Journal of Machine Learning **ISSN**: 2790-2048(e), 2790-203X(p)

Soft-Constrained Distance Preserving t-SNE

Joseph Balderas [1], Li Wang * [1], Andrzej Korzeniowski [1], and Ren-Cang Li [1]

[1]Department of Mathematics, University of Texas at Arlington, Arlington, TX 76019, USA.

Abstract. Dimension reduction is a crucial tool for high-dimensional data analysis. Many dimension reduction techniques have been proposed for preserving different properties of a given dataset. For data visualization, t-distributed stochastic neighborhood embedding (t-SNE) is a popular method due to its ability to produce nicely separated clusters. However, t-SNE suffers from some major drawbacks. In this paper, a distance preserving t-SNE (DPt-SNE) is proposed, aiming to capture the global structure of the data and simultaneously maintain its local cluster separation. The basic idea is to incorporate a set of soft constraints, i.e. relaxing expected pairwise distance preserving constraints in order to regulate the low-dimensional embedding to preserve global structure encoded in the given distance metric of input data. In addition, we introduce a scaling optimization parameter to alleviate potential issues that arise when the difference between high and low dimensional distances is too large to overcome. Experimental results on six datasets positively confirm that our DPt-SNE can better reveal global structure than t-SNE, while retaining competitive clustering separation.

Keywords:
t-SNE,
Distance preservation,
Dimensionality reduction,
Data visualization.

Article Info.:
Volume: X
Number: X
Pages: 1 - 18
Date: /2025
doi.org/10.4208/jml.250414

Article History:
Received: 14/04/2025
Accepted: 20/10/2025

Communicated by:
Qianxiao Li

1 Introduction

Dimension reduction (DR) is the process of embedding high-dimensional data into a low dimensional space while preserving intrinsic features of the original data. In particular, the low-dimensional data can be visualized on a scatter plot when the dimension of the embedding space is just 2 or 3. Data visualization gives out important visual insights into global and local structural properties of a dataset, such as its cluster structure and its distributional characteristics. A low-dimensional embedding is said to preserve local structure if neighboring points in the input space are still neighbors in the embedding space; the embedding is said to preserve global structure if the relative positioning or distances between neighborhoods in the input space are preserved in the embedding space [26]. Unfortunately, low-dimensional embeddings can be misleading sometimes because of unintended false clusters or incorrectly placing clusters far apart in the low-dimensional space [27]. Additionally, many DR methods are adept at preserving local structure but not global structure, or vice-versa [12]. For this reason, DR methods are typically chosen based on the intended task by the user.

Many DR methods have been proposed in the literature, each possessing unique advantages or disadvantages. One of the earliest and most successful DR methods is principal component analysis (PCA) [13], which linearly projects data onto a new coordinate

[*]Corresponding author. li.wang@uta.edu

system pointing in the directions of maximum variance [4]. While PCA is adept at linearly preserving global structure of data [1], it cannot model complex nonlinear relationships. Kernel PCA (KPCA) [20] overcomes this shortcoming by utilizing the kernel trick to handle nonlinearity in data. Manifold learning [6] is an approach to DR based on the assumption that high-dimensional data often lies on a low-dimensional manifold; thus, a low-dimensional representation of data can be obtained by projecting the data onto this manifold. Isometric feature mapping (ISOMAP) [22], one of the earliest manifold learning algorithms, estimates geodesic distances and then uses multidimensional scaling (MDS) [8] to obtain an embedding that preserves these distances. Additionally, local linear embedding (LLE) [19] models the manifold as a collection of patches, and Laplacian Eigenmaps [3] uses the spectral graph theory to obtain an embedding with locality preserving properties. Maximum variance unfolding (MVU) [28] is another local method that finds a kernel matrix by maximizing variance in the feature space while preserving angles and distances between neighbors in the original space.

These classical manifold learning methods focus on preserving raw distances, which can lead to poorly preserved neighborhood structure [26] and inadequate visualizations for complex high-dimensional data [24]. To address this issue, modern DR methods focus on preserving graph structure instead of raw distances [26]. A popular example is t-SNE [24], which models probability distributions over the high-dimensional data and the embedded data and minimizes the Kullback-Leibler (KL) divergence between the two distributions. In particular, t-SNE uses the heavy tailed Student t-distribution to model low-dimensional points, resulting in visualization with nicely separated clusters. Maximum posterior manifold embedding (MPME) [16] is a probabilistic DR framework which learns a distribution function of embedded points given a close prior distribution and preserves expected pairwise distances to additionally uncover any global skeleton structure of data. Uniform manifold approximation and projection (UMAP) [17] is a recent method which produces visualizations that rival t-SNE and additionally preserves more global structure. UMAP works by creating a graph in the high-dimensional space and then finding a graph in the low-dimensional space with the same topological structure. Moving away from local methods is TriMap [1], a modern DR method which uses a triplet constraint to capture global structure more effectively than the commonly used t-SNE and UMAP. Finally, pairwise controlled manifold approximation projection (PaCMAP) [26] is based on the desirable loss function principle and optimized via first focusing on preserving global structure and then focusing on preserving local structure.

Of those DR methods, t-SNE is commonly picked out for its impressive data visualizations. As shown in [12], t-SNE effectively preserves local structure, scoring high compared to other DR methods on metrics that measure local structure preservation like support vector machine (SVM) and k-nearest neighbor (KNN) accuracy. However, t-SNE possesses some major drawbacks, such as poor preservation of global structure, false clustering, and sensitivity to parameter selection and initialization [12]. To improve t-SNE's performance on capturing global structure, we propose incorporating soft-constrained expected pairwise distance preservation constraints to its KL cost function, similar to those of MPME. Additionally, we utilize the probabilities calculated over points in the high dimensional space as weights in the added constraints. Extensive experiments are conducted on six datasets. Experimental results show that our proposed method can reveal global structure better than t-SNE, while maintaining competitive clustering separation.

The rest of this paper is organized as follows. First in Section 2, we introduce t-SNE and MPME which form the basis of our proposed method. Next in Section 3, we introduce the proposed method, presenting motivation, model formulation, and optimization process. We present our numerical experiments and discuss the performance of the proposed method in Section 4, and finally, conclusions are drawn in Section 5.

2 Background on t-SNE and MPME

In this section, we introduce two DR methods: t-SNE and MPME, which form the foundation of our proposed method. Let $\mathbf{x}_1, \ldots, \mathbf{x}_n$ be high-dimensional data points in $\mathbb{R}^D$ and let $\mathbf{y}_1, \ldots, \mathbf{y}_n$ be its low-dimensional representation in $\mathbb{R}^d$ where usually $d \ll D$.

