MNIST-Nd: a set of naturalistic datasets to benchmark clustering across dimensions

Editors: List of editors' names

Abstract

Driven by advances in recording technology, large-scale high-dimensional datasets have emerged across many scientific disciplines. Especially in biology, clustering is often used to gain insights into the structure of such datasets, for instance to understand the organization of different cell types. However, clustering is known to scale poorly to high dimensions, even though the exact impact of dimensionality is unclear as current benchmark datasets are mostly two-dimensional. Here we propose MNIST-Nd, a set of synthetic datasets that share a key property of real-world datasets, namely that individual samples are noisy and clusters do not perfectly separate. MNIST-Nd is obtained by training mixture variational autoencoders with 2 to 64 latent dimensions on MNIST, resulting in six datasets with comparable structure but varying dimensionality. It thus offers the chance to disentangle the impact of dimensionality on clustering. Preliminary common clustering algorithm benchmarks on MNIST-Nd suggest that Leiden is the most robust for growing dimensions.

Keywords: benchmarking datasets; clustering; high dimensional space

1. Introduction

Modern datasets are often high-dimensional, especially with deep learning embeddings (Schroff et al., 2015; Weis et al., 2022; Douze et al., 2024) and advanced recording techniques in natural sciences, i.e. transcriptomics (Qiu et al., 2017; Wolf et al., 2019; Harris et al., 2018; Lause et al., 2024). To uncover their internal structure, dimensionality reduction or clustering is commonly used. While linear methods like PCA miss non-linear patterns, t-SNE (Van der Maaten and Hinton, 2008), UMAP (McInnes et al., 2018) or PHATE (Moon et al., 2019), are popular for visualization but sensitive to hyperparameters (Kobak and Berens, 2019; Kobak and Linderman, 2021), making conclusions based only on them challenging. Alternatively, clustering in the original space avoids information loss, but distinguishing distances or densities in high dimensions is difficult as pairwise distances become more alike (Johnstone and Titterington, 2009).

Many different clustering methods exist, but realistic benchmarks with non-uniform noise and high dimensional data are lacking. Instead, most benchmarking datasets are 2D or 3D (Karypis, 2002; Gagolewski, 2022; Thrun and Ultsch, 2020; Barton, 2015; Laborde et al., 2023). Simple real-world datasets with high-dimensional features are also used for clustering evaluation (Thrun and Ultsch, 2020), but they are not directly comparable across dimensionalities as different datasets have different internal structures. A few artificial datasets with variable dimensions exist, like multidimensional Gaussians (Gagolewski, 2022; Laborde et al., 2023; Sedlmair et al., 2012) worms (Sieranoja and Fränti, 2019), and DENSIRED (Jahn, 2023). However, they lack realistic non-uniform noise and don't scale variance with dimensionality, which makes cluster separation easier in high dimensions as clusters shrink and no longer overlap while overlapping density is common in real data.

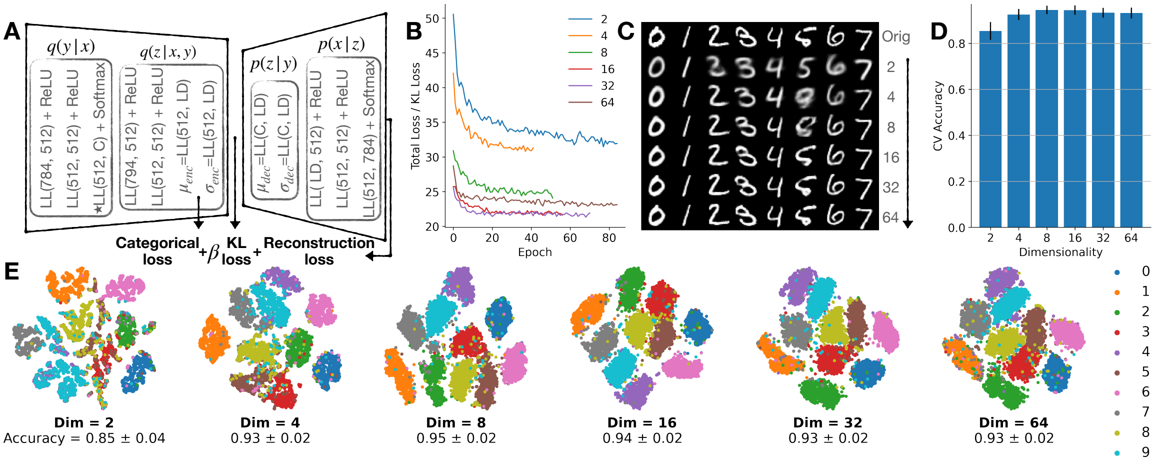

Figure 1: **A**: Mixture VAE architecture. LL: linear layer (+number of in and out channels). LD: latent dim. ⋆-layer: Gumbel softmax. It returns a one-hot-encoding to select the mixture component. The next layer concatenates it to the original input (Appendix A). **B**: Total loss divided by KL-loss is similar across dimensions. KL: Kullback-Leibler Divergence. **C**: Reconstruction examples. **D**: Cross-validated random forest accuracy (±SD). **E**: t-SNE visualizations of the embeddings.

