Revisiting Data Visualization with t-SNE: A Reproduction Study

Jan Tiegges
McGill University

Jonathan Colaço Carr
McGill University

Rishabh Thaney
McGill University

Abstract

We present a reproducibility report for the paper "Visualizing Data using t-SNE". First, we reproduce the t-SNE visualizations on the three datasets considered in the paper, comparing t-SNE to Sammon Mapping, Isomap and Locally Linear Embedding visualizations. We then perform ablation studies that verify the effectiveness of t-SNE in alleviating the crowding problem, the use of PCA reduction in data preprocessing, and generalization error of classifiers trained on the visualization data. Overall, we found it relatively straightforward to reproduce the results. Our ablation studies confirm that t-SNE is a superior data visualization method. For future practitioners, we suggest preprocessing the data by using the principal components that explain 90% of the dataset variance.

1 Introduction

Identifying meaningful relationships in high dimensional data is a critical task in data science and machine learning. The goal of this project was to reproduce t-SNE [vdMH08], a popular method for visualizing high-dimensional data. Specifically, we used t-SNE to visualize three datasets - MNIST, the Olivetti Faces dataset and the COIL-20 dataset[1], comparing t-SNE visualizations with three other methods: Sammon Mapping [Sam69], Isomap [TdSL00] and Locally Linear Embeddings (LLEs) [RS00]. We then performed ablation studies to verify the following claims made in the original t-SNE paper:

(1) The use of t-distributions in t-SNE avoids class clusters being crowded together,

(2) PCA-reducing the data to 30 dimensions is a reasonable preprocessing step,

(3) Classifiers trained on t-SNE visualizations have the lowest generalization error when compared with other visualization methods.

Overall, our ablation studies confirm the paper's results and were relatively easy to implement. We find that t-SNE's use of t-distributions to compare low-dimensional data reduces cluster crowding. Although PCA-reducing the dataset to 30 components is reasonable, we find that using the principal components (PCs) which capture 90% of the explained variance gives slight improvements for two of the three datasets, both quantitatively and qualitatively. We confirm t-SNE reduces generalization error for a variety of classifiers (the original paper considered only the 1-nearest-neighbor classifier).

2 Methodology

2.1 Data Visualization Algorithms

In this section we review the baseline algorithms considered in this work as well as the necessary material on Stochastic Neighborhood Embedding (SNE) required to understand our ablation studies. A more rigorous analysis of t-SNE can be found in [CM22].

Baseline Algorithms. As in the original t-SNE paper, we compared the t-SNE visualizations with three other visualization methods: Sammon Mapping [Sam69], Isomap[TdSL00], and LLE [RS00]. Isomap and LLE were implemented with scikit-learn [BLB+13] using their default hyperparemeters. We used a publicly available implementation of Sammon Mapping[2] and used 200 gradient steps.

[1]The COIL-20 dataset is publicly available at `https://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php`

[2]`https://github.com/tompollard/sammon`

SNE Algorithms. In addition t-SNE, we implemented symmetric SNE (shortened to sym-SNE). Both sym-SNE and t-SNE measure the relatedness of data points $(x_i)_{i=1}^n$ in $\mathbb{R}^s$ (where $s \gg 3$) through joint probabilities $\mathbf{P} = (p_{ij})_{i,j=1}^n$. The probability mass p_{ij} measures the "strength of attraction" between x_i and x_j, and is defined for each $1 \leq i \leq n$ as $p_{ii} = 0$ and for $i \neq j$,

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2n}, \text{ where } \quad p_{j|i} = \frac{\exp(-\|x_i - x_j\|_2^2/(2\sigma_i^2))}{\sum_{k \neq i} \exp\left(-\|x_i - x_l\|_2^2/(2\sigma_i^2)\right)}, \tag{1}$$

where $\|\cdot\|_2^2$ is the squared ℓ_2 norm. Notice that the attraction strength $p_{\cdot|i}$ from point x_i is dispersed in $\mathbb{R}^s$ as a Gaussian distribution centered at x_i with variance σ_i^2. In particular, $p_{j|i}$ decreases monotonically with the Euclidean distance between x_i and x_j, and the hyperparameter σ_i^2 controls the overall strength of attraction between far away data points.

The key distinction between sym-SNE and t-SNE is how the strength of attraction is measured *between map points*. For the points $(y_i)_{i=1}^n$ in $\mathbb{R}^2$ (or $\mathbb{R}^3$), the sym-SNE and t-SNE algorithms measure the attractions $\mathbf{Q}^{\text{symSNE}} = (q_{ij}^{\text{symSNE}})_{i,j=1}^n$ and $\mathbf{Q}^{\text{tSNE}} = (q_{ij}^{\text{tSNE}})_{i,j=1}^n$ as $q_{ii}^{\text{symSNE}} = q_{ii}^{\text{tSNE}} = 0$ and for $i \neq j$:

$$q_{ij}^{\text{symSNE}} = \frac{\exp\left(-\|y_i - y_j\|_2^2\right)}{\sum_{k \neq l} \exp\left(-\|y_k - y_l\|^2\right)} \tag{2a}$$

$$q_{ij}^{\text{tSNE}} = \frac{\left(1 + \|y_i - y_j\|_2^2\right)^{-1}}{\sum_{k \neq l} (1 + \|y_k - y_l\|_2^2)^{-1}}. \tag{2b}$$

Notice that the sym-SNE attractions are dispersed as Gaussian, i.e. the *same* as the high-dimensional attractions. However, the t-SNE attractions are dispersed as a Cauchy distribution (a 1-dimensional t-distribution), which is *different* than the attraction dispersal for the high-dimensional data points. The authors claim that by using a Cauchy distribution rather than a Gaussian distribution to compute map point attractions, they avoid a "crowding problem", where map point clusters are forced to overlap. We verify this claim in Section 3.2.1.

Intuitively, when the data attractions $\mathbf{P}$ are similar the map point attractions $\mathbf{Q}$, the map points should faithfully represent the similarities between data points. Thus, the objective of both the sym-SNE and t-SNE algorithms is to find the map points $(\hat{y}_i)_{i=1}^n$ which minimize the KL divergence between $\mathbf{P}$ and $\mathbf{Q}$,

$$(\hat{y}_1, \ldots, \hat{y}_n) = \underset{(y_1,\ldots,y_n)}{\operatorname{argmin}} D_{KL}(\mathbf{P}, \mathbf{Q}) = \underset{(y_1,\ldots,y_n)}{\operatorname{argmin}} \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}, \tag{3}$$

with the p_{ij}'s defined in Equation 1 the q_{ij}'s for sym-SNE and t-SNE defined in Equations 2a and 2b, respectively. We implemented t-SNE using sci-kit learn [BLB+13] and implemented sym-SNE by adapting a publicly available implementation of t-SNE[3].

