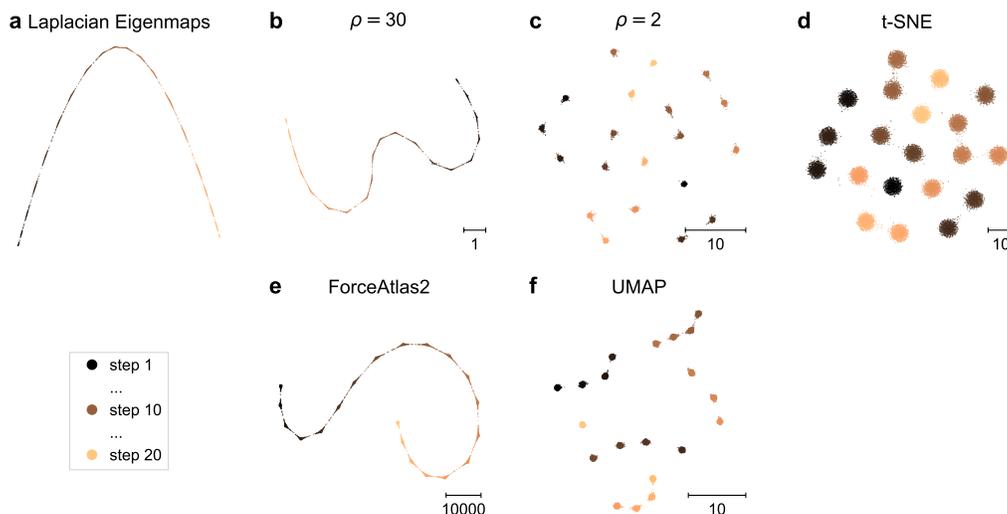


Figure 5: **Simulated data emulating a developmental trajectory.** The points were sampled from 20 isotropic 50-dimensional Gaussians, equally spaced along one axis such that only few inter-cluster edges exist in the k NN graph. Panels (b–f) used a shared random initialization. Panels (b–d) did not use early exaggeration.

to make all embeddings of the same data set visually aligned to each other (Kobak and Berens, 2019). For t-SNE, the initialization was always scaled to have a standard deviation of 0.0001, as suggested by Kobak and Berens (2019) and is default in openTSNE (Poličar et al., 2019). For UMAP, the initialization was scaled to have the range of $[-10, 10]$, as is default in the original implementation. For ForceAtlas2, we scaled the initialization to have a standard deviation of 10 000 to approximately match the scale of final ForceAtlas2 embeddings (we experimented with different values and found this setting to work well and avoid convergence problems). Note that Figure 5 is an exception and uses random initialization. By default, UMAP uses LE (and not PCA) initialization, whereas the fa2 implementation uses random initialization. On the data sets analyzed here, we did not observe any qualitative difference as a result of our non-default (PCA) initialization (cf. Figure 3a,b).

LE was computed using the `scikit-learn` (Pedregosa et al., 2011) implementation, using the `SpectralEmbedding` class and the PyAMG solver 4.0 (Olson and Schroder, 2018). The input graph was the same as the input to FA2. No initialization was needed for LE. We flipped the signs of LE eigenvectors to orient them similarly to other embeddings, whenever necessary.

4. The Attraction-Repulsion Spectrum

We first investigated the relationships between the NE algorithms using the MNIST data set of hand-written digits (sample size $n = 70\,000$; dimensionality $28 \times 28 = 784$, reduced to 50 with PCA; Figure 1). T-SNE produced an embedding where all ten digits were clearly separated into clusters with little white space between them, making it difficult to assess relationships between digits. Increasing attraction to $\rho = 4$ shrank the clusters and strongly increased the amount of white space; it also identified two groups of graphically similar digits: “4/7/9” and “3/5/8”. Further increasing

the attraction to $\rho = 30$ made all clusters connect together: cluster “6” connected to “5” and to “0”. Even higher exaggeration made the embedding similar to Laplacian eigenmaps, in agreement with the theoretical prediction discussed above (Linderman and Steinerberger, 2019). Here similar digit groups like “4/7/9” were entirely overlapping, and could only be separated using higher eigenvectors (Figure 1, insets). On the other side of the spectrum, exaggeration values $0 < \rho < 1$ resulted in inflated coalescing clusters.

The MNIST example suggests that high attraction emphasizes connections between clusters at the cost of within-cluster structure, whereas high repulsion emphasizes the cluster structure at the expense of between-cluster connections. We interpreted this finding as a *continuity-discreteness trade-off*.

We developed a simple toy example to illustrate this trade-off in more detail (Figure 5). For this, we generated data as draws from 20 standard isotropic Gaussians in 50 dimensions, each shifted by 6 standard deviation units from the previous one along one axis (1000 points per Gaussian, so $n = 20\,000$ overall). For this analysis we used random initialization and turned the early exaggeration off, to isolate the effect of each loss function on the ‘unwrapping’ of the random initial configuration.

We found that t-SNE with strong exaggeration ($\rho = 30$) recovered the underlying one-dimensional manifold structure of the data almost as well as LE (Figure 5a,b). In both cases, the individual clusters were almost invisible. In contrast, an embedding with weaker attraction and stronger repulsion (t-SNE with exaggeration $\rho = 2$) showed individual clusters but was unable to fully recover the 1-dimensional structure and only found some chunks of it (Figure 5c). Finally, standard t-SNE clearly showed 20 individual clusters but with the continuous structure entirely lost (Figure 5d).

Further, we analyzed a developmental single-cell transcriptomic data set, where cells were collected from human brain organoids at seven time points between 0 days and 4 months into the development (Kanton et al., 2019). In this kind of data, one expects to find rich cluster structure as well as a strong time-dependent trajectory. As in the other data sets, we found that stronger attraction ($\rho = 30$) better represented the developmental trajectory, whereas stronger repulsion (standard t-SNE) better represented the cluster structure (Figure 6). Using much higher k for the k NN graph construction made the developmental trajectory in high-attraction methods even clearer (Figure A1), in agreement with the FA2-based analysis performed in the original publication. We observed the same pattern in a separate data set obtained from chimpanzee brain organoids (Figures A2, A3).

While high exaggeration helps to preserve continuous structures, this comes with a price of distorting local neighborhoods. To quantify this effect, we computed the fraction of $k = 15$ nearest neighbors in high dimensions that remain among the nearest neighbors in the embedding (‘ k NN recall’). To compute it for a given data point, we found 15 points with the largest affinities in the symmetrized affinity matrix, and determined what fraction of them is among the 15 exact nearest neighbors in the embedding. This was averaged over 10 000 randomly selected points. We found that as ρ increased, the local neighborhood became more and more distorted (Figure 7). For the MNIST data set, the k NN recall of default t-SNE ($\rho = 1$) was 0.34; with $\rho = 4$ it went down to 0.12; with $\rho = 30$ it further dropped to 0.06.

We observed the same fast and monotonic decrease in k NN recall in both brain organoid data sets, as well as in six further data sets (Figure 7): Fashion MNIST (Xiao et al., 2017), Kannada MNIST (Prabhu, 2019), Kuzushiji MNIST (Clanuwat et al., 2018), single-cell data from hydra (Siebert et al., 2019), from zebrafish embryo (Wagner et al., 2018b), and from mouse cortex (Tasic et al., 2018).

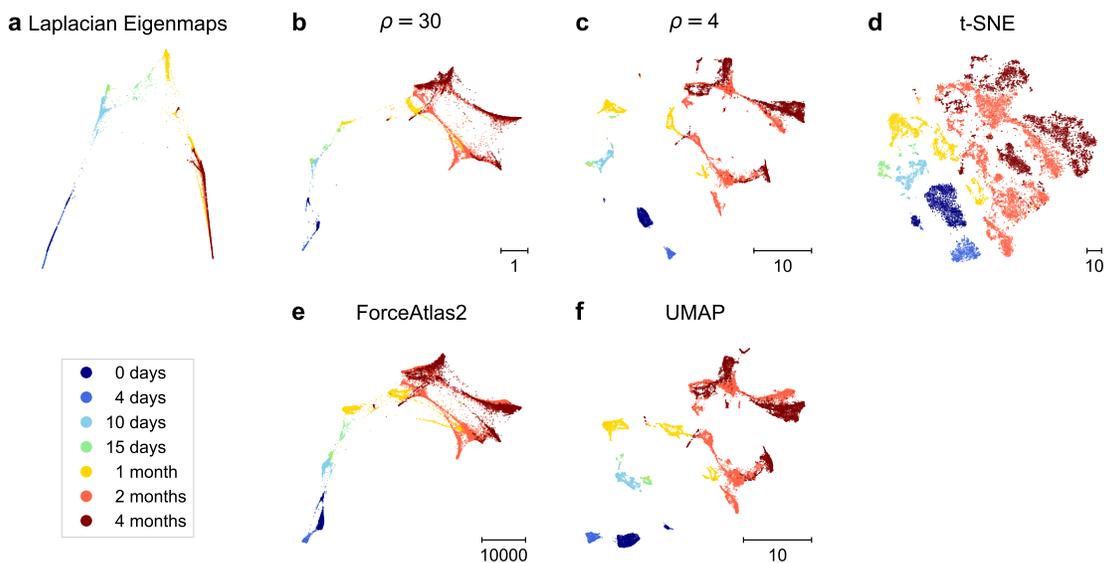


Figure 6: **Neighbor embeddings of the single-cell RNA-seq developmental data.** Cells were sampled from human brain organoids (cell line 409b2) at seven time points between 0 days and 4 months into the development (Kanton et al., 2019). Sample size $n = 20\,272$. Data were reduced with PCA to 50 dimensions. See Appendix B for transcriptomic data preprocessing steps.

5. UMAP and ForceAtlas2 Can Be Placed on the Attraction-Repulsion Spectrum

Using the MNIST data set, we observed that FA2 produced an embedding very similar to t-SNE with $\rho \approx 30$, while UMAP produced an embedding very similar to t-SNE with $\rho \approx 4$ (Figure 1). The same was true for the simulated data set (Figure 5) and for the brain organoid data set (Figure 6), as well as for the seven further data sets that we analyzed in addition (Figures A2, A4, A5, A6, A7, A8, A9).

To quantify this observation, we computed distance correlations (Szekely et al., 2007) between UMAP & FA2 embeddings and t-SNE embeddings with various values of $\rho \in [1, 100]$ (Figure 8). We found that for most data sets the highest correlation between UMAP and t-SNE layouts was achieved at $4 \leq \rho < 15$ (Figure 8a). For FA2, the highest correlation was typically achieved at $20 < \rho < 80$ (Figure 8b). In both cases, the maximum correlations were above 0.94, indicating very similar layouts. Whereas the exact value of ρ yielding the maximum correlation varied between data sets, the correlation values at $\rho = 4$ for UMAP and at $\rho = 30$ for FA2 were always high and close to the maximum correlations. We also observed that $\rho = 4$ yielded a good match to UMAP independent of the sample size (Figure A10).