To overcome these issues, we propose MNIST-Nd, a dataset of embeddings from a mixture variational autoencoder (m-VAE) (Dilokthanakul et al., 2016), trained on MNIST (LeCun, 1998). MNIST-Nd has realistic noise as it appears in learned embeddings and controllable dimensions while maintaining consistent signal-to-noise ratio across dimensions. We benchmark common clustering algorithms (k-means, GMM, TMM, Leiden) on MNIST-Nd with respect to their performance and robustness across dimensions and find that Leiden clustering (Traag et al., 2019) significantly outperforms other methods in higher dimensions for both performance and robustness.

2. Dataset Creation

To create the datasets, we use a mixture VAE (m-VAE; Fig. 1 A; Dilokthanakul et al., 2016), where the prior is a mixture of Gaussians with ten components, which biases the latent space to have ten density modes. We use the β-VAE framework (Higgins et al., 2017) to scale the importance of the Kullback-Leibler (KL) loss. As the KL loss is unbounded and grows with the number of dimensions, we scale β inversely proportional to the dimensionality such that all datasets are similarly regularized to match the prior shape. The ratio of the total loss to the KL term converges to similar values across latent dimensions (Fig. 1 B), suggesting that the impact of the KL loss is indeed comparable. The architecture of the autoencoder as well as all other training hyperparameters (except for β) are fixed. Once the different m-VAE models are trained, we encode the test set of MNIST (Fig. 1 C) to get MNIST-Nd and analyze its embeddings. A (ten-fold) cross-validated random forest classifier (Breiman, 2001) achieves comparable classification accuracy across dimensions ($\sim$ 90%, Fig. 1 D). Similiar

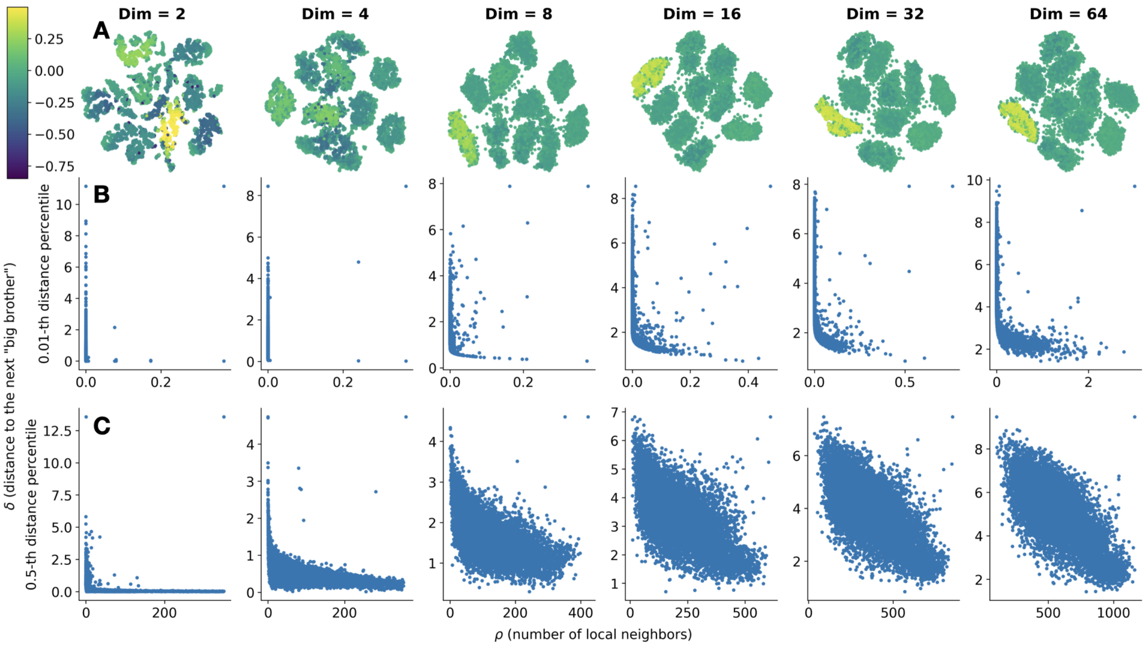

Figure 2: **A**: t-SNE colored with DISCO scores. **B-C**: Density peaks for different local radii. Outliers in the upper-right are the suggested density peaks center points.

accuracies suggest that our embeddings are indeed comparably separable across different dimensions, albeit t-SNE embeddings in higher dimensional datasets look somewhat more condensed (Fig. 1 E). The accuracies are not close to state-of-the-art classifiers by design: realistic biological datasets are imperfect and our datasets share this property.

Density modes and cluster overlap. As many real-world datasets have overlapping density between clusters, we want to ensure our toy datasets has it as well. To check this, we estimated DISCO scores (Anonymous, 2024) (Fig. 2A) and search for density peaks (Rodriguez and Laio (2014)) (Fig. 2B, C). DISCO scores are bounded between −1 and 1, where negative values imply points being better connected to a different cluster than to the assigned one. The majority of points of our embeddings are scored around zero or below, suggesting a noticeable density overlap. An alternative analysis is to count the density peaks. We follow Rodriguez and Laio (2014) and measure density through the number of neighbors ρ within a radius r hypersphere (using an exponential kernel, so values are not integers). Then for each point we compute δ, the distance to the closest point with more neighbors. This way, the outliers in the upper-right corner of the ρ-δ diagram are natural cluster centers as they have many local neighbors and are far away from other points with more local neighbors. The number of clear density modes is smaller than ten for all plots, indicating overlapping clusters. As distances in high-dimensional space tend to be more uniformly distributed, smaller thresholds reveal more density peaks in higher dimensions. We used different thresholds to ensure same qualitative results (Fig. 2 B–C, Appendix E).

