An Empirical Study into Clustering of Unseen Datasets with Self-Supervised Foundation Models

Scott C. Lowe [* 1] **Joakim Bruslund Haurum** [* 2 3] **Sageev Oore** [† 1 4] **Thomas B. Moeslund** [† 2 3]
Graham W. Taylor [† 1 5]

Abstract

Can foundation models generalize to new datasets outside their training domain, without any retraining? Our suite of benchmarking experiments use encoders pretrained solely on ImageNet-1k with either supervised or self-supervised training techniques, clustering image datasets that were not seen during training with conventional clustering algorithms. This evaluation allows us to investigate the impact of the pretraining protocol on a model's ability to generalize outside its training domain, and explore what is natively prioritized by the model in its embeddings in a real-world scenario where novel data lacks labels. We find supervised encoders typically offer more utility than SSL encoders within the training domain, and vice-versa far outside of it, however, fine-tuned SSL encoders demonstrate the opposite trend.

1. Introduction

Self-supervised learning (SSL) has attracted great interest in recent years across almost every machine learning sub-field, due to the promise of being able to harness large quantities of unlabelled data, obtaining generic feature embeddings useful for a variety of downstream tasks (Balestriero et al., 2023), and thus enabling the training of very large *foundation models*. This has, for example, led to the development of impressive large language models (Brown et al., 2020) and computer vision systems trained on 1 billion images (Goyal et al., 2021). However, while the embeddings from a pretrained foundation model can perform well on downstream tasks after fine-tuning, there has been less investigation into the extent to which the embeddings can generalize outside the training domain without fine-tuning the model. Prior work (Vaze et al., 2022; Zhou & Zhang, 2022) suggests SSL feature encoders generate embeddings suitable for clustering, but nonetheless adjust the feature encoders through fine-tuning. Yet, widespread interest in the application of large foundation models on custom datasets, combined with prohibitive cost of compute, make this question important and increasingly urgent.

We find that there has been no investigation into whether pretrained encoders can serve as a foundation for clustering, outside of the training distribution of the encoder. Vaze et al. (2023) showed that features from SSL encoders are typically biased toward shape features and not color, texture, or count when clustered using K-Means. However, this was conducted using a synthetic dataset, where very specific object attributes could be disentangled. In contrast, here we perform a *zero-shot transfer-learning task*, evaluating the performance of a suite of pretrained image encoders across a diverse set of datasets, using various classical clustering methods, yielding the following contributions. We:

- Conduct the first in-depth empirical investigation of clustering of pretrained encoders outside their training domain, finding SSL encoders can produce meaningful clusters on unseen datasets far outside the training domain without per-dataset parameter tuning.
- Establish a comprehensive suite of benchmark evaluations for clustering unseen image datasets.
- Demonstrate that clustering quality is an evaluation method orthogonal to kNN. Additionally, we show that clusterings can be further investigated on multi-labelled datasets to identify which stimulus attributes the encoder prioritizes.
- Discover the representations of SSL-pretrained image models are more heavily impacted by background-foreground disparity than supervised models.
- Find manifold-based reduction of embeddings is essential for performant clustering.
- Find that Agglomerative Clustering clusters embeddings best, though the effect size is small.

---

[*] Equal contribution [1] Vector Institute, Canada [2] Visual Analysis and Perception Lab, Aalborg University, Denmark [3] Pioneer Centre for AI, Denmark [4] Dalhousie University, Canada [5] University of Guelph, Canada. Correspondence to: Scott C. Lowe <scott.lowe@vectorinstitute.ai>.

Published at ICML 2024 Workshop on Foundation Models in the Wild.

- Find that the silhouette score is strongly correlated with the adjusted mutual information score provided the silhouette is measured in UMAP-reduced space, and hence can be a strong proxy of clustering performance without access to ground-truth labels.

2. Experimental Design

We consider the task of **zero-shot clustering** of feature embeddings obtained from pretrained foundation models. The aim is to cluster the embeddings from various as-yet unseen datasets, in a way such that the clusters are intrinsically well-defined and, ideally, match the ground-truth (GT) label assignments, through the transfer of pretraining knowledge and without any domain-adaptation. Our models and clusterers are only tuned on data from a single dataset, ImageNet-1k (IN-1k) (Russakovsky et al., 2015). The clustering methods are deployed on 26 test datasets without retuning any parameters, allowing us to cluster novel datasets without utilizing any training data for the new datasets.

2.1. Feature Encoders

To capture the diverse methodologies within the SSL field, we compare methods from the major SSL paradigms within computer vision (Balestriero et al., 2023). We choose one representative method per paradigm, and compare the clusterability of their features against those of a model pretrained with cross-entropy supervision (X-Ent.) on IN-1k. The SSL models selected are: *Contrastive Learning*, MoCo-v3 (Chen et al., 2021); *Self-Distillation*, DINO (Caron et al., 2021); *Canonical Correlation Analysis*, VICReg (Bardes et al., 2022); and *Masked Image Modelling*, MAE (He et al., 2022). For each method we consider two common backbone architectures, ResNet-50 (He et al., 2016) and ViT-B (Dosovitskiy et al., 2021) as available using publicly available checkpoints trained on the IN-1k dataset.

2.2. Clustering Methods

To cluster the feature embeddings, we considered several classical clustering methods: *K-Means* (Lloyd, 1982), *Spectral Clustering* (Yu & Shi, 2003), *Agglomerative Clustering* (AC) (Everitt et al., 2011), *Affinity Propagation* (AP) (Frey & Dueck, 2007), and *HDBSCAN* (McInnes & Healy, 2017). These were chosen as they have few parameters to tune and cover several clustering paradigms (partition, hierarchical, graph-theory, and density). For each clusterer, we conducted a hyperparameter search by clustering IN-1k training data, described in Appendix E. As K-Means and Spectral require the number of clusters, we assume that this is known *a priori*. In contrast, AC, AP, and HDBSCAN estimate the number of clusters from the data. AC can operate with the number of clusters given or inferred, and we consider both configurations (“AC w/ C” and “AC w/o C”, respectively).