2.2 Datasets

The **MNIST-784** dataset [Den12] has 70,000 images showing handwritten numbers which are split into 60,000 training examples and 10,000 test examples. These digits are numbered from 0 to 9, making it a total of 10 classes. As in the original t-SNE paper, we perform visualizations for 6, 000 randomly selected MNIST images, rather than the full dataset. The **COIL-20** dataset has 1,440 pictures of 20 objects/classes from different angles. These images are grayscale and have a resolution of 32x32 pixels. Finally, the **Olivetti Faces** dataset contains 400 face images of 40 people, with each person having 10 different images varying in lighting, expressions, and facial details, making it a dataset with 40 classes and 10 examples per class. The images are grayscale, originally 92x112 pixels but available in a 64x64 resolution.

2.3 Hyperparameters

Experimenting with hyperparameters for these data visualization tools is difficult since the goal of data visualization is *qualitative*; we want a "good" visualization of the data. As the t-SNE authors note, it is hard to define quantitative measures of data visualization, making it more challenging to fine-tune hyperparameters.

For the t-SNE algorithm we considered the effect of two important hyperparameters: the number of PCs used in PCA-reducing the data in preprocessing and perplexity. The perplexity controls the variance allowed in each class

[3] `https://github.com/nlml/tsne_raw`

cluster (i.e. the σ_i's of Equation 1). In addition to experimenting with 30 PCs as in the original paper, we also considered PCA-preprocessing that used the PCs explaining 25%, 50%, 75%, 90% and 100% of the data (see Section 3.2.2).

Perplexity Experiment. The perplexity in the t-SNE algorithm is a hyperparameter related the width of the Gaussian distributions used to measure pairwise similarities in Equation 1.[WVJ16]. In Figure 1, we show the effect of perplexity on the visualizations of the MNIST dataset. The first thing to note is that for values less than 5, the t-SNE algorithm focuses too much on preserving local structures between individual points and fails to form clusters. The plots for the values 5 to 40 are quite similar, but the separations into the individual clusters are even clearer for higher values. However, local differences are sacrificed and the individual clusters are much denser. At even higher perplexity values (>40), local nuances less clear, which is why the value of 40 chosen by the authors seems reasonable for all experiments.

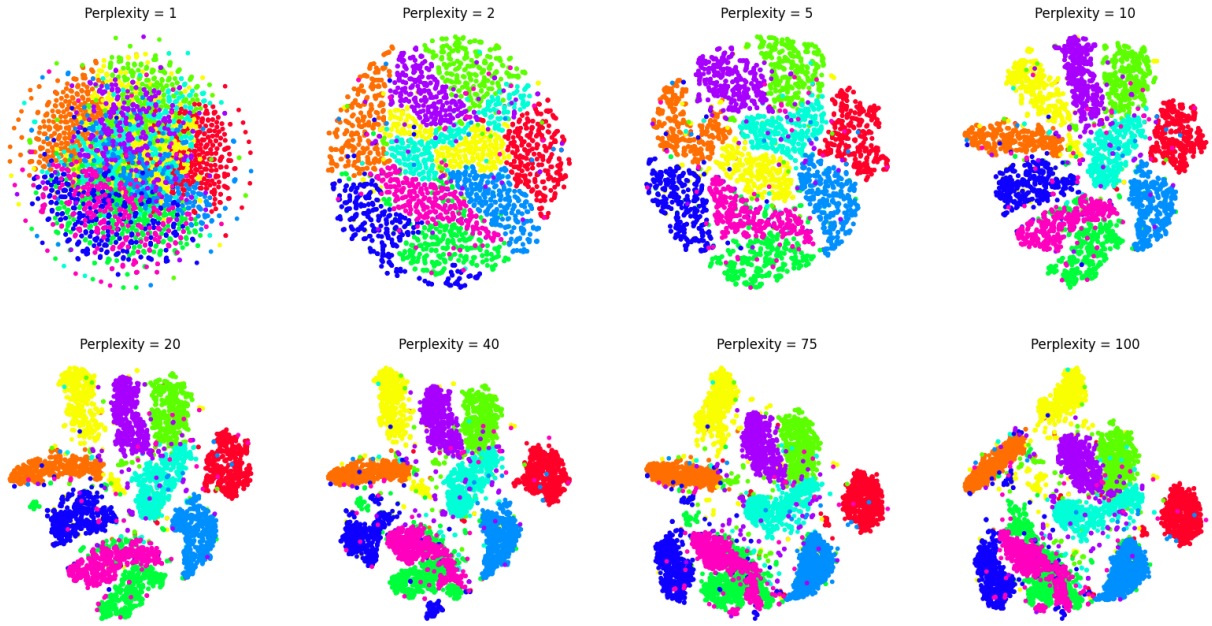


Figure 1: Effect of the Perplexity Parameter on t-SNE visualization of the MNIST dataset.

There are various other hyperparameters used in the first-order optimization method used by t-SNE, but the details of these parameters are not discussed in the original paper. We found that for t-SNE, Isomap and LLE, the default hyperparameters in scikit-learn were sufficient to reproduce the results.

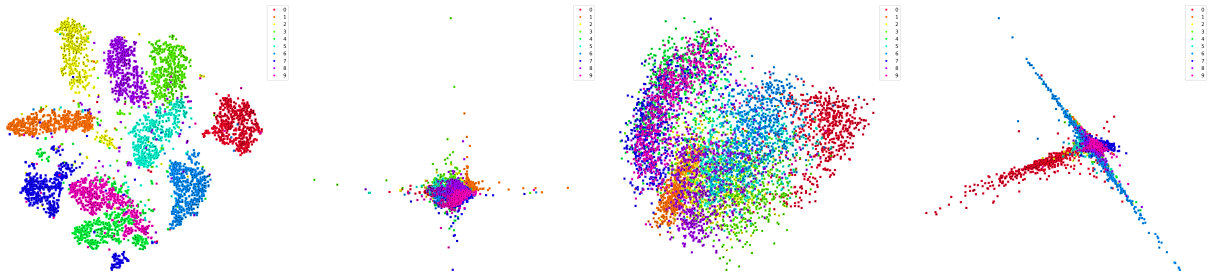
3 Experiments

3.1 Reproducing the original paper

Overall, we were able to reproduce the visualizations of the original paper in an almost identical way, supporting the authors' claims. t-SNE was the only method able to generate meaningful representations for all three data sets, although we think that it was still quite ambiguous in the case of the Olivetti data set. Of the other methods, Isomap stood out somewhat as a possible alternative, while the other two techniques performed rather poorly. The Sammon mapping visualizations differed the most from the results in the paper. Figure 2 shows the visualizations reproduced for the MNIST dataset. The visualizations for the COIL-20 and Olivetti Faces dataset are provided in the appendix.