Note that for all three algorithms we used all default parameters (apart from always using the same PCA initialization and fixing $a = b = 1$ in UMAP), confirming that the differences between t-SNE and UMAP in affinities and in the value of ϵ in the loss function do not play a large role, at least for our data sets. We used the shared PCA initialization to roughly align the embeddings and make distance correlations more meaningful, but all conclusions stayed the same if we used default UMAP/FA2 initializations (see Section 3.6).

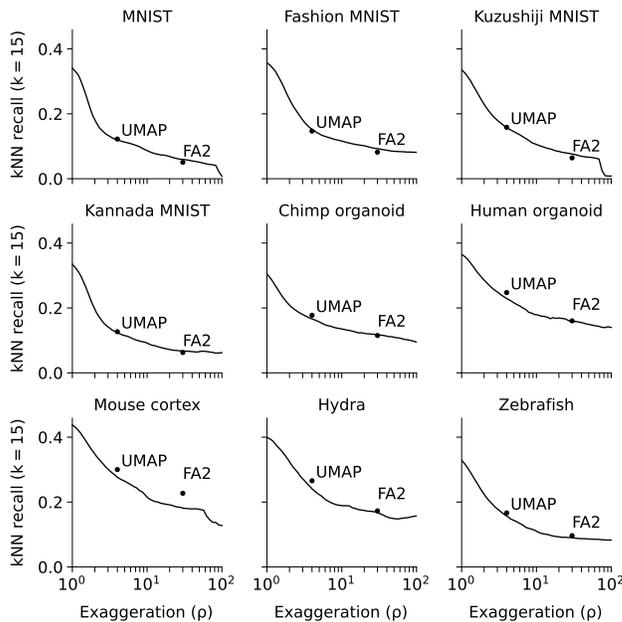


Figure 7: **Nearest neighbors recall as a function of ρ .** The fraction of $k = 15$ nearest neighbors in high dimensions that remain among the nearest neighbors in the embedding (average over 10 000 randomly selected points; see text). The values for UMAP and FA2 are shown at $\rho = 4$ and $\rho = 30$.

A caveat here is that distance correlation metric can be strongly affected by the exact placement of the clusters, and does not always capture the intuitive notion of ‘similarity’. For example, both correlation curves for the Kannada MNIST data set (Figure 8, red lines) peak at almost the same value of ρ , but visual inspection of the embeddings (Figure A5) suggests that $\rho = 4$ is qualitatively closer to UMAP, while $\rho = 30$ is qualitatively closer to FA2, in agreement with all other data sets.

The k NN recall of UMAP and FA2 was also similar to the k NN recall of t-SNE with exaggeration set to $\rho = 4$ and $\rho = 30$ respectively, across all analyzed data sets (Figure 7). This suggests that not only the general layout, as measured by the distance correlation, but also the local structure of the embedding was similar between UMAP/FA2 and t-SNE with appropriate exaggeration.

6. Increased Attraction in UMAP Due to Negative Sampling

As shown above, the gradient of UMAP (Eq. 7) is very similar to the gradient of t-SNE (Eq. 2) but does not contain the ‘normalizing’ n/Z term in front of the repulsive forces. What are the typical values of this coefficient? The normalization term Z in t-SNE evolves during optimization: it starts at $Z \approx n^2$ due to all $d_{ij} \approx 0$ at initialization and decreases as the embedding expands. For a perfect embedding with all $p_{ij} = q_{ij}$ and $v_{ij} = w_{ij}$, Z would be equal to n ; in reality Z usually still exceeds n . We found that for all nine data sets analyzed here, the value of Z in the end of optimization with $\rho = 1$ was in the range $[50n, 120n]$ (Figure 9a). For MNIST, the final Z value was $\sim 100n$, corresponding to the final $n/Z \approx 0.01$ (Figure 9b). Increasing the exaggeration shrinks the embedding and increases the final Z ; it also changes the repulsive factor to $n/(\rho Z)$ (Eq. 3). Across all data sets, the final Z value with $\rho = 4$ was in the $[400n, 2300n]$ range (Figure 9a). For MNIST, it was $\sim 2100n$, corresponding to the final $n/(\rho Z) \approx 0.0001$. This means that UMAP matched t-SNE results with the repulsive factor 0.0001 better than it matched t-SNE results with the repulsive factor 0.01, even though UMAP itself uses repulsive factor $\gamma = 1$ (Eq. 7). How is this possible?

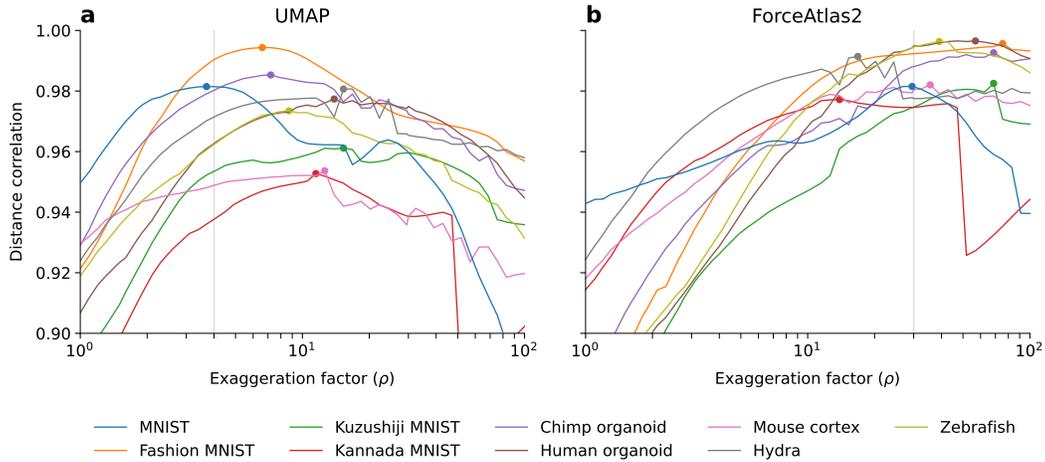


Figure 8: **Distance correlations between UMAP/FA2 and t-SNE.** Exaggeration values $\rho \in [1, 100]$ were evenly distributed on a log-scale, with $\rho = 4$ and $\rho = 30$ added explicitly; 52 points in total. Distance correlation (Szekely et al., 2007) was computed using dcor package (Carreño, 2017) on a random subset ($n = 5\,000$) of the data. Dots mark the maximum of each curve. **(a)** Distance correlation between UMAP and t-SNE. **(b)** Distance correlation between FA2 and t-SNE.

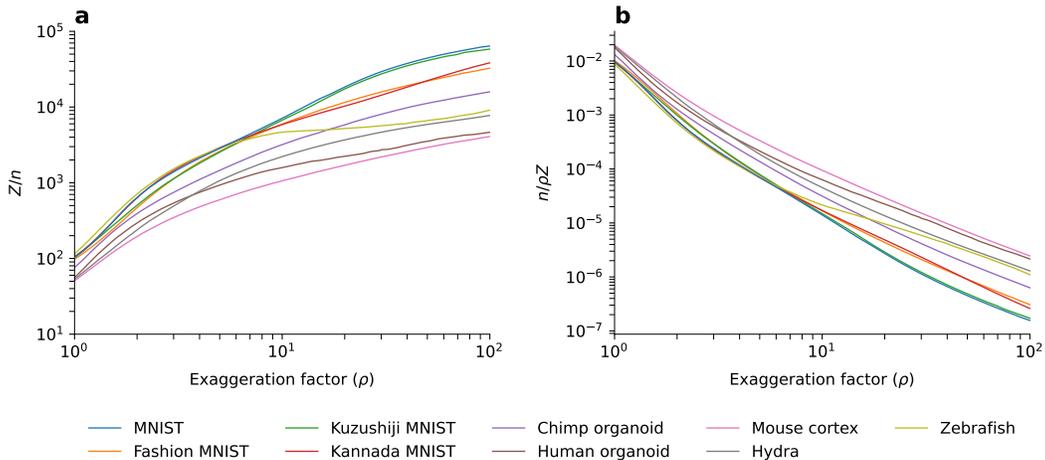


Figure 9: **(a)** The normalization term Z/n computed for all data sets considered in the manuscript, as a function of ρ . **(b)** The term $n/(\rho Z)$ computed for all data sets considered in the manuscript.

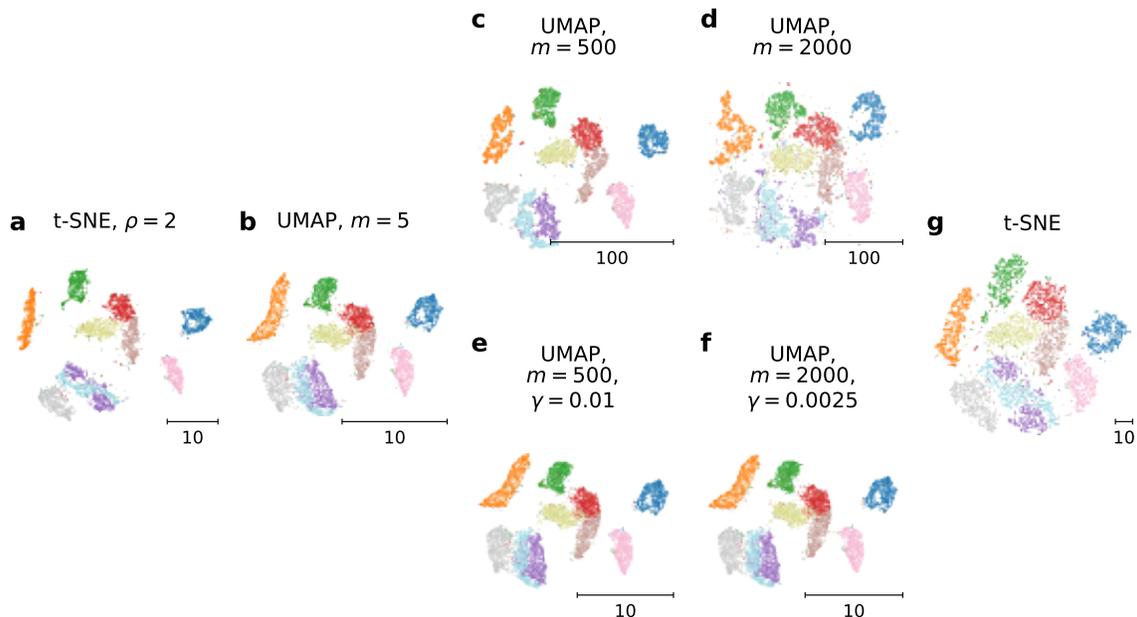


Figure 10: **The effect of negative sampling rate on UMAP embeddings.** MNIST subsample with $n = 6000$. We used a subsample of MNIST because the runtime scales as $\mathcal{O}(mn)$, making it impractical to use $m \approx n$ for large n . UMAP (panels b–f) was run for 3000 epochs to ensure convergence, as large m introduce noise in the optimization. (a) T-SNE embedding with $\rho = 2$. (b–d) UMAP embeddings with $m \in \{5, 500, 2000\}$. Panels (c) and (d) were initialized with the standard UMAP embedding ($m = 5$, 750 epochs) to simulate early exaggeration. (e–f) UMAP embeddings with $m \in \{500, 2000\}$, while keeping the product $\gamma \cdot m$ constant. (g) Standard t-SNE of the same data.

