Diffusion t-SNE for multiscale data visualization

Lan Huong Nguyen[*]
ICME
Stanford University
Stanford, CA 94305
lanhuong@stanford.edu

Susan Holmes
Department of Statistics
Stanford University
Stanford, CA 94305
susan@stat.stanford.edu

Abstract

T-Stochastic Neighbor Embedding (t-SNE) is an effective technique for visualizing high-dimensional data. However, empirical evidence shows that its output is sometimes unpredictable and often sensitive to the choice of perplexity parameter. Varying the parameter value does not consistently lead to a shift of focus from local to global data structure recovery as widely believed. We propose a new embedding method, *Diffusion t-SNE*, which introduces a time-step parameter that allows the recovery of the input data geometry at different scales. We also provide mathematical explanations for why the entropy equalization procedure used in t-SNE results in a loss of information about local variances, leading to data distortions that produce misleading representations with uninformative relative sizes and unidentifiable input data sampling densities and variances. Building upon this analysis, we present a scaling scheme of the pairwise proximities that achieves accurate representations of regional data variances. We apply our proposed methods to both synthetic and real-word datasets and evaluate quality of the output embeddings in terms of both the local and global structure recovery.

1 Introduction

Neighbor embedding methods generate reduced data representations, where points close in the original input are also close in the output but where points originally distant are not necessarily distant in the output space. This property is desirable when one is interested only in recovering the inherent grouping of the data points and obtaining a good cluster separation [1, 13]. t-SNE by Van der Maaten and Hinton [23] is a very popular neighbor embedding method for visualizing high-dimensional data. The algorithm has been successfully used on to many large real datasets and has exposed the underlying data groupings; it performs particularly well when applied to datasets composed of well-separated clusters. A representative example of t-SNE's advantage over other linear and non-linear dimensionality reduction methods is the Mixed National Institute of Standards and Technology (MNIST) handwritten digits dataset [9]. The t-SNE embedding produces a 2D map of 784 dimensional input images with data points clearly separated into clumps corresponding to different digits. Following the success of t-SNE in generating representations of benchmark machine learning image datasets [23, 7], the method is now often used on biological datasets. In particular, t-SNE has been widely adopted by the immunology community, where the method is frequently applied to genomics or mass cytometry data to produce visualizations of groups of cell populations and sub-populations [15, 22, 6].

However, one of the weaknesses of t-SNE is its inability to correctly represent local differences in data density or variance. It has been widely recognized that cluster sizes and large-scale distances are not interpretable in t-SNE output embeddings [24]. While this property might be harmless

[*] Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

Preprint. Under review.

when the input data is inherently categorical (e.g. images of digits or objects), the inability to make comparisons between different regions of the data is a major limitation when visualizing datasets governed by unknown latent continuous factors. Additionally, by design t-SNE stretches and shrinks different fragments of the data at different rates. Data distortions are most striking when the sampling density is highly variable across different regions. Since many datasets, especially biological ones, are not characterized by a latent discrete variable but rather by an unknown continuous process, it is important to use data visualization tools that can accurately capture both the local neighborhoods and the long-range interactions.

In this article, we first characterize the part of the t-SNE algorithm that induces its undesirable properties. Then, we (1) propose a scaling procedure to recover the relative differences in regional variances, and (2) develop a new algorithm, *Diffusion t-SNE*, for generating multiscale data visualizations. More specifically, we show that the dissimilarity-to-similarity conversion procedure adopted by t-SNE and other NE methods is responsible for both the data contraction and expansion at varying degrees across different neighborhoods as well as for the loss of information of the long-range interactions. Our proposed method merges the strengths of two methods: t-SNE and Diffusion Maps [3] to achieve efficient data structure recovery at different scales.

2 Preliminaries

The input to the t-SNE algorithm is either original observations set in high dimensions $\mathbf{x}_i \in \mathbb{R}^p,\ i = 1, \ldots, N$, or data points first projected to a more informative "mid-size" (e.g. 50-100 dimensional) feature space using traditional dimensionality reduction methods such as the principal component analysis (PCA) or neural-network-based feature extraction techniques such as auto-encoders. The algorithm computes pairwise dissimilarities (often Euclidean distances) which is problematic when measure on data in very high dimensions due to the distance concentration [5, 11] – a phenomenon where the distances become very narrowly distributed, making them hardly distinguishable from each other. Applying the initial dimension reduction step can reduce noise levels and alleviate the issue of low-resolution distances, just described. t-SNE converts the pairwise distances to inter-point proximities – *conditional probabilities* of one observation being in a neighborhood of another, $p_{j|i}$, defined by using Gaussian kernels with varying bandwidth parameters, σ_i^2's:

$$p_{j|i} = \frac{\exp(-d_{ij}^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-d_{ik}^2/2\sigma_i^2)},\ \forall j \neq i \ \text{ and } \ p_{i|i} = 0. \tag{2.1}$$

The σ_i^2's are specific to each observation. The values for these parameters are selected to keep the entropy of $p_{\cdot|i}$ constant, determined by a user-specified perplexity parameter, $\eta = \exp(\gamma)$. For each $i = 1, \ldots, N$ we have the following constraint on the corresponding entropy term, H_i:

$$\gamma = H_i = -\sum_{j\neq i} p_{j|i}\log(p_{j|i}) = \sum_{j\neq i}\frac{d_{ij}^2}{2\sigma_i^2}p_{j|i} + \log\sum_{j\neq i}\exp(-d_{ij}^2/2\sigma_i^2). \tag{2.2}$$

There is no closed-form solution to the above expression, and the σ_i's are usually computed through a binary search. Conditional probabilities are then symmetrized to form *joint probabilities*, $(P)_{ij} = \frac{1}{2N}\left(p_{j|i} + p_{i|j}\right)$, representing the final measure of similarity between two data points. The main advantage of using symmetric rather than asymmetric affinities is the computationally simpler form of the gradient (defined later in this section). The novelty of t-SNE over its precursor, SNE, is the use of a different affinity function for the output domain instead of the same Gaussian kernel originally used. The similarities between observations in the output configuration, $\{\mathbf{y}_i \in \mathbb{R}^d,\ d \ll p\}$, are computed using a Student t-distribution (usually with one degree of freedom):

$$q_{ij} = \frac{(1 + \|\mathbf{y}_i + \mathbf{y}_j\|^2)^{-1}}{\sum\limits_{k=1}^{N}\sum\limits_{l\neq k}(1 + \|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}},\ \forall j \neq i \ \text{ and } \ q_{ii} = 0. \tag{2.3}$$

The mismatched affinity functions help t-SNE alleviate the *crowding problem* and prevents data points from collapsing onto each other in the output embedding [23]. Finally, the coordinates of the embedding ($\{\mathbf{y}_i\}$'s) are computed by minimizing the Kullback-Leibler (KL) divergence between the input and output proximity measures, $C = KL(P\|Q) = \sum_{i,j} p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right)$, using gradient descent.

3 Identifying local differences in data variance and density

In this section, we show why the dissimilarity-to-similarity conversion procedure used by t-SNE and other NE techniques suffers from an uncontrolled stretching and shrinking when the input data exhibits non-uniform local variance or sampling density across regions. The t-SNE entropy equalization procedure (2.2) for selecting different kernel bandwidths for each data point was originally introduced to accommodate unequal data density across regions. Previously, when constant bandwidths were used, selecting a σ optimal for the entire dataset was challenging. In most cases, small values were chosen to provide a good neighborhood resolution for dense regions where the bulk of the data resides in. A small constant σ would lead to loss of information on input data in sparsely populated regions because the Gaussian kernel values for data points in these neighborhoods would all be similarly infinitesimal. For every i in a sparse neighborhood, the corresponding proximities (normalized kernels) $p_{j|i}$'s would be roughly constant for each j. Consequently, the ith data point would appear uniformly distant from all other samples, so its output embedding coordinates could not be accurately determined.

Using varying σ_i's alleviates the issue described above, as the resolution of the pairwise proximities can be kept at similar levels across dense and sparse regions by adjusting the scale of the Gaussian kernel individually for each neighborhood. Nevertheless, while entropy-equalizing bandwidths can increase the amount of information encoded in the inter-point proximities in sparse neighborhoods, they can also lead to undesirable distortions in the output embedding.

3.1 Effective neighborhoods

The user-specified perplexity parameter is often referred to as the effective number of neighbors. Since the term has not been precisely determined, we propose the following definition:

Definition 3.1. For computed bandwidth parameters σ_i's, and a chosen error level, $0 < \epsilon < 1$, let the set of *effective neighbors* of the ith data point be $\mathcal{N}_i = \{j \colon \exp(-d_{ij}^2/2\sigma_i^2) > \epsilon/N \text{ and } j \neq i\}$, where the σ_i's are selected to meet the constraints specified in (2.2). Further let $\bar{d}_i^2 = \sum_{j \neq \mathcal{N}_i} d_{ij}^2 p_{j|i}$ be the *effective distance* of data point i to its neighbors.