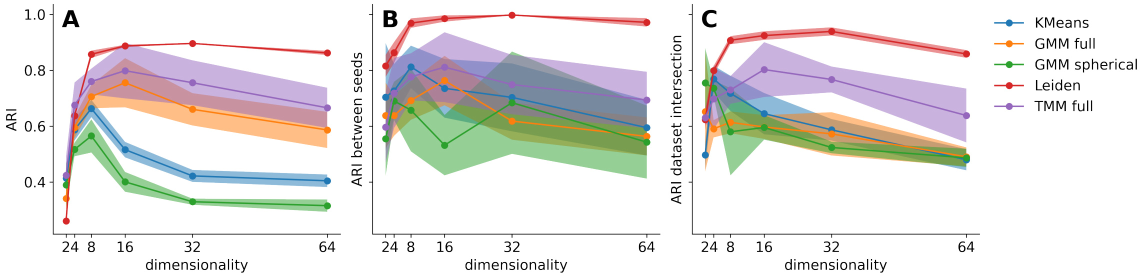

Figure 3: **A**: ARI with ground truth shows clustering performances. **B**: ARI across seeds measures the stability of the clustering methods for different initializations. **C**: ARI for bootstrapped datasets shows clustering robustness for data perturbations.

3. Evaluating clustering methods on MNIST-Nd

Next, we use MNIST-Nd to test the performance and robustness of different clustering algorithms. We choose k-means as an example for distance-based clustering, Gaussian and t-distribution mixture modelling (GMM and TMM) as density-based, and Leiden clustering as a graph-based clustering method. For evaluation, we use the adjusted rand index (ARI) (Hubert and Arabie, 1985), which measures the pairwise similarity of two cluster assignments. It is one when two partitions match exactly up to global label permutations. It is zero when the agreement between the two partitions is consistent with random assignments and negative when consistency is systematically below chance (Chacón and Rastrojo, 2023).

First, we compare the clustering performance by calculating the ARI between cluster predictions and ground truth labels across ten random seeds. The performance of the clustering algorithms decreases with growing dimensions except for Leiden clustering which does not seem to be affected as much (Fig. 3 A). Second, we evaluate the stability of methods by computing the ARI between all pairs of partitions from the previous step. While Leiden clustering remains stable, the ARI decreases with dimensions for the centroid-based methods (Fig. 3 B). Third, we measure robustness to data perturbations using bootstrapping. We create three datasets, each containing 60% of the original data. Of this, 40% is shared across all three datasets for ARI estimation, while 20% is unique to each dataset (Fig. 3 C). The datasets split is consistent across dimensions. GMMs and k-means show the biggest decay along dimensions, while TMM clustering appeared to be more robust. This is the only experiment where we see a decrease of Leiden ARIs for higher dimensions. For all methods ARI values decline with higher dimensions because more points are needed to confidently estimate distances and densities in high-dimensional space.

4. Conclusions and Limitations

We propose a framework to generate realistic noisy multidimensional datasets to isolate the impact of dimensionality on clustering performance. Benchmarking on MNIST embeddings shows that Leiden clustering outperforms other methods in higher dimensions, though this result needs validation on additional datasets. Future work may include generating embeddings for other datasets to explore varying internal structures and validate these findings.

References

Anonymous. Disco. https://anonymous.4open.science/r/DISCO-8E44/README.md, 2024.

Tomas Barton. Clustering benchmarks. https://github.com/deric/clusteringbenchmark, 2015.

Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

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.

José E Chacón and Ana I Rastrojo. Minimum adjusted rand index for two clusterings of a given size. *Advances in Data Analysis and Classification*, 17(1):125–133, 2023.

Nat Dilokthanakul, Pedro AM Mediano, Marta Garnelo, Matthew CH Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2016.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. 2024.

Marek Gagolewski. A framework for benchmarking clustering algorithms. *SoftwareX*, 20: 101270, 2022.

Jana Gauss, Fabian Scheipl, and Moritz Herrmann. Dcsi–an improved measure of cluster separability based on separation and connectedness. *arXiv preprint arXiv:2310.12806*, 2023.

Maria Halkidi and Michalis Vazirgiannis. Clustering validity assessment: Finding the optimal partitioning of a data set. In *Proceedings 2001 IEEE international conference on data mining*, pages 187–194. IEEE, 2001.

Kenneth D Harris, Hannah Hochgerner, Nathan G Skene, Lorenza Magno, Linda Katona, Carolina Bengtsson Gonzales, Peter Somogyi, Nicoletta Kessaris, Sten Linnarsson, and Jens Hjerling-Leffler. Classes and continua of hippocampal ca1 inhibitory neurons revealed by single-cell transcriptomics. *PLoS biology*, 16(6):e2006387, 2018.

Irina Higgins, Loic Matthey, Arka Pal, Christopher P Burgess, Xavier Glorot, Matthew M Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR (Poster)*, 3, 2017.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2: 193–218, 1985.