We hypothesized that this mismatch arises because the UMAP implementation is based on negative sampling and does not in fact optimize its stated loss (Eq. 5). Instead, the negative sampling decreases the repulsion strength, creating an effective $\gamma_{\text{eff}}(m) \ll 1$. We verified that increasing the value of m increased the repulsion strength in UMAP (Figure 10): embeddings grew in size and the amount of between-cluster white space decreased. But when we decreased the γ factor together with increasing m so that their product $\gamma \cdot m$ stayed constant, the embedding did not change at all (Figure 10e,f), confirming that the negative sampling rate m directly controls the repulsion strength.

The repulsion strength in UMAP can also be explicitly controlled by the γ parameter. Decreasing the γ value had the same effect as increasing the ρ value in t-SNE, and moved the UMAP result towards the LE part of the attraction-repulsion spectrum (Figure A11). We found it not possible to increase the repulsion strength by setting $\gamma \gg 1$, likely due to convergence problems.

We can approximately compute the effective repulsion coefficient $\gamma_{\text{eff}}(m)$ arising in UMAP through the negative sampling as follows. The number of repulsive forces per one attractive force is $\sim n/k$ in the full gradient but m with negative sampling. This suggests that $\gamma_{\text{eff}} \approx km/n$ and should decrease with the sample size as $\mathcal{O}(1/n)$.

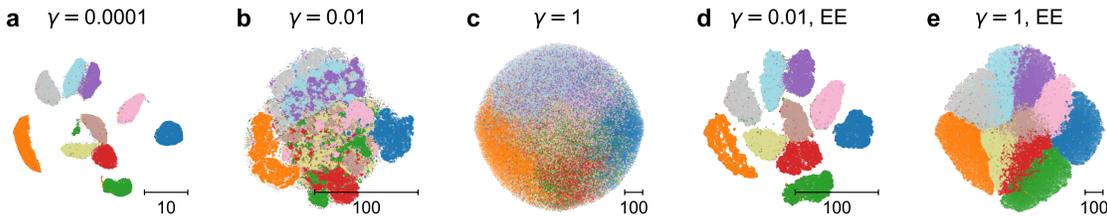


Figure 11: **Barnes–Hut UMAP without negative sampling.** (a–c) Embeddings with $\gamma \in \{0.0001, 0.01, 1\}$. (d–e) Embeddings with gamma values $\gamma \in \{0.01, 1\}$ initialized with the embedding with $\gamma = 0.0001$ [panel (a)], in analogy to early exaggeration in t-SNE.

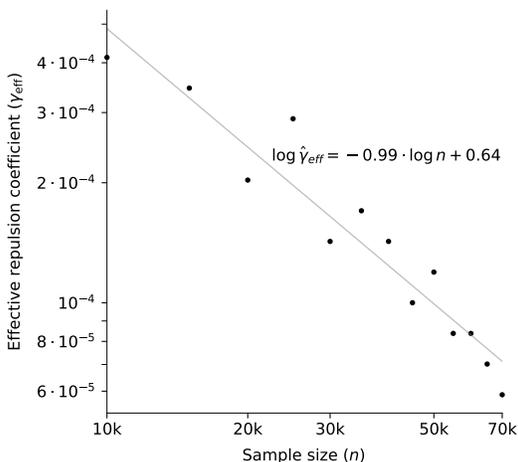


Figure 12: **Estimating the effective repulsion in UMAP.** We ran Barnes–Hut UMAP for subsets of MNIST with sample sizes from 10 000 to 70 000 in the increments of 5000, using a grid of γ values. For each sample size, we found the γ value matching the size of negative-sampling-based UMAP. The grey line shows the linear regression fit to the log-transformed data. Sample sizes below 10 000 led to noisy estimates and were excluded.

To confirm our interpretation, we developed a Barnes–Hut UMAP implementation that optimizes the full UMAP loss without any negative sampling (see Section 3.6). On full MNIST, $\gamma = 0.0001$ yielded an embedding that resembled the standard (negative-sampling-based) UMAP (Figure 11a), while larger values of γ yielded over-repulsed embeddings (Figure 11b,c) and required early exaggeration to produce meaningful results (Figure 11d,e), with $\gamma = 0.01$ resembling t-SNE and $\gamma = 1$ being over-repulsed compared to t-SNE. This suggests that directly optimizing the cross-entropy loss (Eq. 5) leads to an embedding where the repulsive forces strongly dominate visual appearance (Figure 11c,e).

Furthermore, we can use our Barnes–Hut implementation to directly estimate γ_{eff} . To do that, we ran the Barnes–Hut implementation for a range of sample sizes and a range of γ values (40 values of γ evenly log-spaced between 0.01 and 10^{-5}). For each sample size, we found the γ value giving the best match to the size ($\max \mathbf{Y} - \min \mathbf{Y}$) of the negative-sampling-based UMAP (Figure 12). The resulting $\hat{\gamma}_{\text{eff}}$ estimates decreased with the sample size as $1/n$, with regression slope being -0.99 on the log-log plot (Figure 12), confirming our prediction. In a follow-up work, Damrich and Hamprecht (2021) analyzed the UMAP sampling procedure in more detail, showing that $\gamma_{\text{eff}} \approx \log k \cdot m/n$.

In popular expositions (Coenen and Pearce, 2019; Oskolkov, 2019), the success of UMAP and its visually appealing embeddings have been attributed to its cross-entropy loss function and its topological foundations. However, our conclusion is that the more condensed clusters typically

observed in UMAP compared to t-SNE are a serendipitous consequence of UMAP’s negative sampling strategy. The mathematical framework and the cross-entropy loss function developed in the original paper (McInnes et al., 2018) would lead to very different and suboptimal embeddings (Figure 11c,e), if not for the negative sampling. Note that we are not criticizing the actual UMAP embeddings here; our statement is that UMAP’s stated and effective loss functions are qualitatively different.

7. Increased Attraction in FA2 Due to Non-Decaying Attractive Forces

The attractive forces in t-SNE scale as $d_{ij}/(1 + d_{ij}^2)$. When all d_{ij} are small, this becomes an approximately linear dependency on d_{ij} , which is the reason why t-SNE with high exaggeration $\rho \gg 1$ replicates Laplacian eigenmaps (see Section 3.5 and Appendix A). For large distances d_{ij} , attractive forces in t-SNE decay to zero, making default t-SNE very different from LE. In contrast, in FA2, attractive forces always scale as d_{ij} . Thus, the larger the embedding distance between two points, the stronger the attractive force between them. This strong non-decaying attractive force moves FA2 towards Laplacian eigenmaps on the attraction-repulsion spectrum.

While the attractive forces in FA2 are the same as in Laplacian eigenmaps, FA2 has repulsive forces instead of the quadratic constraint of LE. This moves FA2 somewhat away from LE on the attraction-repulsion spectrum. These arguments provide a qualitative explanation for why FA2 behaves similar to t-SNE with strong exaggeration ($\rho \approx 30$, as we empirically showed above), but more quantitative analysis remains for future work. In addition, our arguments suggest that the exact scaling law of the repulsive forces ($1/d_{ij}^2$ or $1/d_{ij}$) may have little qualitative influence on the resulting embedding as long as the attractive forces remain linear in d_{ij} . We leave it for future work to investigate this.

Note that it is not possible to move FA2 embeddings along the attraction-repulsion spectrum by multiplying the attractive or repulsive forces by a constant factor (such as γ in UMAP or ρ in t-SNE). Multiplying attractive forces by any factor a or repulsive forces by any factor $1/a$ only leads to rescaling of the embedding by $1/\sqrt{a}$. Indeed, if all forces are in equilibrium before such multiplication and rescaling, they will stay in equilibrium afterwards. This is a general property of force-directed layouts where both attractive and repulsive forces scale as powers of the embedding distance d_{ij} .

8. Discussion

We showed that changing the balance between attractive and repulsive forces in t-SNE directly affects the trade-off between preserving continuous/global or discrete/local structures. Increasingly strong repulsion ‘brings out’ information from higher Laplacian eigenvectors into the two embedding dimensions (Figure 1). It is remarkable that the repulsive forces, which are data-agnostic and do not depend on the input data (Carreira-Perpiñán, 2010), have so much qualitative influence.

While we only considered the exaggeration factor ρ here, other parameters of t-SNE can also qualitatively affect the resulting embedding. In particular, the tail-heaviness of the low-dimensional similarity kernel (Yang et al., 2009) controls the emphasis put on the fine cluster structure of the data (Kobak et al., 2020). There is thus non-trivial interaction between the exaggeration ρ and the tail-heaviness parameter α , which we illustrate using the MNIST data set in Figure A12, but leave more detailed exploration of this two-dimensional parameter space for future work.

Our results suggest that it is beneficial for high-repulsion embeddings to begin optimization with lower repulsion strength, in order to better preserve global structure. This explains how UMAP benefits from its default initialization with Laplacian eigenmaps (Kobak and Linderman, 2021) and how t-SNE benefits from early exaggeration (Linderman and Steinerberger, 2019) (Figure A13 demonstrates the importance of early exaggeration in t-SNE). Similarly, Carreira-Perpiñán (2010) suggested to gradually increase repulsion strength during optimization of elastic embedding. A promising approach to t-SNE optimization would be to use Laplacian eigenmaps for initialization and replace the early exaggeration phase with gradual annealing of the exaggeration factor ρ from ‘infinity’ down to its final desired value.

Our treatment provides a unified perspective on several well-known NE algorithms that have scalable implementations and that have been shown to successfully embed data sets such as MNIST without coarse-graining the k NN graph. Methods based on coarse-graining, such as PHATE (Moon et al., 2019) or latent variable NE method in Saul (2020) may behave differently. We believe that our treatment may allow to position other NE algorithms on the same spectrum. For example, a recently suggested TriMap algorithm (Amid and Warmuth, 2019), which uses negative sampling similar to UMAP, appears to have stronger attractive forces than UMAP (cf. Figure 5 in the original paper), with some TriMap embeddings, e.g. of the Fashion MNIST data set, looking similar to the ForceAtlas2 embeddings shown in our work. It remains for future work to investigate if and how some of the more recent NE algorithms based on negative sampling fit on the attraction-repulsion spectrum. This includes, for example, IHVD (Minch et al., 2020) and MDE (Agrawal et al., 2021). The latter work developed a flexible NE framework that can combine various attractive and repulsive forces optimized using negative sampling, with the quadratic constraint of Laplacian eigenmaps, resulting in a rich family of embeddings.

