EuroVis Workshop on Visual Analytics (2020)
K. Vrotsou and C. Turkay (Editors)

A Generic Model for Projection Alignment Applied to Neural Network Visualization

Gabriel D. Cantareira[1] and Fernando V. Paulovich[2]

[1]Universidade de São Paulo, Brazil
[2]Dalhousie University, Canada

Abstract

Dimensionality reduction techniques are popular tools for the visualization of neural network models due to their ability to display hidden layer activations and aiding the understanding of how abstract representations are being formed. However, many techniques render poor results when used to compare multiple projections resulted from different feature sets, such as the outputs of different hidden layers or the outputs from different models processing the same data. This problem occurs due to the lack of an alignment factor to ensure that visual differences represent actual differences between the feature sets and not artifacts generated by the technique. In this paper, we propose a generic model to align multiple projections when visualizing different feature sets that can be applied to any gradient descent-based dimensionality reduction technique. We employ this model to generate a variant of the UMAP method and show the results of its application.

1. Introduction

Dimensionality reduction (DR) techniques are mechanisms that embed complex data into low-dimensional spaces while still retaining meaningful relationships between objects, allowing for both computational and human interface operations to be conducted in a faster and more efficient manner. In data visualization and visual analytics, they are generally used to map multivariate data into 2D or 3D spaces, with the goal of exploration, observation, and comprehension of patterns and features.

Commonly called Multidimensional Projections in this context [NA18], these techniques have been used in many different domains and applications, such as document collection exploration, vector field analysis, and multimedia organization. Recently, the use of projections as a tool to aid the understanding of artificial neural network (ANN) models has become an active subject in the research community [RFFT17, RFT17, ERT19], with their use being adopted in several state-of-the-art VA approaches [PHVG*18, KAKC17] to analyzing hidden neural network information.

However, comparing projections generated by different feature sets representing the same data, such as outputs from multiple hidden layers of a neural network, present a few problems: attribute values and ranges may be widely different, and the number of attributes itself may not be the same. Furthermore, many sophisticated projection techniques are non-linear, meaning that there are no guarantees that small variations in the same feature set (e.g., small adjustments in neural network outputs from one training epoch to the next) will not result in widely different projections [WVJ16, RFT16].

While there are a few works dedicated to solving these issues, there are certain problems when generalizing their approaches, and there is no discussion over how to compare how projection techniques perform in the task of generating aligned embeddings of different feature sets so that their properties can be observed. Many projection techniques offer properties that are desirable for specific applications, but cannot be used in this context due to their inability to holding stable projections between observations.

In this short paper, we present a generic model that can be applied to any gradient descent-based projection technique to include an alignment factor that minimizes non-relevant variation between projections. Our model addresses issues that arise with this generalization, such as dimensionality differences and cost function variations. We apply this model to generate an extension of the well-known UMAP technique [MHM18] that can keep alignment over multiple feature sets, and show its functionality in neural network applications. This extension is shown to produce more reliable aligned results while maintaining desirable features of UMAP.

In short, the main contributions of this paper are:

- A generic flexible framework for supporting projection alignment with any gradient descent-based DR technique;
- A variant of the UMAP projection technique that performs alignment.

DOI: 10.2312/eurova.20201089

2. Related Work

Multidimensional Projections can be described as operations that receive an input $X = (x_1, x_2, ..., x_n) \in \mathbf{R}^m$ and produce a representation $X' \in \mathbf{R}^p, p < m$ as output. This representation is designed as to preserve dissimilarity relations δ_{ij} between points x_i and x_j as much as possible. Most visualization-related applications aim to embed data in the visual space, with $p = 2$ or $p = 3$.

There are many techniques to perform projection. Recent surveys [NA18, EMK*19] discuss each technique and classify them according to many parameters, such as linearity, scope of optimization (local or global), data type, or capability of supervision. Among the most known techniques, we can cite the classical MDS [Tor52], PCA [Hot33], LDA [Fis36], LSP [PNML08], LAMP [JCC*11], PLMP [PSN10], t-SNE [VDMH08].