2.1 t-SNE

Van der Maaten and Hinton [24] proposed t-SNE as an improved version of the unsupervised data visualization method SNE [11]. In particular, t-SNE fixed the SNE crowding problem and came with a more efficient optimization process. t-SNE produces a low-dimensional representation for data via modeling probability distributions for the original data and embedded data and by minimizing the KL divergence between the two distributions.

t-SNE uses the conditional similarity

$$p_{j|i} = \begin{cases} \dfrac{\exp\left(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma_i^2\right)}{\sum_{k\neq i}\exp\left(-\|\mathbf{x}_i - \mathbf{x}_k\|^2/2\sigma_i^2\right)}, & i \neq j, \\ 0, & i = j \end{cases} \tag{2.1}$$

to model the probability that the point $\mathbf{x}_j$ is a neighbor to $\mathbf{x}_i$. Here, the bandwidth parameter σ_i satisfies $\mathrm{Perp} = 2^{H(P_i)}$, where perplexity Perp is specified by the user and $H(P_i) = -\sum_j p_{j|i} \log p_{j|i}$ denotes the Shannon entropy for the probability distribution P_i over all of the other data points [24]. These values are then used to compute the corresponding symmetric similarity

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}, \tag{2.2}$$

which defines a probability distribution $P \equiv [p_{ij}]$ over all data pairs. Similarly, a distribution $Q \equiv [q_{ij}]$ over data pairs in the low-dimensional space is defined using the student t-distribution with one degree of freedom

$$q_{ij} = \begin{cases} \dfrac{\left(\|\mathbf{y}_i - \mathbf{y}_j\|^2 + 1\right)^{-1}}{\sum_{k\neq l}\left(\|\mathbf{y}_k - \mathbf{y}_l\|^2 + 1\right)^{-1}}, & i \neq j \\ 0, & i = j. \end{cases} \tag{2.3}$$

To obtain an optimal embedding, t-SNE minimizes the KL divergence of the two distributions

$$\mathrm{KL}(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}, \tag{2.4}$$

usually by gradient descent with momentum.

The t-SNE method is adept at preserving local neighborhoods in the low-dimensional space, resulting in nicely separated clusters for visualization. Variations and extensions [21,23] have also been proposed to address t-SNE's computational issues on large datasets. Despite its wide usage, Huang *et al.* [12] pointed out that t-SNE poorly preserved global structure, was quite sensitive to its perplexity parameter and initialization, and could produce results that were misleading and difficult to interpret. Additionally, t-SNE usually works well for $d \in \{2,3\}$ but is not well suited for general DR tasks for $d > 3$ [24].

2.2 MPME

MPME, proposed by Mao *et al.* [16], is a probabilistic DR method which learns a posterior density function of embedded data such that pairwise distances are preserved in the low-dimensional space. Although t-SNE is also a probabilistic DR method, the posterior density function in MPME treats $\{\mathbf{y}_i\}_{i=1}^n$ as random vectors, while t-SNE views them as deterministic. Once the posterior is obtained, low-dimensional points can be calculated by the maximum a posteriori (MAP) estimation of the distribution.

Write the random matrix

$$Y = [\mathbf{y}_1, \ldots, \mathbf{y}_n] \equiv [\mathbf{f}_1, \ldots, \mathbf{f}_d]^\top \in \mathbb{R}^{d \times n},$$

where $\mathbf{f}_i$ is the i-th row of Y. Also, denote the input data set $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ by $\mathbb{D}$. MPME aims to estimate the posterior density function $p(Y|\mathbb{D})$ under the following assumptions:

1. $p(Y|\mathbb{D})$ is close to a prior distribution $q(Y)$.
2. The rows $\mathbf{f}_i$ of Y are independent, and so we can write $p(Y|\mathbb{D}) = \prod_{i=1}^d p_i(\mathbf{f}_i|\mathbb{D})$ and $q(Y) = \prod_{i=1}^d q_i(\mathbf{f}_i)$.

Additionally, MPME requires that the expected Euclidean distances for points in the low-dimensional space preserve pairwise distances in the high-dimensional space

$$\mathbb{E}_{p(Y|\mathbb{D})}\left[\|\mathbf{y}_i - \mathbf{y}_j\|^2\right] = \phi_{ij}, \quad \forall\, i,j, \tag{2.5}$$

where ϕ_{ij} is a distance measure between $\mathbf{x}_i$ and $\mathbf{x}_j$, such as the squared Euclidean distance $\|\mathbf{x}_i - \mathbf{x}_j\|^2$. The posterior density function is estimated by solving the optimization problem

$$\begin{aligned} \min_{p(Y|\mathbb{D}) \in \mathbb{P}, \{\xi_{ij}\}} \quad & \mathrm{KL}\big(p(Y|\mathbb{D})||q(Y)\big) + C \sum_{i,j} \xi_{ij} \\ \text{s.t.} \quad & \mathbb{E}_{p(Y|\mathbb{D})}\left[\|\mathbf{y}_i - \mathbf{y}_j\|^2\right] \leq \phi_{ij} + \xi_{ij}, \quad \xi_{ij} \geq 0, \quad \forall\, i,j, \end{aligned} \tag{2.6}$$

where the error tolerance variables ξ_{ij} are introduced to account for noises, C is a regularization parameter, and $\mathbb{P}$ is the feasible set of density function p. When the prior $q(Y)$ is

Gaussian, an analytic expression for $p(Y|\mathbb{D})$ can be obtained by solving the dual of (2.6). Finally, the embedded points $\mathbf{y}_1, \ldots, \mathbf{y}_n$ are obtained by the MAP estimation. In general, solving (2.6) is difficult for general priors q.

In addition to DR, the dual variables obtained by solving the dual of (2.6) form a sparse nonnegative similarity matrix W which can be interpreted as a weighted neighborhood graph over points in the embedded space that captures the skeleton structure of the original data [16]. The optimization problem for learning W can be formulated as a box-constrained convex optimization problem, and thus, can be solved efficiently by optimization methods such as the limited-memory Broyden-Fletcher-Goldfarb-Shanno method for box constraints (L-BFGS-B) [5]. Still, the full algorithm for solving MPME outlined in [16] has computational complexity $\mathcal{O}(N^3)$, and so it scales poorly for very large datasets.

3 Proposed model – DPt-SNE

3.1 Motivation

Ideally, a reliable DR method will produce an embedding that can capture local and global structure of a high-dimensional dataset. However, most methods preserve either local or global structure but not both [26]. For example, t-SNE scores high on evaluation metrics for local structure, like the KNN and SVM accuracy, but scores poorly on evaluation metrics for global structure, like the random triplet accuracy, spearman correlation, and k-nearest classes preservation [12].

To address this weakness in t-SNE, we propose to incorporate a pairwise distance preserving constraint like (2.5) to the t-SNE objective. Alternatively, one can interpret such a model as MPME with its continuous distributions replaced with the discrete distributions P and Q of t-SNE. We would also like our model to incorporate the probabilities calculated from input data as weights in distance constraints. These two enhancements to t-SNE are meant to significantly improve t-SNE's ability to preserve global structure of data without sacrificing its ability to preserve local structure. We name our method the distance preserving t-SNE (DPt-SNE).

3.2 Model formulation

Again let $\mathbf{x}_1, \ldots, \mathbf{x}_n$ be the sample data points in $\mathbb{R}^D$ and $\pi = (\pi_1, \ldots, \pi_n)$ be the stationary distribution with respect to the right stochastic matrix $P_c \in \mathbb{R}^{n \times n}$, whose (i, j)-th entry is the conditional probability $p_{j|i}$ given by (2.1). We have

$$\pi P_c = \pi, \quad \sum_{i=1}^{n} \pi_i = 1, \quad \pi_i \geq 0, \quad \forall\, i.$$