We argue that negative sampling (Mikolov et al., 2013), used by LargeVis/UMAP, strongly lowers the effective repulsion, compared to the stated cross-entropy loss function. In a follow-up work, Damrich and Hamprecht (2021) have developed a more formal analysis of negative sampling in UMAP and confirmed our findings.

Negative sampling exhibits some similarity to stochastic gradient descent (SGD), where the gradient is repeatedly computed on small random subsets of the data, known as mini-batches. However, we believe that this analogy is not helpful. SGD iterates over the entire training set, partitioned in mini-batches. Small mini-batches increase the variance of the gradient estimates but do not introduce any bias. Negative sampling, on the other hand, only samples a small subset of the repulsive forces for each attractive force, introducing a systematic bias into the gradient computation.

Negative sampling is closely related to the *noise-contrastive estimation* (NCE) framework (Gutmann and Hyvärinen, 2012). NCE was recently applied to t-SNE under the name of NCVis (Artemenkov and Panov, 2020), and the general NCE theory asserts that it should be asymptotically equivalent to optimizing the full gradient (Gutmann and Hyvärinen, 2012). We consider it an interesting research direction to study the relationship between negative sampling and NCE and their effect on 2D embeddings as well as on higher-dimensional embeddings used in methods like word2vec (Mikolov et al., 2013).

The practical takeaway from our work is not that one of the considered algorithms is the ‘best’. All three algorithms discussed in this manuscript (t-SNE, UMAP, ForceAtlas2) are widely used in several academic fields, like single-cell biology (Becht et al., 2019; Kobak and Berens, 2019) or population genomics (Diaz-Papkovich et al., 2019; Karczewski et al., 2020). But the choice of the method is often done without a solid understanding of why the results may be different or what

trade-offs are at play. Our work highlights that which algorithm is more appropriate may depend on the question one wants to answer.

If the goal of the analysis is to explore the fine clusters and the local neighborhood structure, one should use t-SNE with default exaggeration parameter $\rho = 1$ (Figure 7). If the interest rather lies in exploring the global structure of the data, such as continuous or temporal trajectories, then it may be helpful to increase the exaggeration in t-SNE or to use UMAP or ForceAtlas2. All things considered, t-SNE with exaggeration is able to cover the entire attraction-repulsion spectrum (Figure 1), whereas UMAP with negative sampling is only able to produce visualizations with relatively high attraction. ForceAtlas2 is even more limited in that it cannot move along the spectrum and corresponds to high attraction. In terms of the running speed, modern implementations of t-SNE (Linderman et al., 2019; Artemenkov and Panov, 2020) and UMAP are comparable.

We hope that the treatment developed here will allow researchers to make an informed choice between algorithms in practical applications.

Acknowledgments

We thank Sebastian Damrich, George Linderman, Stefan Steinerberger, James Melville, and Ulrike von Luxburg for helpful discussions; Pavlin Poličar for discussions and openTSNE support; and He Zhisong for the help with loading the organoid transcriptomic data.

This research was funded by the Cyber Valley Research Fund (D.30.28739), the Deutsche Forschungsgemeinschaft through a Heisenberg Professorship (BE5601/4-1), the Excellence Cluster 2064 “Machine Learning: New Perspectives for Science” (390727645), the National Institute Of Mental Health of the National Institutes of Health (U19MH114830), and the German Ministry of Education and Research (01IS18039A, 01GQ1601). The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. The authors declare no financial conflict of interest and did not receive any donations/funding from industry with relationship to this project.

The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Jan Niklas Böhm.

Appendix A. Relationship to Laplacian Eigenmaps

Laplacian eigenmaps Let a $n \times n$ symmetric matrix \mathbf{V} contain pairwise affinities between n points (or edge weights between nodes in an undirected graph). Let the diagonal matrix \mathbf{D} contain row (or, equivalently, column) sums of \mathbf{V} , that is, $D_{ii} = \sum_j V_{ij}$. Then $\mathbf{L} = \mathbf{D} - \mathbf{V}$ is known as the (unnormalized) graph Laplacian. Laplacian eigenmaps (Belkin and Niyogi, 2002) can then be formulated as solving the generalized eigenvector problem

$$\mathbf{L}\mathbf{a} = \lambda\mathbf{D}\mathbf{a}$$

and taking the eigenvectors corresponding to the *smallest* eigenvalues (after discarding the trivial eigenvector $[1, 1, \dots, 1]^\top$ with eigenvalue zero). By multiplying both sides of this equation by \mathbf{D}^{-1} , the problem can be reformulated as finding the eigenvectors of $\mathbf{D}^{-1}\mathbf{V}$ corresponding to the *largest* eigenvectors:

$$\mathbf{D}^{-1}\mathbf{V}\mathbf{a} = (1 - \lambda)\mathbf{a}.$$

The matrix $\mathbf{D}^{-1}\mathbf{V}$ is not symmetric and has rows normalized to 1. It can be interpreted as a diffusion operator on the graph, making Laplacian eigenmaps equivalent to *Diffusion maps* (Coifman and Lafon, 2006). Another equivalent way to rewrite it, is to define normalized Laplacian $\mathbf{L}_{\text{norm}} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$ and solve an eigenvector problem $\mathbf{L}_{\text{norm}}\mathbf{b} = \lambda\mathbf{b}$, where $\mathbf{b} = \mathbf{D}^{1/2}\mathbf{a}$.

t-SNE without repulsion In the limit of $\rho \rightarrow \infty$, the repulsive term in the t-SNE gradient can be dropped, all $w_{ij} \rightarrow 1$, and hence the gradient descent update rule becomes (Linderman and Steinerberger, 2019)

$$\mathbf{y}_i^{t+1} = \mathbf{y}_i^t - \eta \sum_j v_{ij}(\mathbf{y}_i^t - \mathbf{y}_j^t),$$

where t indexes the iteration number and η is the learning rate (including all constant factors in the gradient). Denoting by \mathbf{Y} the $n \times 2$ matrix of the embedding coordinates, this can be rewritten as

$$\begin{aligned} \mathbf{Y}^{t+1} &= (\mathbf{I} - \eta\mathbf{D} + \eta\mathbf{V})\mathbf{Y}^t \\ &= \mathbf{M}\mathbf{Y}^t. \end{aligned}$$

\mathbf{M} is the transition matrix of this Markov chain (note that it is symmetric and its rows and columns sum to 1; its values are all non-negative for small enough η). According to the general theory of Markov chains, the largest eigenvalue of \mathbf{M} is 1, and the corresponding eigenvector is $[1, 1, \dots, 1]^\top$, meaning that the embedding shrinks to a single point (as expected without repulsion). The slowest shrinking eigenvectors correspond to the next eigenvalues. This means that when ρ becomes very large, the embedding will resemble leading nontrivial eigenvectors of \mathbf{M} (as ρ grows, the embedding will become smaller and smaller, but here we ignore its overall size; eigenvectors can have arbitrary length so overall scale of the embedding is not important here). This becomes equivalent to a power iteration algorithm. The eigenvectors of \mathbf{M} are the same as of $\mathbf{L} = \mathbf{D} - \mathbf{V}$, which is the unnormalized graph Laplacian of the symmetric affinity matrix.

Note that this is not precisely what LE computes: as explained above, LE finds eigenvectors of the *normalized* graph Laplacian (von Luxburg et al., 2008). However, for t-SNE affinities, \mathbf{D} is approximately proportional to the identity matrix, because \mathbf{V} is obtained via symmetrization of directed affinities, and those have rows summing to 1 by construction. We can therefore expect that the leading eigenvectors of \mathbf{L} and of \mathbf{L}_{norm} are very close. We verified that for MNIST data they were almost exactly the same. A more abstract perspective is provided by von Luxburg et al. (2008): the unnormalized spectral embedding corresponds to minimizing a graph cut with respect to the number of vertices, whereas LE takes vertex degrees into account. Since all vertices in a k NN or perplexity-based graph have an almost equal degree, the eigenvectors of the normalized and unnormalized Laplacian will be very close.

Note also that nothing prevents different columns of \mathbf{Y} to converge to the same leading eigenvector: each column independently follows its Markov chain. Indeed, we observed that for large enough values of ρ and large enough number of gradient descent iterations, the embedding collapsed to one dimension. This is the expected limiting behaviour when $\rho \rightarrow \infty$. However, for moderate values of ρ (as shown in this manuscript), this typically does not happen, and columns of \mathbf{Y} resemble the two leading non-trivial eigenvectors of the Laplacian. The repulsive force prevents the embedding from collapsing to the leading Laplacian eigenvector. At the same time, a weak repulsive force will only be able to ‘bring out’ the second LE eigenvector. The stronger the contribution of repulsive forces, the more LE eigenvectors it would be able to ‘bring out’ (remember that the attractive force acts stronger on the higher eigenvectors).

Loss function of LE and quadratic constraint The original Laplacian eigenmaps paper (Belkin and Niyogi, 2002) motivates the eigenvector problem by considering

$$\mathcal{L}_{LE} = \sum_{i,j} v_{ij} \|\mathbf{y}_i - \mathbf{y}_j\|^2 = 2 \text{Tr}(\mathbf{Y}^\top \mathbf{L} \mathbf{Y}).$$

This expression can be trivially minimized by setting all $\mathbf{y}_i = \mathbf{0}$, so the authors introduce a quadratic constraint $\mathbf{Y}^\top \mathbf{D} \mathbf{Y} = \mathbf{I}$, yielding the generalized eigenvector problem. We note that a different quadratic constraint $\mathbf{Y}^\top \mathbf{Y} = \mathbf{I}$ would yield a simple eigenvector problem for \mathbf{L} . In any case, the constraint plays the role of the repulsion in t-SNE framework.

Appendix B. Data Sources and Transcriptomic Data Preprocessing

The brain organoid data sets (Kanton et al., 2019) were downloaded from <https://www.ebi.ac.uk/arrayexpress/experiments/E-MTAB-7552/> in form of UMI counts and metadata tables. The metadata table for the chimpanzee data set was taken from the supplementary materials of the original publication. We used gene counts mapped to the consensus genome, and selected all cells that passed quality control by the original authors (`in_FullLineage=TRUE` in metadata tables). For human organoid data, we only used cells from the 409b2 cell line, to simplify the analysis (the original publication combined cells from two cell lines and needed to perform batch correction).

The hydra data set (Figure A7; Siebert et al., 2019) was downloaded in form of UMI counts from <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE121617>.