Recent techniques, such as H-SNE [PHL*16] and UMAP [MHM18], aim to provide non-linear dimensionality reduction that preserves both local neighborhoods and global data structure. These techniques are shown to be of great use in data analysis applications, especially in machine learning. [PHVG*18, CAS*19]

One type of application for projections that have become very popular is in the analysis of (deep) neural networks [RFFT17, RFT17, ERT19, PHVG*18, KAKC17, CPE20]. One typical use of projection is to compare outputs from multiple hidden layers of a neural network, producing a different image for each layer. In this scenario, projections need to be aligned so they can be properly compared. The idea of *Projection Alignment* is to obtain multiple projections from the same number of feature representations of the same data in such a way that projected distances are as similar as possible to distances in the original feature sets while also keeping projections as similar as possible with one another.

Currently, only a few projection techniques support alignment natively, mostly being restricted to using the same initialization parameters [VDMH08] or a fixed set of control points [JCC*11]. To the best of our knowledge, the only techniques capable of actively aligning outputs during projection generation are the Dynamic t-SNE (Dt-SNE [RFT16]) and the Visual Feature Fusion (VFF [HP19]). These two are better detailed in the next section.

3. Aligning Projections

As previously discussed, alignment is essential when comparing projections in an analytical task. Figure 1 illustrates that, presenting an example of projections with and without alignment. We show projections from two feature sets, $f0$ and $f1$, that are produced from the activations of the last two hidden layers of a Convolutional Neural Network (CNN) from 2,000 samples from the MNIST [LBBH98] dataset. The upper row contains projections $p0$ and $p1$ obtained from the standard UMAP technique, and the lower row contains projections from our modified version of UMAP to preserve alignment (*Multi-feature UMAP*). We know beforehand that the two activation sets are quite similar. However, even if initialized with the same parameters, standard UMAP generates projections whose clusters are in different positions, and linear transformations, such as Procrustes analysis, cannot correct this mismatch. This is even more common in projection techniques that are less strict regarding inter-cluster distances, such as t-SNE. Techniques that solve this problem, and therefore form the basis of our generic alignment framework, are presented next.

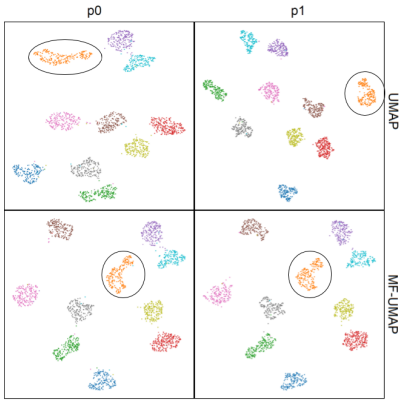

Figure 1: *Projections p_0 and p_1 of two feature sets using standard UMAP (no alignment) and our model (MF-UMAP). The feature sets correspond to activation data from two hidden layers of a CNN. These layers provide similar outputs, so the variation in the projections should be small. However, the unaligned projections cause groups to change places (e.g., the orange group) and may result in misinterpretation.*

3.1. Dynamic t-SNE

When projecting layer activation data from neural networks, Rauber et al. [RFFT17] discussed methods of reducing non-relevant variability in data projected with t-SNE, such as using point-cloud registration and using previous projections as initializations. The authors later proposed the Dt-SNE [RFT16], a variation of t-SNE designed to produce successive projections of a time-varying dataset while minimizing irrelevant projection variability and ensuring *temporal coherence*. This is achieved by adding a term to t-SNE's cost equation, as follows:

$$C_{dtsne}[t] = C_{tsne}[t] + \frac{\lambda}{2N} \sum_{i=1}^{N} \|p_i[t] - p_i[t+1]\|^2 \qquad (1)$$

In this equation, N is the number of data objects in each time step $t \in T$ to be projected, $C_{tsne}[t]$ is the standard t-SNE cost for projection t, $p_i[t]$ is the position of data instance i in projection t, and λ is a parameter that controls the trade-off between temporal alignment and fidelity to the standard t-SNE optimization.

The authors present a discussion on the different approaches to alignment, and their reasoning for using an additional cost function term is applicable to our method as well. While Dt-SNE was designed to align different versions of the same feature set, the algorithm itself offers no such restriction, but distances in each set need to be normalized and comparable. Its main limitation is its temporal focus: projections are only aligned concerning the ones immediately before or after them.

