Uniform Manifold Approximation with Two-phase Optimization
— Supplemental Material #1 (Appendices A to K) —

Hyeon Jeon *, Hyung-Kwon Ko *, Soohyun Lee, Jaemin Jo, and Jinwook Seo
*equal contribution

April 30, 2022

Content

This supplemental material presents more details about the following topics:

Appendix A: Pseudo Code of UMATO

The following is the Pseudo Code of the procedure described in Section 3.

Algorithm 1 Uniform Manifold Approximation with Two-phase Optimization (UMATO)

procedure UMATO(X, k_X, d, *min_dist*, n_h, e_g, e_l)
  Input: High-dimensional data X, number of nearest neighbors k, target dimension d,
    minimum distance in embedding *min_dist*, number of hub points n_h,
    epochs for global and local optimization e_g, e_l
  Output: Low-dimensional projection Y
  Compute k-nearest neighbors of X
  Obtain sorted list using indices' frequency of k-nearest neighbors
  Build k-nearest neighbor graph structure
  Classify points into hubs, expanded nearest neighbors, and outliers
  Optimize $CE(f(X_h)||g(Y_h))$ to preserve global configuration (Equation 4 in Section 2.1.)
  Initialize expanded nearest neighbors using hub locations
  Update k-nearest neighbors & compute weights (Equation 3)
  Optimize $CE(f(X)||g(Y))$ to preserve local configuration (Equation 8)
  Position outliers
  return Y

Appendix B: Hyperparameter Settings of DR Techniques in Experiment

As aforementioned in the document, we generated a projection for each dimensionality reduction technique that has the lowest $KL_{0.1}$ score for the fair comparison. The search was conducted in the following way.

- Isomap: Grid search
  - changed the number of nearest neighbors from 5 to 50 with an interval of 5
- t-SNE, At-SNE: Grid search
  - changed the perplexity from 5 to 50 with an interval of 5
  - changed the learning rate from 0.1 to 1.0 with a log-uniform scale
  - used the default setting for the other hyperparameters
- UMAP: Grid search
  - changed the number of nearest neighbors from 5 to 50 with an interval of 5
  - changed the minimum distance between points from 0.1 to 1.0 with an interval of 0.1
  - used the default setting for the other hyperparameters
- TopoAE: utilized scikit-optimize [2] library

In UMATO, we used the same setting for the hyperparameters that share with UMAP. For the original hyperparameters of UMATO, our setting is as follows:

- Number of hub points: 200 (for Spheres), 300 (for Real-world datasets)
- Number of epochs: 100 for global optimization and 50 for local optimization
- Global learning rate: 0.0065
- Local learning rate: 0.01

Note that we used fewer hub points for the Spheres since it has only 10,000 data points in total, while the other datasets have 60,000 data points.

Appendix C: Local Quality Metric Scores with Different k Values

In our main manuscript, we reported the score of local quality metrics (Trustworthiness & Continuity, Mean Relative Rank Errors) with hyperparameter $k = 5$. Here, we report the result with $k = 10$ and 15. As we can check from the table below, while the values fluctuate slightly, the ranks are mostly robust over diverse k values.