Philipp Jahn. Densired. https://github.com/PhilJahn/DENSIRED, 2023.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Iain M Johnstone and D Michael Titterington. Statistical challenges of high-dimensional data, 2009.

George Karypis. Cluto-a clustering toolkit. 2002.

Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Dmitry Kobak and Philipp Berens. The art of using t-sne for single-cell transcriptomics. *Nature communications*, 10(1):5416, 2019.

Dmitry Kobak and George C Linderman. Initialization is critical for preserving global data structure in both t-sne and umap. *Nature biotechnology*, 39(2):156–157, 2021.

Jose Laborde, Paul A Stewart, Zhihua Chen, Yian A Chen, and Naomi C Brownstein. Sparse clusterability: testing for cluster structure in high dimensions. *BMC bioinformatics*, 24 (1):125, 2023.

Jan Lause, Philipp Berens, and Dmitry Kobak. The art of seeing the elephant in the room: 2d embeddings of single-cell data do make sense. *bioRxiv*, pages 2024–03, 2024.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

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

Kevin R. Moon, David van Dijk, Zheng Wang, Scott Gigante, Daniel B. Burkhardt, William S. Chen, Kristina Yim, Antonia van den Elzen, Matthew J. Hirn, Ronald R. Coifman, Natalia B. Ivanova, Guy Wolf, and Smita Krishnaswamy. Visualizing structure and transitions for biological data exploration. *bioRxiv*, 2019. doi: 10.1101/120378. URL https://www.biorxiv.org/content/early/2019/04/04/120378.

Davoud Moulavi, Pablo A Jaskowiak, Ricardo JGB Campello, Arthur Zimek, and Jörg Sander. Density-based clustering validation. In *Proceedings of the 2014 SIAM international conference on data mining*, pages 839–847. SIAM, 2014.

Xiaojie Qiu, Qi Mao, Ying Tang, Li Wang, Raghav Chawla, Hannah A Pliner, and Cole Trapnell. Reversed graph embedding resolves complex single-cell trajectories. *Nature methods*, 14(10):979–982, 2017.

Alex Rodriguez and Alessandro Laio. Clustering by fast search and find of density peaks. *science*, 344(6191):1492–1496, 2014.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

Michael Sedlmair, Andrada Tatu, Tamara Munzner, and Melanie Tory. A taxonomy of visual cluster separation factors. In *Computer graphics forum*, volume 31, pages 1335–1344. Wiley Online Library, 2012.

Sami Sieranoja and Pasi Fränti. Fast and general density peaks clustering. *Pattern recognition letters*, 128:551–558, 2019.

Michael C Thrun and Alfred Ultsch. Clustering benchmark datasets exploiting the fundamental clustering problems. *Data in brief*, 30:105501, 2020.

VA Traag, L Waltman, and NJ Van Eck. From louvain to leiden: guaranteeing well-connected communities. sci. rep. 9, 5233, 2019.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Marissa A Weis, Stelios Papadopoulos, Laura Hansel, Timo Lüddecke, Brendan Celii, Paul G Fahey, J Alexander Bae, Agnes L Bodor, Derrick Brittain, J Buchanan, et al. Large-scale unsupervised discovery of excitatory morphological cell types in mouse visual cortex. 2022.

F Alexander Wolf, Philipp Angerer, and Fabian J Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome biology*, 19:1–5, 2018.

F Alexander Wolf, Fiona K Hamey, Mireya Plass, Jordi Solana, Joakim S Dahlin, Berthold Göttgens, Nikolaus Rajewsky, Lukas Simon, and Fabian J Theis. Paga: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. *Genome biology*, 20:1–9, 2019.

Appendix A. m-VAE and its Losses

The m-VAE model has two main components: an encoder and a decoder. The encoder itself has two parts. The first estimates the probability $p(y|x)$, where x is the input and y is the 'class label'. To achieve it, the it assigns a point to a mixture component using the Gumbel Softmax layer (Jang et al., 2016), which enables differentiable sampling from a categorical distribution without ground truth labels. It computes the log probabilities of all the classes in the distribution and adds them to noise from the Gumbel distribution, similar to the reparametrization trick in a classic VAE (Kingma, 2013). The class with the highest value is treated as a one-hot label. The second part concatenates this pseudo-label y with the input x to estimate the latent variable z from a Gaussian prior, $p(z|x, y)$. The decoder then reconstructs the input from z, similar to a standard VAE. The loss includes KL and reconstruction losses are same as for β-VAE, and the categorical loss, which is the entropy between logits and probabilities from the Gumbel Softmax layer. Note that the ground truth labels are not used anywhere during learning.

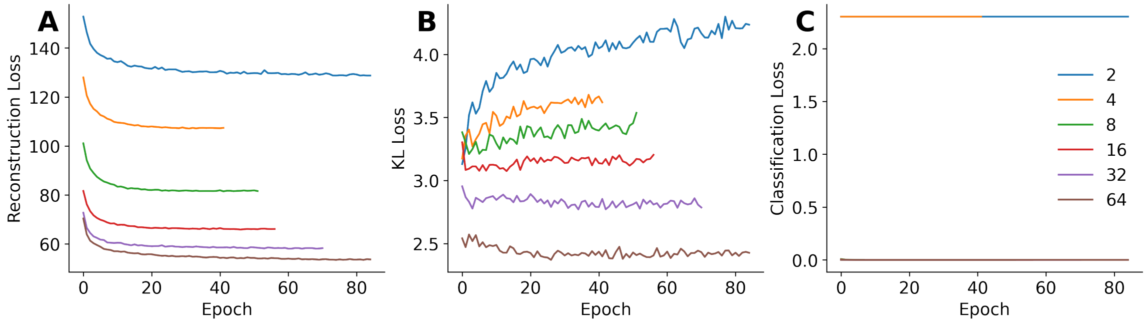

Figure 4: **A:** Reconstruction loss. **B:** KL loss. We assumed Gaussian distribution. **C:** Classification Loss. It seems like latent dimensions of 2 and 4 are too shallow to separate the groups linearly.