The zebrafish data set (Figure A8; Wagner et al., 2018b) was downloaded in form of UMI counts from https://kleintools.hms.harvard.edu/paper_websites/wagner_zebrafish_timecourse2018/WagnerScience2018.h5ad.

The adult mouse cortex data set (Figure A9; Tasic et al., 2018) was downloaded in form of read counts from http://celltypes.brain-map.org/api/v2/well_known_file_download/694413985 and http://celltypes.brain-map.org/api/v2/well_known_file_download/694413179 for the VISp and ALM cortical areas, respectively. Only exon counts were used here. The cluster labels and cluster colors were retrieved from <http://celltypes.brain-map.org/rnaseq/mouse/v1-alm>.

To preprocess each data set, we selected the 1000 most variable genes using the procedure from Kobak and Berens (2019) with default parameters (for the mouse cortex data set we used 3000 genes and `threshold=32`; Kobak and Berens, 2019) and followed the preprocessing pipeline from the same paper: normalized all counts by cell sequencing depth (sum of gene counts in each cell), multiplied by the median cell depth (or 1 million in case of mouse cortex data), applied $\log_2(x + 1)$ transformation, did PCA, and retained 50 leading PCs.

The data sets shown in Figures A4, A5, and A6 have been published explicitly to function as drop-in replacements for the hand-written MNIST data set. The data set variants that we used here all consist of a total $n = 70\,000$ images of 28×28 pixels, in 10 balanced classes. The input was preprocessed like the original MNIST data set by reducing the dimensions to 50 via PCA. Fashion and Kuzushiji MNIST were downloaded via OpenML with the keys `Fashion-MNIST` (<https://www.openml.org/d/40996>) and `Kuzushiji-MNIST` (<https://www.openml.org/d/41982>), respectively. Kannada MNIST was downloaded from https://github.com/vinayprabhu/Kannada_MNIST.

Appendix C. Supporting Experiments

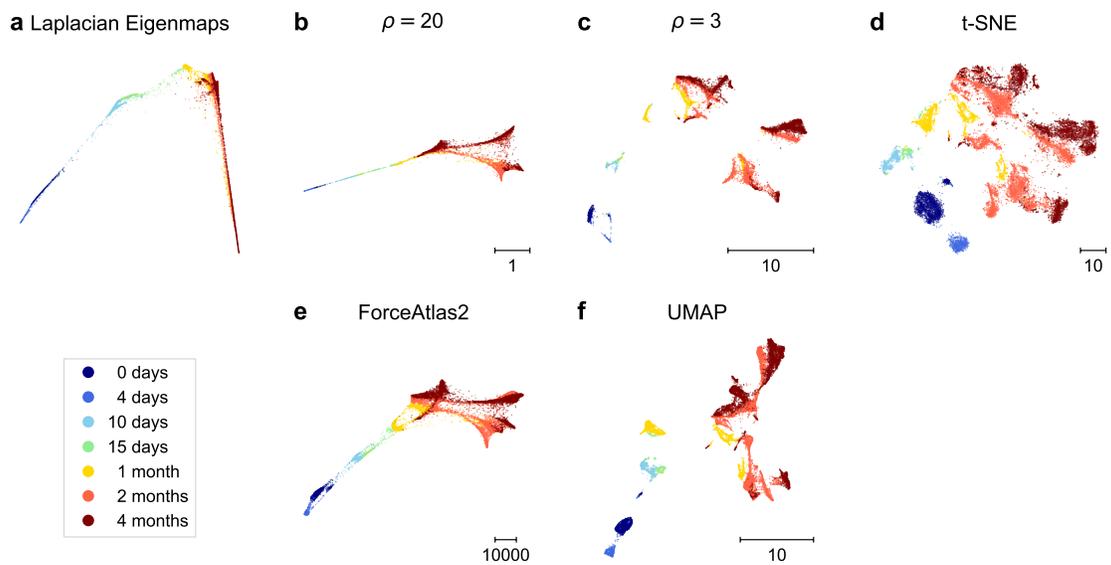


Figure A1: **Neighbor embeddings of the single-cell RNA-seq developmental data (human, high k).** The same as Figure 6, but LE, FA2, and UMAP used $k = 150$ (instead of our default $k = 15$), while t-SNE used perplexity 300 (instead of our default 30).

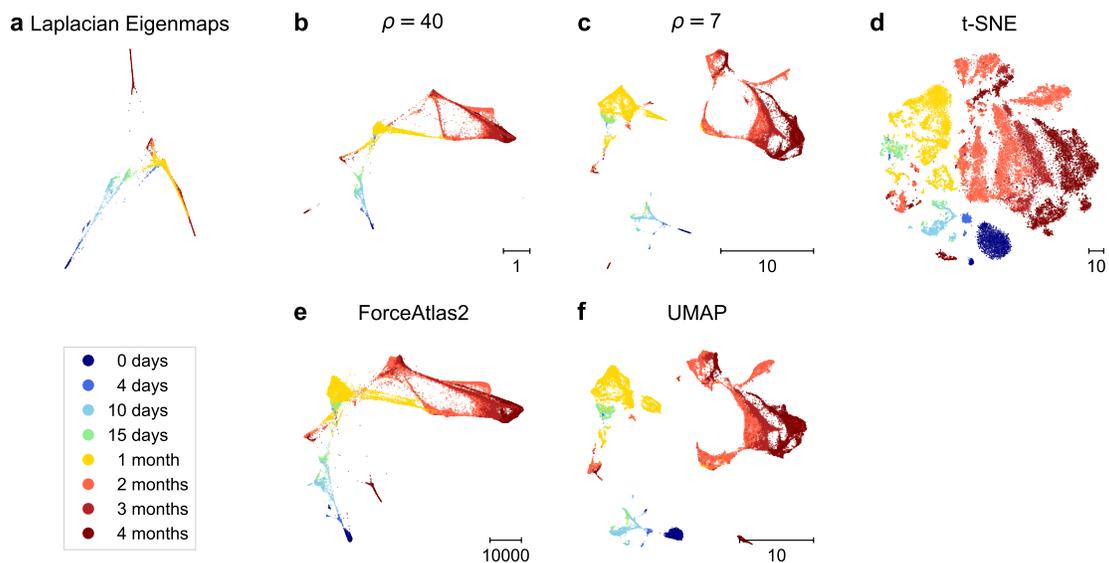


Figure A2: **Neighbor embeddings of the single-cell RNA-seq developmental data (chimpanzee).** Cells were sampled from chimpanzee brain organoids at eight time points between 0 days and 4 months into the development (Kanton et al., 2019). Sample size $n = 36884$. Data were reduced with PCA to 50 dimensions. See Appendix B for transcriptomic data preprocessing steps.

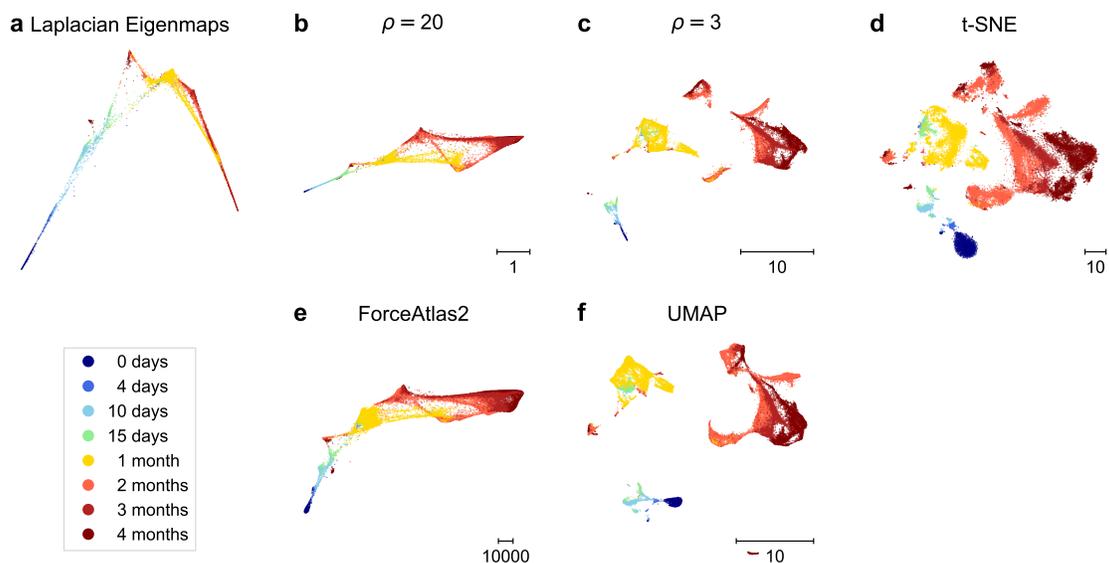


Figure A3: **Neighbor embeddings of the single-cell RNA-seq developmental data (chimpanzee, high k).** The same as Figure A2, but LE, FA2, and UMAP used $k = 150$ (instead of our default $k = 15$), while t-SNE used perplexity 300 (instead of our default 30).

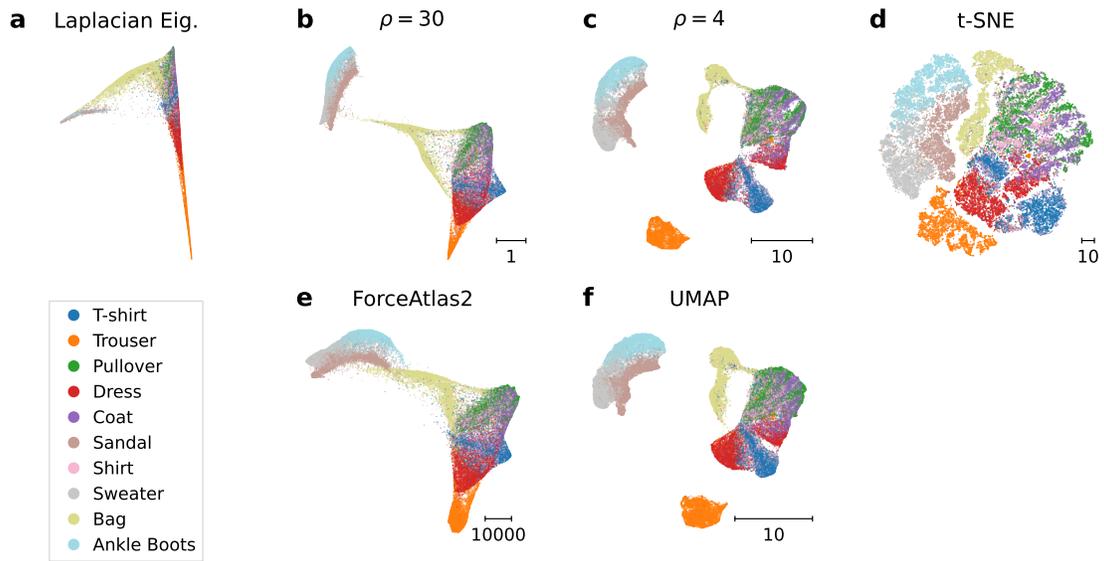


Figure A4: **Fashion MNIST data set (Xiao et al., 2017)**. Sample size $n = 70\,000$. Dimensionality was reduced to 50 with PCA. Colors correspond to 10 classes, see legend.