Let $Y_1, \ldots, Y_n$ be i.i.d. column vectors in $\mathbb{R}^d$ obtained by sampling from π over a set of low-dimensional points $\mathbf{y}_1, \ldots, \mathbf{y}_n$ so that $\Pr(Y_i = \mathbf{y}_j) = \pi_j$, $\forall i, j$. Since π reflects the density of the input dataset [7], if the low-dimensional points preserve neighborhoods, then the sample ought to preserve destiny regions of high dimensional points. The average distance of point $\mathbf{y}_i$ to the other embedded points is given by

$$\frac{1}{n}\sum_{j=1}^{n}\mathbb{E}\|\mathbf{y}_i - Y_j\|^2 = \mathbb{E}\|\mathbf{y}_i - Y_1\|^2 = \sum_{j=1}^{n}\|\mathbf{y}_i - \mathbf{y}_j\|^2 \mathrm{Pr}(Y_j = \mathbf{y}_j), \tag{3.1}$$

where the first and second equalities follow from the fact that $Y_1, \ldots, Y_n$ are i.i.d. This leads to the first constraint which imposes that the average distance of each data point to all other points is preserved in the embedded space

$$\frac{1}{n}\sum_{j=1}^{n}\mathbb{E}\|\mathbf{y}_i - Y_j\|^2 = \sum_{j=1}^{n}\|\mathbf{y}_i - \mathbf{y}_j\|^2 \mathrm{Pr}(Y_j = \mathbf{y}_j) = \frac{1}{n}\sum_{j=1}^{n}\phi_{ij}, \quad i = 1, \ldots, n, \tag{3.2}$$

where ϕ_{ij} is the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. Additionally, we can impose the average of all pairwise distances to be preserved in the embedded space

$$\mathbb{E}\|Y_i - Y_j\|^2 = \sum_{k=1}^{n}\sum_{l=1}^{n}\|\mathbf{y}_k - \mathbf{y}_l\|^2 \mathrm{Pr}(Y_i = \mathbf{y}_k)\mathrm{Pr}(Y_j = \mathbf{y}_l) = \frac{1}{n^2}\sum_{k,l}\phi_{kl}. \tag{3.3}$$

Our proposed model aims to find embedded data points as the solution of the optimization problem

$$\begin{aligned} \min_{\{\mathbf{y}_1,\ldots,\mathbf{y}_n\}} \quad & \Big\{\mathrm{KL}(P||Q) := \sum_{i\neq j} p_{ij}\mathrm{log}\frac{p_{ij}}{q_{ij}}\Big\} \\ \text{s.t.} \quad & (3.2) \quad \text{and} \quad (3.3), \end{aligned} \tag{3.4}$$

where p_{ij} is defined as in (2.2) and q_{ij} as in (2.3).

Alternatively, constraints (3.2) and (3.3) can be dealt with more effectively via regularization. Let $D_Y \in \mathbb{R}^{n\times n}$ with the (i,j)-th entry $\|\mathbf{y}_i - \mathbf{y}_j\|^2$ and let $\Phi = [\phi_{ij}] \in \mathbb{R}^{n\times n}$. Corresponding to constraints (3.2) and (3.3), we define two loss terms by relaxing these hard constraints as soft constraints via square losses as follows:

$$\mathrm{Loss}_1 = \sum_{i=1}^{n}\left[\sum_{j=1}^{n}\left(\pi_j\|\mathbf{y}_i - \mathbf{y}_j\|^2 - \frac{1}{n}\phi_{ij}\right)\right]^2 = \mathbf{1}_n^\top D_1^\top D_1 \mathbf{1}_n, \tag{3.5}$$

where $D_1 = D_Y\mathrm{diag}(\pi) - \Phi/n$ and $\mathbf{1}_n \in \mathbb{R}^n$ is the column vector of all ones, and

$$\mathrm{Loss}_2 = \left[\sum_{i,j}\left(\pi_i\pi_j\|\mathbf{y}_i - \mathbf{y}_j\|^2 - \frac{1}{n^2}\phi_{ij}\right)\right]^2 = \left(\mathbf{1}_n^\top D_2 \mathbf{1}_n\right)^2, \tag{3.6}$$

where $D_2 = \mathrm{diag}(\pi)D_Y\mathrm{diag}(\pi) - \Phi/n^2$. Incorporating these two loss functions, we arrive at the following unconstrained optimization problem, as a variant of (3.4):

$$\min_{\{\mathbf{y}_1,\ldots,\mathbf{y}_n\}}\left\{f(Y) := \sum_{i\neq j} p_{ij}\log\frac{p_{ij}}{q_{ij}} + \sum_{i=1}^{2} C_i \cdot \mathrm{Loss}_i\right\}, \tag{3.7}$$

where $C_i > 0$ for $i = 1, 2$ are regularization parameters. Notice that solving (3.7) requires calculating the stationary distribution π, which is not feasible for large n. Instead, we approximate π with the distribution

$$\hat{\pi} = \frac{1}{n}\mathbf{1}_n^\top P_c. \tag{3.8}$$

The loss functions in (3.5) and (3.6) can become very large if the two types of distances mismatch greatly in scale, and that can cause numerical issues, such as overflow, and degrade model quality. To alleviate these issues, we introduce a scaling variable γ and modify the two loss functions to

$$\text{Loss}_1^\gamma = \sum_{i=1}^n \left[\sum_{j=1}^n \left(\pi_j \|\mathbf{y}_i - \mathbf{y}_j\|^2 - \frac{\gamma}{n}\phi_{ij} \right) \right]^2 = \mathbf{1}_n^T \left(D_1^\gamma\right)^\top D_1^\gamma \mathbf{1}_n, \tag{3.9}$$

$$\text{Loss}_2^\gamma = \left[\sum_{i,j} \left(\pi_i \pi_j \|\mathbf{y}_i - \mathbf{y}_j\|^2 - \frac{\gamma}{n^2}\phi_{ij} \right) \right]^2 = \left(\mathbf{1}_n^\top D_2^\gamma \mathbf{1}_n\right)^2, \tag{3.10}$$

where

$$D_1^\gamma = D_Y \text{diag}(\pi) - \frac{\gamma}{n}\Phi, \tag{3.11}$$

$$D_2^\gamma = \text{diag}(\pi) D_Y \text{diag}(\pi) - \frac{\gamma}{n^2}\Phi. \tag{3.12}$$

Finally, our proposed model becomes

$$\min_{\{\mathbf{y}_1,\ldots,\mathbf{y}_n\},\gamma} \left\{ f(Y,\gamma) := \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}} + \sum_{i=1}^2 C_i \cdot \text{Loss}_i^\gamma \right\}. \tag{3.13}$$

3.3 Optimization method

In order to apply the gradient descent with momentum to solve (3.13), we need to calculate the gradient of $f(Y,\gamma)$ with respect to $\mathbf{y}_k, k \in \{1,\ldots,n\}$. From [24], we have the KL gradient

$$\frac{\partial}{\partial \mathbf{y}_k} \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}} = 4 \sum_{j \neq k} (p_{kj} - q_{kj}) Z_{kj} (\mathbf{y}_k - \mathbf{y}_j), \tag{3.14}$$

where $Z_{kj} = (\|\mathbf{y}_k - \mathbf{y}_j\|^2 + 1)^{-1}$. Next, we compute the gradients of the two loss terms. Recall D_1^γ in (3.11) and D_2^γ in (3.12) and let $D_{1;(j,:)}^\gamma$ be the j-th row of D_1^γ. We have

$$\begin{aligned} \frac{\partial \text{Loss}_1^\gamma}{\partial \mathbf{y}_k} &= \frac{\partial}{\partial \mathbf{y}_k} \sum_{i=1}^n \left[\sum_{j=1}^n \left(\pi_j \|\mathbf{y}_i - \mathbf{y}_j\|^2 - \frac{\gamma}{n}\phi_{ij} \right) \right]^2 \\ &= \frac{\partial}{\partial \mathbf{y}_k} \left[\sum_{j=1}^n \left(\pi_j \|\mathbf{y}_k - \mathbf{y}_j\|^2 - \frac{\gamma}{n}\phi_{kj} \right) \right]^2 \end{aligned}$$