Appendix B. ARI explained

To estimate clustering robustness, we use adjusted rand index (ARI) (Hubert and Arabie, 1985), which measures the pairwise similarity of two cluster assignments X and Y.

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] \Big/ \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] \Big/ \binom{n}{2}} . \tag{1}$$

Here, a_i is the number of data points in cluster i of partition X, b_j is the number of data points in cluster j of partition Y, n_{ij} is the number of data points in clusters i and j of partition X and Y, respectively, and n is the total number of data points.

ARI is an cluster validation index, which computes a score for a pair of partitions. It checks for all pairs of points whether they are grouped together (end up in the same cluster) in both partitions. If they are, it is counted as ‘agreement’ while otherwise (the pair is in the same cluster in one partition and in different clusters in the other) the two partitions disagree. It computes a score that reflects the proportion of agreements that are not due to random chance, meaning it accounts for the fact that some agreements would occur randomly. A score of 1 means perfect agreement (i.e. the clusterings are equivalent up to change of labels), 0 indicates that there are not more agreements between the two partitions than there would be between two random partitions.

Appendix C. Other clustering evaluation metrics

There are other metrics than ARI to evaluate clustering partitions. ARI is a metric between two clustering partitions as it evaluates if pairs of points end up in the same groups across partitions. *Fowlkes-Mallows* also compares two clustering partitions but, in contrast to ARI, without adjusting for chance. The score is the geometric mean of the precision and recall across the whole dataset (true positive: pairs of points grouped together in both partitions). The other three set-based metrics are homogeneity, completeness, and v-measure. These three scores are asymmetric, one partition is being evaluated while the other provides (pseudo)labels. Ideally, this ‘other’ partition would be the ground-truth assignment. A clustering result satisfies homogeneity if all of its clusters contain only data points which are members of a single class. The score is achieved by computing the fraction of points with the ‘correct’ label, averaged over all clusters. A clustering result satisfies completeness if all the data points that are members of a given class are elements of the same cluster. Both metrics are not symmetric: switching ground-truth and predicted labels for completeness will return the homogeneity, and vice versa. V-measure is equivalent to the normalized mutual information (NMI) and it’s the harmonic mean of completeness and homogeneity.

We see that the trends and line order in Fig. 5 is consistent with ARI across all the metrics, meaning that the results based on ARI are compatible with those other cluster evaluation metrics.

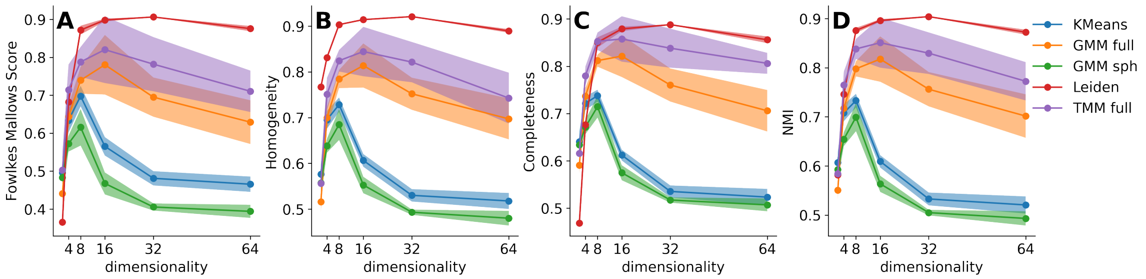

Figure 5: **A:** Fowlkes-Mallows score. **B:** Homogeneity. **C:** Completeness. **D:** NMI.

Appendix D. Cluster Validity Indices on MNIST-Nd

D.1. Internal Cluster Validation

In the main paper we provided DISCO scores which are effectively Silhouette scores adapted to a density-based setting. In contrast to Silhouette which strictly favors ball-shaped clusters, density-based cluster validation indices such as DBCV (Moulavi et al., 2014), DCSI (Gauss et al., 2023), and DISCO only look at the 'gap' between clusters but work for clusters of arbitrary shapes.

We chose to use DISCO scores in Fig. 2 as it outputs a point-wise score, allowing us to visualize that essentially all clusters are slightly overlapping, as we desired.

For comparison, we provide (overall) scores computed with different density-based cluster validation indices in Table 1 and S_Dbw (Halkidi and Vazirgiannis, 2001) as a centroid-based CVI (↓ indicates that for S_Dbw lower values are better). From the table, we can see that both DISCO and DBCV evaluate the clustering as 'bad' (DBCV is more prone to output -1 compared to DISCO). While DISCO and S_Dbw consider all of the embeddings equally bad, DBCV and DCSI clearly prefer the 8-64 dimensional embeddings, even though they do not look more easily separable in t-SNE (see Fig. 2). Overall, the values indicate that there is still a significant or large amount of overlap between clusters as desired.