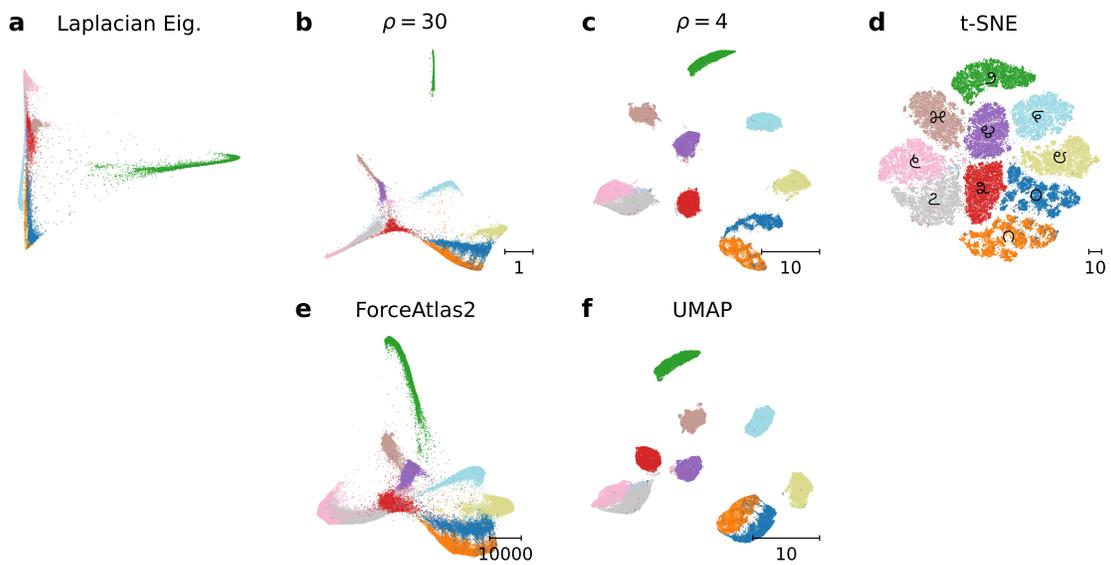


Figure A5: **Kannada MNIST data set (Prabhu, 2019)**. Sample size $n = 70\,000$. Dimensionality was reduced to 50 with PCA. Colors correspond to 10 Kannada digits shown in panel (d).

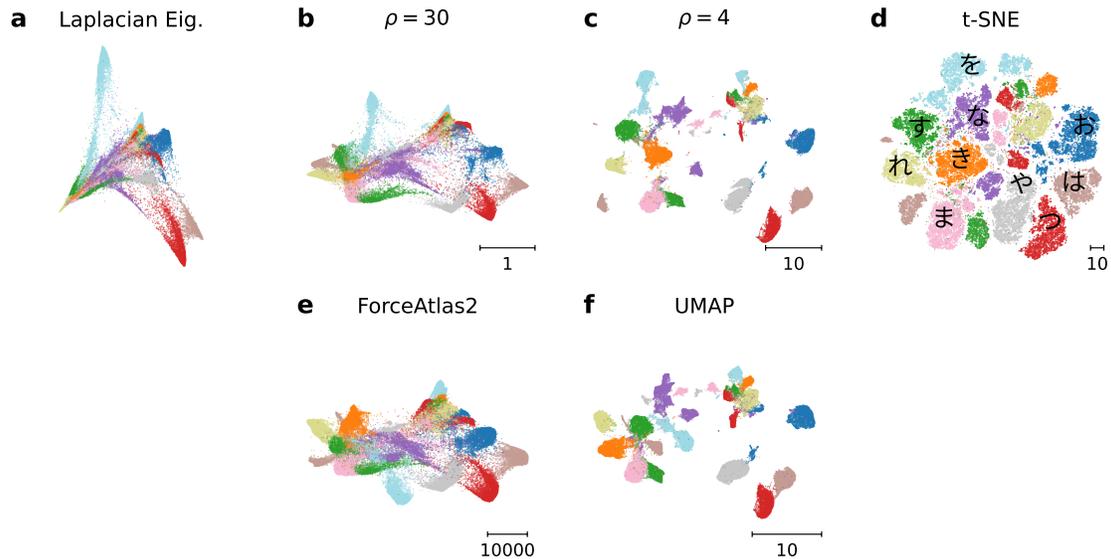


Figure A6: **Kuzushiji MNIST data set (Clanuwat et al., 2018)**. Sample size $n = 70\,000$. Dimensionality was reduced to 50 with PCA. Colors correspond to 10 Kanji characters shown in panel (d).

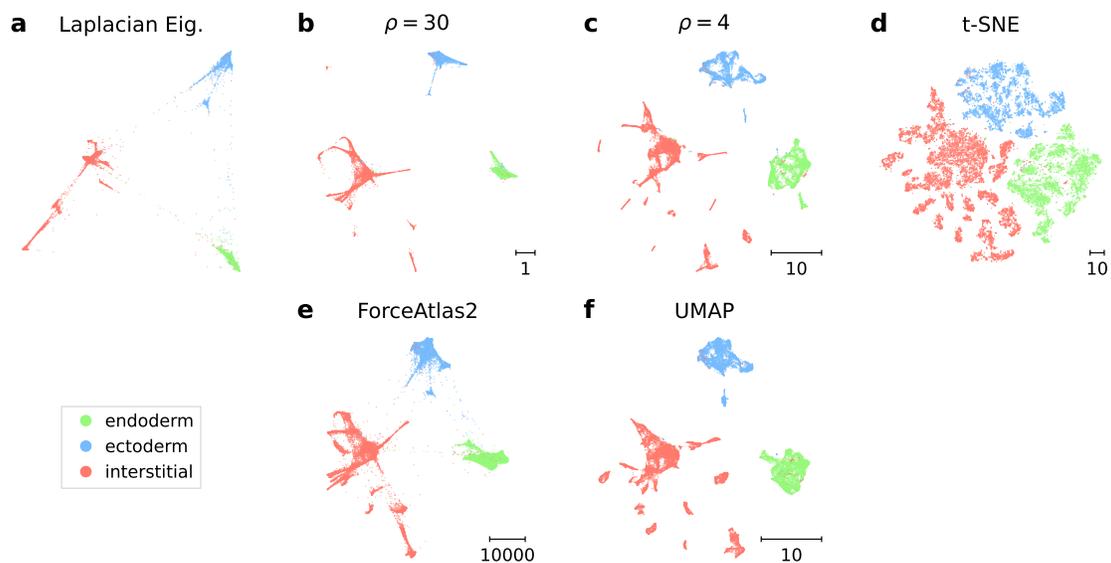


Figure A7: **Single-cell RNA-seq data of a hydra (Siebert et al., 2019)**. Sample size $n = 24\,985$. Dimensionality was reduced to 50 with PCA. See Appendix B for transcriptomic data preprocessing steps. Color corresponds to cell classes.

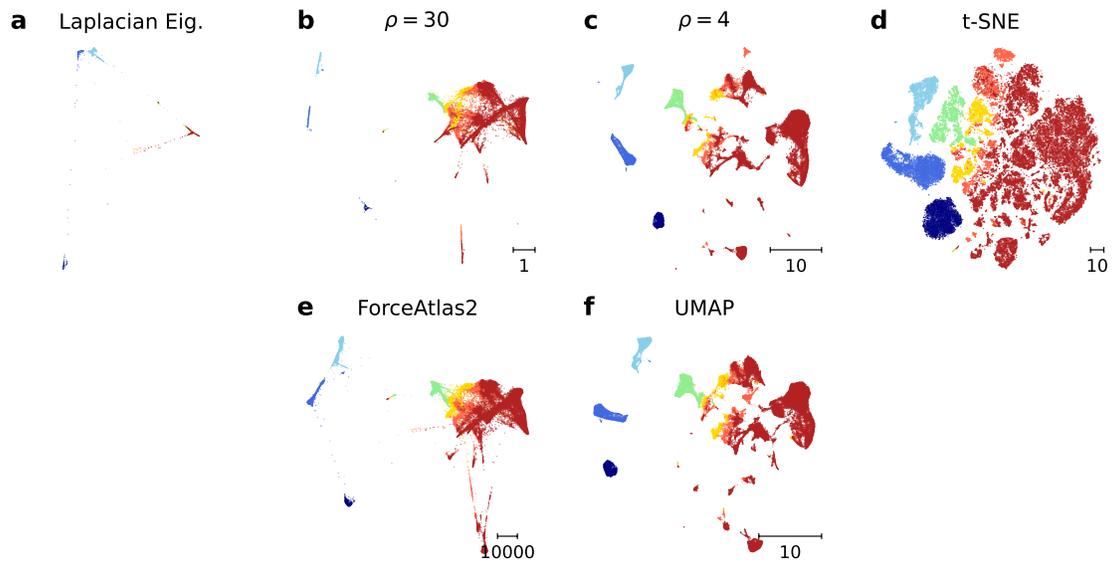


Figure A8: **Single-cell RNA-seq data of a zebrafish embryo (Wagner et al., 2018b)**. Sample size $n = 63\,530$. Dimensionality was reduced to 50 with PCA. See Appendix B for transcriptomic data preprocessing steps. Color corresponds to the developmental stage, indicating the hours post fertilization (hpf).