$$
\begin{aligned}
&+ \frac{\partial}{\partial \mathbf{y}_k} \sum_{i \neq k} \left[\sum_{j=1}^{n} \left(\pi_j \|\mathbf{y}_i - \mathbf{y}_j\|^2 - \frac{\gamma}{n}\phi_{ij} \right) \right]^2 \\
&= 4 \sum_{j=1}^{n} \left(\pi_j \|\mathbf{y}_k - \mathbf{y}_j\|^2 - \frac{\gamma}{n}\phi_{kj} \right) \sum_{j=1}^{n} \pi_j (\mathbf{y}_k - \mathbf{y}_j) \\
&\quad + 4 \sum_{i \neq k} \sum_{j=1}^{n} \left(\pi_j \|\mathbf{y}_i - \mathbf{y}_j\|^2 - \frac{\gamma}{n}\phi_{ij} \right) \pi_k (\mathbf{y}_k - \mathbf{y}_i) \\
&= 4 \sum_{j \neq k} \left[\pi_j D^{\gamma}_{1;(k,:)} \mathbf{1}_n + \pi_k D^{\gamma}_{1;(j,:)} \mathbf{1}_n \right] (\mathbf{y}_k - \mathbf{y}_j),
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial \mathrm{Loss}_2^{\gamma}}{\partial \mathbf{y}_k} &= \frac{\partial}{\partial \mathbf{y}_k} \left[\sum_{i,j} \left(\pi_i \pi_j \|\mathbf{y}_i - \mathbf{y}_j\|^2 - \frac{\gamma}{n^2}\phi_{ij} \right) \right]^2 \\
&= 8 \sum_{i,j} \left(\pi_i \pi_j \|\mathbf{y}_i - \mathbf{y}_j\|^2 - \frac{\gamma}{n^2}\phi_{ij} \right) \times \sum_{j=1}^{n} \pi_k \pi_j (\mathbf{y}_k - \mathbf{y}_j) \\
&= 8 \big(\mathbf{1}_n^{\top} D_2^{\gamma} \mathbf{1}_n \big) \sum_{j \neq k} \pi_k \pi_j (\mathbf{y}_k - \mathbf{y}_j).
\end{aligned}
$$

Putting all three gradients together yields

$$
\begin{aligned}
\frac{\partial f(Y,\gamma)}{\partial \mathbf{y}_k} = \sum_{j \neq k} \Big\{ & 4(p_{kj} - q_{kj}) Z_{kj} \\
& + 4C_1 \left[\pi_j D^{\gamma}_{1;(k,:)} \mathbf{1}_n + \pi_k D^{\gamma}_{1;(j,:)} \mathbf{1}_n \right] \\
& + 8C_2 \pi_k \pi_j \big(\mathbf{1}_n^{T} D_2^{\gamma} \mathbf{1}_n \big) \Big\} (\mathbf{y}_k - \mathbf{y}_j). \qquad (3.15)
\end{aligned}
$$

Write

$$
\frac{\partial f(Y,\gamma)}{\partial Y} = \left[\frac{\partial f(Y,\gamma)}{\partial \mathbf{y}_1}, \dots, \frac{\partial f(Y,\gamma)}{\partial \mathbf{y}_n} \right]. \qquad (3.16)
$$

Finally, we solve (3.13) alternatingly over Y and γ.

$$
\begin{aligned}
\overline{\mathbf{y}}_i &= \sum_{j=1}^{n} \pi_j \|\mathbf{y}_i - \mathbf{y}_j\|^2, \quad \overline{\phi}_i = \frac{1}{n} \sum_{j} \phi_{i,j}, \\
\hat{y} &= \sum_{i,j} \pi_i \pi_j \|\mathbf{y}_i - \mathbf{y}_j\|^2, \quad \hat{\phi} = \frac{1}{n^2} \sum_{i,j} \phi_{ij},
\end{aligned}
$$

and let $\alpha = C_2 / C_1$. Fixing Y, (3.13) is equivalent to

$$
\min_{\gamma} \mathrm{Loss}_1^{\gamma} + \alpha \mathrm{Loss}_2^{\gamma} = \min_{\gamma} \sum_{i} (\overline{\mathbf{y}}_i - \gamma \overline{\phi}_i)^2 + \alpha (\hat{y} - \gamma \hat{\phi})^2,
$$

whose optimal solution is explicitly given by

$$\gamma = \frac{\sum_i \overline{\mathbf{y}}_i \overline{\phi}_i + \alpha \hat{y} \hat{\phi}}{\sum_i \overline{\phi}_i^2 + \alpha \hat{\phi}^2}. \tag{3.17}$$

On the other hand, fixing γ, we update $\mathbf{y}_k$ by the gradient decent with momentum

$$\mathbf{y}_k^{(t)} = \mathbf{y}_k^{(t-1)} - \eta_t \frac{\partial f(Y)}{\partial \mathbf{y}_k} + \alpha_t \left(\mathbf{y}_k^{(t-1)} - \mathbf{y}_k^{(t-2)} \right) \tag{3.18}$$

for the t-th iteration, as in [24], where $\eta_t > 0$ is the learning rate and $\alpha_t \in (0, 1)$ is the momentum rate. Algorithm 1 outlines the procedure for solving (3.13) via gradient descent with momentum. In our experiments in the next section, for simplicity, we use $\eta_t = 500$ and $\alpha_t = 0.8$, which follows the setting used in the original python implementation of t-SNE[1].

Algorithm 1 DPt-SNE: Distance Preserving t-SNE.

```
Input: Data matrix $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, perplexity Perp, regularization parameters $C_1, C_2 > 0$,
    maximal number of iterations $N$, learning rate $\eta_t$ and momentum rate $\alpha_t$ for the $t$-th
    iteration.
Output: Embedded data matrix $Y_N$.
Compute distance matrix $\Phi = [\phi_{i,j}] \in \mathbb{R}^{n \times n}$ of $X$.
Compute stochastic matrix $P_c = [p_{j|i}]$ using (2.1).
Compute $p_{ij}$ using (2.2).
Estimate stationary distribution $\pi$ using (3.8).
Initialize $Y_0 \equiv [\mathbf{y}_1^{(0)}, \ldots, \mathbf{y}_n^{(0)}]$ by performing PCA on $X$ and set $V_{-1} = 0$.
for $t = 0$ to $N - 1$ do
    Compute the squared Euclidean distance matrix $D_Y$ for $Y = Y_t \equiv [\mathbf{y}_1^{(t)}, \ldots, \mathbf{y}_n^{(t)}]$.
    Compute $q_{ij}$ using (2.3).
    Compute $\gamma$ using (3.17).
    Compute $\left. \frac{\partial f(Y, \gamma)}{\partial Y} \right|_{Y = Y_t}$ using (3.16) with (3.15).
    $V_t = -\eta_t \left. \frac{\partial f(Y, \gamma)}{\partial Y} \right|_{Y = Y_t} + \alpha_t V_{t-1}$.
    $Y_{t+1} = Y_t + V_t$.
end for
return $Y_N$, the embedded data matrix.
```

It is worth noting that the proposed optimization method solves (3.13) alternatingly over Y and γ. This is because their gradient values could have very different scale depending on the input data, so the momentum-based gradient descent implemented in t-SNE may not be effective any longer. In contrast, the proposed alternating method has

[1] `https://lvdmaaten.github.io/tsne/code/tsne_python.zip`

two advantages: (i) the optimizer implemented in t-SNE is reusable, and (ii) optimizing γ enjoys a simple and closed-form solution in (3.17).