2.3. Datasets

We evaluated the different permutations of pretrained models and clustering methods on a diverse set of 26 datasets, detailed in Table 1. The datasets span tasks with differing levels of label granularity, number of classes and samples, domain shifts, and degree of class imbalance. Only the IN-1k training split was used for training encoders and clustering hyperparameter selection, hence other datasets are indicative of performance that would be seen in-the-wild. We divided the datasets into five groups: In-domain (ID), Domain-shifted (DS), Near-out-of-domain (Near-OOD), Fine-grained near-OOD (FG) and Far-out-of-domain (Far-OOD). For more details see Appendix B.

2.4. Evaluation Metrics

We evaluated the performance of a clustering using adjusted mutual information (AMI) (Vinh et al., 2010) and silhouette score (Rousseeuw, 1987), defined in Appendix C. AMI measures the agreement between the constructed clusters and the GT labels, corrected for chance-level agreement. Silhouette score measures how well-defined the clusters are intrinsically, without reference to a GT clustering.

2.5. Experimental Methodology

For each test dataset, we preprocessed the images by resizing to 224 pixels (shortest-side) and taking a centered square 224×224 crop. For each encoder, images were standardized using the RGB mean and stdev used to train it, then passed through the encoder to create embedding vectors of 2048-d (RN50) or 768-d (ViT-B). We clustered the embeddings with each clusterer, using parameters fit on IN-1k training data (see Appendix E). When using UMAP or PCA for dim reduction, this was fit separately for each test dataset.

3. Experimental Results

We report the clustering capabilities of the considered encoders and clusterers measured by AMI on datasets of varying distances from the training domain. For additional experiments see Appendices M–R.

3.1. Comparison of SSL Encoders

For each dataset (Appendix B), we measured the AMI between the clustered embeddings and the GT labels (averaged over clusterers). Using the IN-1k supervised model as a baseline, we took the difference between its AMI and that of the SSL models, averaged within each group of datasets.

As shown in Fig. 1, the performance of the SSL encoders is lower than that of the supervised network on ID, DS, and near-OOD datasets; though the effect size is often large (in the order of 10 p.p.), the difference is generally not

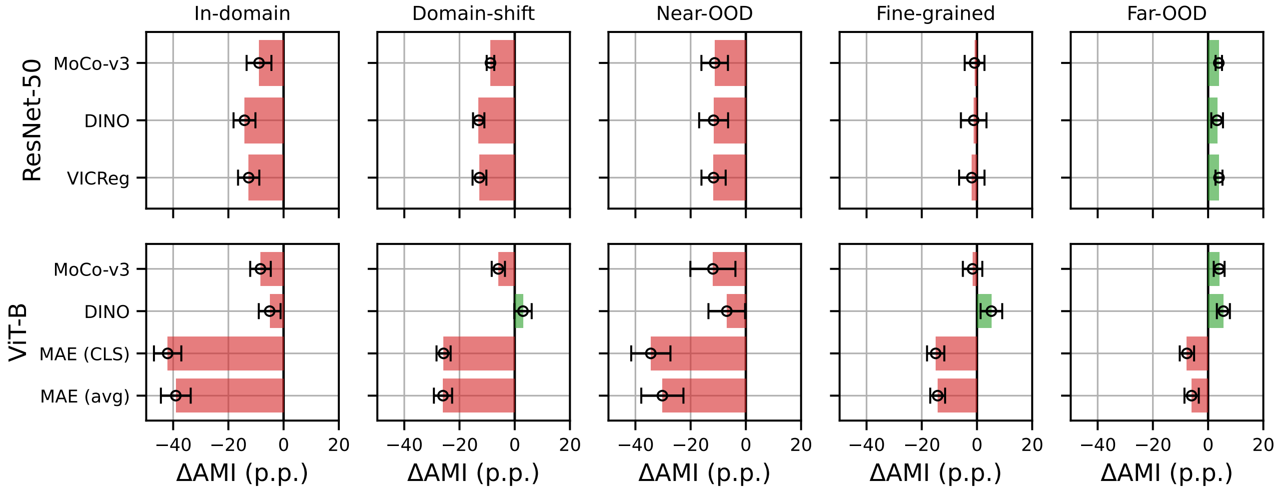


Figure 1: **Percentage-point (p.p.) difference in AMI between clusters formed from SSL encoder embeddings versus supervised encoder embeddings.** We compare the clustering quality of each dataset (avg 6 clusterers) using SSL encoders against encoders supervised on IN-1k. We present the mean across $3 \leq N \leq 8$ datasets in each group (bars: ±1 stderr).

significant due to the limited number of datasets per category and the variance between them (significant for MoCo-v3 and DINO on DS; $p < 0.05$, Bonferroni-corrected paired t-test). The MAE encoder (either using the CLS token embedding, or average for image patches) performed especially poorly (significantly worse than supervised ViT-B on DS and Near-OOD; $p < 0.05$). This is congruent with the observation that MAE-trained models possess details about the pixel-level contents of the stimulus, but need fine-tuning to perform well at whole-image classification (He et al., 2022).

For FG datasets, overall, SSL encoders are comparable in performance to supervised (except MAE, lower than sup., $p < 0.05$), but when we explore the results on a per-dataset basis, we find supervised encoders perform best on Stanford Cars and NABirds by a reasonable margin, whilst SSL encoders perform best on Aircraft and Oxford Flowers datasets (see Appendix H). We speculate this difference between FG datasets is caused by their (dis)similarity with IN-1k imagery. On Far-OOD datasets, SSL-encoders (except MAE) outperform supervised networks (not significant). The effect of dataset granularity is explored further in Appendix N.

Our results demonstrate supervised encoders perform better at clustering unseen datasets similar to the training distribution, which we suspect is due to alignment with the training task. But as the data moves further from the training data, the adaptability of SSL encoders triumphs over supervised models when they leave their training niche. Within the SSL encoders, DINO produced the best encoder for ViT-B, but the worst for RN50. We believe this is because DINO's training process, unlike other SSL methods, is able to take advantage of ViT's attention mechanism to focus solely on the subject, which we explore further in Appendix O.