	DISCO	DBCV	DCSI	S_Dbw (↓)
2d	-0.05	-1	0.16	0.87
4d	-0.10	-0.92	0.35	0.78
8d	-0.02	-0.77	0.48	0.68
16d	-0.01	-0.65	0.46	0.81
32d	-0.02	-0.63	0.46	0.86
64d	-0.02	-0.56	0.44	0.88

Table 1: CVI scores across dimensionalities. There is significant overlap between the clusters.

Appendix E. Fast search density peak analysis

E.1. Different thresholds

The main hyperparameter of fast density peak search algorithm (Rodriguez and Laio, 2014) is radius r for local density estimation. As distances become more uniform in the higher dimensions, it is crucial to scale the radius with it. To avoid being biased for the choice of r hyperparameter, we performed this analysis using $[0.01, 0.1, 0.3, 0.5, 1, 2, 3, 5]$-th distances percentiles as r. As expected, smaller distances reveal more density peaks for higher dimensions but generally we see below ten clear density modes within a range of thresholds.

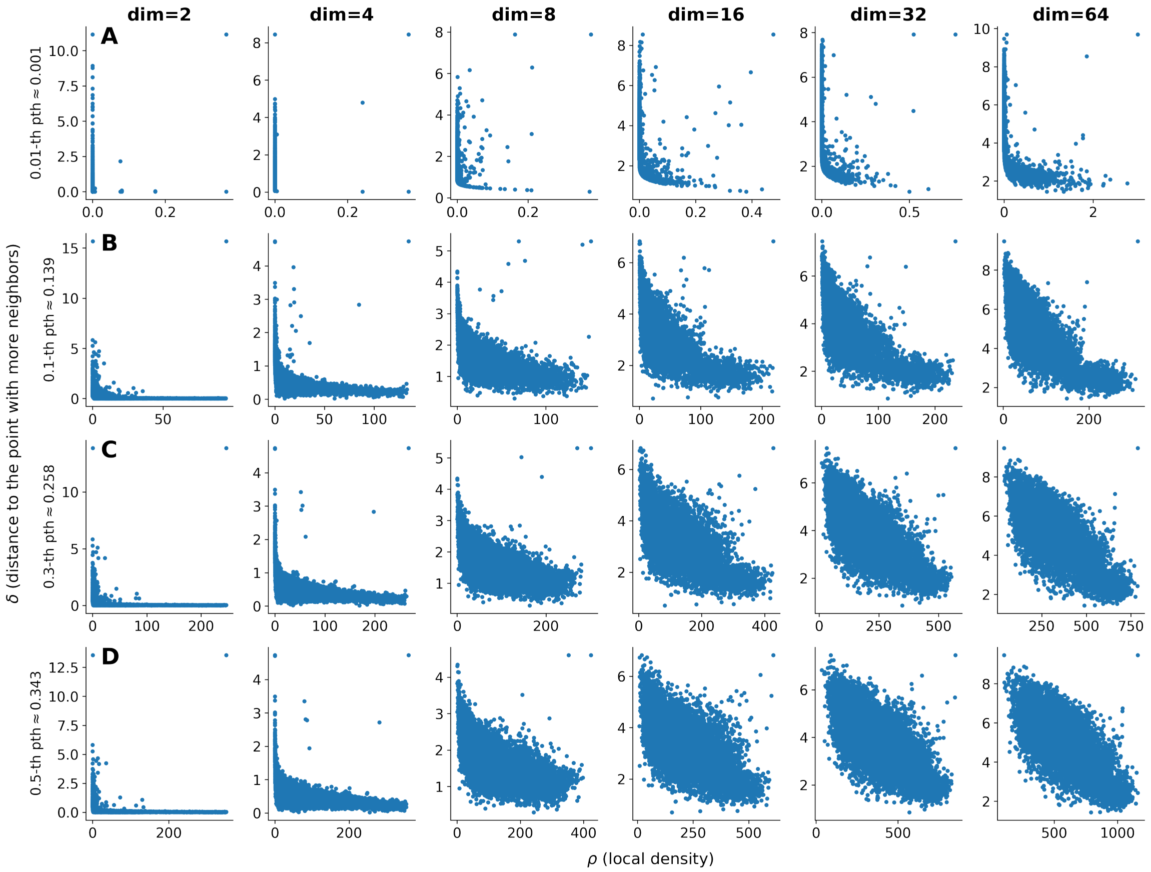

Figure 6: ρ-δ plots for different threshold values. Thresholds are percentiles from the distribution of all distances. In this plot the percentiles below one percent. Pth stays for percentile. The clear separated points are natural cluster centers.

E.2. Gamma cluster centers selection

Following Rodriguez and Laio (2014) one could use a product of local density and distances $\gamma = \rho \cdot \delta$ to the neighbor with bigger amount of neighbors to select the clusters centers. If the number of clusters is defined as n, one would just take n points with the biggest γ. Otherwise, order points by γ decreasing and see how many of the first points have substantially big gaps between each other and select these as cluster centers. Please note that this strategy works nicely for balanced clusters, which is the case for us, otherwise, other strategies should be applied Sieranoja and Fränti (2019). Below we provide plots for the points with 30 biggest γ values for different thresholds.