3.2. Visual Feature Fusion

Proposed by Hilasaca et al. [HP19], VFF itself is a system designed for adapting and fusing different feature representations of the same

data. During its user interaction phase, the system displays projections of a sample of the data according to each feature set, which needs to be aligned as to highlight differences in how they describe the data instances. To achieve this goal, the technique first generates a projection $\bar{p}$ using the mean pairwise distances for data points in all feature sets. Each projection t is then obtained by minimizing

$$C_{ff}[t] = (1-\lambda)\cdot C_{stress}[t] + \frac{\lambda}{N^2}\sum_{i}^{N}\sum_{j}^{N}\left(d(\bar{p}_i,\bar{p}_j) - \left\|\bar{p}_i - p_j[t]\right\|\right)^2 \quad (2)$$

where $C_{stress}[t]$ is the normalized stress [EMK*19] for projection t, $p_j[t]$ is the position of object j in projection t, $\bar{p}_i$ and $\bar{p}_j$ are the positions of objects j and i in $\bar{p}$, $d(\bar{p}_i,\bar{p}_j)$ is the distance between points i and j in $\bar{p}$, and λ is the trade-off controlling parameter.

While VFF is fully designed with the intent of merging different feature sets, its global stress-based optimization makes it not ideal for tasks that require neighborhood preservation and highlighting clusters. The alignment component is also more limiting as it attracts all points towards a previously set projection.

4. Generic Projection Alignment Model

Our generic model for projection alignment uses a cost function modification that can be included in any projection method that employs gradient descent-based optimization, such as the traditional MDS [Tor52], force schemes [Ead84, Cha96], SNE-based methods [VDMH08, IM15, PHL*16], among others. In our model, to make distances comparable between feature sets $F[t]$, we first apply the following normalization for data instance $F[t]_i$:

$$F'[t]_i = N\frac{F[t]_i}{\sum_{j=1}^{N}||F[t]_j||_2} \quad (3)$$

ensuring that distance between vectors is bounded, despite the scale of each feature set or the number of attributes they contain. We opt for the mean of $F[t]$ norms instead of maximum to avoid distortion by outliers that may appear during feature generation. After that, the projection optimization algorithm is executed, with the total cost function for each projection t given by

$$C[t] = (1-\lambda)\cdot C_{projection}[t] + \lambda\cdot\gamma\cdot C_{alignment}[t] \quad (4)$$

where $C_{projection}[t]$ is the standard cost function for the technique being used (e.g., t-SNE, UMAP, and so on), λ is the parameter that controls how much the alignment interferes in the projection, and γ is a scaling parameter. $C_{alignment}[t]$ is the penalty for moving away from other projections, given by

$$C_{alignment}[t] = \frac{1}{2N}\sum_{i=1}^{N} d(p_{i[t]},\bar{p}_i) \quad (5)$$

where $d(p_{i[t]},\bar{p}_i)$ is the distance between the point p_i in the projection t and the mean position of the same point between all projections $\bar{p}_i$. When Euclidean distance is employed, the differentiation of $C_{alignment}$ is simple and has a low computing cost, resulting in the following gradient matrix to update the projection $p[t]$

$$\nabla C[t] = (1-\lambda)\nabla C_{projection}[t] + \lambda\gamma[(p[t]-\bar{p})] \quad (6)$$

The parameter γ was added as a way to compensate for the difference in magnitude between cost function values in different projection techniques and bring the norms of projection and alignment gradients to a similar scale, increasing comparability between techniques and avoiding a trial-and-error process in finding an appropriate λ value. We set it to $||\nabla C_{projection}[t]|| \;/\; ||\nabla C_{alignment}[t]||$ so the two gradients always have the same norm and λ can explicitly control the ratio between forces in each point. It can, however, be fixed to $\gamma = 1$, just requiring the user to bear in mind that the projection's behavior according to λ will likely be unique for each case.

4.1. Multiple Feature UMAP (MF-UMAP)

UMAP has been increasingly adopted for visualization in data analysis tasks in recent works due to its ability to better represent global distance relationships (smaller *stress* values) when compared to t-SNE, while retaining similar properties [MHM18]. Its optimization is based on minimizing the cross-entropy between neighborhood probability distributions in projected and original spaces.