MNIST Visualizations. For MNIST, our plots essentially draw the same conclusions as the original paper, where t-SNE clearly emerges as the strongest method, being the only one that manages to produce different clusters for each digit (Figure 2). Looking at the digits that break out of the clusters (Figure 9), the difficulty of the MNIST dataset also becomes clear, where some digits are very difficult to recognize (e.g. between 4 and 9). As in the original paper, the Sammon mapping also produces a ball in our experiments, but with significantly more outliers. This may be attributed to the fact that we use a different optimization method. The visualizations of Isomap and LLE are very similar to those in the paper.

Olivetti Faces Visualizations. For the Olivetti Faces data set (Appendix D), the authors argue that the t-SNE visualization alone enables a division into clusters, depending on the variation in head direction, facial expression or



(a) t-SNE (b) Sammon Mapping (c) Isomap (d) LLE

Figure 2: Comparison of data visualization methods on the MNIST data set.

glasses [vdMH08]. Despite the fact that we achieve almost identical results, upon closer inspection the individual clusters do not stand out clearly (Figure 13).

COIL-20 Visualizations. For the COIL-20 data set (Appendix E), the paper describes how t-SNE's mapping represents the one-dimensional structure of viewpoints as a closed loop for many of the 20 objects in the COIL-20 dataset. In cases where objects look similar from the front and back, t-SNE distorts the loop by mapping images from the front and back to nearby points. In comparison, Isomap and LLE are less effective at drawing clear dividing lines between distributions corresponding to different objects in COIL-20 [vdMH08]. Our results underline these findings and this data set particularly highlights the strengths of t-SNE.

3.2 Ablation Studies

3.2.1 The crowding problem: symmetric SNE vs t-SNE

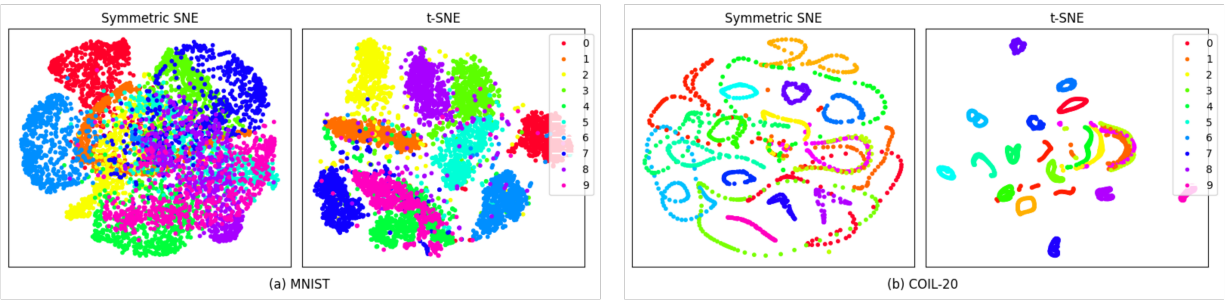


Figure 3: Symmetric SNE and t-SNE visualizations for the MNIST (left) and COIL-20 (right) datasets.

Our first ablation study verified the "crowding problem", which describes when visualization clusters crowd together in a plot. The original paper claims that sym-SNE suffers from the crowding problem, and that by using t-distributions in Equation 2b, t-SNE alleviates this problem. However, there are no plots of symmetric SNE in the paper, making it unclear whether sym-SNE suffers from this problem, or if t-SNE truly solves it. In all three of our experiments, we confirm that sym-SNE suffers from the crowding problem, and that the t-SNE method mitigates it. The results from the MNIST and COIL-20 datasets are shown in Figure 3 and the Olivetti Faces results are provided in Appendix A. We observe that the symmetric SNE map points force class clusters to overlap. Furthermore, t-SNE is clearly better at separating class clusters, while also still effectively showing intra-class variance. We conclude that the difference in sym-SNE and t-SNE distributions described in Equations 2a and 2b is an effective solution to the crowding problem.

3.2.2 Effect of PCA reduction

In our second ablation study, we considered the effect of PCA reduction on the solutions obtained from t-SNE, both quantitatively and qualitatively. When producing t-SNE visualizations, the paper PCA-reduces all of their datasets to 30 dimensions. Thus, they obtain an approximate attraction matrix $\mathbf{P}_{\text{PCA}}$ when computing the attractions in Equation1, and subsequently find attractions $\hat{\mathbf{Q}}_{\text{PCA}}$ that minimize $D_{\text{KL}}(\mathbf{P}_{\text{PCA}}, \mathbf{Q})$ as opposed to $D_{\text{KL}}(\mathbf{P}, \mathbf{Q})$. In this study we investigate the effect of this PCA preprocessing step, both quantitatively and qualitatively.

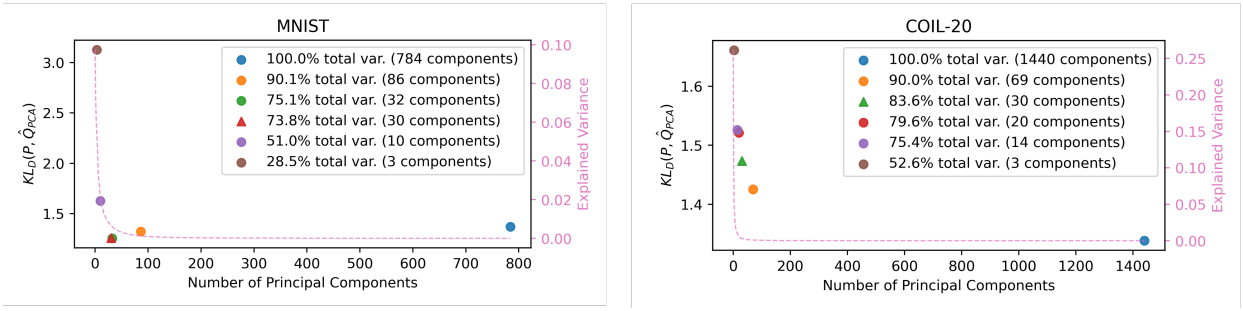


Figure 4: Effect of PCA reduction on the KL Divergence between $\mathbf{P}$ and $\hat{\mathbf{Q}}_{\text{PCA}}$ for the MNIST (left) and COIL-20 (right) datasets. Pink lines show the explained variance of each component (scale shown on the right axis). The triangle points (with 30 components) are the number of PCA components used in the original t-SNE paper.