It can be shown that for $\Delta_i = \max_{j \in \mathcal{N}_i} d_{ij}^2 - \min_{j \in \mathcal{N}_i} d_{ij}^2$, the spread of distances from i to its effective neighbors, the following holds (see supplementary material for derivations):

$$|\mathcal{N}_i| \exp(-\Delta_i^2/2\sigma_i^2) \;\leq \eta \;\leq\; |\mathcal{N}_i| \exp(\Delta_i^2/2\sigma_i^2) + \mathcal{O}(\epsilon). \tag{3.1}$$

It follows that, when distances in the neighborhood do not vary much, we have $|\mathcal{N}_i| \approx \eta$. Similarly, using the entropy formula in 2.2, we can convert the formula for pairwise proximities for all $i = 1, \ldots, N$ as follows:

$$p_{j|i} = \frac{\exp(-d_{ij}^2/2\sigma_i^2)}{\sum_{j \neq i} \exp(-d_{ij}^2/2\sigma_i^2)} = \frac{\exp(-d_{ij}^2/2\sigma_i^2)}{\exp(\gamma - \bar{d}_i^2/2\sigma_i^2)} = \frac{1}{\eta} \exp\left(-\frac{d_{ij}^2 - \bar{d}_i^2}{2\sigma_i^2}\right)$$

The above shows that the t-SNE dissimilarity-to-similarity conversion involves "centering and scaling" of the distances using different factors for different neighborhoods of the input data. In other words, by design the pairwise dissimilarities are calibrated differently across regions. Since for every i, the corresponding entropy of $p_{\cdot|i}$ is constrained to be equal to γ, the "standardized" distances $Z_j^{(i)} = (d_{ij}^2 - \bar{d}_i^2)/2\sigma_i^2$ need to follow a similar distribution. Thus, the t-SNE maps often consists of data patches that seem uniformly dense, regardless of variation in the input data density. The usage of varying shift, $\bar{d}_i$, and scale, σ_i, factors makes cross-comparisons between $p_{\cdot|i}$'s and

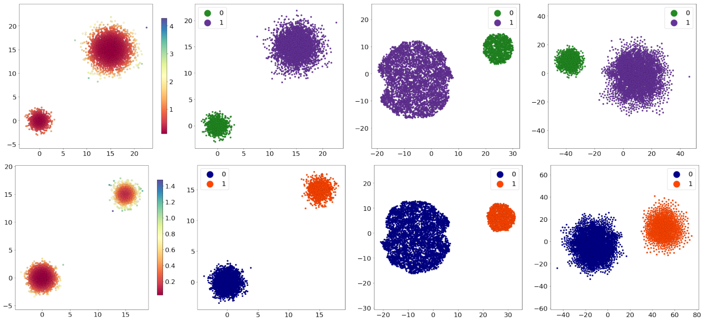


Figure 1: Two clusters with different variances and the same density (top) appear the same as two clusters of the same variance but different density (bottom) on a standard t-SNE map. Scaling conditional probabilities can recover information on local differences in variance. Original data shown in two left-most columns, where data points are colored by the estimated bandwidths (first column) and true cluster membership (second column); then standard t-SNE embedding (third column) and our scaled t-SNE embedding (fourth column).

$p_{\cdot|k}$'s meaningless; more specifically, while differences in values of $p_{j|i}$ and $p_{j'|i}$ are informative, comparisons between $p_{j|i}$ and $p_{j|k}$ or $p_{j|i}$ and $p_{l|k}$ cannot be interpreted. It follows that density and variance are unidentifiable in the t-SNE embeddings.

Consider the two examples in Fig. 1. In the top row, we show two Gaussian 2D clusters with the same sampling density but different volumes. More specifically, we generated data points: $\mathbf{x}_i^{\text{(purp)}} \sim \mathcal{N}(\mu_{\text{purp}}, \lambda I)$, and $\mathbf{x}_k^{\text{(green)}} \sim \mathcal{N}(\mu_{\text{green}}, 2\lambda I)$, where $N_{\text{purp}} = 4N_{\text{green}}$. In the bottom row, the two clusters have the same volume but different sampling density: $\mathbf{x}_i^{\text{(blue)}} \sim \mathcal{N}(\mu_{\text{blue}}, \gamma I)$, and $\mathbf{x}_k^{\text{(orng)}} \sim \mathcal{N}(\mu_{\text{orng}}, \gamma I)$, where $N_{\text{blue}} = 4N_{\text{orng}}$. Note that the resulting t-SNE maps for the two cases (third column of Fig. 1) appear almost the same, suggesting that the size of the cluster in the output embeddings depends only on the number of member points.

Overall, we observe that cluster sizes are uninformative in t-SNE maps. While variable bandwidth improves the resolution of similarities between data points in sparse regions, it leads to distortions in the output embeddings. The shrinkage and expansion of different regions of the data is a direct result of the enforced dissimilarity-to-similarity conversion procedure. Since the inter-point affinity computations involve shifting and scaling of the dissimilarity values at unequal rates, the information on local differences in variance and density is lost. Intuitively, t-SNE treats the k-nearest-neighbors in a very sparse region the same as the k-nearest neighbors in a dense region, even though the distances between neighbors in these two regions can vary significantly. In order to account for the differences in variance, the values of conditional probabilities must provide an adequate comparison between pairwise affinities for points residing in different neighborhoods.

3.2 Scaled t-SNE preserves local data variance

To alleviate this loss of information, we suggest multiplying the rows of the similarity matrix $(P)_{ij} = p_{j|i}$ by factors inversely proportional to the selected bandwidth value. In other words, we use the following formula for scaled conditional probabilities:

$$\tilde{p}_{j|i} = \alpha_i \frac{\exp(-\beta_i d_{ij})}{\sum_{k \neq i} \exp(-\beta_i d_{ik})}, \quad \text{where} \quad \alpha_i = N \frac{\beta_i}{\sum_k \beta_k}, \quad \text{and} \quad \beta_i = \frac{1}{2\sigma_i^2}. \tag{3.2}$$

Scaling $p_{\cdot|i}$ terms by an α_i factor recovers relative differences in the sizes of the neighborhoods considered for each data point i. The resulting similarities between i and its neighbors, $\mathcal{N}_i$, are larger if i is in a dense region (with a corresponding smaller bandwidth σ_i), and smaller if it is in a sparse region (with larger σ_i). Intuitively, larger similarities should be assigned to data points with smaller mean distance to the k-nearest neighbors. Using α_i factors and β_i bandwidths is simply a smooth alternative to using k-nearest neighbors to evaluate inter-point similarities.

Our choice of the above scaling factor is guided by the work of Berry and Harlim [2], who showed that the graph Laplacian ($L = I - P$) constructed with varying bandwidths kernels scaled by the same

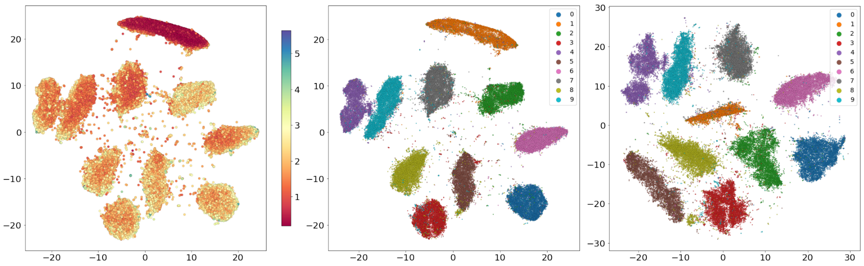


Figure 2: Full MNIST dataset (70,000 dataset) embedding with standard and scaled t-SNE. The middle plot shows the entropy-equalizing t-SNE bandwidths. Scaled t-SNE calibrates the pairwise proximities using these bandwidths to obtain an output embedding with accurate clusters sizes. Scalable approximate t-SNE implementation, FIt-SNE [14], (with perplexity 100) was applied to the top 50 principal components of the input image data (784 pixels). Late exaggeration was used to better separate the clusters as advised in [13].

bandwidths converge to the Laplace-Beltrami (LB) operator that links the local and global geometry [12]. The described scaling procedure allows the differences in local densities and variances in the input data to be reflected in computed inter-point similarities. As a result, t-SNE embeddings can be generated with the relative cluster sizes consistent with the original cluster volumes.