These properties make UMAP a desirable approach to projecting multiple feature set data, especially relating to neural network visualization: for instance, global distances contain information such as similarity between classes, or groups that get closer or further apart in different observations. Therefore, we chose this technique to apply the general alignment model and evaluate its capabilities. The implementation was done in Python, using scikit-learn, scipy and numpy libraries, following the UMAP algorithm proposed in [MHM18]. As the extra terms have low computational cost compared to the projection optimization itself, UMAP's performance does not suffer a significant impact.

5. Results

To verify how the aforementioned concepts can be employed in machine learning visualization, we performed experiments projecting activations from hidden layers of ANNs. We used CIFAR-10 [KH*09] and MNIST [LBBH98] datasets, projecting samples of 2,000 instances from test data. We used three architectures: a 4-layer MLP with 20 processing units per layer; a CNN with 2 convolutional layers, 2 dense layers (20, 20, 40, and 40 units respectively), Batch Normalization and Max Pooling 2×2 between the convolutional layers; the VGG16 [SZ14] model pre-trained on Imagenet dataset [DDS*09] without top layers, which were substituted by two layers of 1,024 dense units. All networks were trained using Adam [KB14] optimizers, with starting learning rate = 0.001.

First, we observe how knowledge is acquired between hidden layers. Figure 2 shows projections of activations from the 4 layers of the CNN over samples from the MNIST data set (convolutional layer activations were captured after the max-pooling layers). These layers have widely different output sizes (3,920 dimensions in the first layer and 40 in the fourth), but the resulting projections are comparable. The separation between certain classes is unclear in the first layer and gets more explicit in later ones. Observing label information, we notice classes that the network considers to be more similar, such as the numbers 4, 7, and 9 in the top region and 3, 5, and 8 at the bottom area. This information remains in the last projections, although with clearer separation.

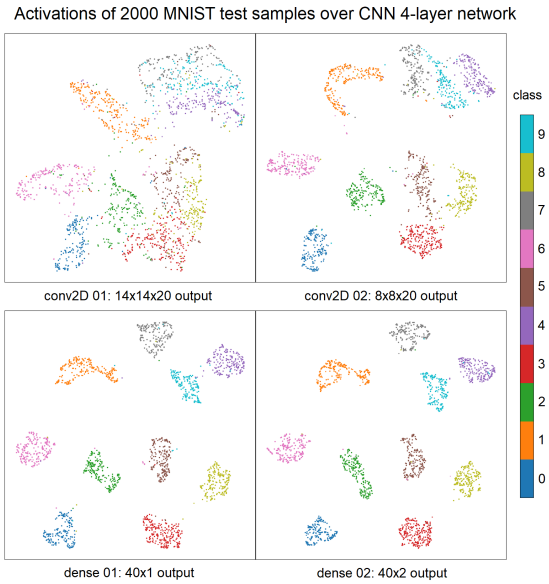

Figure 2: *Projections of activations of* 4 *sequential layers of a CNN. The shapes identified by the first two convolutional layers are similar for specific classes, as they are not fully separated. The last two organize this information and divide the space among classes.*

Our second experiment is to compare different ANNs. Figure 3 shows a comparison between the last layers of CNN and MLP models, using MNIST data activations. We included additional images of projections generated using Dt-SNE, to highlight a relevant aspect of UMAP-based alignment. Both networks are well-trained and capable of separating data at the last layer, but the MLP shows areas (A and B) where a few threads of connection remain between classes. Dt-SNE projection misses this information.

We calculated the centroid for each class and a distance matrix between all centroids in all projections, and then obtained the difference in centroid distances between the projections and the original feature data. In MF-UMAP projections, the resulting mean difference was $= 0.3688$, while the mean difference in the Dt-SNE projections was $= 0.5015$, showing that the first better preserves inter-group distances.

Finally, we compare VGG16 and CNN models. These models were both trained using the CIFAR-10 dataset and obtained similar results (accuracy scores of 74% and 70%, respectively). The VGG16 model was frozen and only had the new top layers trained. Figure 4 shows the resulting projections from the last hidden layers from both models. Both projections show a central area with mixed classes, representing instances with uncertain outputs. However, VGG16 shows more concentrated areas, and is able to isolate a higher amount of green, red, and brown points, indicating a better grasp of what composes these classes. This more specialized output can, however, lead to overfitting.