Dataset	Algorithm	Cont	Trust	$MRRE_X$	$MRRE_Z$	Dataset	Algorithm	Cont	Trust	$MRRE_X$	$MRRE_Z$
Spheres $k = 10$	PCA	0.7965	0.6111	0.7976	0.6089	MNIST $k = 10$	PCA	0.9519	0.7341	0.9574	0.7342
	Isomap	**0.8847**	0.6268	**0.8953**	0.6267		Isomap	0.9713	0.7531	0.9743	0.7530
	t-SNE	**0.8752**	**0.6803**	**0.8944**	**0.7082**		*t*-SNE	**0.9779**	**0.9930**	**0.9838**	**0.9951**
	UMAP	0.8688	0.6508	0.8764	0.6497		UMAP	**0.9858**	0.9543	**0.9889**	0.9545
	TopoAE	0.8309	0.6354	0.8312	0.6335		TopoAE	0.9686	0.9429	0.9716	0.9429
	A*t*-SNE	0.8645	0.6453	0.8724	0.6434		A*t*-SNE	0.9688	0.9734	0.9782	0.9761
	UMATO	0.7875	**0.6564**	0.7881	**0.6559**		UMATO	0.9753	0.8422	0.9792	0.8422
Spheres $k = 15$	PCA	0.7952	0.6120	0.7969	0.6094	MNIST $k = 15$	PCA	0.9481	0.7341	0.9555	0.7342
	Isomap	**0.8774**	0.6282	**0.8914**	0.6271		Isomap	0.9692	0.7529	0.9732	0.7529
	t-SNE	**0.8668**	**0.6723**	**0.8891**	**0.7021**		*t*-SNE	**0.9746**	**0.9908**	**0.9819**	**0.9940**
	UMAP	0.8639	0.6534	0.8737	0.6506		UMAP	**0.9834**	0.9542	**0.9877**	0.9545
	TopoAE	0.8304	0.6364	0.8309	0.6339		TopoAE	0.9666	0.9429	0.9705	0.9429
	A*t*-SNE	0.8603	0.6461	0.8699	0.6438		A*t*-SNE	0.9642	0.9707	0.9755	0.9749
	UMATO	0.7875	**0.6568**	0.7879	**0.6560**		UMATO	0.9728	0.8417	0.9778	0.8420
Fashion MNIST $k = 10$	PCA	0.9827	0.9120	0.9843	0.9116	Kuzushiji MNIST $k = 10$	PCA	0.9313	0.7218	0.9382	0.7214
	Isomap	0.9854	0.9196	0.9865	0.9196		Isomap	0.9546	0.7639	0.9574	0.7636
	t-SNE	0.9873	**0.9936**	0.9903	**0.9948**		*t*-SNE	0.9794	**0.9677**	0.9844	**0.9687**
	UMAP	**0.9941**	0.9586	**0.9950**	0.9585		UMAP	**0.9861**	0.9556	**0.9891**	0.9561
	TopoAE	**0.9898**	0.9591	0.9908	0.9590		TopoAE	0.9732	0.9442	0.9755	0.9441
	A*t*-SNE	0.9792	**0.9843**	0.9829	**0.9846**		A*t*-SNE	0.9723	**0.9659**	0.9789	**0.9670**
	UMATO	0.9897	0.9498	**0.9911**	0.9500		UMATO	**0.9836**	0.8880	**0.9864**	0.8890
Fashion MNIST $k = 15$	PCA	0.9815	0.9121	0.9837	0.9117	Kuzushiji MNIST $k = 15$	PCA	0.9266	0.7220	0.9358	0.7215
	Isomap	0.9838	0.9197	0.9858	0.9196		Isomap	0.9497	0.7640	0.9553	0.7636
	t-SNE	0.9858	**0.9894**	**0.9944**	**0.9927**		*t*-SNE	0.9758	**0.9668**	0.9825	**0.9683**
	UMAP	**0.9934**	0.9585	**0.9946**	0.9584		UMAP	**0.9836**	0.9551	**0.9879**	0.9558
	TopoAE	**0.9892**	0.9591	0.9904	0.9590		TopoAE	0.9715	0.9442	0.9746	0.9441
	A*t*-SNE	0.9769	**0.9840**	0.9817	**0.9845**		A*t*-SNE	0.9678	**0.9648**	0.9765	**0.9665**
	UMATO	0.9888	0.9495	0.9906	0.9499		UMATO	**0.9814**	0.8876	**0.9853**	0.8887

Table 1: Local quality metrics scores of UMATO and the baseline techniques. Although the values are changing slightly depending on the number of nearest neighbors k, the ranks barely change. The winner is in bold and underlined, and the runner-up is in bold.