3.2. Comparison of Fine-Tuned SSL Encoders

There is often more than one way in which a collection of images can be grouped together, depending on which high-level properties within the images are prioritized. We explore breaking down information about the clusterings by utilizing datasets with multiple annotation types in Appendices O, P, Q.

We also considered that clustering the embeddings produced by the SSL-pretrained encoders may sometimes result in "legitimate" clusterings that are consistent with particular semantic features of the images, just not aligned with categorization used for the GT annotations. Moreover, previous work has shown that MAE requires fine-tuning (FT) to be able to be able to perform whole-frame classification (He et al., 2022). Consequently, we investigated whether fine-tuning the pretrained encoders on an IN-1k classification task would make their embeddings more aligned with the classification typically employed in machine learning tasks. We fine-tuned each of the SSL-pretrained encoders on IN-1k following the methodology of He et al. (2022), repeated the clustering parameter search for the FT encoders, then clustered their embeddings of each test dataset.

As shown in Fig. 2, we found fine-tuning unsurprisingly increases performance on in-domain and domain-shifted datasets, where target classes are the same as (or subset of) the IN-1k classes used for the FT task. The gain in performance was sufficient that SSL-encoders tended to beat the supervised network on these datasets, though the difference was not significant. Furthermore, with a RN50 backbone the FT SSL-encoders beat the supervised baseline on Near-OOD data, whilst with a ViT-B backbone the FT SSL-encoders beat the supervised baseline on FG datasets.

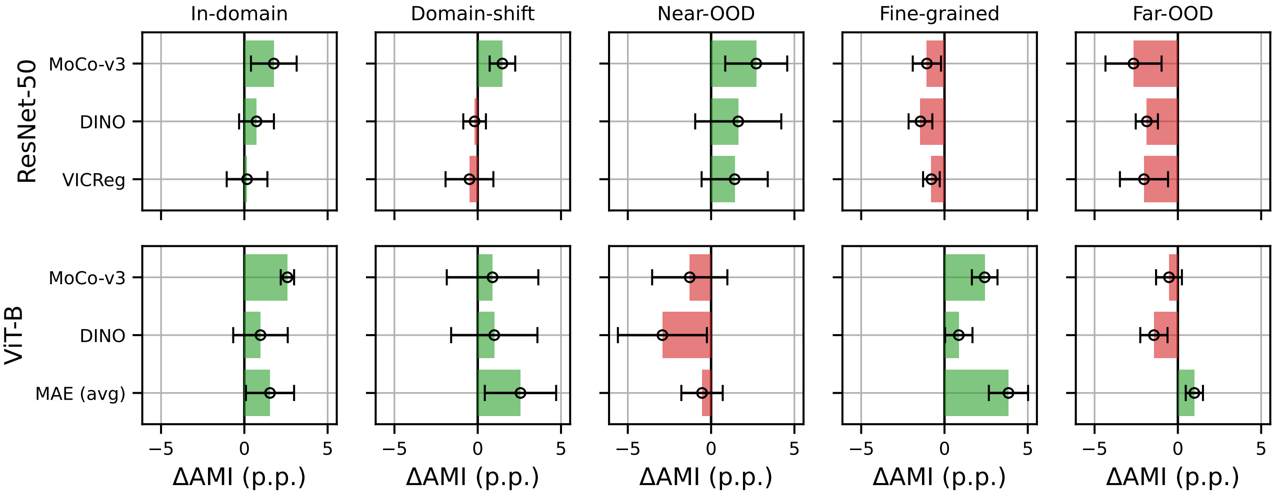


Figure 2: **Percentage-point (p.p.) difference in AMI between clusters formed from embeddings of SSL-pretrained networks fine-tuned on IN-1k versus fully-supervised networks.** We measure the difference in AMI (avg 6 clusterers) with fine-tuned SSL encoders as compared to encoders trained solely with cross-entropy on IN-1k (error bars: ±1 stderr; $3 \leq N \leq 8$ datasets). *Note:* The x-scale differs from that used in Fig. 1, but the baseline (0 values) are the same.

However, the performance on Far-OOD datasets declined post-FT, enough that the performance of SSL-encoders became worse than supervised encoders. The only exception to this was MAE, which greatly increased its performance on all types of dataset. Across the supervised and FT encoders, MAE was the best performing encoder on every group of datasets, though its performance of Far-OOD data was still below that of the non-FT SSL-encoders.

3.3. Correlation between AMI and Silhouette Score

So far, we focused on the AMI metric. However, in the context of clustering data in-the-wild this can be problematic as it requires GT labels, and novel data is unlikely to be extensively labelled. Therefore, we considered whether the silhouette score, S, calculated from just the predicted clusters could provide a valuable metric.

We compared the AMI and S for each clusterer-model-dataset combination, shown in Fig. 9 and Fig. 10. We computed the Spearman's rank correlation coefficient, ρ, using S in either the original embedding space, or the UMAP-reduced space. We find that AMI and S are strongly correlated, with high S implying a high AMI, especially in UMAP-reduced space ($0.76 \leq \rho \leq 0.88$) where S expressed a much larger range of values. Previous work found S to be an inconsistent metric during model training (Xu et al., 2022); our results are consistent with this as we find a high correlation for fully-trained networks only. We conclude silhouette score can be a good proxy for cluster quality for trained networks when GT labels are lacking.

4. Conclusion

We empirically investigated how well the embeddings produced by pretrained models can be clustered in-the-wild for data unseen during training at different distances from the training distribution. We considered two architectures trained using one of 5 methodologies (1 supervised, 4 SSL), on 26 datasets, using 5 distinct types of clusterer.

To cluster embeddings of a novel dataset, we suggest dimensionality reduction with UMAP (5–100 dim, we chose 50d), then use AC with L2, Ward, on the reduced embeddings. UMAP-reduction works best for all clusterers except Spectral, despite not being distance-preserving. We also show promising results that silhouette score can be used to compare methods on real-world data where no GT is available, especially when applied on UMAP-reduced embeddings. These results are indicative of the embedded dataset lying on a low-dimensional, non-linear manifold.