The last column of Fig. 1 shows that our scaling procedure results in output embeddings in which cluster sizes are consistent with the underlying truth. Unlike with the standard t-SNE, the local differences in variances and data sampling densities are distinguishable in the scaled t-SNE. We also demonstrate the effectiveness of this procedure on the MNIST dataset. This dataset is well balanced, containing roughly the same number of samples for each digit category, and t-SNE embedding nicely cluster the digits in the set. However, data variance across digit classes is not uniform, suggesting the local data sampling density varies across the input data space. Consequently, the cluster sizes are uninformative on the standard t-SNE map (middle plot in Fig. 2). For example, while most people write the number one in a similar way, i.e. the images of “1” should be the least variable of all digits, the digit “1” (orange) cluster appears the largest on the standard t-SNE output. On the other hand, when α_i factors are applied to corresponding conditional probabilities we see that the size of cluster ‘1’ becomes the smallest (right-most plot in Fig. 2), as expected.

4 Recovering multiscale structures

While t-SNE maps preserve local neighborhoods, grouping similar observations close together, they usually provide a poor representation of the large-scale structures in the input data. Since the t-SNE’s dissimilarity-to-similarity conversion procedure involves applying a Gaussian kernel to the pairwise distances (2.1) measured on the input data, the algorithm implicitly assumes that these distances cannot accurately capture the long-range interactions between observations. The Gaussian kernel decays exponentially and converts all large distances between non-neighboring points into negligible inter-point proximities. Consequently, the relationship between distant data points is not encoded in the computed similarities. Since the KL divergence minimization problem for finding the optimal output embedding configurations has access only to the pairwise proximities, there is not enough of information available to recover the global landscape of the input data.

Learning global geometry is most challenging when the data lies approximately on a manifold that is non-Euclidean in large scales. In these cases, the geometry of the input data cannot be recovered by simply applying linear projections or rigid motions (such as PCA) because these data transformations are unable to recover any curvature present in the data [19]. When non-linear structures are present, one must “unfold” the underlying manifold and learn its global geometry either by estimating the “true” intrinsic distances defined on the manifold or by integrating or propagating the local information. This means that expanding the size of considered neighborhoods by increasing the perplexity value, would not help t-SNE obtain an accurate depiction of the latent manifold.

4.1 Diffusion t-SNE Algorithm

First, we note that the t-SNE dissimilarity-to-similarity transformation (2.1) for the input data is conceptually very similar to the one used in diffusion maps (DM) [3]. DM defines a random walk on a dataset using a (heat/Gaussian) kernel specifying the local geometry. The corresponding *transition probability matrix* characterizes the directions of fast and slow propagation and approximates the relationships between pairs of points in terms of their connectivity. Running the Markov chain forward is equivalent to propagating and integrating the local information to obtain a global profile of the entire dataset [3]. DM finds a configuration in a low-dimensional space that preserves *diffusion distances* defined as, $D_t(x, x')^2 \triangleq \|p_t(x, \cdot) - p_t(x', \cdot)\|^2_{L^2, d\mu/\pi} = \int_{\mathcal{M}} (p_t(x, u) - p_t(x', u))^2 \frac{d\mu(u)}{\pi(u)}$, where $p_t(x, y)$ is a transition probability from point x to point y in t time steps ($t \in \mathbb{N}$), μ is the distribution of the data points on $\mathcal{M}$, and π is the stationary distribution on the associated Markov chain. The diffusion distance, $D_t(x, x')$ integrates the local geometry over all paths of length t traversed by a Markov chain on the data manifold. For discrete data points, transition probabilities are constructed by row-normalizing Gaussian kernels, $(P_{\text{trans}})_{ij} = p_{t=1}(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/h)$. Note that this procedure is similar to the dissimilarity-to-similarity transformation implemented in t-SNE to obtain $p_{j|i}$'s (with the difference that DM uses constant bandwidths).

The main difference between the two methods is that t-SNE finds embedding coordinates by solving a non-convex minimization problem, while DM computes scaled top K eigenvectors of P^t_{trans}. Euclidean distances between pairs of points in the DM embedding approximate diffusion distances. Note that the number of output dimensions necessary for a good approximation of the diffusion distance depends closely on the rate of decay of the eigenvalues. Thus, while DM usually performs well as a general-purpose technique for dimensionality reduction it may not be well suited for data visualization, where only two or three dimensional embedding maps are required.

In Diffusion t-SNE, we propose to use P^t_{trans} as input to the KL minimization problem to find an optimal 2D or 3D embedding. The t-SNE KL minimization problem is better suited for visualization purposes, because its objective is to find an optimal representation in the user designated plotting space (most often $\mathbb{R}^2$ or $\mathbb{R}^3$) not restricted by an orthogonally constraint imposed on the axes like in DM's eigendecomposition. Moreover the embedding method's adoption of a heavy-tail Student t-distribution, rather than the Gaussian distribution, for computing proximities in the output space helps countering the crowding problem, which is critical when plotting large datasets. All steps of the procedure are listed in Alg.7 provided below. Just like in the standard t-SNE, step (1) of the algorithm can be performed exactly, or by using approximation methods such as vantage-point tree or ANNOY[2]. Step (7) is the same as in standard t-SNE. A $\mathcal{O}(N)$-runtime interpolation-based approximation for step (7) was developed by Linderman et al. [14]. We incorporate FIt-SNE in our Diffusion t-SNE implementation to perform step (7) computations. Additional details on the algorithm, including the row-sum threshold pruning procedure of [20] in step (4) are provided in the supplementary material.

Diffusion t-SNE allows the user to generate different views of the data by computing multiple embeddings with different choices of t. However, the value of t should not be chosen too large e.g. $t > 100$. Conceptually, this is because as $t \to \infty$ the random walk on the data will approach stationarity, and $P^t_{\text{trans.}}$ can become rank one. As expected, taking too large values of t might result in merged clusters or mixing of data points in a way that information on the local inter-point proximities is lost and all signal is diffused.

5 Experiments

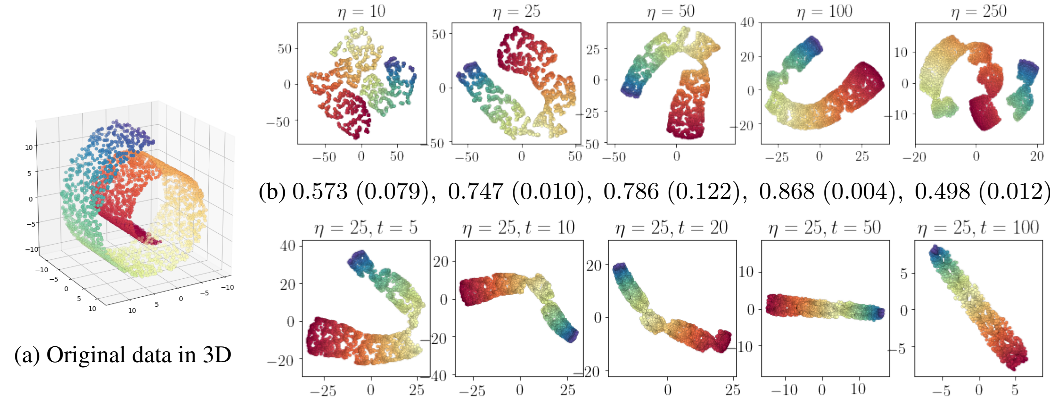
The examples discussed in this section are intended to illustrate and compare the behavior of the standard t-SNE and our proposed diffusion t-SNE algorithm. Supplementary material includes comparison to additional methods. Our results for the standard t-SNE were generated using FIt-SNE implementation. Diffusion t-SNE program and the code to generate all results presented here are available in the supplementary material[3].

[2] "Approximate Nearest Neighbors Oh Yeah" method by Bernhardsson: https://github.com/spotify/annoy

[3] Code will be available online later

Algorithm 1: Diffusion t-SNE

```
input  : Input data $\mathbf{x}_i \in \mathbb{R}^p, i = 1, \dots, N$, perplexity $\eta$ and time step $t$.
output : Output embedding coordinates $\mathbf{y}_i \in \mathbb{R}^d$.
Compute $d(\mathbf{x}_i, \mathbf{x}_j)$ for neighboring pairs $(i, j)$);
For all $i$, find $\sigma_i^2$ and $p_{\cdot|i}$ with $H_i = \log(\eta)$; $(P_{\text{trans}})_{ij} \leftarrow p_{j|i}$;
Compute $P^t_{\text{trans}}$ ;
(Optional) Prune rows of $P^t_{\text{trans}}$ using Alg. S2;
(Optional) Scale $P^t_{\text{trans}}$ using $\alpha_i$'s, $P^t_{\text{trans}} \leftarrow P^t_{\text{trans}}\Lambda$, where $\Lambda = \text{diag}(\alpha_i)$;
Symmetrize transition probabilities, $P^t_{\text{sym}} \leftarrow (P^t_{\text{trans}} + (P^t_{\text{trans}})^T)/2N$;
Find $\{\mathbf{y}_i\}$'s by minimizing $KL(P^t_{\text{sym}}|Q)$ using gradient descent;
```



(a) Original data in 3D

(b) $0.573\ (0.079),\ 0.747\ (0.010),\ 0.786\ (0.122),\ 0.868\ (0.004),\ 0.498\ (0.012)$

(c) $0.901\ (0.029),\ 0.989\ (0.002),\ 0.989\ (0.001),\ 0.989\ (0.001),\ 0.987\ (0.0001)$

Figure 3: Swiss roll (a) embedding. Comparing effect of varying perplexity value in standard t-SNE (b) and varying time step parameter in Diffusion t-SNE (c). Mean, $\bar{\rho}$, and standard deviation, s_ρ, of the Spearman rank correlation between distances measured on the true latent coordinates and embedding configuration are reported below the plots.