Quantitative Analysis. First, we found that for these datasets, 30 PCs captured approximately 70%-85% of the total explained variance in the data. In Figure 4, we show the KL divergence between $\mathbf{P}$ (computed with from the un-reduced data) and $\hat{\mathbf{Q}}_{\text{PCA}}$ (obtained by first PCA-reducing the data and then performing t-SNE) for varying numbers of PCs used to compute $\mathbf{P}_{\text{PCA}}$. We chose PCs that captured approximately[4] 25%, 50%, 75%, 90% and 100% of the explained variance. Overall, we found that the KL divergence between $\mathbf{P}$ and $\hat{\mathbf{Q}}_{\text{PCA}}$ decreased with the number of PCAs kept (see Figure 4). This is consistent with the theory of PCA reduction seen in class, which highlighted that PCA reduction systematically removes the variance in the data [PS]. However, for the MNIST dataset, the relationship between $D_{\text{KL}}(\mathbf{P}, \hat{\mathbf{Q}}_{\text{PCA}})$ and explained variance was not a strictly monotone relationship; the PCA-reduced solutions with 30 and 32 PCs actually reduced the KL divergence *more* than solutions that did not preprocess the data. However, this non-monotonicity may be due to other non-linear effects that arise when using first-order optimization methods to minimize Equation 3 during t-SNE, which is not a convex objective function. For all datasets, using the PCs that explain 90% of the variance in the data seems to reduce the KL divergence sufficiently, while still being a manageable (<90) number of PCs.

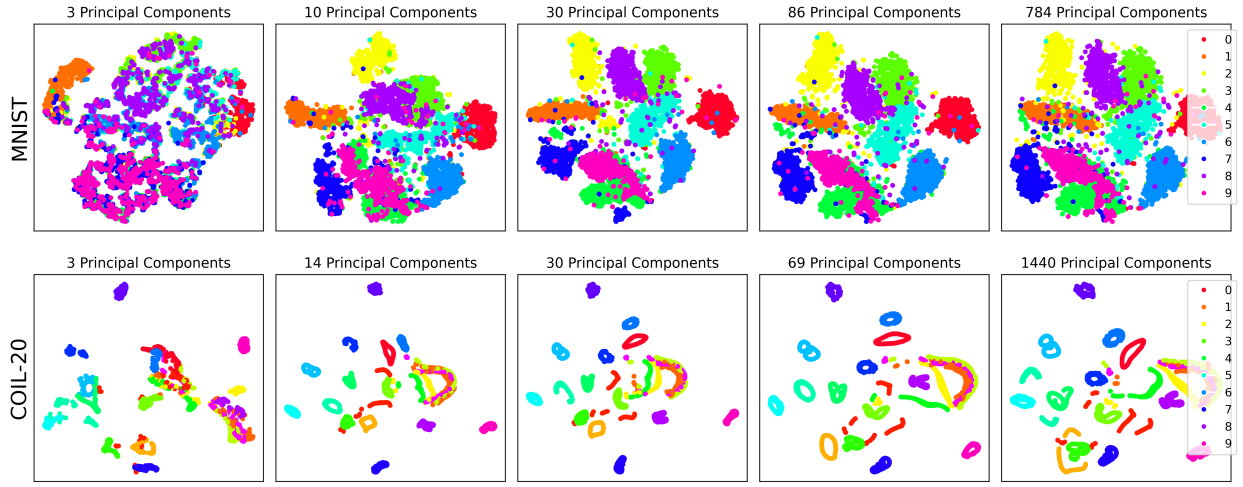


Figure 5: t-SNE plots for different PCA-reductions on the MNIST and COIL-20 datasets. The number of principal components correspond roughly to 25%, 50%, 75%, 90% and 100% of the explained variance (see the legend in Figure 4).

Qualitative Analysis. While our quantitative analysis identifies an interesting connection between PCA reduction and the KL divergence minmized during t-SNE, it does not immediately answer the question of which PC value is fundamentally "best" for visualizing data. Therefore, in we also plotted the solutions obtained from the different PCA reductions in Figures 5 and 8. Although 30 PCs typically finds a good visualization, we observe that the

[4]t-SNE requires a minimum of 3 components, which corresponds to more than 25% of the explained variance for the COIL-20 and Olivetti datasets.

visualization with 69 PCs (90% of the explained variance) for the COIL-20 dataset slightly outperforms that of the 30 PC visualization, with a few more local structures identified.

3.2.3 Generalization error for different classifiers.

The authors underline the strong performance of t-SNE by considering with the generalization error (measured using 10-fold cross validation) of a 1-nearest-neighbor classifier that is trained on the low-dimensional data [vdMH08]. This is compared to the error of a classifier trained on the original dataset. We wanted to investigate this in more detail and see how this generalization error behaves for other classifiers, as well as for the other dimensionality reduction techniques presented. Therefore, we trained different classifiers with the representation of LLE, Sammon and Isomap as well as for PCA (30 components). In addition to k-nearest-neighbor ($k \in \{1, 3, 5\}$), we used logistic regression and random forest as classifiers. The default parameters of sklearn were used for each of the classifiers. The results can be seen in Table 1.

Method	1-NN	3-NN	5-NN	Logistic Regression	Random Forest
PCA	0.055	0.056	0.056	**0.105**	0.078
Isomap	0.590	0.556	0.536	0.544	0.550
LLE	0.501	0.461	0.432	0.523	0.430
Sammon	0.634	0.609	0.583	0.556	0.599
t-SNE	**0.046**	**0.050**	**0.052**	0.157	**0.044**
Original	0.098	0.096	0.097	0.108	0.058

Table 1: Generalization error of different classifiers trained with different low-dimensional data representations.

The results once again demonstrate the superiority of t-SNE to other data-reduction methods. The generalization error is lowest for the t-SNE method for all classifiers except for Logistic Regression, where PCA is slightly ahead. The t-SNE generalization error significantly better (by roughly one order of magnitude) to Isomap, LLE and Sammon Mapping. This experiment further shows the potential of t-SNE not only as a visualization method but also as a general technique for dimensionality reduction, as it is able to transport local characteristics well across dimensions. However, it should be noted that t-SNE is particularly suitable for the reduction to 2 dimensions and the computational effort for each additional dimension increases significantly[vdMH08]. As a result, it may not be able to keep up with a technique such as PCA in higher dimensions.