Analyzing the performance of SSL encoders with clustering enables us to investigate what the embeddings represent in-of-themselves, without imposing a training objective aligned with the evaluation. We find the clustering AMI is only weakly correlated with the kNN accuracy (see Appendix S), suggesting it is an encoder evaluation orthogonal to existing benchmarks in the literature.

For datasets far outside the original training domain, SSL encoders provide clustering in best agreement with the data annotations. For images near the training domain, SSL encoders fine-tuned on class-labels from the training domain perform best, but this gain in performance comes at a cost, greatly reducing the performance on Far-OOD data. Our work emphasizes the importance of the alignment between a foundation model's training task and the downstream task on which its embeddings are applied.

References

Ankerst, M., Breunig, M. M., Kriegel, H.-P., and Sander, J. OPTICS: Ordering points to identify the clustering structure. *SIGMOD Rec.*, 28(2):49–60, jun 1999. ISSN 0163-5808. doi:10.1145/304181.304187.

Arthur, D. and Vassilvitskii, S. K-Means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pp. 1027–1035, USA, 2007. Society for Industrial and Applied Mathematics. ISBN 9780898716245. doi:10.1145/1283383.1283494.

Balestriero, R., Ibrahim, M., Sobal, V., Morcos, A., Shekhar, S., Goldstein, T., Bordes, F., Bardes, A., Mialon, G., Tian, Y., Schwarzschild, A., Wilson, A. G., Geiping, J., Garrido, Q., Fernandez, P., Bar, A., Pirsiavash, H., LeCun, Y., and Goldblum, M. A cookbook of self-supervised learning, 2023. arXiv: 2304.12210. doi:10.48550/arxiv.2304.12210.

Bao, H., Dong, L., Piao, S., and Wei, F. BEiT: BERT pre-training of image transformers. In *10th International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=p-BhZSz59o4`.

Bardes, A., Ponce, J., and LeCun, Y. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *10th International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=xm6YD62D1Ub`.

Bellman, R., Corporation, R., and Collection, K. M. R. *Dynamic Programming*. Rand Corporation research study. Princeton University Press, 1957. ISBN 9780691079516.

Bezdek, J. C., Ehrlich, R., and Full, W. FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2):191–203, 1984. ISSN 0098-3004. doi:10.1016/0098-3004(84)90020-7.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf`.

Caron, M., Bojanowski, P., Joulin, A., and Douze, M. Deep clustering for unsupervised learning of visual features. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y. (eds.), *Computer Vision – ECCV 2018*, pp. 139–156, Cham, 2018. Springer International Publishing. ISBN 978-3-030-01264-9. doi:10.1007/978-3-030-01264-9_9.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9912–9924. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/70feb62b69f16e0238f741fab228fec2-Paper.pdf`.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9630–9640, 2021. doi:10.1109/ICCV48922.2021.00951.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/chen20j.html`.

Chen, X. and He, K. Exploring simple siamese representation learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15745–15753, 2021. doi:10.1109/CVPR46437.2021.01549.

Chen, X., Xie, S., and He, K. An empirical study of training self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9620–9629, 2021. doi:10.1109/ICCV48922.2021.00950.

Chopra, S., Hadsell, R., and LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pp. 539–546 vol. 1, 2005. doi:10.1109/CVPR.2005.202.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., and Vedaldi, A. Describing textures in the wild. In *2014 IEEE*

Conference on Computer Vision and Pattern Recognition, pp. 3606–3613, 2014. doi:10.1109/CVPR.2014.461.

Cole, E., Yang, X., Wilber, K., Aodha, O. M., and Belongie, S. When does contrastive visual representation learning work? In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 01–10, 2022. doi:10.1109/CVPR52688.2022.01434.

Cui, Y., Gu, Z., Mahajan, D., van der Maaten, L., Belongie, S., and Lim, S.-N. Measuring dataset granularity, 2019. arXiv: 1912.10154. doi:10.48550/arxiv.1912.10154.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=YicbFdNTTy`.

Ermolov, A., Siarohin, A., Sangineto, E., and Sebe, N. Whitening for self-supervised representation learning. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3015–3024. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/ermolov21a.html`.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pp. 226–231. AAAI Press, 1996. URL `https://cdn.aaai.org/KDD/1996/KDD96-037.pdf`.

Everitt, B. S., Landau, S., Leese, M., and Stahl, D. *Cluster Analysis*. Wiley, January 2011. doi:10.1002/9780470977811.

Frey, B. J. and Dueck, D. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007. doi:10.1126/science.1136800.

Gharaee, Z., Gong, Z., Pellegrino, N., Zarubiieva, I., Haurum, J. B., Lowe, S., McKeown, J., Ho, C., McLeod, J., Wei, Y.-Y., Agda, J., Ratnasingham, S., Steinke, D., Chang, A., Taylor, G. W., and Fieguth, P. A step towards worldwide biodiversity assessment: The BIOSCAN-1M insect dataset. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 43593–43619. Curran Associates, Inc., 2023. URL `https://proceedings.neurips.cc/paper_files/paper/2023/file/87dbbdc3a685a97ad28489a1d57c45c1-Paper-Datasets_and_Benchmarks.pdf`.

Gong, Z., Wang, A. T., Haurum, J. B., Lowe, S. C., Taylor, G. W., and Chang, A. X. BIOSCAN-CLIP: Bridging vision and genomics for biodiversity monitoring at scale, 2024. arXiv: 2405.17537. doi:10.48550/arxiv.2405.17537.

Goyal, P., Caron, M., Lefaudeux, B., Xu, M., Wang, P., Pai, V., Singh, M., Liptchinsky, V., Misra, I., Joulin, A., and Bojanowski, P. Self-supervised pretraining of visual features in the wild, 2021. arXiv: 2103.01988. doi:10.48550/arxiv.2103.01988.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., Piot, B., kavukcuoglu, k., Munos, R., and Valko, M. Bootstrap your own latent - a new approach to self-supervised learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 21271–21284. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/f3ada80d5c4ee70142b17b8192b2958e-Paper.pdf`.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi:10.1109/CVPR.2016.90.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9726–9735, 2020. doi:10.1109/CVPR42600.2020.00975.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15979–15988, 2022. doi:10.1109/CVPR52688.2022.01553.