Appendix D: Analysis of Embeddings' Stability Over the Number of Epochs

One of UMAP's critical problems is that embedding results are susceptible to the number of epochs; changing it can lead to the exaggerated or reduced distance between clusters (Figure 1, 2). As the proper number of epochs should be different for every dataset, users cannot be sure which value is appropriate and can readily make a wrong selection of epochs. This can cause a misinterpretation about the structure of data as users often interpret the distance between clusters in the embedding as their real dissimilarity [3]. The two-phase optimization of UMATO solves the problem since the global optimization (first phase) is easy to converge as it runs only with a small portion of points. Because we set the hub points not to be affected much by the points added in the local optimization, the second phase does not harm the distance between clusters.

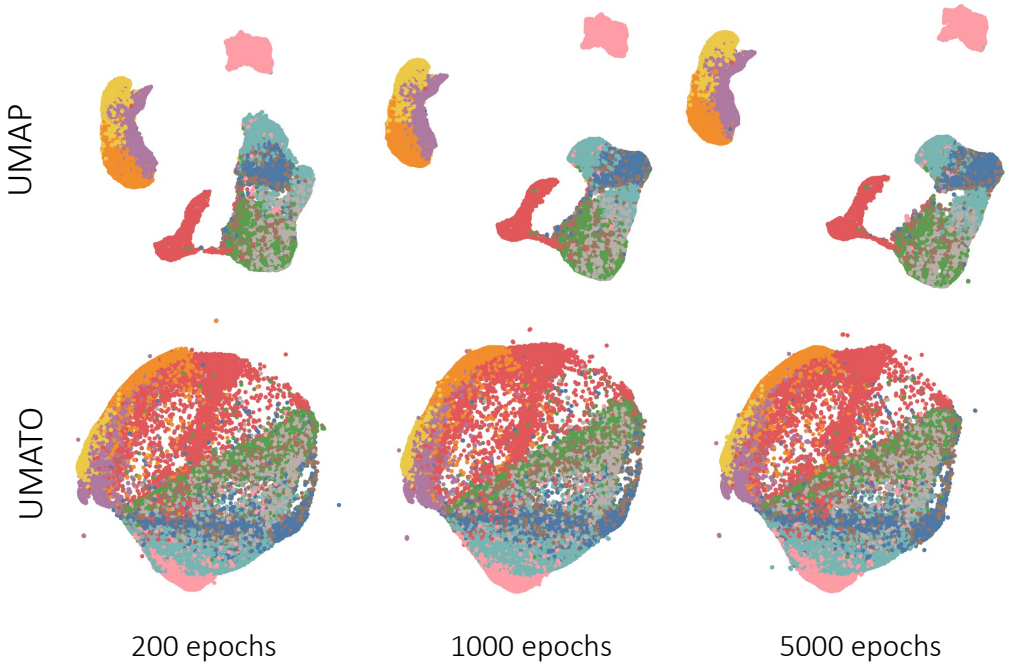

Figure 1: UMATO and UMAP embeddings with a varying number of epochs. (Top row) UMAP is susceptible to the number of epochs so that the clusters get dispersed as the epochs increase. (Bottom row) On the other hand, regardless of the number of epochs in the global optimization, UMATO results almost the same embedding result.