5.1 Swiss roll

A Swiss roll dataset was generated by sampling 3000 points on a flat rectangle uniformly at random and mapping them into a 3D configuration (Fig. 3(a)). Manifold learning techniques are often tested on their ability to unfold a Swiss roll. While this dataset is a synthetic example, it helps develop intuition about the behavior of t-SNE and highlights the underlying issues related to embedding of non-Euclidean (curved) datasets. Fig. 3 (b) shows the standard t-SNE output embeddings of a Swiss roll using different perplexity settings. For small values of η, t-SNE outputs many disconnected, unordered clusters of small groups of neighbors. For large perplexity values, t-SNE begins to recognize larger pieces of the manifold, but still is unable to unfold the Swiss roll and returns disconnected parts of the latent sheet. Essentially, for all selected values of perplexity, t-SNE fails to unfold the flat surface curved in 3D.

This simple example challenges the popular belief that increasing perplexity values improves the global structure recovery [8]. In practice, raising perplexity simply increases the size of the average bandwidth used by t-SNE to compute the conditional probabilities. Using larger bandwidths is equivalent to expanding the areas considered the local neighborhoods. Implicitly, this means that we place more trust in larger distances measured on the input data. Unfortunately, this assumption is usually unreliable for high-dimensional data: as mentioned in Section 2, dissimilarities tend to concentrate, with moderate and large distances having "low-resolution." As a result, increasing perplexity should not be expected to improve the recovery of the large-scale structures of non-linear manifolds with high curvature.

Diffusion t-SNE utilizes powers of the transition matrix and is able to reveal the structure of the Swiss roll scales. Below the embedding plots in Fig. 3, we also report the mean and standard deviation (over 5 embedding runs) of the average (over data points) Spearman rank correlation between the true distances observed in the natural parametrization space and the ones measured in the output embedding, $\rho_r = \frac{1}{N}\sum_{i=1}^{N} \mathrm{cor}_{\text{Spear.}}(d^X_{i,\cdot}, d^Y_{i,\cdot})$. Keeping a small perplexity value (resulting in small bandwidths) and increasing the value of the time step parameter allows the methods to successfully unfold the curved manifold. The unscaled diffusion t-SNE maps show an adequate representation of the large-scale structure; however, the method still shrinks the blue end of the Swiss roll. This behavior is related to the phenomenon discussed in Section 3; since the outer side of a Swiss roll appears to be sparser in $\mathbb{R}^3$, the estimated t-SNE bandwidths are larger resulting, in contraction of the region in the output embedding. In the supplementary Fig. S4, we show that using scaled diffusion t-SNE recovers the latent rectangular sheet, where both ends are of equal size. Using a ranked-based metric, $Q_{NX}(K)$, developed by Lee and Verleysen [10], we also evaluate the quality of the embeddings in terms of their ability to preserve local neighborhoods, Coranking curves are reported in supplementary Fig. S1, where we see that when limited to very small neighborhoods ($K < 10$), the standard t-SNE performs better than the diffusion t-SNE, but the quality scores converge and become roughly similar ($Q_{NX}(K) \in [0.75, 0.85]$) for all SNE methods and $K < 40$.

5.2 Embryoid Body differentiation

We also test our method on a single cell expression dataset from a 27-day time course study of the embryoid body (EB) differentiation generated by Moon et al. [18]. The dataset [4] contains scRNA-seq samples from human embryonic stem cells (hESCs) differentiating as embryoid bodies (EB), collected in 3-day intervals. We use the same data pre-processing steps as the ones performed by the authors of the original paper. [5] We then compute the first 50 principal components on the processed single cell data and apply the standard and diffusion t-SNE to generate 2D embeddings.

Since the dataset is governed by a differentiation process, the latent structure should involve gradual, continuous changes rather than distinct clusters, as in cases of categorical image classes. It is thus important to recover accurate structures present in the input data at different scales, i.e. it is of value to both show the relationship between cells collected on the same days (small-scale) and to depict how cells are organized across differentiation stages (large-scale). In Fig. 4, diffusion t-SNE generates data visualizations consistently at varying-scale by controlling the time step parameter, t, whereas the standard t-SNE produces embeddings with disconnected patches even at a perplexity value of $\eta = 1000$. Fig. 4 (f) shows that even using a large perplexity value cannot connect stem cells (red points). Diffusion t-SNE consistently shows all cells at day $0 - 3$ closer together. Cells at later stages are then spread out across the embedding space. The embedding configuration becomes more connected as the value of t increases, implying a shift of focus from local to global. In the supplementary material, we include the embeddings generated with other methods including PHATE[6] [18] – the method developed by the owners of the data – and the increasing popular UMAP[7] [16, 17], to show that diffusion t-SNE produces better multi-scale views of the data. Additionally, in the supplementary material we show the effectiveness of diffusion t-SNE on another scRNA-seq dataset from a study by Farrell at al. [4], describing the developmental trajectories during zebrafish embryogenesis (not discussed here due to space limitations).

6 Discussion and conclusion

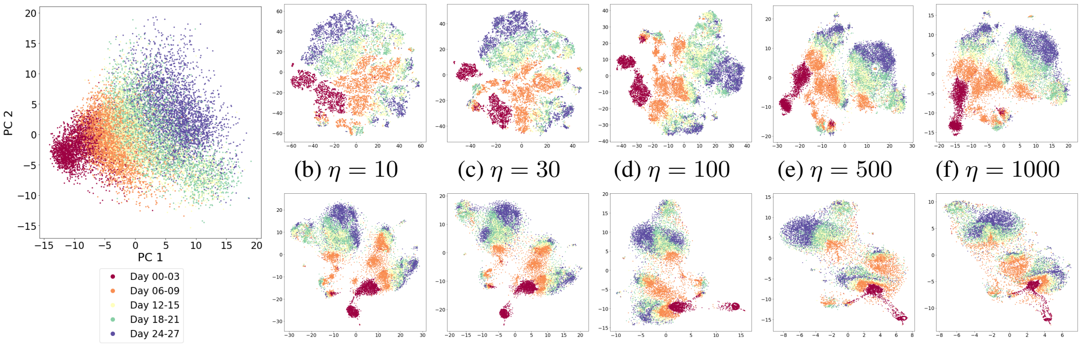
In this paper, we study the properties of t-SNE, focusing on the limitations due to the method's dissimilarity-to-similarity transformation procedure. We note that similar methods of converting distances to proximities are exploited by many embedding techniques, including LargeVis [21] or UMAP. In particular, the entropy-equalization for choosing Gaussian kernel bandwidth is widely implemented. As derived in this article, the t-SNE's conditional probabilities can be expressed as inverse exponential on shifted and scaled pairwise distances, where the shifting and scaling are applied using different factors for each data point. This results in distortions in the output embedding with the most noticeable effects occurring when the input data is unevenly sampled across the observed space. In general, variance and density are unidentifiable on the t-SNE maps, and the cluster sizes

[4] EB data downloaded from: `https://data.mendeley.com/datasets/v6n743h5ng/1`

[5] EB data pre-processing steps: `https://github.com/KrishnaswamyLab/PHATE`

[6] Potential of Heat-diffusion for Affinity-based Transition Embedding

[7] Uniform Manifold Approximation and Projection



(a) PCA

(b) $\eta = 10$ (c) $\eta = 30$ (d) $\eta = 100$ (e) $\eta = 500$ (f) $\eta = 1000$

(g) $\eta = 20; t = 5$ (h) $\eta = 20; t = 10$ (i) $\eta = 20; t = 20$ (j) $\eta = 20; t = 30$ (k) $\eta = 20; t = 50$

Figure 4: Embryoid Body dataset visualization with PCA (a), standard t-SNE (a-f) and diffusion t-SNE (g-k) for different choices parameters.

and the distances between cluster centers that cannot be interpreted from the output embedding. We introduce two effective modifications to t-SNE that help alleviate these undesired characteristics: (1) a scaling scheme that makes the inter-point proximities retain information on the regional fluctuations in variance and (2) a Diffusion t-SNE algorithm that, by varying a time step parameter, can recover the data geometry at different scales, including the global structure.