Helber, P., Bischke, B., Dengel, A., and Borth, D. EuroSAT: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019. doi:10.1109/JSTARS.2019.2918242.

Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., Song, D., Steinhardt, J., and Gilmer, J. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8320–8329, 2021a. doi:10.1109/ICCV48922.2021.00823.

Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., and Song, D. Natural adversarial examples. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15257–15266, 2021b. doi:10.1109/CVPR46437.2021.01501.

Howard, J. ImageNette, ImageWoof, and ImageWang, 2019. URL `https://github.com/fastai/imagenette/`. Revision: ba8db39.

Krause, J., Stark, M., Deng, J., and Fei-Fei, L. 3d object representations for fine-grained categorization. In *2013 IEEE International Conference on Computer Vision Workshops*, pp. 554–561, 2013. doi:10.1109/ICCVW.2013.77.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL `https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf`.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi:10.1109/5.726791.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 3730–3738, December 2015. doi:10.1109/ICCV.2015.425.

Lloyd, S. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. doi:10.1109/TIT.1982.1056489.

Maji, S., Kannala, J., Rahtu, E., Blaschko, M., and Vedaldi, A. Fine-grained visual classification of aircraft, 2013. arXiv: 1306.5151. doi:10.48550/arxiv.1306.5151.

McInnes, L. Comparing python clustering algorithms. `https://hdbscan.readthedocs.io/en/latest/comparing_clustering_algorithms.html`, 2016. Revision: 98928d0c. Accessed: 2023-09-22. Last updated: 2022-03-28.

McInnes, L. Using UMAP for clustering. `https://umap-learn.readthedocs.io/en/latest/clustering.html`, 2018. Revision: 5f7512b9. Accessed: 2023-09-22. Last updated: 2018-06-08.

McInnes, L. and Healy, J. Accelerated hierarchical density based clustering. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 33–42, 2017. doi:10.1109/ICDMW.2017.12.

McInnes, L., Healy, J., and Melville, J. UMAP: Uniform manifold approximation and projection for dimension reduction, February 2018. arXiv: 1802.03426. doi:10.48550/arxiv.1802.03426.

McInnes, L., Healy, J., Saul, N., and Großberger, L. UMAP: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861, 2018. doi:10.21105/joss.00861.

Millán Arias, P., Alipour, F., Hill, K. A., and Kari, L. DeLUCS: Deep learning for unsupervised clustering of DNA sequences. *PLOS ONE*, 17(1):1–25, 01 2022. doi:10.1371/journal.pone.0261531.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. URL `http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf`.

Nilsback, M.-E. and Zisserman, A. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729, 2008. doi:10.1109/ICVGIP.2008.47.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H. V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Howes, R., Huang, P.-Y., Xu, H., Sharma, V., Li, S.-W., Galuba, W., Rabbat, M., Assran, M., Ballas, N., Synnaeve, G., Misra, I., Jegou, H., Mairal, J., Labatut, P., Joulin, A., and Bojanowski, P. DINOv2: Learning robust visual features without supervision, 2023. arXiv: 2304.07193. doi:10.48550/arxiv.2304.07193.

Pakman, A., Wang, Y., Mitelut, C., Lee, J., and Paninski, L. Neural clustering processes. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7455–7465. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/pakman20a.html`.

Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., and Efros, A. A. Context encoders: Feature learning by inpainting. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2536–2544, 2016. doi:10.1109/CVPR.2016.278.

Pearson, K. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11): 559–572, 1901. doi:10.1080/14786440109462720.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011. URL `http://jmlr.org/papers/v12/pedregosa11a.html`.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. Do ImageNet classifiers generalize to ImageNet? In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5389–5400. PMLR, 09–15 Jun 2019. URL `https://proceedings.mlr.press/v97/recht19a.html`.

Romano, S., Vinh, N. X., Bailey, J., and Verspoor, K. Adjusting for chance clustering comparison measures. *Journal of Machine Learning Research*, 17(134):1–32, 2016. URL `http://jmlr.org/papers/v17/15-627.html`.

Ronen, M., Finder, S. E., and Freifeld, O. DeepDPM: Deep clustering with an unknown number of clusters. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9851–9860, 2022. doi:10.1109/CVPR52688.2022.00963.

Rousseeuw, P. J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987. ISSN 0377-0427. doi:10.1016/0377-0427(87)90125-7.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, April 2015. doi:10.1007/s11263-015-0816-y.

Sohn, K. Improved deep metric learning with multi-class N-pair loss objective. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL `https://proceedings.neurips.cc/paper_files/paper/2016/file/6b180037abbebea991d8b1232f8a8ca9-Paper.pdf`.

Sokal, R. R. and Michener, C. D. A statistical method for evaluating systematic relationships. *University of Kansas science bulletin*, 38:1409–1438, 1958.

Song, H., Xiang, Y., Jegelka, S., and Savarese, S. Deep metric learning via lifted structured feature embedding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4004–4012, Los Alamitos, CA, USA, jun 2016. IEEE Computer Society. doi:10.1109/CVPR.2016.434.

Spanhol, F. A., Oliveira, L. S., Petitjean, C., and Heutte, L. A dataset for breast cancer histopathological image classification. *IEEE Transactions on Biomedical Engineering*, 63(7):1455–1462, 2016. doi:10.1109/TBME.2015.2496264.

Tapaswi, M., Law, M., and Fidler, S. Video face clustering with unknown number of clusters. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5026–5035, 2019. doi:10.1109/ICCV.2019.00513.

Van Gansbeke, W., Vandenhende, S., Georgoulis, S., Proesmans, M., and Van Gool, L. SCAN: Learning to classify images without labels. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.-M. (eds.), *Computer Vision – ECCV 2020*, pp. 268–285, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58607-2. doi:10.1007/978-3-030-58607-2_16.