3.4 Initialization and distance metrics

The low-dimensional embedding matrix Y_0 can be initialized for Algorithm 1 by performing PCA on X. PCA which is usually computed by Cuppen's divide-and conquer method implemented in LAPACK [2,9,10,14,18] is computationally efficient and produces starting embeddings which maintain much of the original data's global structure. There are many choices for the distance matrix Φ, such as squared Euclidean, normalized, Cosine [25], and Gaussian. In this paper, we only explore the squared Euclidean distances to model the global structure of the input data.

4 Experiments

4.1 Experimental setting

The proposed method, DPt-SNE, aims to enhance the popular t-SNE so as to preserve pairwise distances of the input data. Naturally, the original t-SNE is the most appropriate baseline for comparison. Previous t-SNE variants, such as BH t-SNE [23] and FIt-SNE [15], focus on improving computational speed and scalability rather than anything else, and in fact, they still rely on the same conceptual idea of the original t-SNE algorithm [26] and retain the same KL-divergence objective function, which preserves local neighborhoods. Additionally, as shown in [12], those variants perform comparably with respect to classification and random triplet accuracy. For completeness, we also include the comparisons using BH t-SNE and FIt-SNE methods, which are implemented in openTSNE[2]. We surmise that the proposed approach of integrating pairwise distances can be incorporated into those t-SNE variants without much difficulty, but such extensions will not be explored in this paper.

Since t-SNE is mainly devised for data visualization, the reduced dimension is often set to either 2 or 3. In our experiments, both 2 and 3 are used as the reduced dimensions for study. For the ease of presentation, we name t-SNE with reduced dimension m by tSNE-m and our DPt-SNE with reduced dimension m by DPt-SNE-m, where $m \in \{2, 3\}$. The proposed model (3.13) has two more hyper-parameters C_1 and C_2 in addition to the perplexity from t-SNE. To reduce the complexity of hyper-parameter tuning, we set $C_1 = C_2 = C$ and tune C in the range $[10^{-8}, 1]$. Note that if $C = 0$, DPt-SNE (3.13) is equivalent to t-SNE. The perplexity of both t-SNE and DPt-SNE is tuned from a predefined set $\{10, 20, 30, 40, 50\}$.

It is well known that t-SNE can generate embeddings with good clustering patterns; so we evaluate the compared methods on datasets for classification tasks where clustering patterns can be properly validated in terms of class labels. Six publicly available benchmark datasets are used in the experiments. The statistics of these datasets are shown in Table 4.1.

[2] `https://github.com/pavlin-policar/openTSNE`

Table 4.1: The six datasets.

Dataset	n	d	c
banknote	1372	4	2
COIL20	1440	1024	20
YALE-B	2414	1024	38
mnist2500	2500	784	10
pendigits	3498	16	10
mammoth_3d	10000	3	11

* n is the number of samples, d is the number of features, and c is the number of classes.

Two evaluation criteria are used to measure the quality of the compared methods. Classification accuracy is achieved by the one-nearest neighbor classifier where the input dataset is split randomly into $q\%$ for testing and $(100 - q)\%$ for training with $q \in \{90, 80, 70, 60, 50, 40, 30\}$. The classification accuracy of the testing set over 10 random splits of the input data is reported. Another evaluation criterion is random triplet accuracy, which is the percentage of triplets whose relative distance in the high dimensional space is preserved in the low dimensional space [26]. To calculate random triplet accuracy, we take each data point $\mathbf{x}_i$, find 5 random triplets of data points which include $\mathbf{x}_i$, and then calculate the relative distances for each triplet in the high dimensional space and for the corresponding triplet in the low dimensional space. The accuracy is calculated by counting the number of triplets whose order is preserved in the low dimensional space. Notice that the random triplet accuracy is merely a heuristic principle. Due to the random nature of this metric, we repeat this procedure 10 times and take the average as the final accuracy. For both evaluation metrics, we report the highest average scores for t-SNE and DPt-SNE achieved over their respective parameter spaces mentioned above. Notice that classification accuracy focuses on the quality of clustering patterns, and thus, is viewed as a measure for local neighborhood preservation. On the other hand, random triplet accuracy focuses on the relative positioning of neighborhoods throughout the entire dataset, and, thus, is viewed as a measure for global structure preservation [12,26].

4.2 Performance evaluation

The classification accuracy and standard deviation averaged over 10 random splitings of 10% for training and 90% for testing on the six datasets are shown in Table 4.2. The random triplet accuracy and standard deviation averaged over 10 random repeats are reported in Table 4.3. We have the following observations:

1. DPt-SNE can achieve competitive classification accuracy when compared to t-SNE, BH t-SNE, and FIt-SNE, with marginally better performance with respect to both reduced dimensions.

2. DPt-SNE shows significantly better random triplet accuracy than t-SNE with respect to both reduced dimensions. The improvement is up to 14% for $m = 2$ and 16% for $m = 3$. The biggest improvement is observed on dataset banknote, while only

marginal improvements are observed on dataset minist2500. Similar results can be observed when comparing DPt-SNE with BH t-SNE and FIt-SNE, except that FIt-SNE performs best for mammoth_3d in 2-D embedding and BH t-SNE performs best for mnist2500 in 3-D embedding.

In addition, we further explore how the classification accuracy is affected by the ratio of training/testing splitting. The experimental results in Fig. 4.1 show that (i) more portion of data for training can generally help increase classification accuracy; (ii) DPt-SNE can often perform better than t-SNE when the training portion of data is relatively small.

Table 4.2: Classification accuracy via one-nearest neighbor classifier on 2-D and 3-D embeddings.

Dataset	tSNE-2	Barnes-Hut-2	FIt-SNE-2	DPt-SNE-2
YALE-B	0.6780 ± 0.0154	0.7087 ± 0.0085	**0.7491 ± 0.0131**	0.6719 ± 0.0147
mnist2500	0.8801 ± 0.0111	0.8603 ± 0.0070	0.8732 ± 0.0104	**0.8841 ± 0.0107**
banknote	0.9939 ± 0.0036	0.9950 ± 0.0037	**0.9952 ± 0.0037**	0.9949 ± 0.0051
pendigits	0.9784 ± 0.0051	0.9785 ± 0.0059	0.9782 ± 0.0044	**0.9788 ± 0.0061**
mammoth_3d	0.9650 ± 0.0031	**0.9651 ± 0.0036**	0.9632 ± 0.0035	0.9649 ± 0.0036
COIL20	0.9620 ± 0.0082	0.9370 ± 0.0145	0.9527 ± 0.0095	**0.9682 ± 0.0177**

Dataset	tSNE-3	Barnes-Hut-3	DPt-SNE-3
YALE-B	0.7038 ± 0.0171	0.6665 ± 0.0118	**0.7052 ± 0.0131**
mnist2500	0.8884 ± 0.0104	0.8657 ± 0.0089	**0.8967 ± 0.0081**
banknote	0.9947 ± 0.0034	0.9963 ± 0.0030	**0.9964 ± 0.0036**
pendigits	0.9774 ± 0.0035	0.9774 ± 0.0065	**0.9794 ± 0.0050**
mammoth_3d	0.9637 ± 0.0038	0.9646 ± 0.0039	**0.9647 ± 0.0033**
COIL20	0.9691 ± 0.0091	0.9417 ± 0.0116	**0.9724 ± 0.0081**

* Data splitting: 10% for training and 90% for testing. FIt-SNE in openTSNE is not well supported for 3-D embedding.

Table 4.3: Random triplet mean accuracy and standard deviation.