References

[1] Sanjeev Arora, Wei Hu, and Pravesh K. Kothari. An analysis of the t-sne algorithm for data visualization. *CoRR*, abs/1803.01768, 2018.

[2] Tyrus Berry and John Harlim. Variable bandwidth diffusion kernels. *Applied and Computational Harmonic Analysis*, 40(1):68 – 96, 2016.

[3] Ronald R. Coifman and Stéphane Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5 – 30, 2006. Special Issue: Diffusion Maps and Wavelets.

[4] Jeffrey A. Farrell, Yiqun Wang, Samantha J. Riesenfeld, Karthik Shekhar, Aviv Regev, and Alexander F. Schier. Single-cell reconstruction of developmental trajectories during zebrafish embryogenesis. *Science*, 360(6392), 2018.

[5] Damien Francois, Vincent Wertz, and Michel Verleysen. The concentration of fractional distances. *IEEE Trans. on Knowl. and Data Eng.*, 19(7):873–886, July 2007.

[6] Sinisa Hrvatin, Daniel R. Hochbaum, M. Aurel Nagy, Marcelo Cicconet, Keiramarie Robertson, Lucas Cheadle, Rapolas Zilionis, Alex Ratner, Rebeca Borges-Monroy, Allon M. Klein, Bernardo L. Sabatini, and Michael E. Greenberg. Single-cell analysis of experience-dependent transcriptomic states in the mouse visual cortex. *Nature Neuroscience*, 21(1):120–129, 2018.

[7] Andrei Karpathy. t-sne visualization of cnn codes.

[8] Dmitry Kobak and Philipp Berens. The art of using t-sne for single-cell transcriptomics. *bioRxiv*, 2018.

[9] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. MNIST handwritten digit database. 1998.

[10] John A. Lee and Michel Verleysen. Quality assessment of dimensionality reduction: Rank-based criteria. *Neurocomput.*, 72(7-9):1431–1443, March 2009.

[11] John A. Lee and Michel Verleysen. Shift-invariant similarities circumvent distance concentration in stochastic neighbor embedding and variants. *Procedia Computer Science*, 4:538 – 547, 2011.

[12] J. Liang, R. Lai, T. W. Wong, and H. Zhao. Geometric understanding of point clouds using laplace-beltrami operator. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 214–221, June 2012.

[13] George C. Linderman, Manas Rachh, Jeremy G. Hoskins, Stefan Steinerberger, and Yuval Kluger. Efficient algorithms for t-distributed stochastic neighborhood embedding. *CoRR*, abs/1712.09005, 2017.

[14] 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):243–245, 2019.

[15] Evan Z. Macosko, Anindita Basu, Rahul Satija, James Nemesh, Karthik Shekhar, Melissa Goldman, Itay Tirosh, Allison R. Bialas, Nolan Kamitaki, Emily M. Martersteck, John J. Trombetta, David A. Weitz, Joshua R. Sanes, Alex K. Shalek, Aviv Regev, and Steven A. McCarroll. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell*, 161(5):1202 – 1214, 2015.

[16] L. McInnes, J. Healy, and J. Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *ArXiv e-prints*, February 2018.

[17] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861, 2018.

[18] 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, Natalia B. Ivanova, Guy Wolf, and Smita Krishnaswamy. Visualizing structure and transitions for biological data exploration. *bioRxiv*, 2019.

[19] Dominique Perrault-Joncas and Marina Meila. Non-linear dimensionality reduction: Riemannian metric estimation and the problem of geometric discovery. *arXiv e-prints*, page arXiv:1305.7255, May 2013.

[20] Katja Reichel, Valentin Bahier, Cédric Midoux, Nicolas Parisey, Jean-Pierre Masson, and Solenn Stoeckel. Interpretation and approximation tools for big, dense markov chain transition matrices in population genetics. *Algorithms for Molecular Biology*, 10(1):31, Dec 2015.

[21] Jian Tang, Jingzhou Liu, Ming Zhang, and Qiaozhu Mei. Visualizing large-scale and high-dimensional data. In *Proceedings of the 25th International Conference on World Wide Web*, pages 287–297. International World Wide Web Conferences Steering Committee, 2016.

[22] Bosiljka Tasic, Zizhen Yao, Lucas T. Graybuck, Kimberly A. Smith, Thuc Nghi Nguyen, Darren Bertagnolli, Jeff Goldy, Emma Garren, Michael N. Economo, Sarada Viswanathan, Osnat Penn, Trygve Bakken, Vilas Menon, Jeremy Miller, Olivia Fong, Karla E. Hirokawa, Kanan Lathia, Christine Rimorin, Michael Tieu, Rachael Larsen, Tamara Casper, Eliza Barkan, Matthew Kroll, Sheana Parry, Nadiya V. Shapovalova, Daniel Hirschstein, Julie Pendergraft, Heather A. Sullivan, Tae Kyung Kim, Aaron Szafer, Nick Dee, Peter Groblewski, Ian Wickersham, Ali Cetin, Julie A. Harris, Boaz P. Levi, Susan M. Sunkin, Linda Madisen, Tanya L. Daigle, Loren Looger, Amy Bernard, John Phillips, Ed Lein, Michael Hawrylycz, Karel Svoboda, Allan R. Jones, Christof Koch, and Hongkui Zeng. Shared and distinct transcriptomic cell types across neocortical areas. *Nature*, 563(7729):72–78, 2018.

[23] Laurens J. P. van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[24] Martin Wattenberg, Fernanda Viégas, and Ian Johnson. How to use t-sne effectively. *Distill*, 2016.

Supplementary Material

Appendix A

Here we show the relationship between the perplexity parameter, η, and the effective number of neighbors. First, we expand the formula for entropy of $p_{\cdot|i}$:

$$\begin{aligned}
H_i &= \log\left(\sum_{j\neq i}\exp(-d_{ij}^2/2\sigma_i^2)\right) + \sum_{j\neq i}\frac{d_{ij}^2}{2\sigma_i^2}p_{j|i} \\
&= \log\left(\sum_{j\neq i}\exp(-d_{ij}^2/2\sigma_i^2)\right) + \log\left(\exp\left(\bar{d_i^2}/\sigma_i^2\right)\right) \\
&= \log\left(\sum_{j\neq i}\exp(-(d_{ij}^2-\bar{d_i^2})/2\sigma_i^2)\right) \\
&= \log\left(\sum_{j\in\mathcal{N}_i}\exp(-(d_{ij}^2-\bar{d_i^2})/2\sigma_i^2) + \sum_{j\notin\mathcal{N}_i}\exp(-(d_{ij}^2-\bar{d_i^2})/2\sigma_i^2)\right)
\end{aligned}$$

where $\bar{d_i^2} = \sum_{j\neq i} d_{ij}^2 p_{j|i}$.

Since $H_i = \log(\eta)\ \text{ for all} i = 1,\ldots,N$:

$$\eta = \underbrace{\sum_{j\in\mathcal{N}_i}\exp(-(d_{ij}^2-\bar{d_i^2})/2\sigma_i^2)}_{A} + \underbrace{\sum_{j\notin\mathcal{N}_i}\exp(-(d_{ij}^2-\bar{d_i^2})/2\sigma_i^2)}_{B}$$

Then, we have:

$$\begin{aligned}
|\mathcal{N}_i|\exp((\bar{d_i^2}-d_{i,\max}^2)/2\sigma_i^2)\ &\leq A\ \leq\ |\mathcal{N}_i|\exp((\bar{d_i^2}-d_{i,\min}^2)/2\sigma_i^2) \\
0\ &\leq B\ \leq\ \epsilon((N-|\mathcal{N}_i|)/N)\exp(\bar{d_i^2}/2\sigma_i^2)
\end{aligned}$$

where $d_{i,\min}^2 = \min_{j\in\mathcal{N}_i} d_{ij}^2$ and $d_{i,\max}^2 = \max_{j\in\mathcal{N}_i} d_{ij}^2$, the minimum and maximum distance of i to its effective neighbors. Note that $(\bar{d_i^2}-d_{i,\min}^2)/2\sigma_i^2 \ll 1$, since $\bar{d_i^2}$ are highly skewed towards small squared distances d_{ij}^2s. Therefore, the term $\exp((\bar{d_i^2}-d_{i,\min}^2)/2\sigma_i^2)\approx 1$. Let $\Delta_i = d_{i,\max}^2 - d_{i,\min}^2$ be the spread of distances in the neighborhood of i. Then we have:

$$|\mathcal{N}_i|\exp(-\Delta_i^2/2\sigma_i^2)\ \leq\eta\ \leq\ |\mathcal{N}_i|\exp(\Delta_i^2/2\sigma_i^2)+\mathcal{O}(\epsilon)$$

when distances to nearby neighbors are similar, $\Delta_i^2 \ll 1$ is small, we have $\eta\approx|\mathcal{N}_i|$.