Van Horn, G., Branson, S., Farrell, R., Haber, S., Barry, J., Ipeirotis, P., Perona, P., and Belongie, S. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 595–604, 2015. doi:10.1109/CVPR.2015.7298658.

Van Horn, G., Cole, E., Beery, S., Wilber, K., Belongie, S., and Mac Aodha, O. Benchmarking representation learning for natural world image collections. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12879–12888, Los Alamitos, CA, USA, jun 2021. IEEE Computer Society. doi:10.1109/CVPR46437.2021.01269.

Vaze, S., Hant, K., Vedaldi, A., and Zisserman, A. Generalized category discovery. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7482–7491, 2022. doi:10.1109/CVPR52688.2022.00734.

Vaze, S., Vedaldi, A., and Zisserman, A. No representation rules them all in category discovery. In *Advances in Neural Information Processing Systems*, volume 37, 2023. URL `https://openreview.net/forum?id=5ytypAqAsR`.

Vinh, N. X., Epps, J., and Bailey, J. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance.

Journal of Machine Learning Research, 11(95):2837–2854, 2010. URL http://jmlr.org/papers/v11/vinh10a.html.

Wang, H., Ge, S., Lipton, Z., and Xing, E. P. Learning robust global representations by penalizing local predictive power. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32, pp. 10506–10518. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/3eefceb8087e964f89c2d59e8a249915-Paper.pdf.

Wang, Y., Huang, H., Rudin, C., and Shaposhnik, Y. Understanding how dimension reduction tools work: An empirical approach to deciphering t-SNE, UMAP, TriMap, and PaCMAP for data visualization. *Journal of Machine Learning Research*, 22(201):1–73, 2021. URL http://jmlr.org/papers/v22/20-1061.html.

Ward, J. H. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963. ISSN 01621459. URL http://www.jstor.org/stable/2282967.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms, 2017. arXiv: 1708.07747. doi:10.48550/arxiv.1708.07747.

Xiao, K., Engstrom, L., Ilyas, A., and Madry, A. Noise or signal: The role of image backgrounds in object recognition, 2020. arXiv: 2006.09994. doi:10.48550/arxiv.2006.09994.

Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., and Hu, H. SimMIM: A simple framework for masked image modeling. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9653–9663, 2022. doi:10.1109/CVPR52688.2022.00943.

Xu, D. and Tian, Y. A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2(2):165–193, June 2015. doi:10.1007/s40745-015-0040-1.

Xu, I., Lowe, S., and Trappenberg, T. Label-free monitoring of self-supervised learning progress. In *2022 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pp. 78–84, 2022. doi:10.1109/CCECE49351.2022.9918377.

Yang, B., Fu, X., Sidiropoulos, N. D., and Hong, M. Towards K-means-friendly spaces: Simultaneous deep learning and clustering. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3861–3870. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/yang17b.html.

Yu, F., Zhang, Y., Song, S., Seff, A., and Xiao, J. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop, 2015. arXiv: 1506.03365. doi:10.48550/arxiv.1506.03365.

Yu, S. and Shi, J. Multiclass spectral clustering. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pp. 313–319 vol.1, 2003. doi:10.1109/ICCV.2003.1238361.

Zbontar, J., Jing, L., Misra, I., LeCun, Y., and Deny, S. Barlow twins: Self-supervised learning via redundancy reduction. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12310–12320. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/zbontar21a.html.

Zhang, Z., Song, Y., and Qi, H. Age progression/regression by conditional adversarial autoencoder. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4352–4360, Los Alamitos, CA, USA, jul 2017. IEEE Computer Society. doi:10.1109/CVPR.2017.463.

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., and Torralba, A. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2018. doi:10.1109/TPAMI.2017.2723009.

Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., and Kong, T. iBOT: Image BERT pre-training with online tokenizer, 2022a. arXiv: 2111.07832. doi:10.48550/arxiv.2111.07832.

Zhou, S., Xu, H., Zheng, Z., Chen, J., li, Z., Bu, J., Wu, J., Wang, X., Zhu, W., and Ester, M. A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions, 2022b. arXiv: 2206.07579. doi:10.48550/arxiv.2206.07579.

Zhou, X. and Zhang, N. L. Deep clustering with features from self-supervised pretraining, 2022. arXiv: 2207.13364. doi:10.48550/arxiv.2207.13364.

Appendices

A. Related Works

Self-Supervised Learning (SSL) has recently received an increasing amount of interest from the computer vision domain, in part due to its promising results in natural language processing (Brown et al., 2020). Whilst SSL has a long history of research, currently dominant methods can be divided into four general categories (Balestriero et al., 2023): (1) Contrastive Learning approaches, which build on metric learning, in which embeddings of multiple views of the same instance are brought together and embeddings from different instances are pushed apart (Chopra et al., 2005; Song et al., 2016; Sohn, 2016; Chen et al., 2020; He et al., 2020; Chen et al., 2021); (2) Self-Distillation approaches, where a student and teacher encoder process an input image with distinct transforms applied, and the student is tasked with predicting embeddings of the teacher (Grill et al., 2020; Chen & He, 2021; Caron et al., 2021; Zhou et al., 2022a; Oquab et al., 2023); (3) Canonical Correlation Analysis approaches, where feature embeddings are analyzed in terms of the cross-covariance matrix, through mechanisms such as minimizing covariance across feature dimensions and minimizing correlation across feature embeddings for different inputs (Caron et al., 2020; Zbontar et al., 2021; Ermolov et al., 2021; Bardes et al., 2022); (4) Masked Image Modelling approaches, where large parts of the input image are masked out and have to be reconstructed in image-space (Pathak et al., 2016; He et al., 2022; Bao et al., 2022; Xie et al., 2022).