Dataset	tSNE-2	Barnes-Hut-2	FIt-SNE-2	DPt-SNE-2
YALE-B	0.7700 ± 0.0028	0.7710 ± 0.0035	0.7851 ± 0.0026	**0.8042 ± 0.0027**
mnist2500	0.6529 ± 0.0027	0.6559 ± 0.0039	0.6429 ± 0.0033	**0.6567 ± 0.0046**
banknote	0.7138 ± 0.0042	0.7999 ± 0.0031	0.7844 ± 0.0025	**0.8504 ± 0.0046**
pendigits	0.6223 ± 0.0023	0.6379 ± 0.0027	0.6308 ± 0.0044	**0.6809 ± 0.0035**
mammoth_3d	0.6600 ± 0.0023	0.7035 ± 0.0018	**0.7765 ± 0.0024**	0.7138 ± 0.0018
COIL20	0.6542 ± 0.0056	0.6770 ± 0.0076	0.6955 ± 0.0040	**0.7306 ± 0.0064**

Dataset	tSNE-3	Barnes-Hut-3	DPt-SNE-3
YALE-B	0.7707 ± 0.0042	0.7809 ± 0.0027	**0.7853 ± 0.0044**
mnist2500	0.6725 ± 0.0033	**0.6805 ± 0.0058**	0.6765 ± 0.0040
banknote	0.7506 ± 0.0033	0.7497 ± 0.0055	**0.9152 ± 0.0035**
pendigits	0.6782 ± 0.0049	0.6810 ± 0.0030	**0.6838 ± 0.0038**
mammoth_3d	0.6879 ± 0.0022	0.7337 ± 0.0019	**0.7387 ± 0.0020**
COIL20	0.6966 ± 0.0066	0.6998 ± 0.0046	**0.7317 ± 0.0044**

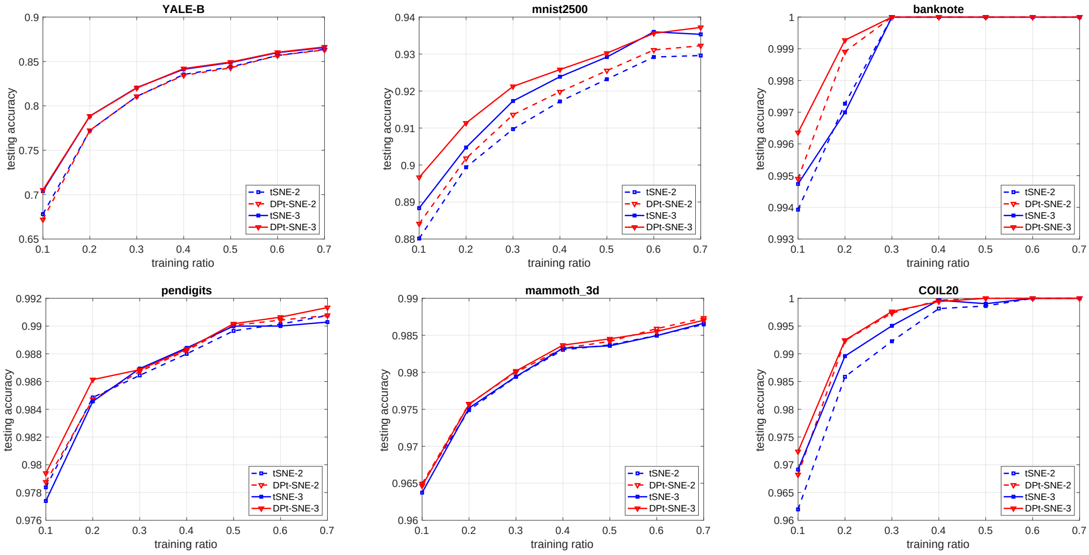

Figure 4.1: Classification accuracy via one-nearest neighbor classifier on 2-D and 3-D embeddings by t-SNE and by our DPt-SNE as the training portion of data varies from 10% to 70%.

4.3 Parameter sensitivity analysis

Hyper-parameter C_1 and C_2 are the key factors that have impact on the DPt-SNE embeddings. In our experiments, we set $C = C_1 = C_2$. In addition, a suitable choice of perplexity is important for both DPt-SNE and t-SNE. For parameter sensitivity, we examine the performances by both methods in terms of the two aforementioned metrics. It is worth noting that DPt-SNE with $C = 0$ is equivalent to t-SNE. Fig. 4.2 shows the performance of t-SNE ($C = 0$) and DPt-SNE ($C > 0$) on the six datasets with perplexity values in the range $\{10, 20, 30, 40, 50\}$. For each given perplexity, with properly tuned $C > 0$, DPt-SNE outperforms t-SNE with respect to the two different metrics. For those datasets, $C \leq 10^{-4}$ can often achieve good triplet mean accuracy and competitive classification accuracy.

4.4 Data visualization

Since t-SNE is mainly used for data visualization, the 2-D embeddings of t-SNE and DPt-SNE on the six datasets are shown in Fig. 4.3. In order to show the effect of the parameter C, we demonstrate embeddings of DPt-SNE in three different scenarios: DPt-SNE (tradeoff), DPt-SNE (best accuracy) and DPt-SNE (best triplet). Specifically, DPt-SNE (best accuracy) and DPt-SNE (best triplet) are DPt-SNE with the best classification accuracy and best triplet mean accuracy respectively by performing model selection over C. DPt-SNE (tradeoff) is the tradeoff between classification accuracy and triplet mean accuracy, which is achieved by performing model selection over C with respect to the F_1-like metric[3] over

[3] The F_1-like metric reuses the F_1 metric defined for binary classification with respect to precision and recall to measure the quality of embeddings with respect to classification accuracy and triplet mean accuracy. Its purpose is to select some embeddings which are balanced between the two metrics for visual demonstration.

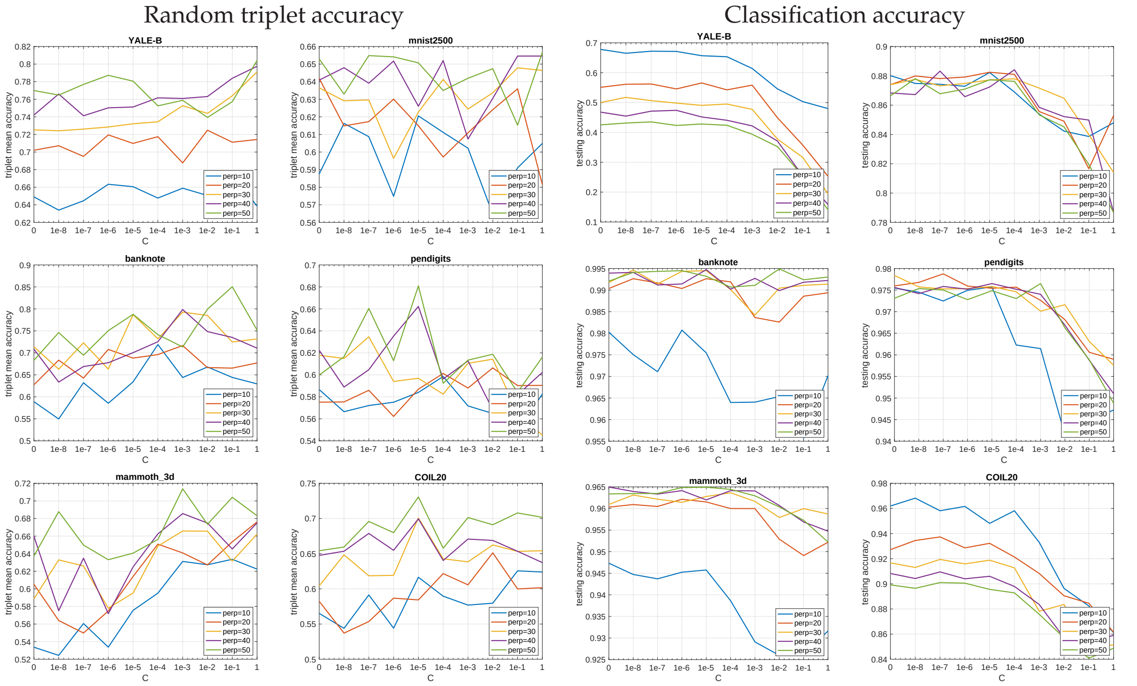

Figure 4.2: Ablation study on 2-D embeddings with respect to both testing accuracy and triplet mean accuracy.