Appendix B

Algorithm details

We observe that one might potentially run into scalability issues when computing $P^t_{\text{trans.}}$ in step (3). Despite being technically $\mathcal{O}(N^3)$ in runtime, today dense-dense matrix multiplication is highly parallelized and very fast in practice. While this is true for medium-size matrices ($N \leq 10,000$), working with larger matrices on might run into limitations both in time and space. Thus, when computing diffusion t-SNE for big datasets one needs scalable approaches for estimating the entries of $P^t_{\text{trans.}}$. We suggest two approximation methods:

(A) a random walk Monte Carlo approximation,

(B) a landmark approach like the one used by Moon et al. [18].

In the Monte Carlo method (A), N_{walks} random walks of length t are generated starting from each data point, and the resulting frequency of endpoints used for quantifying t-step transition probabilities. Usually N_{walks} is set to be $1/\epsilon$, where ϵ is the desire level of precision. Steps of random walks can be drawn using alias sampling which requires $\mathcal{O}(N)$ of runtime to set-up an alias table from the original $P_{\text{trans.}}$, ($t = 1$), and $\mathcal{O}(1)$ for each draw of a random walk. Overall, the approximation procedure has $\mathcal{O}(Nt/\epsilon)$ order runtime, but is massively parallelizable, as each random walk chain can be sampled independently. Thus, the random walk approximation can be very fast if implemented on a GPU.

In the landmark method (B), $P^t_{\text{trans.}}$ is computed as transition probability matrix for a random walk of length t constrained to pass through a landmark every other step, i.e.

$$P^t_{\text{trans.}} \approx P_{NM}(P_{MN}P_{NM})\dots(P_{MN}P_{NM})P_{MN} = P_{NM}P_{MM}^{\lfloor \frac{t}{2} \rfloor}P_{MN}$$

where $P_{MM} = P_{MN}P_{NM}$ denotes a transition matrix between landmark points or landmark partitions. Since $M \ll N$, P^t_{MM} is fast to compute. Method (B) requires $\mathcal{O}(NM^2 + \log(t)M^3)$ for computing $P^t_{\text{trans.}}$ using the landmark approximation formula above. To select the landmarks one might consider partitioning the dataset into M clusters using k-means on top K eigenvectors of $P^t_{\text{trans.}}$ computed using a randomized SVD algorithm. The (i, m)-th entry of P_{NM} is then the probability of a data point i transitioning to the m-th landmark partition. Equivalently, the (m, i)-th entry of P_{MN} denotes the probability of transitioning from the m-th landmark to a data point i. Both can be computed by aggregating the original transition matrix $P_{\text{trans.}}$ over the M-partitions.

Both methods require $\mathcal{O}(Nk)$ of storage space, where k is the average number of non-zeros in $P^t_{\text{trans.}}$. Note that for small t, the matrix $P^t_{\text{trans.}}$ might remain sparse, however it will fill up quickly, and pruning method like the one described in the next subsection might be required for scalability.

Sparse approximation of $P^t_{\text{trans.}}$

Below we provide the algorithm for sparsifying a stochastic matrix (with row-sums equal 1) adapted from the procedure by Reichel et al. [20]. The algorithm takes $\mathcal{O}(N^2 \log(N))$ runtime, and $\mathcal{O}(Nk \log(k))$ if the the matrix is initially already relatively sparse, with k equal the average number of non-zero terms in a row.

Algorithm S 2: Pruning transition matrices

```
input  : A stochastic matrix, $P$, and a row-sum threshold, $\omega$.
output : Sparse approximation of $P$.
for $i \leftarrow 1$ to $n$ do
    sort the row entries in a decreasing order (saving their corresponding column indices):
      $P_{ii_1} \geq P_{ii_2} \geq \cdots \geq P_{ii_n}$;
    find the minimal index with a cumulative sum of ordered row-entries at least $\omega$:
      $r \leftarrow arg\min_k\{\sum_{k=1}^{n} P_{ii_k} \geq \omega\}$;
    keep at least the two biggest values per row: $r \leftarrow \max(2, r)$;
    keep all values of equal rank: while $P_{ii_r} = P_{ii_{r+1}}$ do $r \leftarrow r + 1$;
    set $P_{ii_k} \leftarrow 0$ for all $k > r$;
    rescale the row to sum to one: $P_{ij} = P_{ij} / \sum_k P_{ik}$;
end
```

Appendix C

Swiss roll dataset

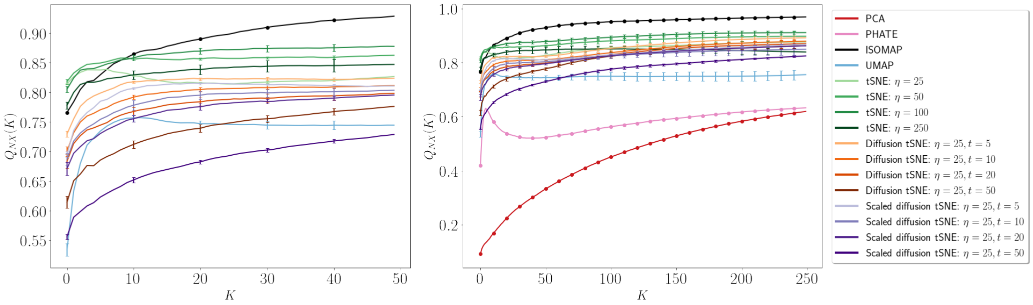
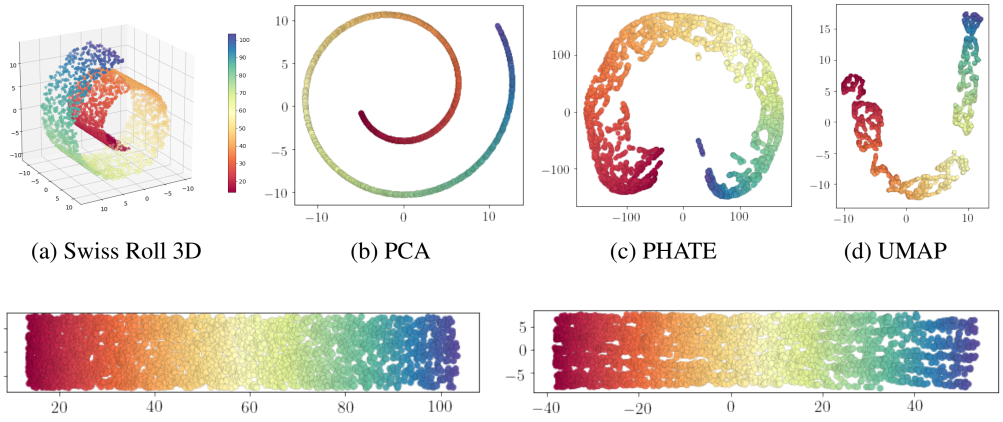


Figure S1: Co-ranking, $Q_{NX}(K)$, curves for varying choices of K-nearrest neighbors (x-axis), measuring embeddings' quality for recovering local neighborhoods. Left plot is a zoomed-in subset of the right plot. Ground truth natural parametrization (latent rectangle) was used for computing $Q_{NX}(K)$.



(a) Swiss Roll 3D

(b) PCA

(c) PHATE

(d) UMAP

(e) True latent parametrization

(f) ISOMAP

Figure S2: Swiss roll with other methods.