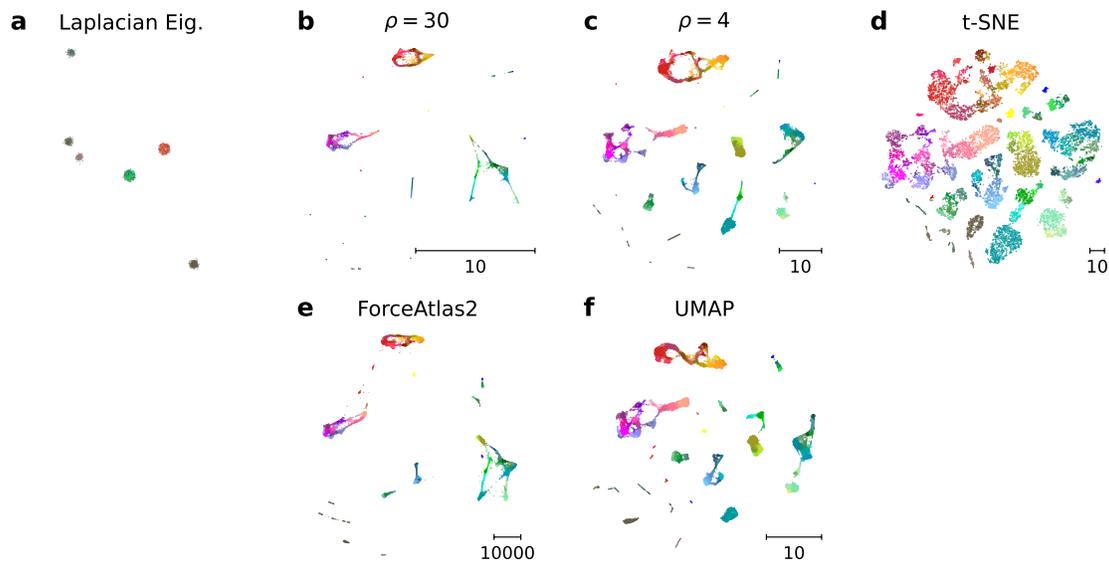


Figure A9: **Single-cell RNA-seq data of adult mouse cortex (Tasic et al., 2018)**. Sample size $n = 23\,822$. Dimensionality was reduced to 50 with PCA. See Appendix B for transcriptomic data preprocessing steps. Colors are taken from the original publication (warm colors: inhibitory neurons; cold colors: excitatory neurons; grey/brown: non-neural cells). We added Gaussian noise to the LE embedding in panel (a) to make the clusters more visible. In this data set, the k NN graph is disconnected and has 6 components, resulting in 6 distinct points in the LE embedding.

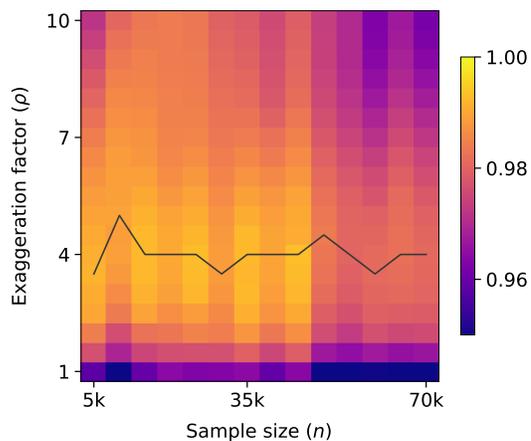


Figure A10: Distance correlations between t-SNE with $\rho \in [1, 10]$ and UMAP depending on the sample size, for MNIST subsets of size $n \in [5\,000, 70\,000]$. Black line indicates best matching ρ values.

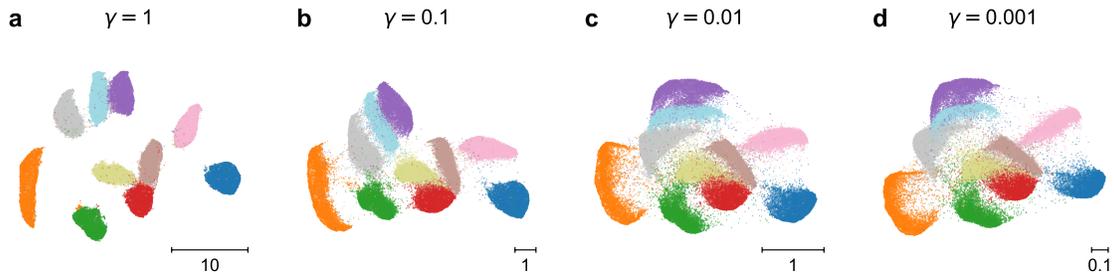


Figure A11: **Decreasing the repulsion in UMAP.** (a) UMAP embedding of MNIST with $\gamma = 1$ (default). (b–d) Decreasing γ produces the same effect as increasing the exaggeration ρ in t-SNE. Values $\gamma > 1$ are not shown because we could not achieve a well-converged embedding for $\gamma \gg 1$.

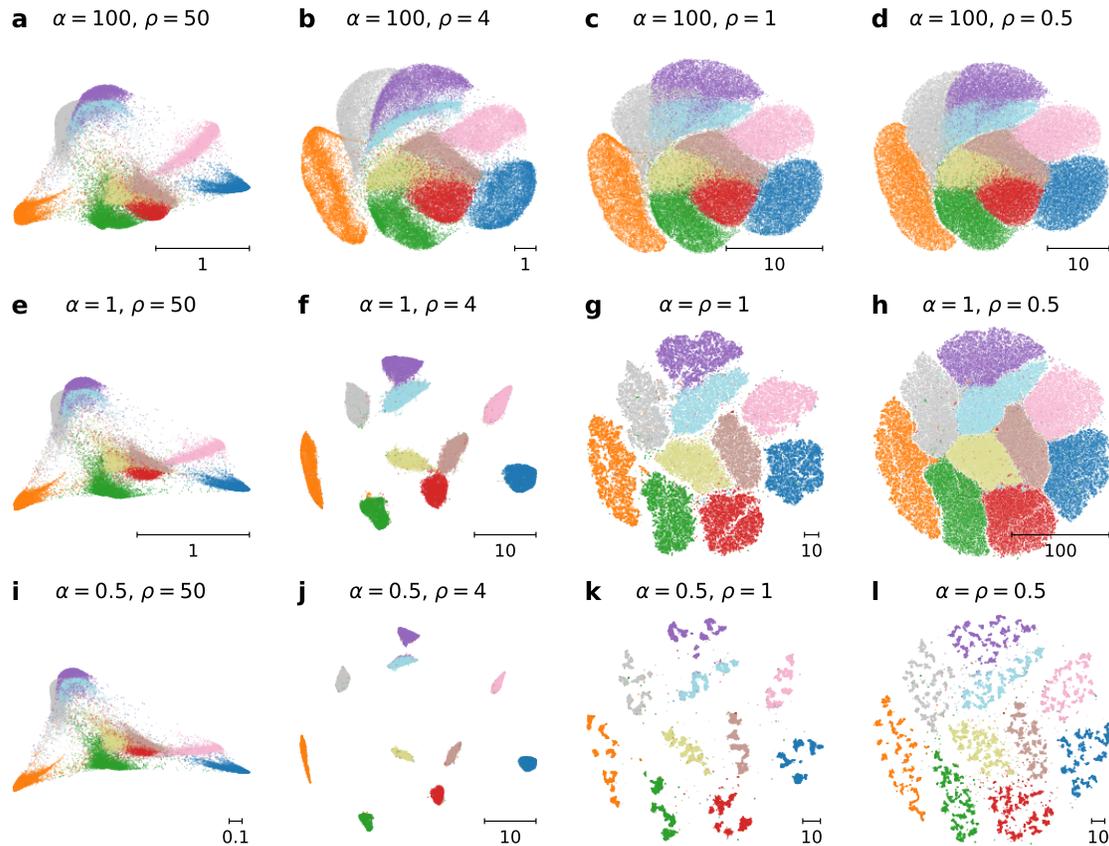


Figure A12: **Varying the tail-heaviness and exaggeration.** Changes in the layout for t-SNE of the MNIST data set when varying the tail-heaviness (Kobak et al., 2020; Yang et al., 2009) $\alpha \in \{100, 1, 0.5\}$ and the exaggeration factor $\rho \in \{50, 4, 1, 0.5\}$.

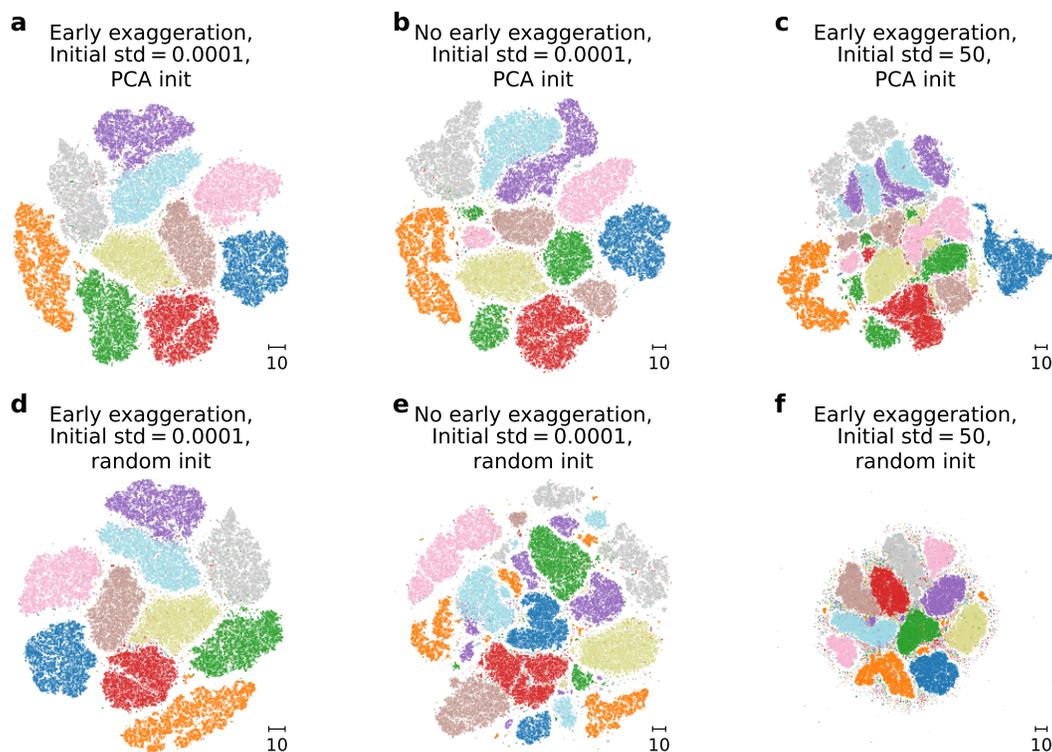


Figure A13: **The effect of early exaggeration on t-SNE.** (a) Default t-SNE embedding of MNIST. This uses early exaggeration and sets the standard deviation of PCA initialization to 0.0001. (b) T-SNE embedding without early exaggeration. This embedding is stuck in a suboptimal local minimum with some clusters split into multiple parts. (c) T-SNE embedding with early exaggeration, but with initial standard deviation set to 50. The attractive forces are too weak to pull the clusters together during the early exaggeration phase. (d) Default t-SNE with random initialization. The cluster structure is recovered, but the placement of the clusters is different from (a). (e) Same experiment as in (b), but with random initialization. The clusters are more fragmented due to less structure in the initialization and the lack of early exaggeration. (f) Same experiment as (c), but with random initialization. Here again, the attractive forces are too weak to pull the clusters together, and in addition there are points on the periphery that got stuck there due to the large initial distances.