classification accuracy and triplet mean accuracy. From Fig. 4.3, we have the following observations: (i) compared to t-SNE, DPt-SNE (tradeoff) can achieve competitive classification accuracy and better triplet mean accuracy; (ii) DPt-SNE (best accuracy) outperforms t-SNE; (iii) DPt-SNE (best triplet) produces quite different clustering patterns from t-SNE, better representing the global structure in data. These observations imply that DPt-SNE can achieve better performance in terms of both evaluation metrics with properly tuned C and is highly effective in integrating global structure into the original t-SNE modeling.

4.5 Approximation analysis of the stationary distribution

The proposed method depends on the stationary distribution π, which is approximated by (3.8). It is noted that the proposed approximation could be interpreted as the result of evolving the uniform probability mass function forward for only one time step under the Markov chain. Intuitively, the approximation could be made more accurate if multiple steps are performed. In this section, we explore how the number of steps could have impact on the testing performance of the proposed method. Let r be the number of steps. An approximation of the stationary distribution π with r steps is

$$\hat{\pi} = \frac{1}{n}\mathbf{1}_n^\top P_c^r.$$

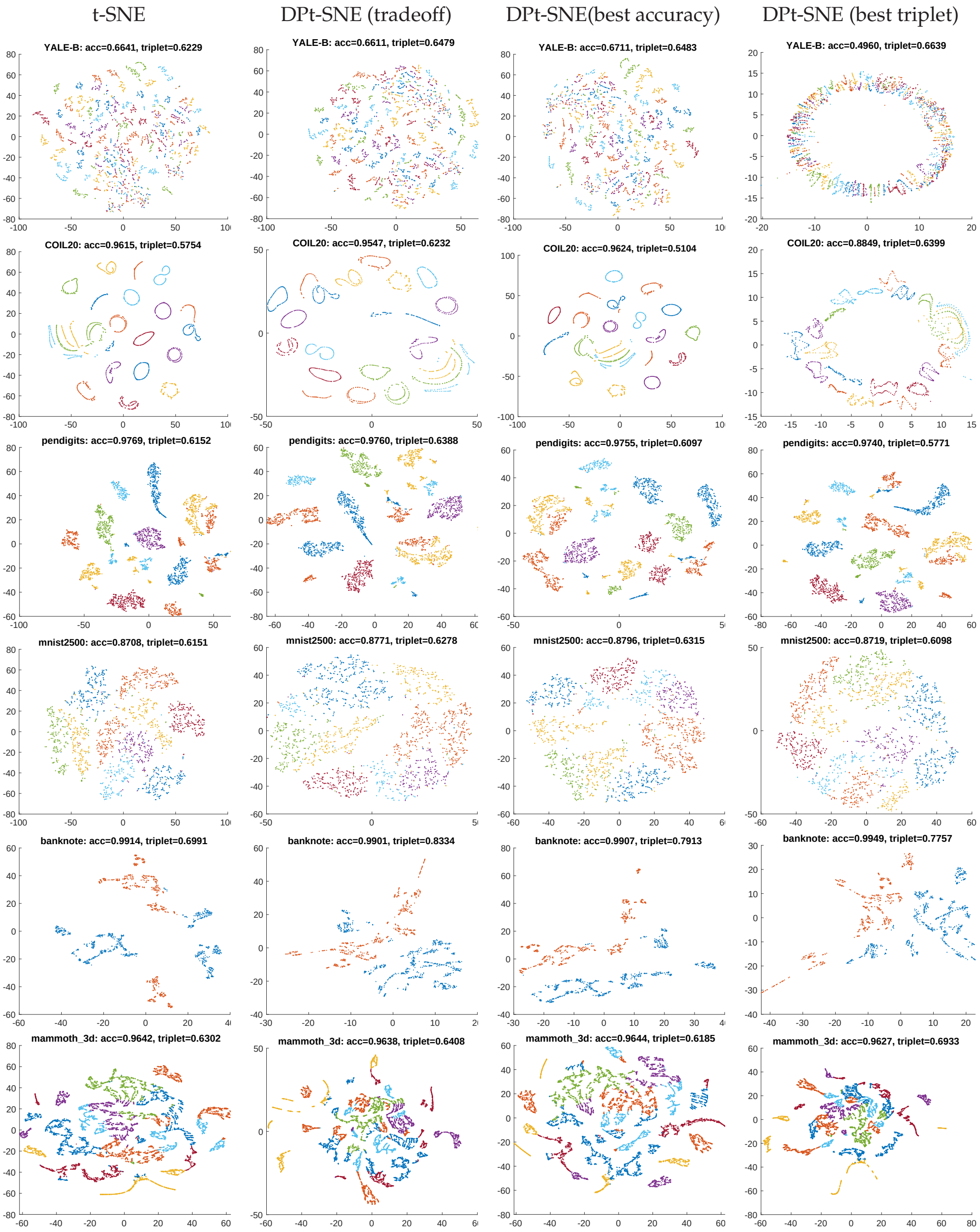

Figure 4.3: 2-D embeddings of the six datasets by t-SNE and by DPt-SNE.

We performed the same experiments as shown in Section 4.2 by varying $r \in \{1, 2, 3, 5, 10\}$. The experimental results of DPt-SNE-2 and DPt-SNE-3 in terms of both testing accuracy and triplet mean accuracy on the six datasets are shown in Fig. 4.4. It is observed that