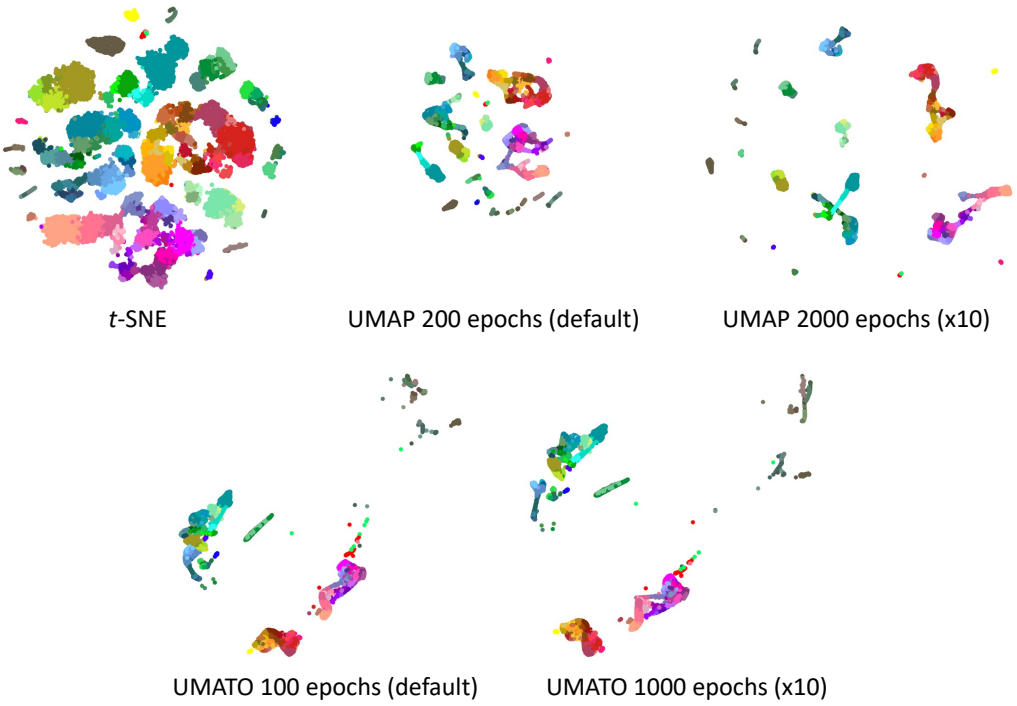

Figure 2: Embeddings of UMATO, UMAP, and *t*-SNE on mouse neocortex dataset [7]. Compared to *t*-SNE, UMAP and UMATO show a more clear cluster structure. However, in the case of UMAP, when the number of epochs is changed from the default value (200) to 2,000, the distance between clusters becomes exaggerated. UMATO generates more consistent embeddings regardless of the number of epochs, which does not induce any bias in analyzing inter-cluster distances.

Appendix E: Analysis of Embeddings' Stability Against Subsampling

When the data size grows, we usually want to sample a portion of it to speed up the embedding process. However, the concern is whether the projection that runs with the sampled points is consistent with the corresponding part of the embedding made with the entire dataset. The two projections should contain the least bias possible if the algorithm can generate stable and consistent results.

To compute the projection stability of dimensionality reduction techniques, we used the normalized Procrustes distance to measure the distance between two comparable distributions. We can test stability under subsampling by examining the normalized Procrustes distance between the embedding of a subsample, and the corresponding sub-sample of an embedding of the full dataset. Specifically, given two datasets $X = \{x_1, x_2, \dots, x_n\}$ and $Y = \{y_1, y_2, \dots, y_n\}$, we define the Procrustes distance between the two distributions as

$$d_P(X, Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i')^2}.$$

To compute Procrustes distances between embedding of subsample $Y' \subset Y$ and corresponding subsample of an embedding of entire dataset $X' \subset X$, we optimized Y' using translation, uniform scaling, and rotation to minimize the squared error $\sum_{i=1}^{n} (x_i' - y_i')^2$.

To replicate the experiment by Mcinnes et al. [5], we used the same Flow Cytometry dataset [6, 1], and ran optimal translation, uniform scaling, and rotation to minimize the Procrustes distance between the two distributions. As shown in the table below, UMATO outperformed *t*-SNE and A*t*-SNE for all sub-sample sizes. Moreover, although UMAP is known to be stable among existing algorithms, UMATO showed even better (lower) Procrustes distance except for one sub-sample size (60%). From this result, we can acknowledge that UMATO can generate more stable and consistent results regardless of sub-sample size than other dimensionality reduction techniques.