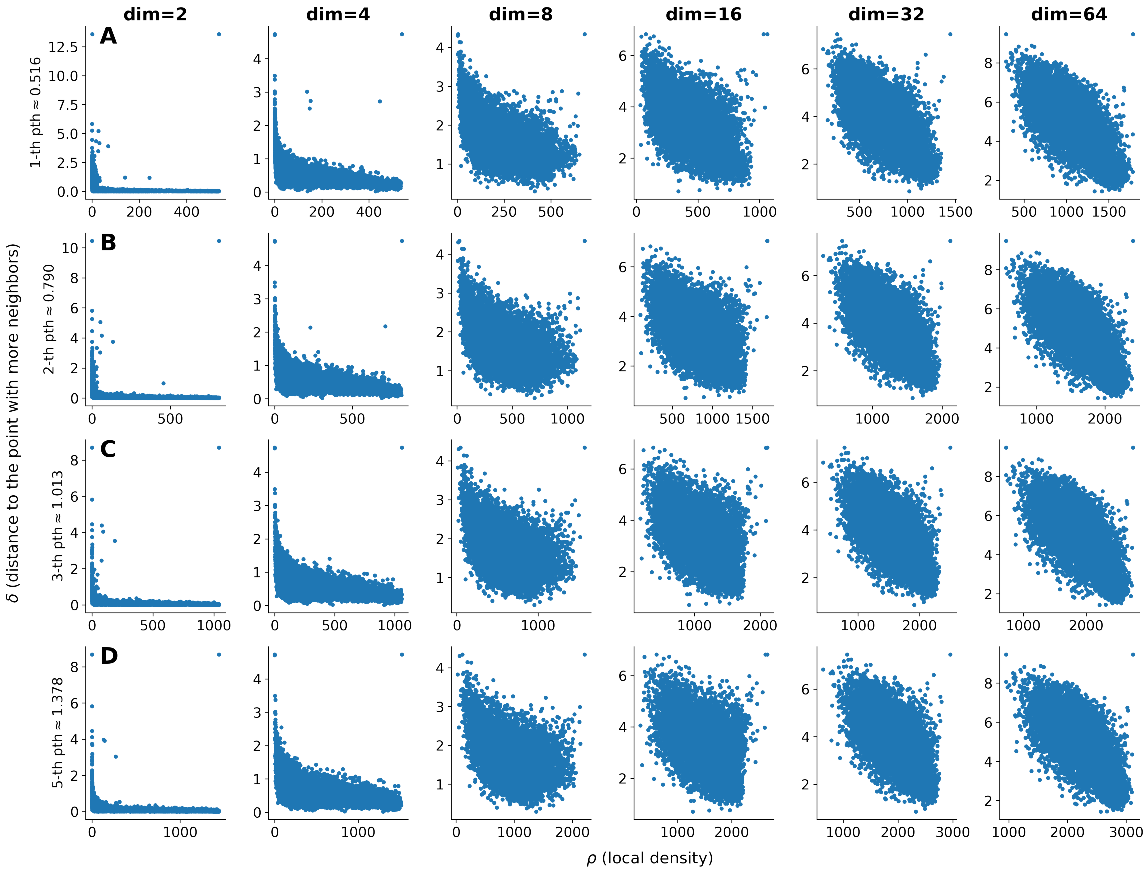

Figure 7: Same as Fig. 6 but percentiles are now above one percent.

Appendix F. Code sources acknowledgement

- We adjusted the PyTorch version of the following repo to train a m-VAE https://github.com/jariasf/GMVAE.git
- We also adjusted code from here for early stopping https://github.com/Bjarten/early-stopping-pytorch
- For TMM we used the implementation from https://github.com/jlparki/mix_T.
- We used Leiden cluster implementation from ‘scanpy‘ package, (Wolf et al., 2018) https://scanpy.readthedocs.io/en/stable/,
- K-means and GMM were used from the ‘scikit-learn‘ package (Buitinck et al., 2013)