4 Discussion & Conclusion

Our reproduction study showed that the authors' claims are largely correct and that t-SNE is superior to the compared dimensionality reduction methods for visualization. We confirmed that sym-SNE suffers from the crowding problem, while t-SNE does not. For PCA reduction, we observed a decrease in KL divergence with increasing number of PCs, with an interesting non-monotonic behavior in the MNIST dataset. For future practitioners using t-SNE, we suggest preprocessing data by using PCs that capture 90% of the explained variance, since this lead to both quantiative and qualitative improvements for two of the datasets. The result of our experiment regarding the generalization error for different classifiers underscores the efficacy of t-SNE, and shows that t-SNE visualizations reduce the generalization error accross a number of different classifiers.

What was easy/difficult. While the original paper itself was easy to follow, the mathematical details of the method in particular were a hurdle for us at the beginning. In particular, the fact that some concepts in computational geometry and manifold learning were presumed required some additional reading. However, the actual implementation of these methods was relatively simple, as libraries such as scikit-learn were used. Only the Sammon mapping and symmetric SNE algorithms required custom implementations. Contrary to the authors of that time, we had few problems with the fast execution of the experiments due to the computational resources available today.

Our analysis confirms the potential for t-SNE in data visualization. However, we also point out the difficulties of comparing and evaluating visualizations, since the end-goal is to obtain qualitatively "good" results. Further studies could investigate the effect of additional parameters in the t-SNE optimization routine in more detail. Future research should also investigate the scalability of t-SNE for larger data sets and compare it with other more recent techniques (e.g. UMAP [MHM18]).

Statement of Contribution All authors contributed to report writing. JT reproduced the original paper results and performed the third ablation study. JCC performed the first ablation study. RT and JCC performed the second ablation study.

References

[BLB+13] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[CM22] T Tony Cai and Rong Ma. Theoretical foundations of t-sne for visualizing high-dimensional clustered data. *The Journal of Machine Learning Research*, 23(1):13581–13634, 2022.

[Den12] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[MHM18] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[PS] Isabeau Prémont-Schwarz. Dimensionality reduction.

[RS00] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[Sam69] J.W. Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18(5):401–409, 1969.

[TdSL00] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[vdMH08] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[WVJ16] Martin Wattenberg, Fernanda Viégas, and Ian Johnson. How to use t-sne effectively. *Distill*, 1(10):e2, 2016.

A Crowding Problem for Olivetti Faces Dataset

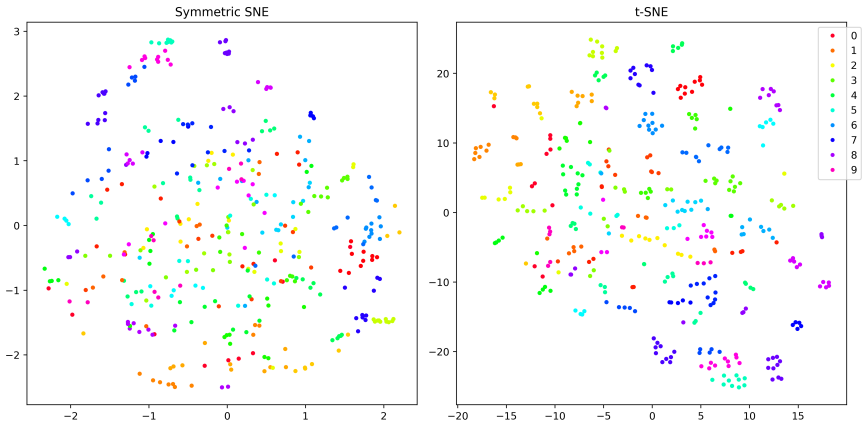


Figure 6: Crowding Problem for Olivetti Faces Dataset

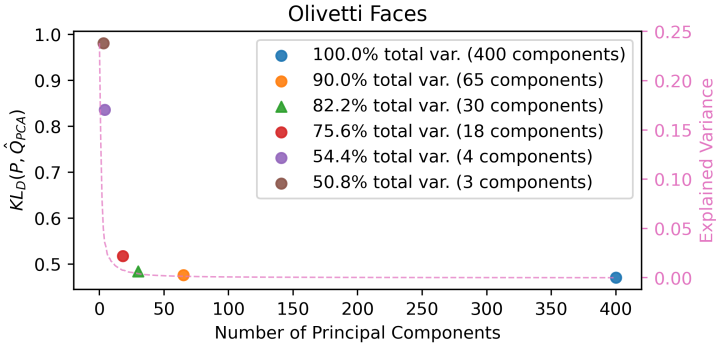


Figure 7: KL Divergence between attractions $\mathbf{P}$ and the attractions $\hat{\mathbf{Q}}$ found for different PCA-reduced data.