6. Conclusions

In this paper, we proposed a generic method for producing aligned projections of multiple feature sets, allowing the visualization to focus on the actual differences between each set. Our method improves on existing alignment solutions by supporting the flexibility needed by different projection techniques as they may offer advantages for specific tasks. We used this method to create a variation of UMAP that enabled alignment and showed its usefulness in neural network visualization applications. Future research directions may include developing a quantitative evaluation method to determine how well different techniques fit in the trade-off between projection fidelity and alignment, and how alignment interferes with other quality measures.

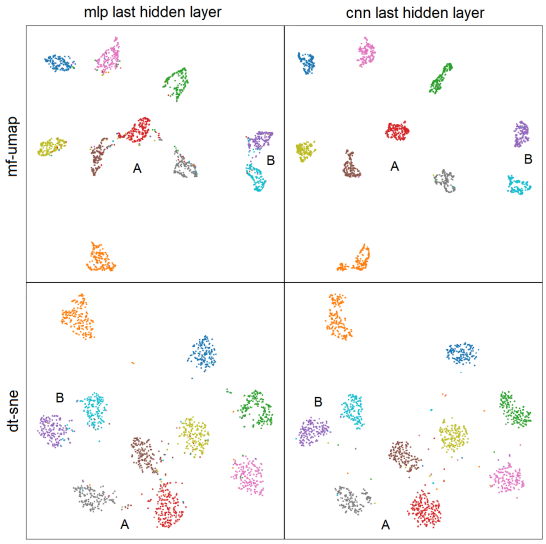

Figure 3: *Comparison between activations of the last hidden dense layers of MLP and a CNN using MF-UMAP and Dt-SNE. The CNN network separates data more explicitly, to the point of overfitting, but only MF-UMAP is capable of showing areas where separation improved (A and B).*

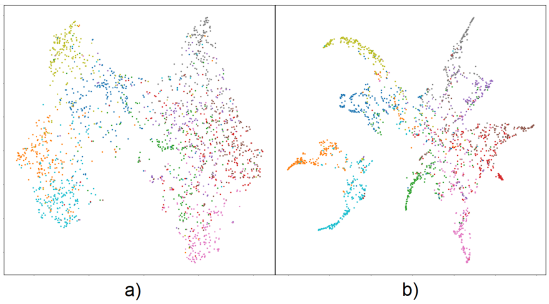

Figure 4: *Last hidden layer activations from two ANNs using CIFAR-10 data. a) CNN and b)VGG16. While their accuracy results are somewhat similar (*0.70 *and* 0.74*, respectively), there are clear differences in how each network perceives data, and how the extra complexity present in the VGG16 model can generate more refined, concentrated outputs.*

7. Acknowledgements

We would like to thank CAPES and FAPESP (2015/08118-6) for the financial support.

References

[CAS*19] CARTER S., ARMSTRONG Z., SCHUBERT L., JOHNSON I., OLAH C.: Activation atlas. *Distill* (2019). https://distill.pub/2019/activation-atlas. doi:10.23915/distill.00015. 2

[Cha96] CHALMERS M.: A linear iteration time layout algorithm for visualising high-dimensional data. In *Proceedings of Seventh Annual IEEE Visualization'96* (1996), IEEE, pp. 127–131. 3

[CPE20] CANTAREIRA G. D., PAULOVICH. F. V., ETEMAD. E.: Visualizing learning space in neural network hidden layers. In *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 3: IVAPP,* (2020), INSTICC, SciTePress, pp. 110–121. doi:10.5220/0009168901100121. 2

[DDS*09] DENG J., DONG W., SOCHER R., LI L.-J., LI K., FEI-FEI L.: ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09* (2009). 3

[Ead84] EADES P.: A heuristic for graph drawing. *Congressus numerantium 42* (1984), 149–160. 3

[EMK*19] ESPADOTO M., MARTINS R. M., KERREN A., HIRATA N. S., TELEA A. C.: Towards a quantitative survey of dimension reduction techniques. *IEEE Transactions on Visualization and Computer Graphics* (2019). 2, 3

[ERT19] ESPADOTO M., RODRIGUES F. C. M., TELEA A. C.: Visual analytics of multidimensional projections for constructing classifier decision boundary maps. In *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)* (2019), vol. 3, pp. 28–38. 1, 2