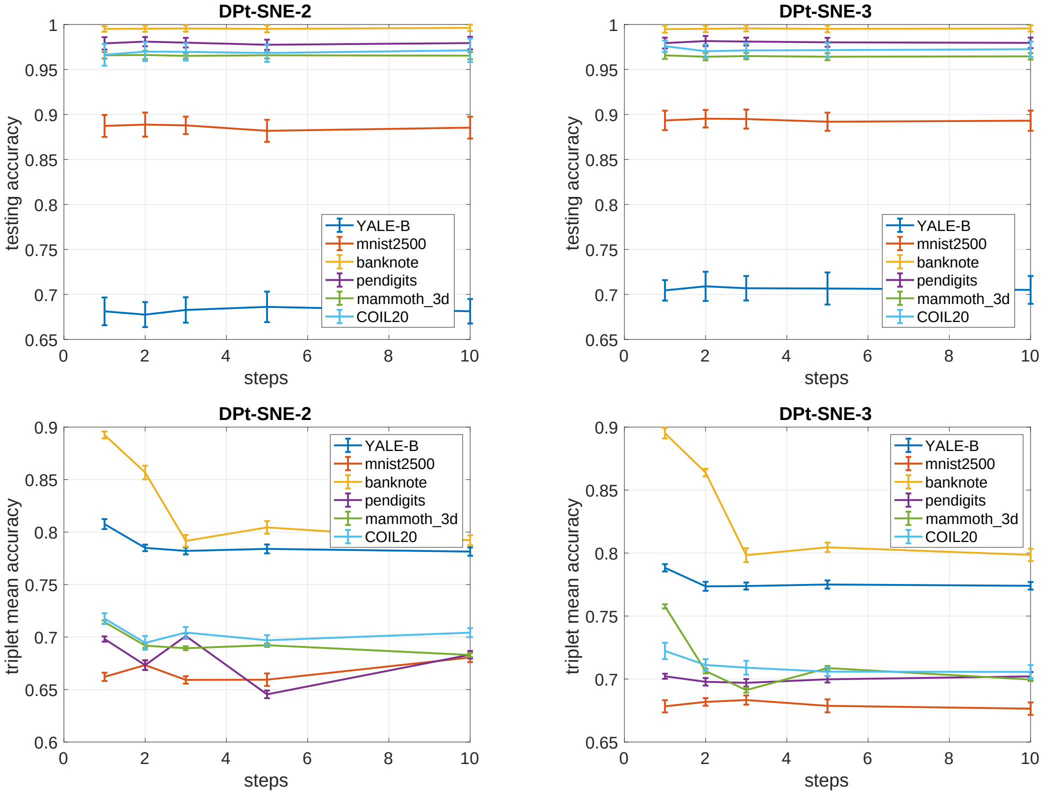

Figure 4.4: The impact of approximating stationary distribution over multi-steps via the standard deviation over 10 random experiments.

(i) the number of steps have marginal impact in terms of testing accuracy, and (ii) the triplet mean accuracy varies with respect to the underlying dataset, often with good performance at $r = 1$, indicating the current approximation by (3.8) is often adequate.

5 Conclusion

We propose a new variant of t-SNE: DPt-SNE, which can better capture the global structure in the input data than the original t-SNE while maintaining its power of cluster separation. The use of expected pairwise distances over discrete distributions truly pays off as the bearer of global structure. Our experiments were conducted on the six datasets in Table 4.1 to explore the importance of preserving global and local structures in terms of classification accuracy and triplet mean accuracy. The experimental results demonstrate that DPt-SNE can achieve better triplet mean accuracy with competitive classification accuracy. In addition, the visualization results show quite different patterns of global and local structures, and that further validate the effectiveness of preserving pairwise distances in revealing both global and local structures.

Acknowledgments

The authors wish to thank the anonymous reviewers for their constructive suggestions and comments that have been incorporated to improve the presentation considerably.

Dr. Li Wang is supported in part by the NSF (Grant No. DMS-2407692) and by the NIH (Grant No. R01GM160515). Dr. Ren-Cang Li is supported in part by the NSF (Grant No. DMS-2407692).

References

[1] E. Amid and M. K. Warmuth, TriMap: Large-scale dimensionality reduction using triplets, *arXiv: 1910.00204*, 2019.

[2] E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, S. Ostrouchov, and D. Sorensen, *LAPACK Users' Guide*, SIAM, 1999.

[3] M. Belkin and P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, in: *Advances in Neural Information Processing Systems*, Vol. 14, MIT Press, 585–591, 2001.

[4] C. J. C. Burges, Dimension reduction: A guided tour, *Found. Trends Mach. Learn.*, 2(4):275–365, 2010.

[5] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, A limited memory algorithm for bound constrained optimization, *SIAM J. Sci. Comput.*, 16(5):1190–1200, 1995.

[6] L. Cayton, *Algorithms for Manifold Learning*, Technical Report, University of California San Diego, 2005.

[7] R. R. Coifman and S. Lafon, Diffusion maps, *Appl. Comput. Harmon. Anal.*, 21(1):5–30, 2006.

[8] T. F. Cox and M. A. A. Cox, *Multidimensional Scaling*, CRC Press, 2000.

[9] J. J. M. Cuppen, A divide and conquer method for the symmetric eigenproblem, *Numer. Math.*, 36:177–195, 1981.

[10] M. Gu and S. C. Eisenstat, A stable and efficient algorithm for the rank-one modification of the symmetric eigenproblem, *SIAM J. Matrix Anal. Appl.*, 15(4):1266–1276, 1994.

[11] G. E. Hinton and S. Roweis, Stochastic neighbor embedding, in: *Advances in Neural Information Processing Systems*, Vol. 15, MIT Press, 857–864, 2002.

[12] H. Huang, Y. Wang, C. Rudin, and E. P. Browne, Towards a comprehensive evaluation of dimension reduction methods for transcriptomic data visualization, *Commun. Biol.*, 5(1):719, 2022.

[13] I. T. Jolliffe, *Principal Component Analysis*, Springer, 2002.

[14] R.-C. Li, Solving secular equations stably and efficiently, Technical Report UCB/CSD-94-851, University of California, 1993.

[15] G. C. Linderman, M. Rachh, J. G. Hoskins, S. Steinerberger, and Y. Kluger, Fast interpolation-based t-SNE for improved visualization of single-cell RNA-seq data, *Nat. Methods*, 16(3):243–245, 2019.

[16] Q. Mao, L. Wang, and I. W. Tsang, A unified probabilistic framework for robust manifold learning and embedding, *Mach. Learn.*, 106:627–650, 2017.

[17] L. McInnes, J. Healy, and J. Melville, UMAP: Uniform manifold approximation and projection for dimension reduction, *arXiv:1802.03426*, 2018.

[18] J. D. Rutter, A serial implementation of Cuppen's divide and conquer algorithm for the symmetric eigenvalue problem, Technical Report UCB/CSD-94-799, University of California, 1994.

[19] L. K. Saul and S. T. Roweis, Think globally, fit locally: Unsupervised learning of low dimensional manifolds, *J. Mach. Learn. Res.*, 4:119–155, 2003.

[20] B. Schölkopf, A. Smola, and K.-R. Müller, Kernel principal component analysis, in: *International Conference on Artificial Neural Networks*, Springer, 583–588, 1997.

[21] J. Tang, J. Liu, M. Zhang, and Q. Mei, Visualizing large-scale and high-dimensional data, in: *Proceedings of the 25th International Conference on World Wide Web*, International World Wide Web Conferences Steering Committee, 287–297, 2016.

[22] J. B. Tenenbaum, V. de Silva, and J. C. Langford, A global geometric framework for nonlinear dimensionality reduction, *Science*, 290(5500):2319–2323, 2000.

[23] L. Van Der Maaten, Accelerating t-SNE using tree-based algorithms, *J. Mach. Learn. Res.*, 15(1):3221–3245, 2014.

[24] L. Van Der Maaten and G. Hinton, Visualizing data using t-SNE, *J. Mach. Learn. Res.*, 9:2579–2605, 2008.

[25] L. Wang, H. Yin, and J. Zhang, Density-based distance preserving graph: Theoretical and practical analyses, *IEEE Trans. Neural Networks Learn. Syst.*, 34(9):6642–6649, 2021.

[26] Y. Wang, H. Huang, C. Rudin, and Y. Shaposhnik, Understanding how dimension reduction tools work: An empirical approach to deciphering t-SNE, UMAP, TriMAP, and PaCMAP for data visualization, *J. Mach. Learn. Res.*, 22(201):1–73, 2021.

[27] M. Wattenberg, F. Viégas, and I. Johnson, How to use t-SNE effectively, *Distill*, 1(10):e2, 2016.

[28] K. Q. Weinberger, F. Sha, and L. K. Saul, Learning a kernel matrix for nonlinear dimensionality reduction, in: *Proceedings of the 21st International Conference on Machine Learning*, ACM, 839–846, 2004.