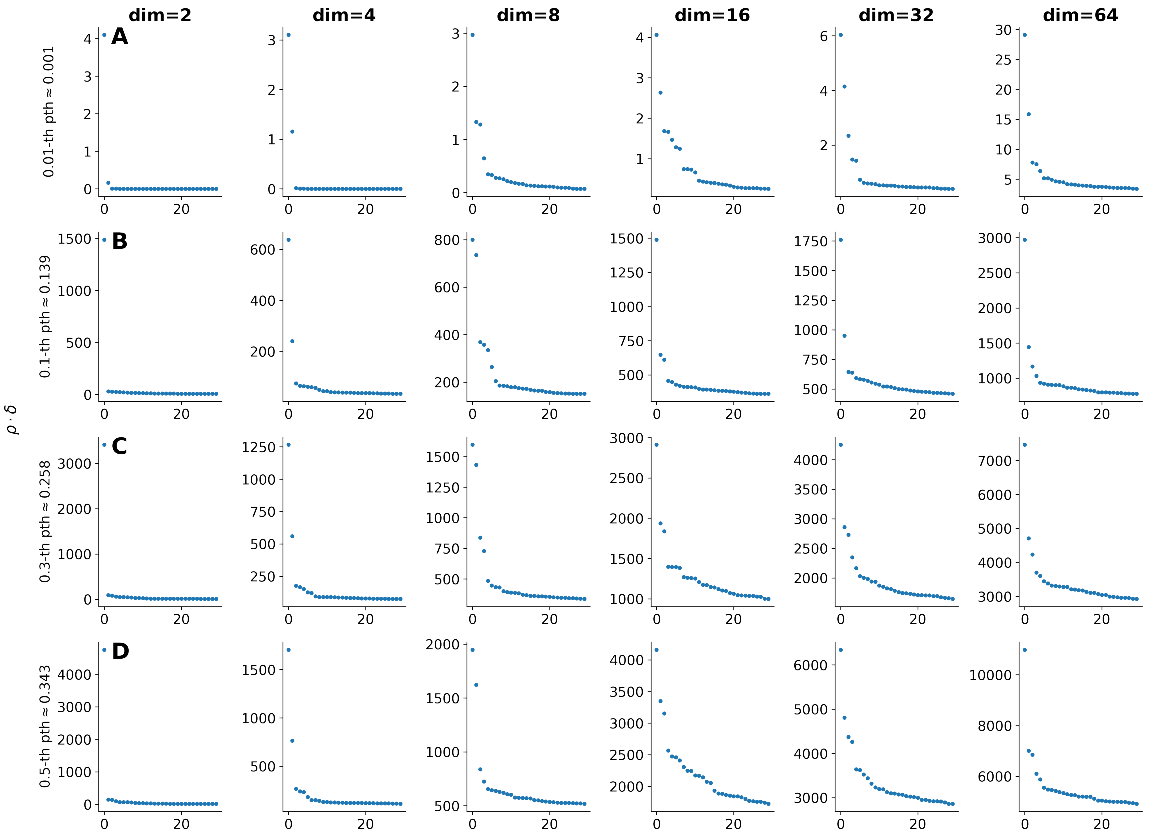

Figure 8: γ values for different threshold values. Thresholds are percentiles from the distribution of all distances. In this plot the percentiles below one percent. Pth stays for percentile. The clear separated points are natural cluster centers.

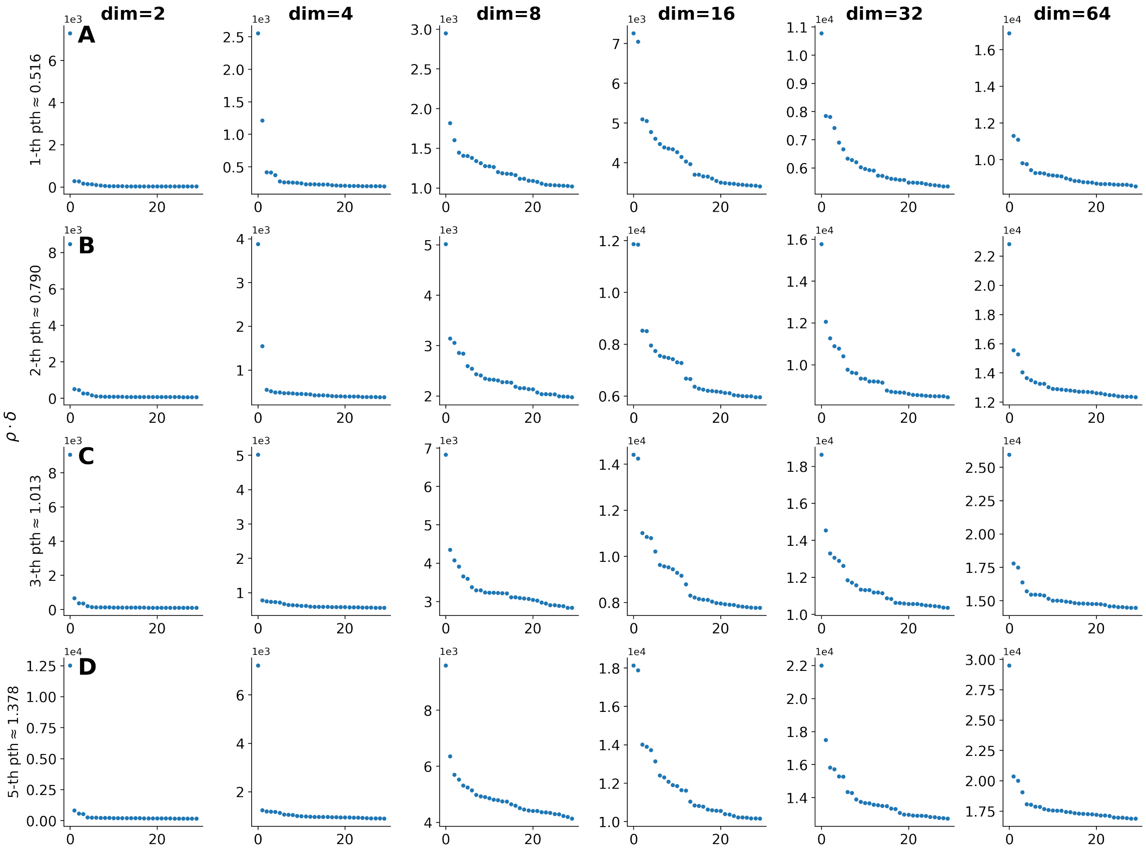

Figure 9: Same as Fig. 8 but percentiles are now above one percent.