Clustering is one of the most common tasks in a large variety of applications and can be defined as the task of finding local structures that are homogeneous and separated without explicit label supervision (Everitt et al., 2011). This problem has been studied for centuries resulting in methods using clustering criteria based on partitioning (Lloyd, 1982; Arthur & Vassilvitskii, 2007), fuzzy theory (Bezdek et al., 1984), graph theory (Yu & Shi, 2003; Frey & Dueck, 2007), density (Ester et al., 1996; Ankerst et al., 1999; McInnes & Healy, 2017), hierarchies (Sokal & Michener, 1958; Ward, 1963), and many more (Xu & Tian, 2015). These methods have traditionally necessitated a disjointed processing pipeline, as the clustering algorithms have been optimized independently of the feature generators. However, in recent years several methods have been proposed to jointly learn feature extractors and clustering processes (Ronen et al., 2022; Caron et al., 2018; Tapaswi et al., 2019; Pakman et al., 2020; Yang et al., 2017; Van Gansbeke et al., 2020; Millán Arias et al., 2022).

B. Datasets

We evaluated the different permutations of feature encoders and clustering methods on a diverse set of datasets, detailed in Table 1. These datasets span tasks with differing levels of label granularity, number of classes and samples, domain shifts, and degree of class imbalance. Out of all these datasets, only the IN-1k training split was present during training of the feature encoders and used to optimize the parameters of the clustering methods. No other datasets have been observed by the networks, and the methodology was not tuned on them. We divided the datasets into five groups as follows:

- **In-domain (ID).** Images and class labels lie within the IN-1k domain.
- **Domain-shifted (DS).** Class labels are aligned with IN-1k, but the images are changed *e.g.* background removed or replaced; images of artwork representing the class.
- **Near-out-of-domain (Near-OOD).** Images look like IN-1k images, and the classification task is similar but with new classes and distributional shift.
- **Fine-grained near-out-of-domain (FG).** Natural images resembling a subdomain of IN-1k, but labelled at a much finer-level of granularity *e.g.* plant species.
- **Far-out-of-domain (Far-OOD).** Images which lie outside the domain of IN-1k, with especially different objectives *e.g.* textures, text, faces, microscopy slides.

C. Evaluation Metrics Details

C.1. Adjusted Mutual Information

Since we are evaluating the clustering on annotated datasets, we evaluated a candidate clustering assignment against the “ground-truth” cluster labels, from an information theoretic perspective. The Normalized Mutual Information (NMI) between two label assignments V an U is defined as

$$\mathrm{NMI}(U, V) = \frac{\mathrm{MI}(U, V)}{\mathrm{mean}(\mathrm{H}(U) + \mathrm{H}(V))}, \tag{1}$$

where $\mathrm{MI}(U, V)$ is the mutual information between label assignments V an U, and $\mathrm{H}(\cdot)$ is the Shannon entropy of the considered label assignment (Zhou et al., 2022b). NMI is a relative measure of the amount of information between two label sets, and hence is bounded between 0 and 1, with 1 occurring for a perfect match and 0 occurring when there is absolutely no mutual information between the label assignments. NMI has commonly been used to evaluate deep-learning based clustering methods, together with the clustering accuracy (Zhou et al., 2022b).

However, NMI is not corrected for chance so its value can increase merely by increasing the number of clusters used

Table 1: **Dataset overview.** We evaluate on a diverse set of experiments of differing levels of task granularity, number of classes and samples, domain shift, and class imbalance. We report the number of samples and GT classes contained in the subset of the dataset that was clustered; where possible this was the publicly available test partition (see Appendix H for more details). The class imbalance, ρ, is the ratio between the most and least frequent classes.

Type	Dataset	Reference	№ Sample	№ Class	ρ	Description
In-Domain	ImageNet-1k	Russakovsky et al. (2015)	50 000	1 000	1.00	Diverse general objects
	ImageNet-v2	Recht et al. (2019)	10 000	1 000	1.00	Diverse general objects
	CIFAR-10	Krizhevsky (2009)	10 000	10	1.00	Diverse general objects
	CIFAR-100	Krizhevsky (2009)	10 000	100	1.00	Diverse general objects
	ImageNet-9 originals	Xiao et al. (2020)	4 050	9	1.00	Diverse general objects
Domain-shift	ImageNet-9 FG-only	Xiao et al. (2020)	4 050	9	1.00	Isolated foregrounds
	ImageNet-9 MixRand	Xiao et al. (2020)	4 050	9	1.00	Remixed fore/background
	ImageNet-R	Hendrycks et al. (2021a)	30 000	200	8.43	Art/sculptures of objects
	ImageNet-Sketch	Wang et al. (2019)	50 889	1 000	1.02	Sketches of objects
Near-OOD	ImageNet-O	Hendrycks et al. (2021b)	2 000	200	6.00	Diverse general objects
	LSUN	Yu et al. (2015)	10 000	10	1.00	Urban/indoor scenes
	Places365	Zhou et al. (2018)	36 500	365	1.00	Scenes
Fine-grained	FGVC Aircraft	Maji et al. (2013)	3 333	100	1.03	Aircraft variants
	Stanford Cars	Krause et al. (2013)	8 041	196	2.83	Car variants
	Oxford Flowers	Nilsback & Zisserman (2008)	6 149	102	11.90	Flower variants
	NABirds	Van Horn et al. (2015)	24 633	555	6.67	Bird species
	BIOSCAN-1M	Gharaee et al. (2023)	24 799	2 688	782.50	Insect species
	iNaturalist-2021	Van Horn et al. (2021)	100 000	10 000	1.00	Plant & animal species
Far-OOD	CelebA	Liu et al. (2015)	19 962	1 000	32.00	Human faces (identity)
	UTKFace	Zhang et al. (2017)	5 925	101	549.00	Human faces (age)
	BreakHis	Spanhol et al. (2016)	3 164	32	8.60	Tumor tissue microscopy
	DTD	Cimpoi et al. (2014)	1 880	47	1.00	Texture descriptions
	EuroSAT	Helber et al. (2019)	4 050	10	1.50	Satellite RGB images
	MNIST	LeCun et al. (1998)	10 000	10	1.27	Handwritten digits
	Fashion MNIST	Xiao et al. (2017)	10 000	10	1.00	Clothing articles
	SVHN	Netzer et al. (2011)	26 032	10	3.20	House numbers

(Vinh et al., 2010). In order to account for this, we use the adjusted mutual information metric proposed by Vinh et al. (2010), defined as

$$\mathrm{AMI}(U,V) = \frac{\mathrm{MI}(U,V) - \mathbb{E}[\mathrm{MI}(U,V)]}{\mathrm{mean}(\mathrm{H}(U) + \mathrm{H}(V)) - \mathbb{E}[\mathrm{MI}(U,V)]}, \tag{2}$$

where $\mathbb{E}[\mathrm{MI}(U,V)]$ is the expected value of the mutual information between the considered label assignments. Similar to NMI, an AMI of 1 represents a perfect agreement between label assignments, but a score of 0 indicates the typical score for a completely random label assignment (negative AMI scores are possible).