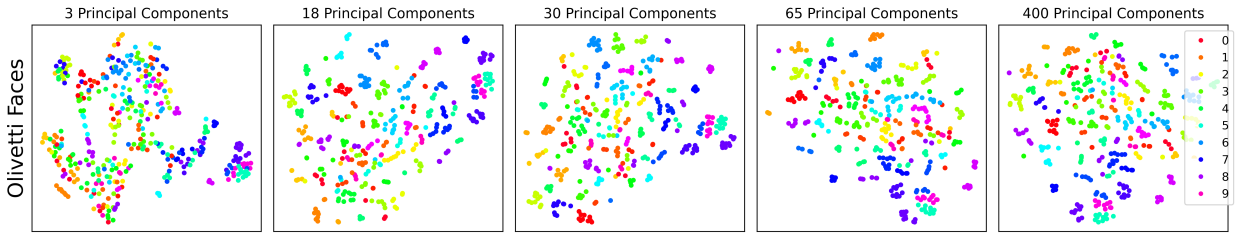


Figure 8: Plots for Olivetti Faces Dataset

B PCA Ablation Study for Olivetti Faces Data

C Visualizations of the MNIST data set

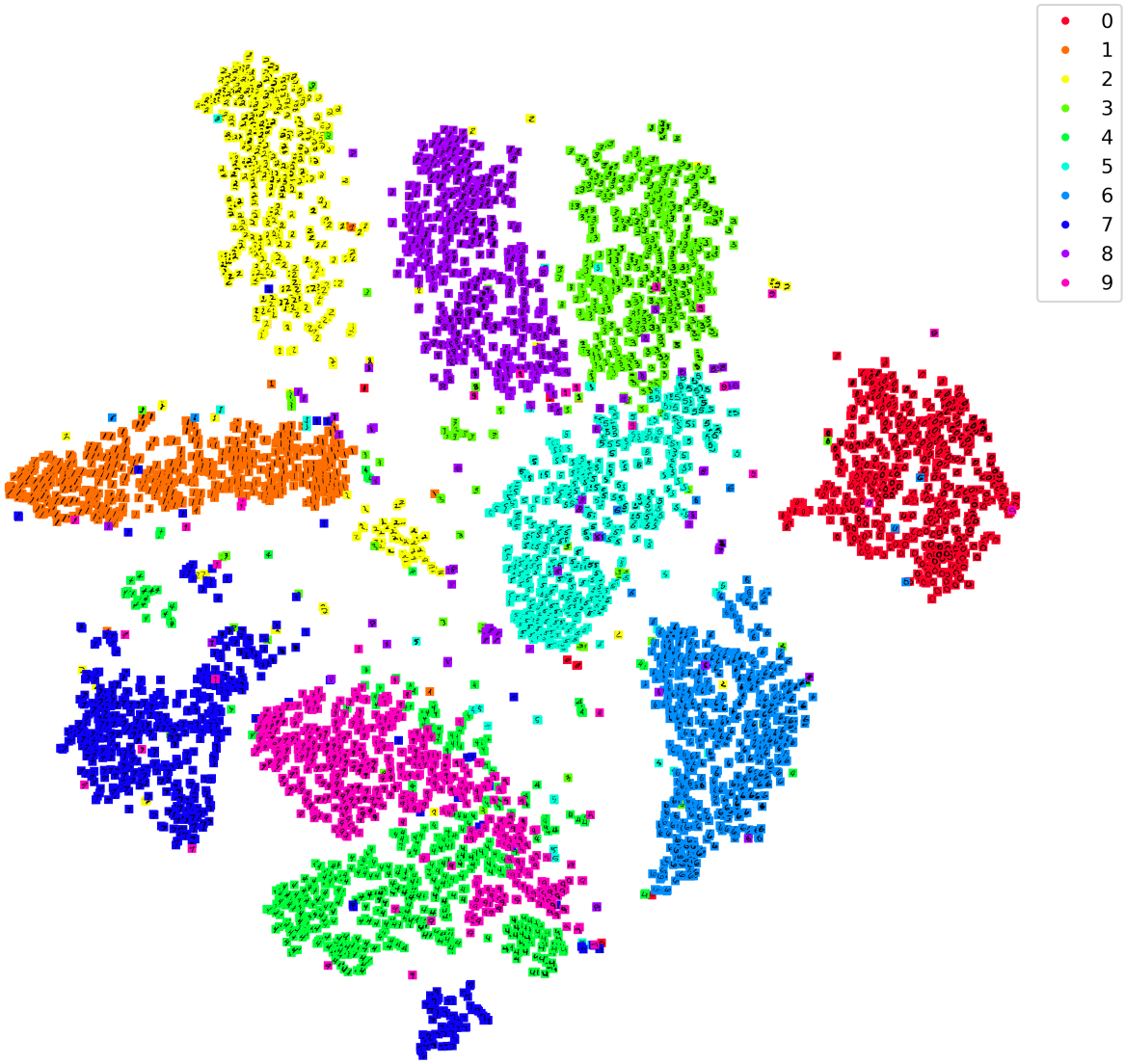


Figure 9: Visualization by t-SNE.

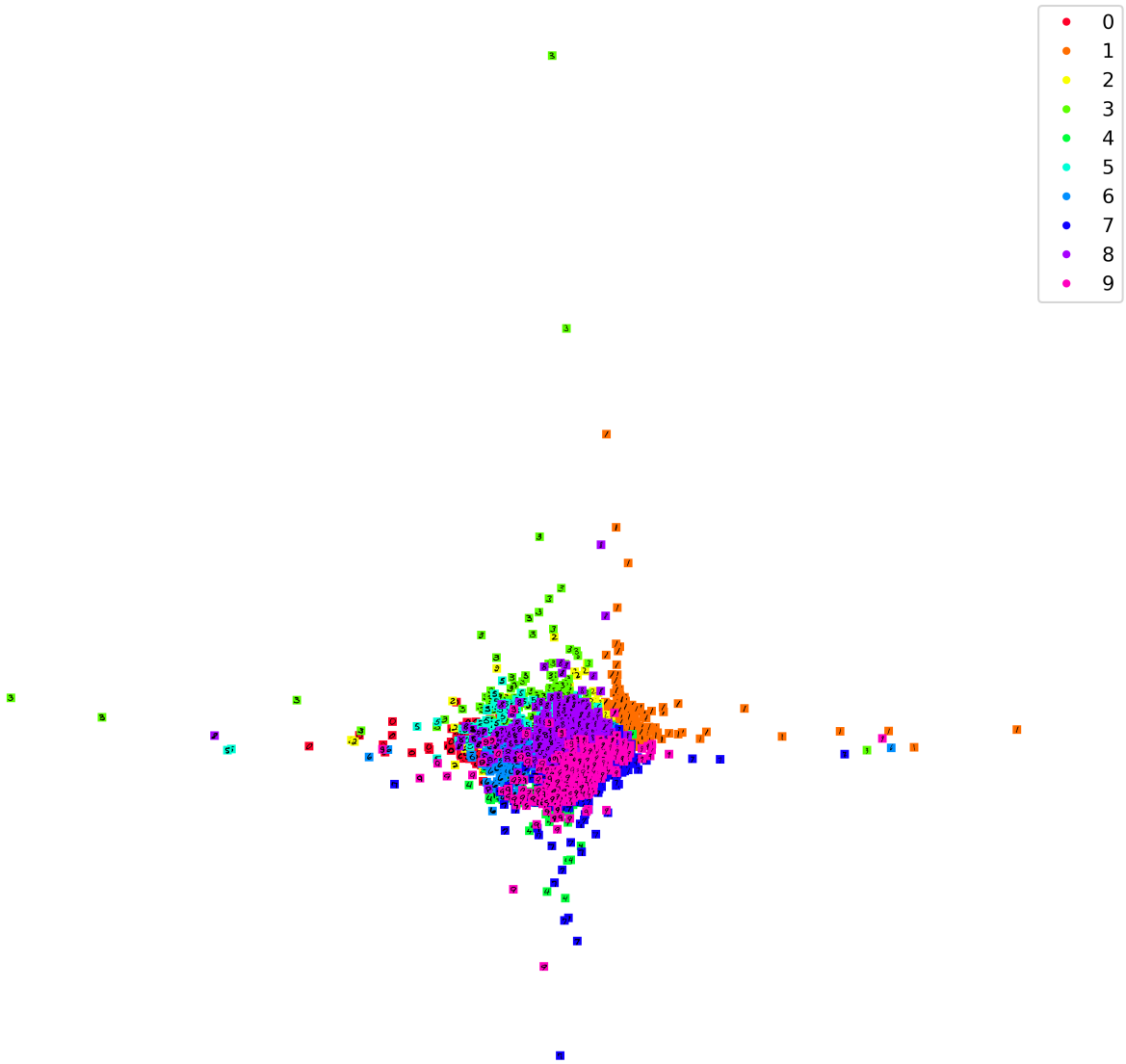


Figure 10: Visualization by Sammon Mapping.