Sample (%)	1	2	5	10	20
t-SNE	0.9835	0.9944	0.9544	0.9736	0.9824
UMAP	0.4002	0.3319	0.2341	0.1324	0.1327
A*t*-SNE	0.8958	0.9510	0.7593	0.7980	0.9062
UMATO (ours)	**0.3153**	**0.1528**	**0.1206**	**0.0988**	**0.0520**
Sample (%)	30	50	60	80	100
t-SNE	0.9924	0.9959	0.9819	0.9944	0.9765
UMAP	0.1577	0.1109	**0.0713**	0.0951	0.0597
A*t*-SNE	0.9376	0.9999	0.9460	0.9599	0.9999
UMATO (ours)	**0.1411**	**0.0526**	0.1732	**0.0529**	**0.0535**

Table 2: The normalized Procrustes distance between the embedding of subsamples and the corresponding subsamples of the entire embedding. From four DR techniques, we measured the normalized Procrustes distance to check the projection stability using the Flow Cytometry dataset. The winner is in bold.

Appendix F: Analysis of Embeddings' Robustness Over Diverse Initialization Methods

We tested the robustness of UMATO, *t*-SNE, and UMAP over various initialization methods such as PCA, spectral embedding, random position, and class-wise separation. In class-wise separation, we initialized each class with a non-overlapping random position in 2-dimensional space, adding random Gaussian noise. Robustness was then measured as an average of Procrustes distances between embeddings with different initialization methods. The results (Table 3) show that UMATO is more robust against diverse initialization methods compared to *t*-SNE and UMAP. Although it did not outperform the competitors for Spheres dataset, as we can see in Figure 3, the global and local structures of the Spheres data can be manifested by UMATO regardless of different initialization methods.

Algorithm	Spheres	MNIST	FMNIST	KMNIST
t-SNE	0.7878	0.8665	0.8284	0.8668
UMAP	**0.7726**	0.7767	0.7793	0.8213
UMATO (ours)	0.9504	**0.4808**	**0.0120**	**0.2037**

Table 3: The average value of the normalized Procrustes distance between diverse dimensionality reduction techniques over four datasets. UMATO has shown the most robust embedding results over different initialization methods in all real-world datasets. Although the UMATO results in the highest normalized Procrustes distance in the Spheres dataset, the embedding results look quite similar (Figure 3). The winner is in bold.

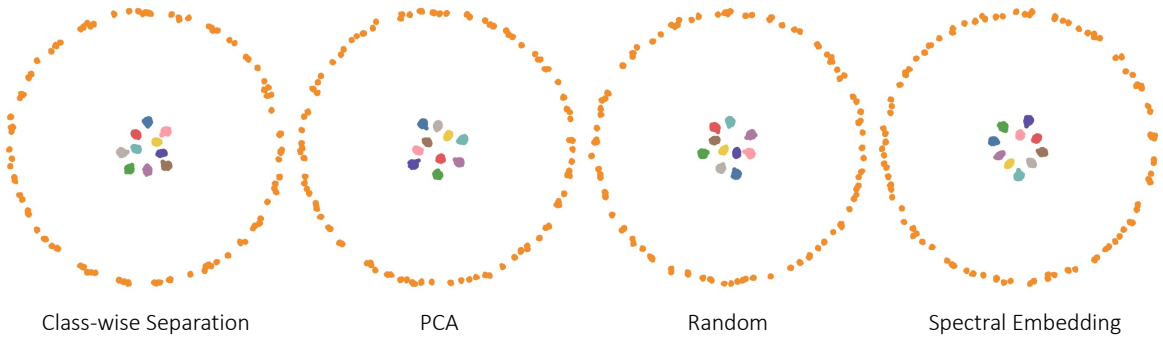

Figure 3: UMATO results on the Spheres dataset using different initialization methods. Although the average value of the normalized Procrustes distance of UMATO results is higher than the baselines, both global and local structures are well-captured with all different initialization methods. Best viewed in color.

Appnedix G: Runtime Analysis

We compared the running time of UMATO dataset against three nonlinear DR techniques (UMAP, *t*-SNE, and Isomap) using MNIST dataset. Note that we used Multicore *t*-SNE [8] implementation for *t*-SNE. As a result, we found that the running time of UMATO was faster than *t*-SNE and Isomap but about three times slower than UMAP implementation [4]. We tested the techniques on a Linux server with 40-core Intel Xeon Silver 4210 CPUs.