References

- Akshay Agrawal, Alnur Ali, and Stephen Boyd. Minimum-distortion embedding. *Foundations and Trends in Machine Learning*, 14(3):211–378, 2021.
- Ehsan Amid and Manfred K Warmuth. TriMap: Large-scale dimensionality reduction using triplets. *arXiv preprint arXiv:1910.00204*, 2019.
- Ehsan Amid, Onur Dikmen, and Erkki Oja. Optimizing the information retrieval trade-off in data visualization using α -divergence. *arXiv preprint arXiv:1505.05821*, 2015.
- Sanjeev Arora, Wei Hu, and Pravesh K Kothari. An analysis of the t-SNE algorithm for data visualization. In *Conference On Learning Theory*, pages 1455–1462, 2018.
- Aleksandr Artemenkov and Maxim Panov. NCVis: Noise contrastive approach for scalable visualization. In *The Web Conference*, pages 2941–2947, 2020.
- Etienne Becht, Leland McInnes, John Healy, Charles-Antoine Dutertre, Immanuel WH Kwok, Lai Guan Ng, Florent Ginhoux, and Evan W Newell. Dimensionality reduction for visualizing single-cell data using UMAP. *Nature Biotechnology*, 37(1):38, 2019.
- S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D.S. Seljebotn, and K. Smith. Cython: The best of both worlds. *Computing in Science Engineering*, 13(2):31–39, 3 2011. ISSN 1521-9615.
- Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems*, pages 585–591, 2002.
- Anna C Belkina, Christopher O Ciccolella, Rina Anno, Richard Halpert, Josef Spidlen, and Jennifer E Snyder-Cappione. Automated optimized parameters for T-distributed stochastic neighbor embedding improve visualization and analysis of large datasets. *Nature Communications*, 10(1): 1–12, 2019.
- Erik Bernhardsson. Annoy. <https://github.com/spotify/annoy>, 2013.
- Miguel A Carreira-Perpiñán. The elastic embedding algorithm for dimensionality reduction. In *International Conference on Machine Learning*, volume 10, pages 167–174, 2010.
- Carlos Ramos Carreño. dcor: distance correlation and related E-statistics in Python. <https://github.com/vnmabus/dcor>, 2017.
- Bhargav Chippada. forceatlas2: Fastest Gephi’s ForceAtlas2 graph layout algorithm implemented for Python and NetworkX. <https://github.com/bhargavchippada/forceatlas2>, 2017.
- Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical Japanese literature. *arXiv preprint arXiv:1812.01718*, 2018.
- Andy Coenen and Adam Pearce. Understanding UMAP. <https://pair-code.github.io/understanding-umap>, 2019.
- Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006.

- Sesbastian Damrich and Fred Hamprecht. On UMAP’s true loss function. In *Advances in Neural Information Processing Systems*, 2021.
- Alex Diaz-Papkovich, Luke Anderson-Trocme, Chief Ben-Eghan, and Simon Gravel. UMAP reveals cryptic population structure and phenotype heterogeneity in large genomic cohorts. *PLoS Genetics*, 15(11):e1008432, 2019.
- Thomas M. J. Fruchterman and Edward M. Reingold. Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164, 1991.
- Emden R. Gansner, Yifan Hu, and Stephen North. A maxent-stress model for graph layout. In *Proceedings of IEEE Pacific Visualization Symposium*, pages 73–80, 2012.
- Michael U. Gutmann and Aapo Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13(11):307–361, 2012.
- Geoffrey E Hinton and Sam T Roweis. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems*, pages 857–864, 2003.
- Daniel Jiwoong Im, Nakul Verma, and Kristin Branson. Stochastic neighbor embedding under F-divergences. *arXiv preprint arXiv:1811.01247*, 2018.
- Mathieu Jacomy, Tommaso Venturini, Sebastien Heymann, and Mathieu Bastian. ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software. *PloS One*, 9(6), 2014.
- Sabina Kanton, Michael James Boyle, Zhisong He, Malgorzata Santel, Anne Weigert, Fátima Sanchís-Calleja, Patricia Guijarro, Leila Sidow, Jonas Simon Fleck, Dingding Han, et al. Organoid single-cell genomic atlas uncovers human-specific features of brain development. *Nature*, 574(7778):418–422, 2019.
- Konrad J Karczewski, Laurent C Francioli, Grace Tiao, Beryl B Cummings, Jessica Alföldi, Qingbo Wang, Ryan L Collins, Kristen M Laricchia, Andrea Ganna, Daniel P Birnbaum, et al. The mutational constraint spectrum quantified from variation in 141,456 humans. *Nature*, 581:434–443, 2020.
- Dmitry Kobak and Philipp Berens. The art of using t-SNE for single-cell transcriptomics. *Nature Communications*, 10:5416, 2019.
- Dmitry Kobak and George Linderman. Initialization is critical for preserving global data structure in both t-SNE and UMAP. *Nature Biotechnology*, 39:156–157, 2021.
- Dmitry Kobak, George Linderman, Stefan Steinerberger, Yuval Kluger, and Philipp Berens. Heavy-tailed kernels reveal a finer cluster structure in t-SNE visualisations. In *Machine Learning and Knowledge Discovery in Databases*, pages 124–139. Springer International Publishing, 2020.
- George C Linderman and Stefan Steinerberger. Clustering with t-SNE, provably. *SIAM Journal on Mathematics of Data Science*, 1(2):313–332, 2019.

- 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(3):243, 2019.
- Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv:1802.03426*, 2018.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.
- Bartosz Minch, Mateusz Nowak, Rafał Wcisło, and Witold Dzwinel. GPU-embedding of kNN-graph representing large and high-dimensional data. In *International Conference on Computational Science*, pages 322–336. Springer, 2020.
- Kevin R Moon, David van Dijk, Zheng Wang, Scott Gigante, Daniel B Burkhardt, William S Chen, Kristina Yim, Antonia van den Elzen, Matthew J Hirn, Ronald R Coifman, et al. Visualizing structure and transitions in high-dimensional biological data. *Nature Biotechnology*, 37(12):1482–1492, 2019.
- Karthik S Narayan, Ali Punjani, and Pieter Abbeel. Alpha-beta divergences discover micro and macro structures in data. In *International Conference on Machine Learning*, pages 796–804, 2015.
- Andreas Noack. Energy models for graph clustering. *Journal of Graph Algorithms and Applications*, 11(2):453–480, 2007.
- Andreas Noack. Modularity clustering is force-directed layout. *Physical Review E*, 79(2):026102, 2009.
- L. N. Olson and J. B. Schroder. PyAMG: Algebraic multigrid solvers in Python v4.0, 2018. URL <https://github.com/pyamg/pyamg>.
- Nikolay Oskolkov. How exactly UMAP works. <https://towardsdatascience.com/how-exactly-umap-works-13e3040e1668>, 2019.
- Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.
- Pavlin Gregor Poličar, Martin Strazar, and Blaz Zupan. openTSNE: a modular Python library for t-SNE dimensionality reduction and embedding. *bioRxiv:731877*, 2019.
- Vinay Uday Prabhu. Kannada-MNIST: A new handwritten digits dataset for the Kannada language. *arXiv preprint arXiv:1908.01242*, 2019.
- Milos Radovanovic, Alexandros Nanopoulos, and Mirjana Ivanovic. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11(sept):2487–2531, 2010.
- Tim Sainburg, Leland McInnes, and Timothy Q Gentner. Parametric UMAP embeddings for representation and semisupervised learning. *Neural Computation*, 33(11):2881–2907, 2021.

- Lawrence K Saul. A tractable latent variable model for nonlinear dimensionality reduction. *Proceedings of the National Academy of Sciences*, 117(27):15403–15408, 2020.
- Nikhil Sharma, Kali Flaherty, Karina Lezgiyeva, Daniel E Wagner, Allon M Klein, and David D Ginty. The emergence of transcriptional identity in somatosensory neurons. *Nature*, pages 1–7, 2020.
- Stefan Siebert, Jeffrey A. Farrell, Jack F. Cazet, Yashodara Abeykoon, Abby S. Primack, Christine E. Schnitzler, and Celina E. Juliano. Stem cell differentiation trajectories in hydra resolved at single-cell resolution. *Science*, 365(6451), 2019.
- Gabor Szekely, Maria Rizzo, and Nail Bakirov. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35:2769–2794, 2007.
- Jian Tang, Jingzhou Liu, Ming Zhang, and Qiaozhu Mei. Visualizing large-scale and high-dimensional data. In *International Conference on World Wide Web*, pages 287–297, 2016.
- 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):72–78, 2018.
- Betsabeh Khoramian Tusi, Samuel L Wolock, Caleb Weinreb, Yung Hwang, Daniel Hidalgo, Rapolas Zilionis, Ari Waisman, Jun R Huh, Allon M Klein, and Merav Socolovsky. Population snapshots predict early haematopoietic and erythroid hierarchies. *Nature*, 555(7694):54–60, 2018.
- Laurens van der Maaten. Accelerating t-SNE using tree-based algorithms. *The 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(Nov):2579–2605, 2008.
- Jarkko Venna, Jaakko Peltonen, Kristian Nybo, Helena Aidos, and Samuel Kaski. Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *Journal of Machine Learning Research*, 11:451–490, March 2010.
- Ulrike von Luxburg, Mikhail Belkin, and Olivier Bousquet. Consistency of spectral clustering. *The Annals of Statistics*, pages 555–586, 2008.
- Daniel E Wagner, Caleb Weinreb, Zach M Collins, James A Briggs, Sean G Megason, and Allon M Klein. Single-cell mapping of gene expression landscapes and lineage in the zebrafish embryo. *Science*, 360(6392):981–987, 2018a.
- Daniel E. Wagner, Caleb Weinreb, Zach M. Collins, James A. Briggs, Sean G. Megason, and Allon M. Klein. Single-cell mapping of gene expression landscapes and lineage in the zebrafish embryo. *Science*, 360(6392):981–987, 2018b. ISSN 0036-8075.
- Caleb Weinreb, Samuel Wolock, and Allon M Klein. SPRING: a kinetic interface for visualizing high dimensional single-cell expression data. *Bioinformatics*, 34(7):1246–1248, 2018.

- Caleb Weinreb, Alejo Rodriguez-Fraticelli, Fernando D Camargo, and Allon M Klein. Lineage tracing on transcriptional landscapes links state to fate during differentiation. *Science*, 367(6479), 2020.
- Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.
- Zhirong Yang, Irwin King, Zenglin Xu, and Erkki Oja. Heavy-tailed symmetric stochastic neighbor embedding. In *Advances in Neural Information Processing Systems*, pages 2169–2177, 2009.
- Zhirong Yang, Jaakko Peltonen, and Samuel Kaski. Scalable optimization of neighbor embedding for visualization. In *International Conference on Machine Learning*, pages 127–135, 2013.
- Zhirong Yang, Jaakko Peltonen, and Samuel Kaski. Optimization equivalence of divergences improves neighbor embedding. In *International Conference on Machine Learning*, pages 460–468, 2014.