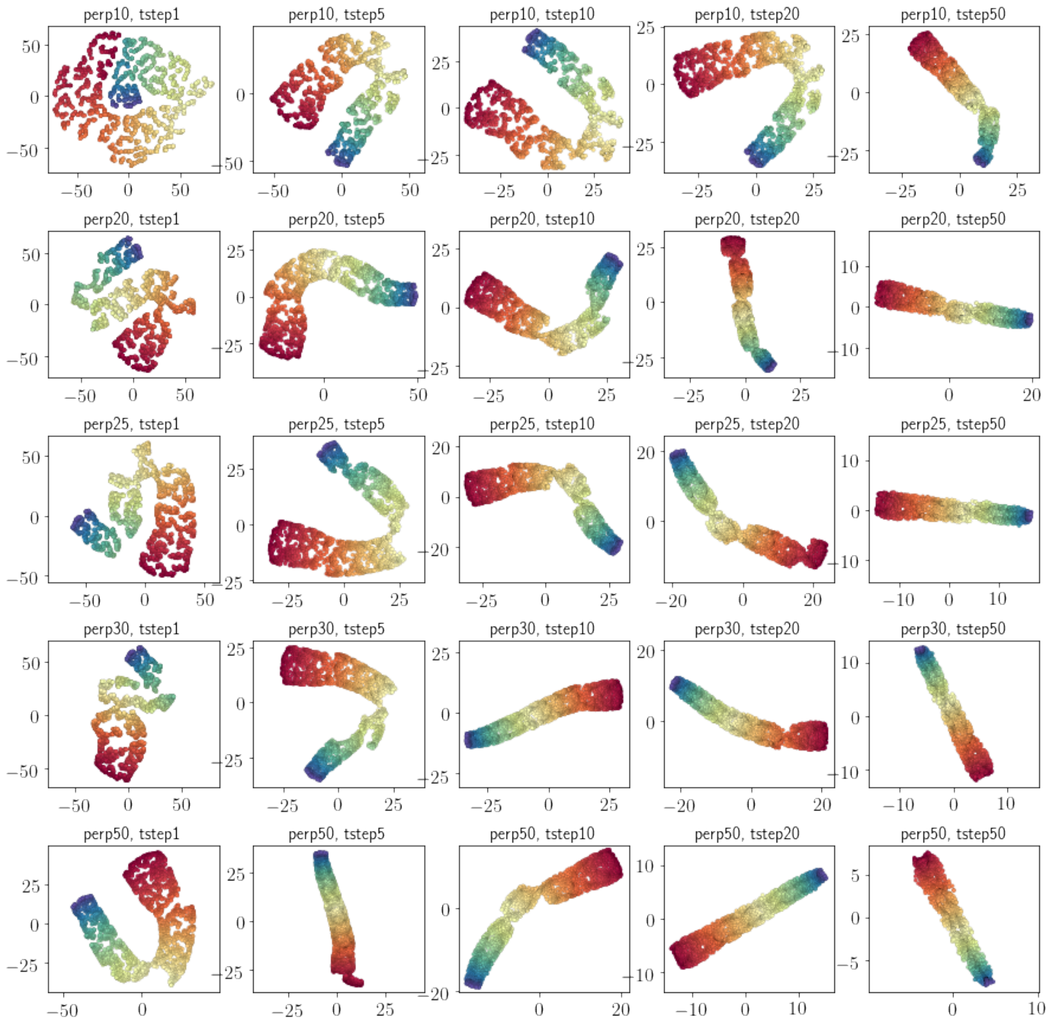


Figure S3: Diffusion t-SNE with different perplexity and time step values.

Single cell datasets

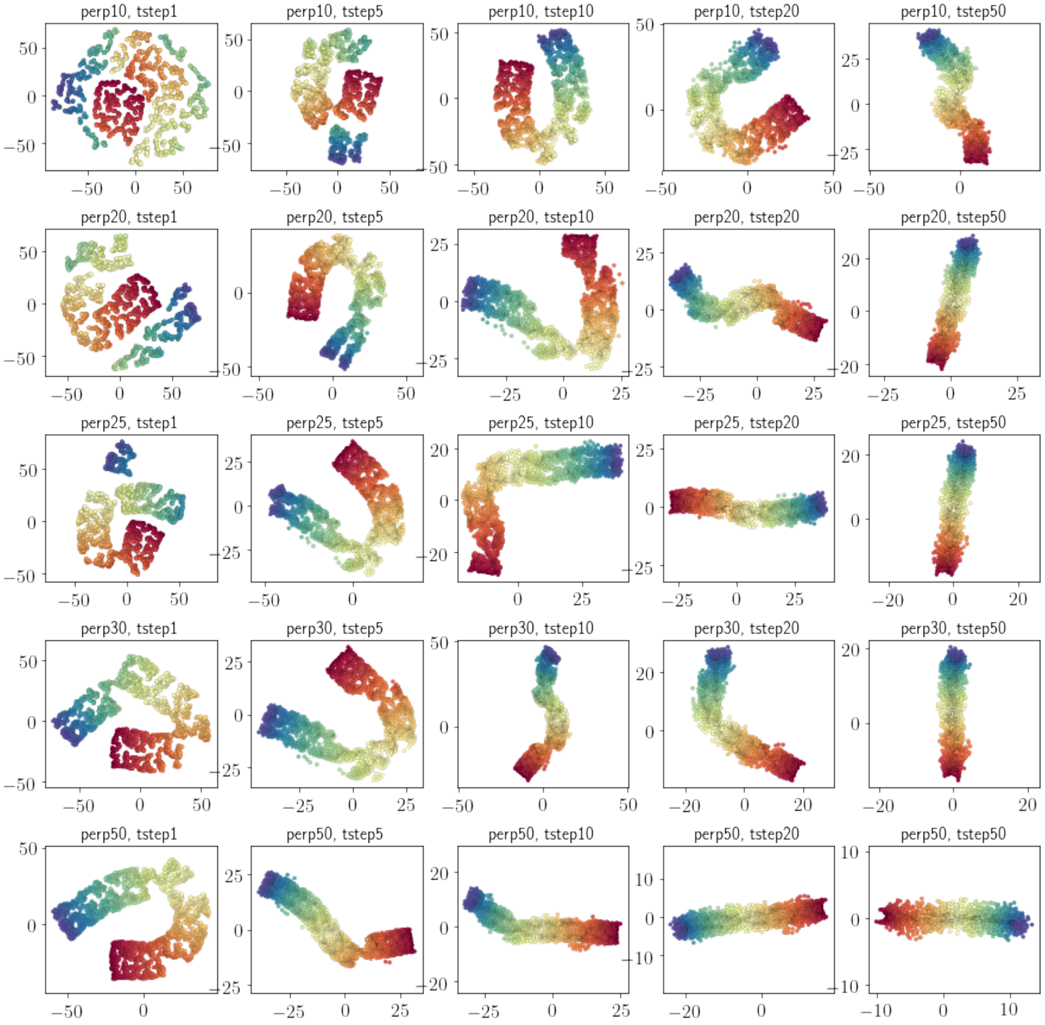


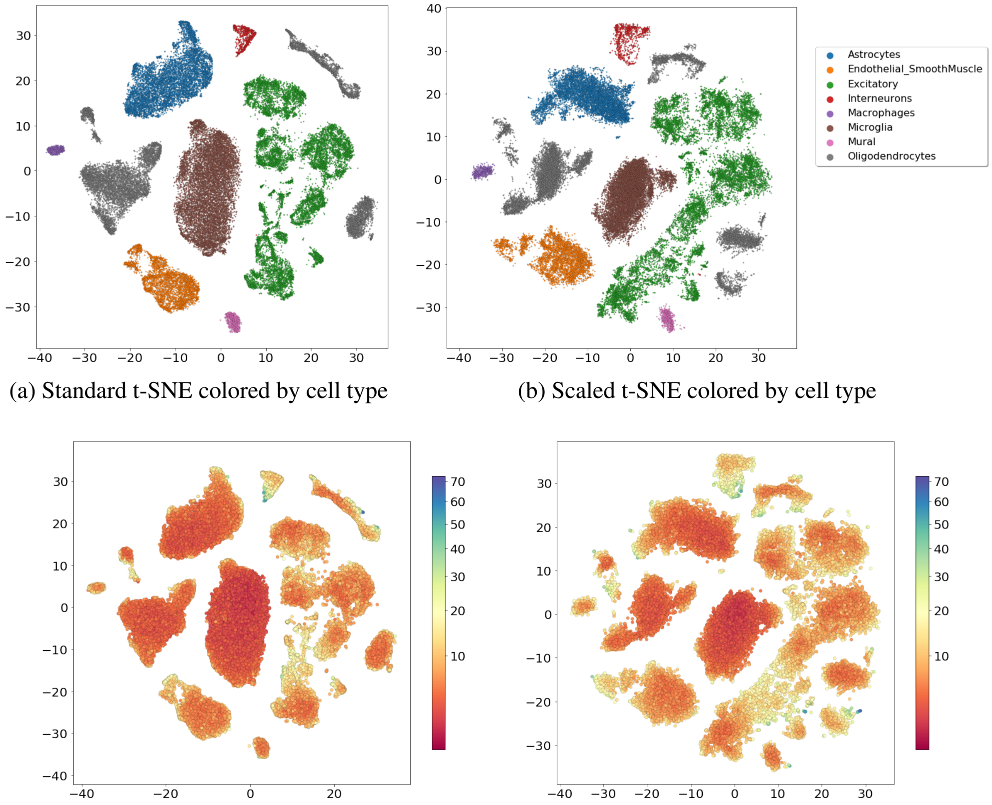
Figure S4: Scaled diffusion t-SNE with different perplexity and time step values.