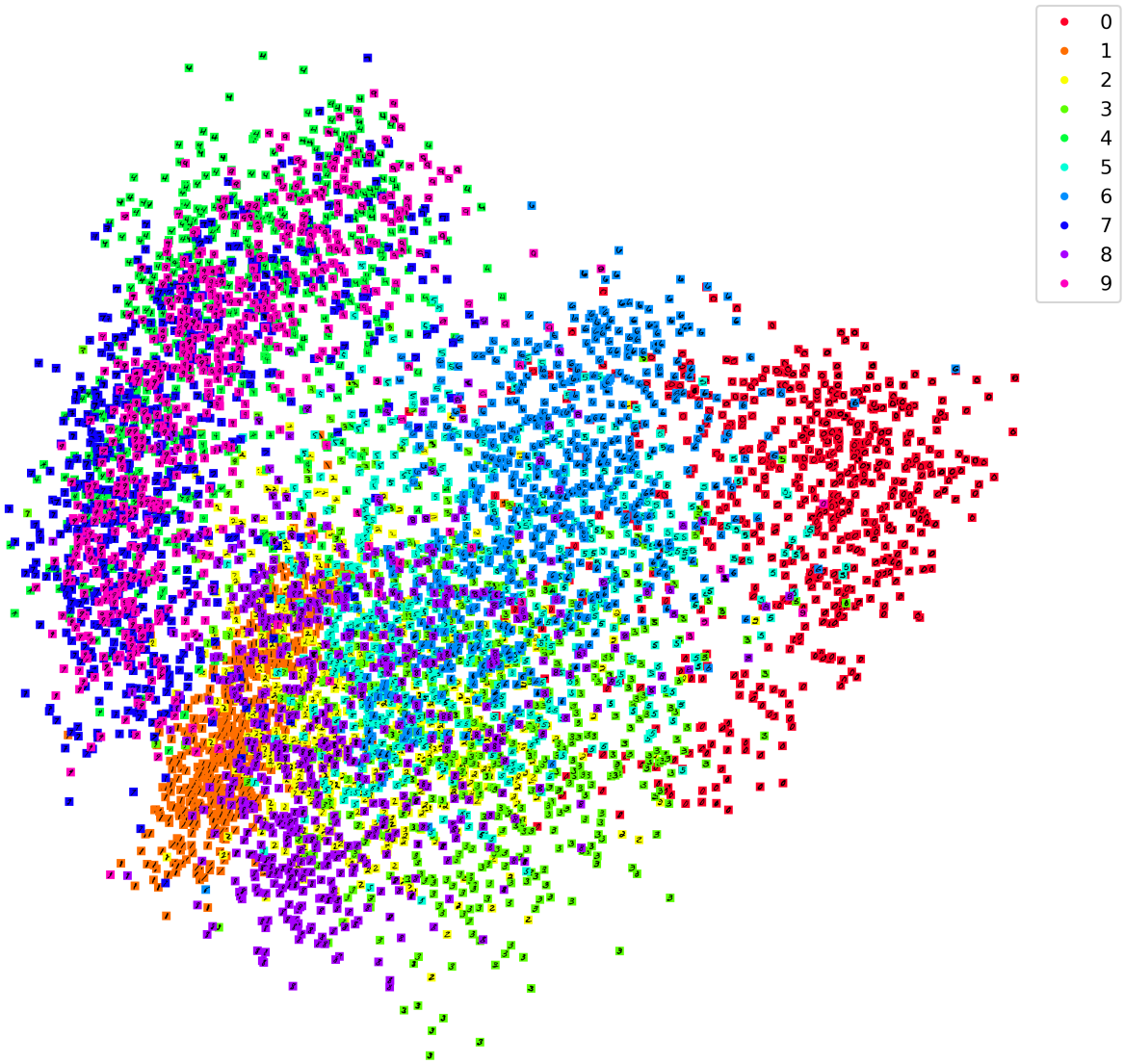


Figure 11: Visualization by Isomap.