Algorithm	Runtime (s)
Isomap	3 hours >
t-SNE	374.85 ± 11.38
UMAP	26.10 ± 3.97
UMATO (ours)	73.32 ± 8.39

Table 4: The runtime for each algorithm in embedding MNIST dataset. UMAP and UMATO take much less time than *t*-SNE. The runtimes are averaged over ten runs. Isomap took more than three hours to get the embedding result.

Appendix H: Multi-Phase Analysis

UMATO can be easily expanded to multiple phases (e.g., three or more phases). Since we have a recursive procedure to expand the nearest neighbors, we can insert the optimization process each time we expand the neighbors to create a multi-phase algorithm. However, our experiment using the Fashion MNIST dataset showed that there is no big difference between two-phase optimization and that with more than two phases.

We report the experimental result of multi-phase optimization (e.g., three and four-phase) using the Fashion MNIST dataset both quantitatively (Table 5) and qualitatively (Figure 4). As in Figure 4, there are no significant differences between the 2D projections, although some outliers are located in different places. In the quantitative results (Table 5), the original UMATO (with two phases) is the winner in $KL_{0.1}$, KL_1, and continuity but came last in other quality metrics. However, the qualitative difference between the original UMATO and the multi-phase optimizations was imperceptible. Therefore, we concluded that developing a multi-phase optimization for UMATO does not bring about any notable improvement in the projection result while consuming more computation.

Algorithm	$KL_{0.01}$	$KL_{0.1}$	KL_1	Cont	Trust
2 Phases (UMATO)	0.6852	**0.0342**	**0.0008**	**0.9911**	0.950
3 Phases	0.6546	0.0343	**0.0008**	0.9900	**0.9556**
4 Phases	**0.6533**	0.0359	**0.0008**	0.9895	0.9532

Table 5: Quantitative evaluation of UMATO and UMATO with multi-phase optimizations. Although the optimization process of UMATO can be simply expandable for multiple phases, no apparent distinctions are found in the results with different numbers of optimization phases. The winner is in bold.

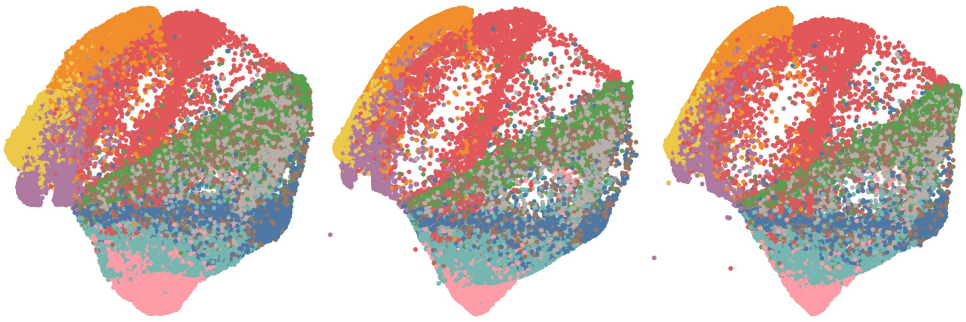

Figure 4: 2D embeddings of the Fashion MNIST dataset using UMATO with different levels of optimizations. Although there is a small difference, such as the locations of outliers, we observed that the embedding results were quite similar.

Appendix I: More Experiments on Synthetic Datasets

We leveraged the 3-dimensional S-curve and Swiss roll datasets to test whether UMATO can preserve both the global and local structures of original datasets. As the visualization shows, only PCA and UMATO were able to capture the global structures of original datasets. UMATO was also able to capture local manifolds. Isomap, *t*-SNE, and UMAP could capture the local manifolds of original datasets, but high-level global structures were not reflected.