Table S1: Mean and standard deviation (across 5 runs) for the average (over all data points) Spearman rank correlation coefficient between the distances measured in the output 2D embedding and the distances measured on the true natural parameters of the Swiss roll. Default parameters were used for UMAP[*] and PHATE[**] embeddings.

	Swiss Roll
PCA	0.412
Isomap	0.999
PHATE	0.759
UMAP	0.879 (0.0161)
t-SNE	
$\eta = 10$	0.573 (0.0794)
$\eta = 20$	0.739 (0.0189)
$\eta = 25$	0.747 (0.0095)
$\eta = 50$	0.786 (0.1222)
$\eta = 100$	0.868 (0.0036)
$\eta = 250$	0.498 (0.0121)
$\eta = 300$	0.459 (0.0152)
$\eta = 500$	0.403 (0.0134)
Diffusion t-SNE, $\eta = 25$	
$t = 5$	0.901 (0.0288)
$t = 10$	0.989 (0.0015)
$t = 20$	0.989 (0.0013)
$t = 50$	0.989 (0.0013)
$t = 100$	0.987 (0.0002)
Diffusion t-SNE, $\eta = 50$	
$t = 5$	0.992 (0.0023)
$t = 10$	0.989 (0.0013)
$t = 20$	0.991 (0.0018)
$t = 50$	0.987 (0.0008)
$t = 100$	0.979 (0.0001)
Scaled diffusion t-SNE, $\eta = 25$	
$t = 5$	0.836(0.0102)
$t = 10$	0.969(0.0084)
$t = 20$	0.991(0.0015)
$t = 50$	0.991(0.0007)
$t = 100$	0.988(0.0008)
Scaled diffusion t-SNE, $\eta = 50$	
$t = 5$	0.989 (0.0030)
$t = 10$	0.994 (0.0023)
$t = 20$	0.994 (0.0016)
$t = 50$	0.989 (0.0005)
$t = 100$	0.979 (0.0002)

[*] implementation used: `https://github.com/lmcinnes/umap`

[**] implementation used: `https://github.com/KrishnaswamyLab/PHATE`



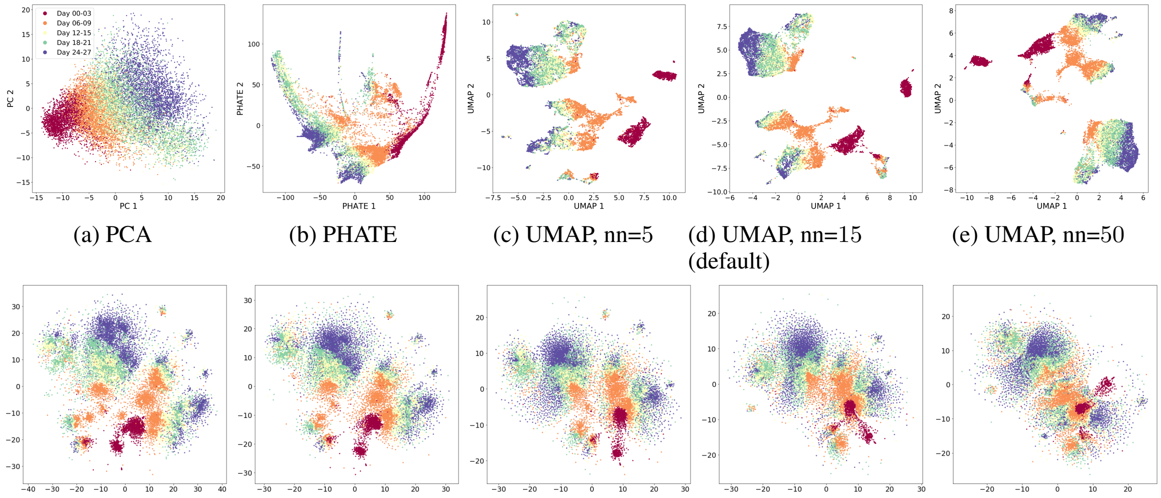
(a) Standard t-SNE colored by cell type

(b) Scaled t-SNE colored by cell type

(c) Standard t-SNE colored by bandwidth

(d) Diffusion t-SNE colored by bandwidth

Figure S5: Single cell expression dataset from a study of the mouse visual cortex by Hrvatin et al. [6]. We see that the microglia cluster containing many data points is artificially expanded in the standard embedding (small bandwidths lead to smaller the proximities for the same distance). Scaled t-SNE allows varying density across the output space, and shows cluster with sizes reflecting the local differences in variance present in the input data. Both algorithms where used the first 50 PCs as input data and 100 as perplexity value.



(a) PCA

(b) PHATE

(c) UMAP, nn=5

(d) UMAP, nn=15 (default)

(e) UMAP, nn=50

(f) Scaled diffusion t-SNE $\eta = 20; t = 5$

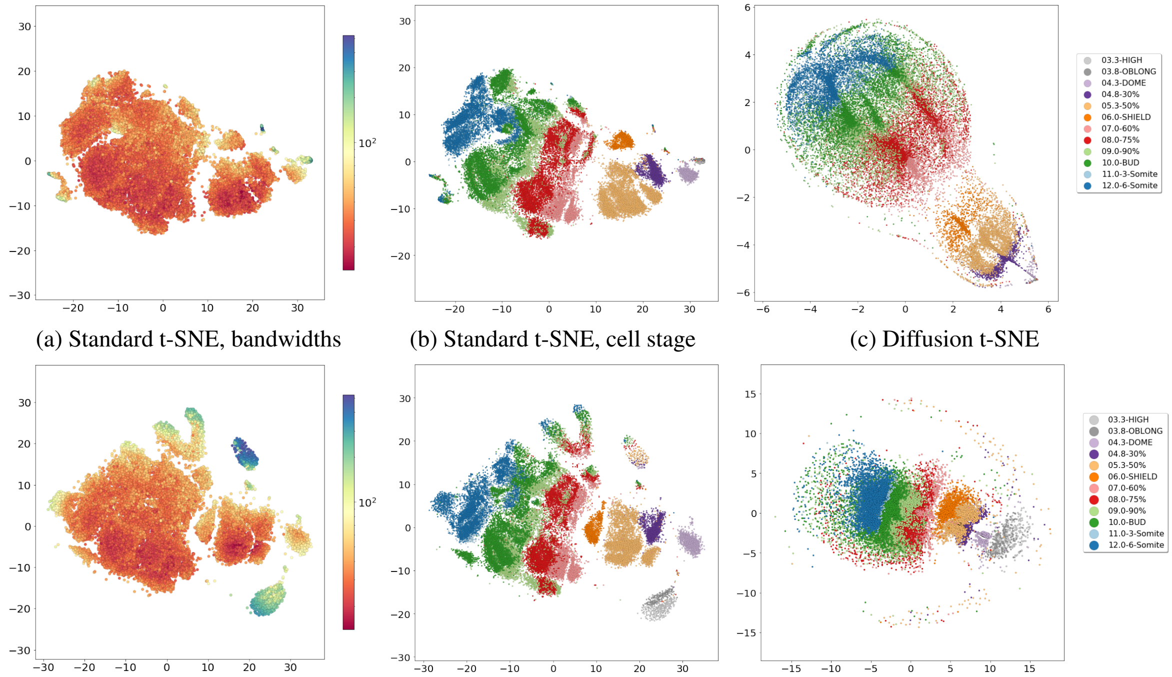
(g) Scaled diffusion t-SNE $\eta = 20; t = 10$

(h) Scaled Diffusion t-SNE $\eta = 20; t = 20$

(i) Scaled diffusion t-SNE $\eta = 20; t = 30$

(j) Scaled diffusion t-SNE $\eta = 20; t = 50$

Figure S6: Embryoid Body dataset from [18]. Additional embedding provided using PHATE, UMAP, and scaled diffusion t-SNE.



(a) Standard t-SNE, bandwidths

(b) Standard t-SNE, cell stage

(c) Diffusion t-SNE

(d) Scaled t-SNE, bandwidths

(e) Scaled t-SNE, cell stage

(f) Scaled diffusion t-SNE

Figure S7: Single cell RNA-seq dataset collected by Farrell et al. [4] to study the zebrafish embryogenesis. Standard t-SNE embedding colored by estimated bandwidths (a) and cell stage (b). Diffusion t-SNE ($t = 10$), both unscaled (c) and scaled (f) is better at recovering the embryonic developmental trajectories showing a more consistent progression through different stages unlike the standard t-SNE which outputs disconnected patches. Scaling conditional probabilities achieves a better embedding resolution for sparser regions (d-f). Highly variable regions occupy more space and are not clustered together. Large perplexity ($\eta = 1000$) was set for all plots, as the dataset is large ($> 38,000$ cells) and highly disconnected.