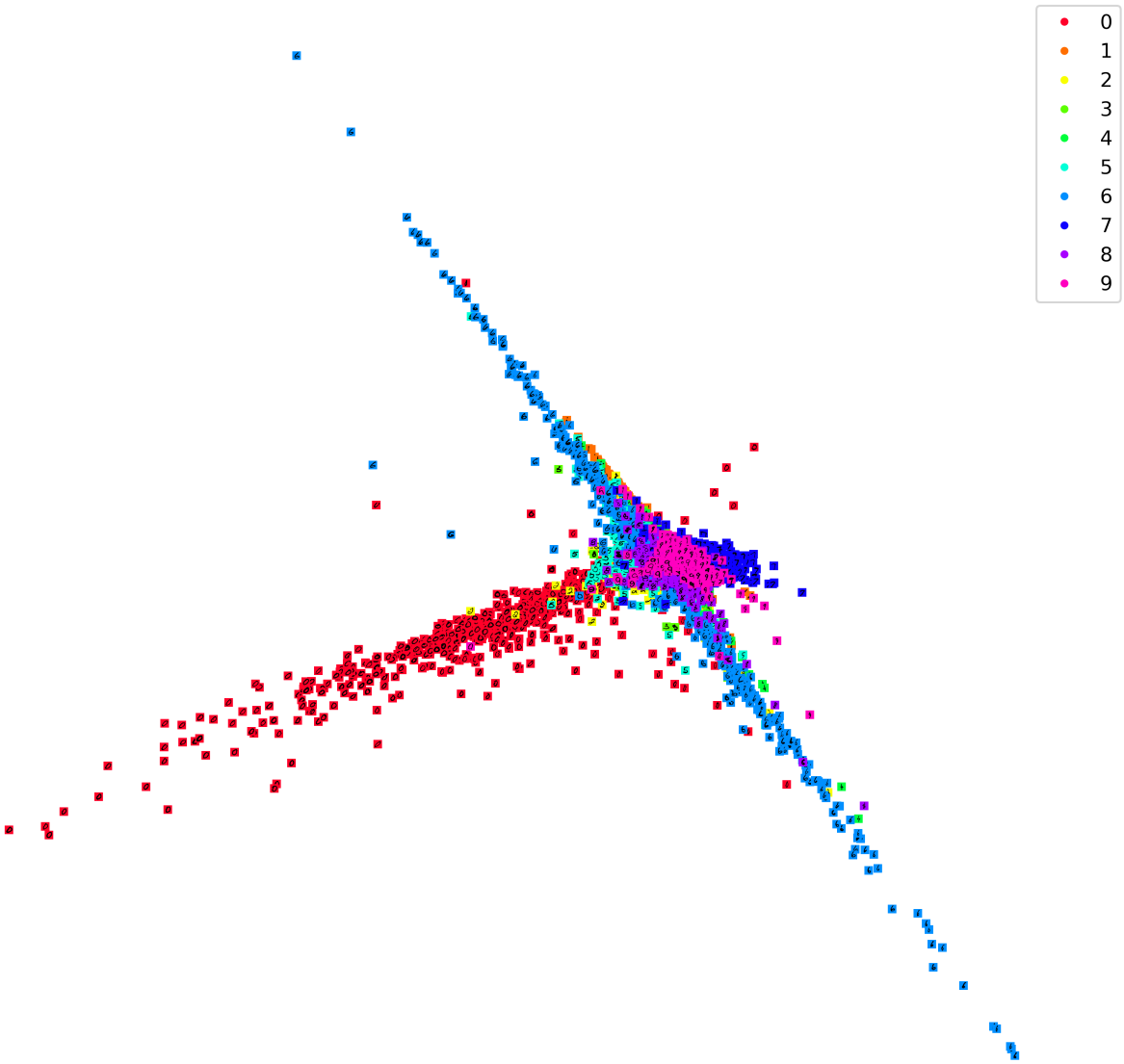


Figure 12: Visualization by LLE.

D Visualizations of the Olivetti Faces data set

(a) Visualization by t-SNE.

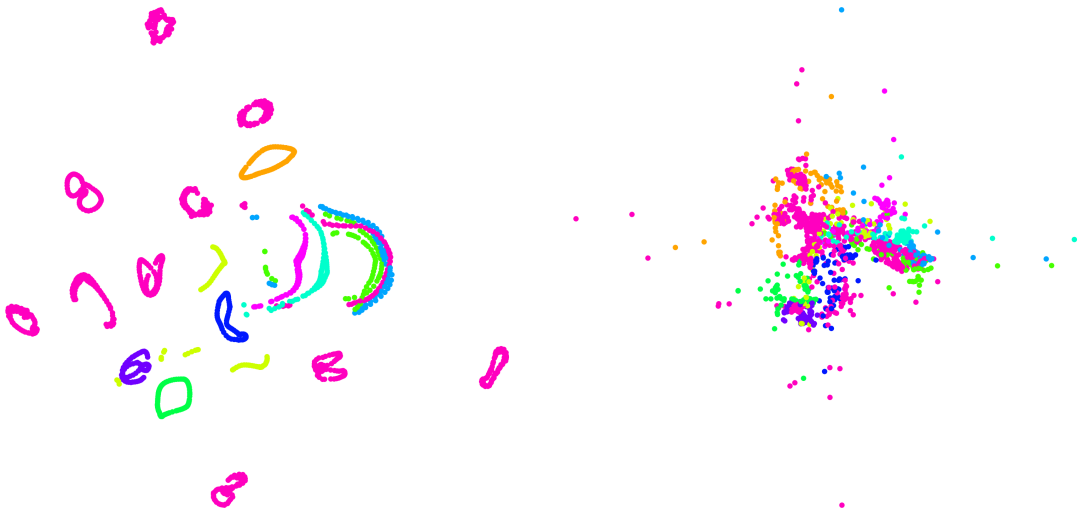
(b) Visualization by Sammon Mapping.

(c) Visualization by Isomap.

(d) Visualization by LLE.

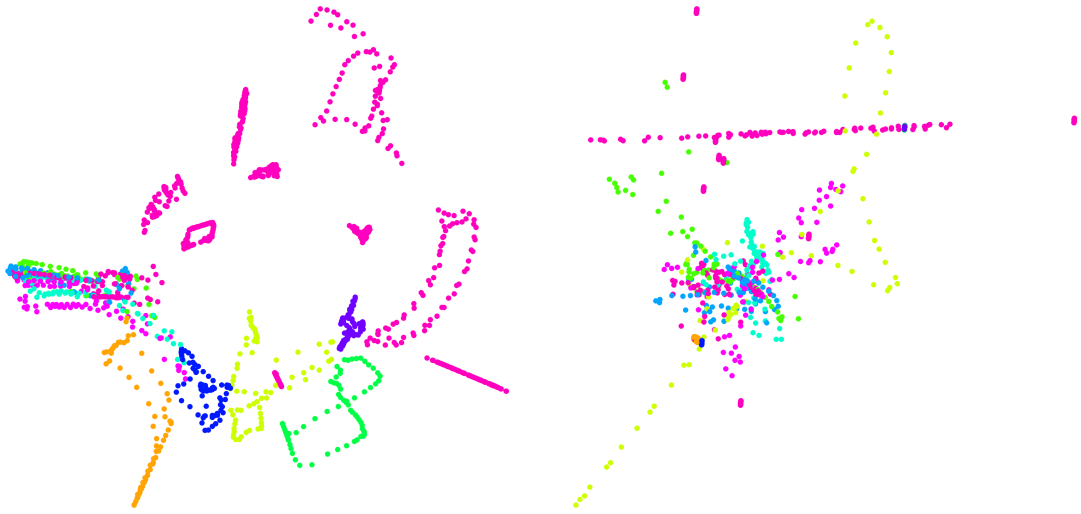
Figure 13: Visualizations of the Olivetti Faces dataset

E Visualizations of the COIL-20 data set



(a) Visualization by t-SNE.

(b) Visualization by Sammon Mapping.



(c) Visualization by Isomap.

(d) Visualization by LLE.

Figure 14: Visualizations of the COIL-20 dataset