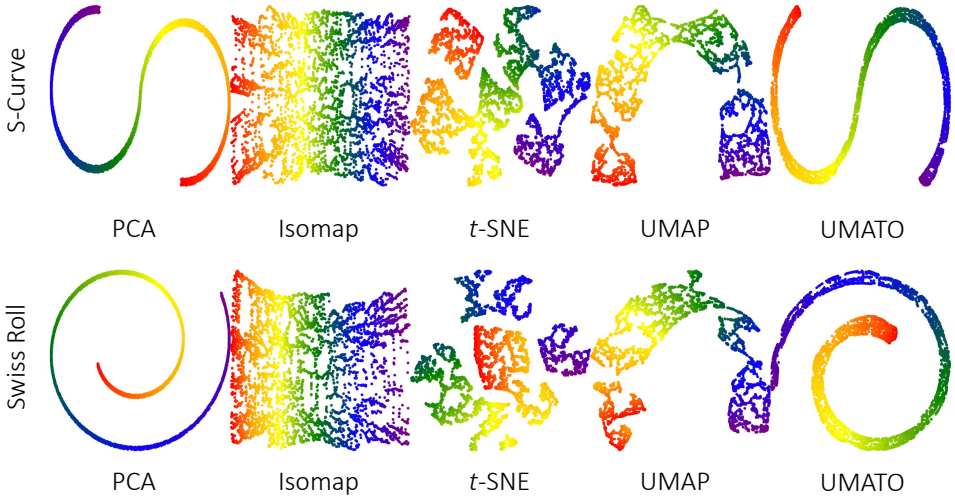

Figure 5: 2D embeddings produced by UMATO and four baseline algorithms. 3-dimensional S-curve and Swiss roll datasets are used for five different techniques.

Appendix J: A High-Resolution Figure of Embeddings used in the Quantitative Experiment

Refer to the end of this document (Figure 6) for the high-resolution image of embeddings used in the quantitative experiment (Section 4).

References

[1] Tess Brodie, Elena Brenna, and Federica Sallusto. Omip-018: Chemokine receptor expression on human t helper cells. *Cytometry Part A*, 83(6):530–532, 2013.

[2] Tim Head, MechCoder, Gilles Louppe, Iaroslav Shcherbatyi, fcharras, Zé Vinícius, cmmalone, Christopher Schröder, nel215, Nuno Campos, Todd Young, Stefano Cereda, Thomas Fan, rene rex, Kejia (KJ) Shi, Justus Schwabedal, carlosdanielcsantos, Hvass-Labs, Mikhail Pak, SoManyUsernamesTaken, Fred Callaway, Loïc Estève, Lilian Besson, Mehdi Cherti, Karlson Pfannschmidt, Fabian Linzberger, Christophe Cauet, Anna Gut, Andreas Mueller, and Alexander Fabisch. scikit-optimize/scikit-optimize: v0.5.2, March 2018.

[3] Hyeon Jeon, Hyung-Kwon Ko, Jaemin Jo, Youngtaek Kim, and Jinwook Seo. Measuring and explaining the inter-cluster reliability of multidimensional projections. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):551–561, 2022.

[4] Leland McInnes, John Healy, and James Melville. Umap. `https://github.com/lmcinnes/umap`, last accessed: 2020-02-20.

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

[6] Josef Spidlen, Karin Breuer, Chad Rosenberg, Nikesh Kotecha, and Ryan R Brinkman. Flowrepository: A resource of annotated flow cytometry datasets associated with peer-reviewed publications. *Cytometry Part A*, 81(9):727–731, 2012.

[7] Bosiljka Tasic, Zizhen Yao, Lucas T Graybuck, Kimberly A Smith, Thuc Nghi Nguyen, Darren Bertagnolli, Jeff Goldy, Emma Garren, Michael N Economo, Sarada Viswanathan, et al. Shared and distinct transcriptomic cell types across neocortical areas. *Nature*, 563(7729):72–78, 2018.

[8] Dmitry Ulyanov. Multicore-tsne. `https://github.com/DmitryUlyanov/Multicore-TSNE`, 2016.