[Fis36] FISHER R. A.: The use of multiple measurements in taxonomic problems. *Annals of eugenics 7*, 2 (1936), 179–188. 2

[Hot33] HOTELLING H.: Analysis of a complex of statistical variables into principal components. *Journal of educational psychology 24*, 6 (1933), 417. 2

[HP19] HILASACA G., PAULOVICH F.: Visual feature fusion and its application to support unsupervised clustering tasks. *arXiv preprint arXiv:1901.05556* (2019). 2

[IM15] INGRAM S., MUNZNER T.: Dimensionality reduction for documents with nearest neighbor queries. *Neurocomputing 150* (2015), 557–569. 3

[JCC*11] JOIA P., COIMBRA D., CUMINATO J. A., PAULOVICH F. V., NONATO L. G.: Local affine multidimensional projection. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2563–2571. 2

[KAKC17] KAHNG M., ANDREWS P. Y., KALRO A., CHAU D. H. P.: Activis: Visual exploration of industry-scale deep neural network models. *IEEE transactions on visualization and computer graphics 24*, 1 (2017), 88–97. 1, 2

[KB14] KINGMA D. P., BA J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014). 3

[KH*09] KRIZHEVSKY A., HINTON G., ET AL.: *Learning multiple layers of features from tiny images*. Tech. rep., Citeseer, 2009. 3

[LBBH98] LECUN Y., BOTTOU L., BENGIO Y., HAFFNER P.: Gradient-based learning applied to document recognition. *Proceedings of the IEEE 86*, 11 (November 1998), 2278–2324. 2, 3

[MHM18] MCINNES L., HEALY J., MELVILLE J.: Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018). 1, 2, 3

[NA18] NONATO L. G., AUPETIT M.: Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment. *IEEE Transactions on Visualization and Computer Graphics* (2018), 1–1. doi:10.1109/TVCG.2018.2846735. 1, 2

[PHL*16] PEZZOTTI N., HÖLLT T., LELIEVELDT B., EISEMANN E., VILANOVA A.: Hierarchical stochastic neighbor embedding. In *Computer Graphics Forum* (2016), vol. 35, Wiley Online Library, pp. 21–30. 2, 3

[PHVG*18] PEZZOTTI N., HÖLLT T., VAN GEMERT J., LELIEVELDT B. P., EISEMANN E., VILANOVA A.: Deepeyes: Progressive visual analytics for designing deep neural networks. *IEEE transactions on visualization and computer graphics 24*, 1 (2018), 98–108. 1, 2

[PNML08] PAULOVICH F. V., NONATO L. G., MINGHIM R., LEVKOWITZ H.: Least square projection: A fast high-precision multidimensional projection technique and its application to document mapping. *IEEE Transactions on Visualization and Computer Graphics 14*, 3 (2008), 564–575. 2

[PSN10] PAULOVICH F. V., SILVA C. T., NONATO L. G.: Two-phase mapping for projecting massive data sets. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1281–1290. 2

[RFFT17] RAUBER P. E., FADEL S. G., FALCAO A. X., TELEA A. C.: Visualizing the hidden activity of artificial neural networks. *IEEE transactions on visualization and computer graphics 23*, 1 (2017), 101–110. 1, 2

[RFT16] RAUBER P. E., FALCÃO A. X., TELEA A. C.: Visualizing time-dependent data using dynamic t-sne. In *Proceedings of the Eurographics/IEEE VGTC Conference on Visualization: Short Papers* (2016), Eurographics Association, pp. 73–77. 1, 2

[RFT17] RAUBER P. E., FALCAO A. X., TELEA A. C.: Projections as visual aids for classification system design. *Information Visualization* (2017). 1, 2

[SZ14] SIMONYAN K., ZISSERMAN A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014). 3

[Tor52] TORGERSON W. S.: Multidimensional scaling: I. theory and method. *Psychometrika 17*, 4 (1952), 401–419. 2, 3

[VDMH08] VAN DER MAATEN L., HINTON G.: Visualizing high-dimensional data using t-sne. journal of machine learning research. *J Mach Learn Res 9* (2008), 26. 2, 3

[WVJ16] WATTENBERG M., VIÉGAS F., JOHNSON I.: How to use t-sne effectively. *Distill 1*, 10 (2016), e2. 1