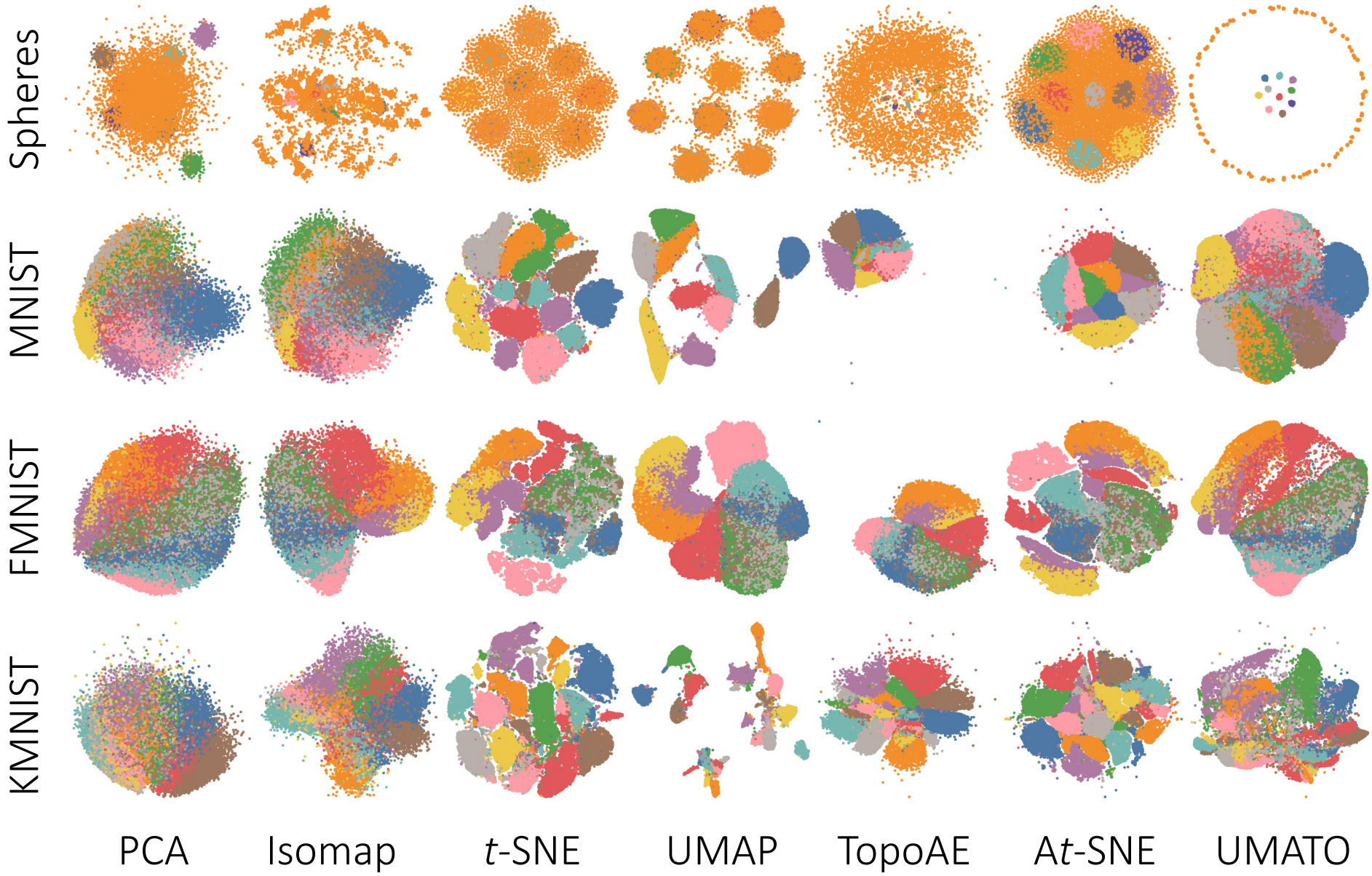

Figure 6: High-resolution image of embeddings used in our quantitative Experiment (Section 4)