We use the AMI metric instead of adjusted Rand index because AMI works better in the regime of unbalanced GT clusters (Romano et al., 2016), common in real-world data scenarios seen in-the-wild and true of half our evaluation datasets, but our findings would be unchanged otherwise (see Table 7).

C.2. Silhouette Score

The silhouette score, S, is a clustering measure based on the intrinsic structure of the created clusters (Rousseeuw, 1987), defined as

$$S = \frac{1}{N} \sum_{i}^{N} \frac{a_i - b_i}{\max(a_i, b_i)}, \tag{3}$$

where N is the total number of data points, a_i is the average distance between data point i and all other points assigned in the same cluster, and b_i is the average distance from i to all points in the next nearest cluster. S is bounded between -1 and 1. A score near 0 indicates that clusters are overlapping, as the data points are equally close to several clusters. A score of 1 indicates that the clusters are dense with little within-cluster distance, and thereby well-clustered. Negative values may indicate an inaccurate clustering. Since S is defined based on the relative distances of data points, it can be computed without reference to a set of ground-truth cluster assignments.

D. Encoder Training Details

The supervised encoders were obtained from the `torchvision` library. We use the weights defined in the following enums:

- ResNet-50 [recipe][1]: `torchvision.models .ResNet50_Weights.IMAGENET1K_V2`
- ViT-B [recipe][2]: `torchvision.models .ViT_B_16_Weights.IMAGENET1K_V1`

For the training details of each of the SSL encoders, please refer to their respective papers, cited accordingly in §2.1.

For our fine-tuning step, we used the method defined by He et al. (2022). When fine-tuning the ResNet architectures, we modified the method to omit the per-layer LR scaling.

E. Clustering Parameter Search Details

In order to maximize performance of each permutation of the feature encoder and clustering methods, we conducted a staggered sweep over relevant clustering parameters. This was conducted using subsets of training splits of IN-1k, Imagenette, and Imagewoof (Howard, 2019). Imagenette and Imagewoof are coarse- and fine-grained subsets of IN-1k, resp., with 10 classes each. These datasets were selected to find parameters robust against changing the number of classes and their granularity, whilst *only* optimizing clustering performance on data within the encoder's original training set. For each of these three, we created a validation set as a class-stratified random subset of the training set with the same number of samples as in the datasets' test set (50 000, 3 925, and 3 929 resp.). The same split was used across all encoders, clusterers, and stages of the parameter search.

As the curse of dimensionality can negatively affect performance of the considered clustering methods (Bellman et al., 1957), we searched for an appropriate dim. reduction process to apply before clustering. We considered using PCA (controlled either by number of reduced dimensions, or fraction of variance explained) (Pearson, 1901), UMAP (McInnes et al., 2018), and PaCMAP (Wang et al., 2021), and compared the performance to using the original (unreduced) embeddings. We found that raw images and embeddings through randomized (untrained) networks were typically best clustered when reduced with PCA. Embeddings with pretrained networks were typically best clustered with some form of manifold-based reduction. For Spectral clustering, a manifold-based reduction step is already included in its method and it benefitted from seeing the original or PCA-reduced embeddings for this process. For others, clustering was best with UMAP-reduction, and the number of reduced dimensions was unimportant across the range 5–200 dims. The findings are consistent with the idea that neural networks embed stimuli onto a low-dimensional, non-linear manifold within their embedding space.

For the full array of selected clustering parameters, see Table 2 and Table 3.

E.1. Preliminary Configuration

In our initial explorations, we found that the performance of most clusterers worked reasonably well with their default parameters, and thus initialized our search using the default parameters for the clusterers. There were three exceptions to this. (1) For Spectral Clustering, the default affinity matrix computation method was using a radial basis function which could not scale to the size of the data. We thus changed the affinity calculation method to use a graph of nearest neighbors instead, which scales better with dimensionality and number of samples. An initial search over the number of neighbors to use indicated the method would perform well across a range of values. Additionally, we changed the default label assignment method from `kmeans` to `cluster_qr`, as the latter has no tuning parameters and is known to perform consistently well. (2) For Affinity Propagation, we found that although PCA-reduced embeddings were insensitive to the choice of damping, we found that UMAP- and PaCMAP-reduced embeddings would not converge when using the default damping value of 0.5. An initial search over the damping using 20-d reduced embeddings with PCA, UMAP, and PaCMAP indicated that a damping value of 0.9 would give robust performance across all dim-reduction methods and all pretrained encoders, hence we adopted this as our default value. Furthermore, we increased the maximum number of iterations for K-Means and Affinity Propagation to 1 000 (from 300 and 200), to help ensure convergence of the algorithms. (3) For HDBSCAN, we noticed that for some encoders the clusterer would select very few clusters for Imagenette and Imagewoof, which reduced its performance. We verified, by clustering the full embeddings, that decreasing the maximum cluster size mitigated this problem. We thus set the maximum cluster size to be a generous 20% of the number of samples throughout the remainder of the search and subsequent experiments, so as to ensure HDBSCAN would not produce a degenerate solution but without forcing it to produce a certain number of clusters.

Our parameter search was conducted as a series of line-searches, in which we optimized one clustering parameter at a time. Once a parameter was optimized, it was frozen and we progressed to optimizing the next parameter. To

---

[1] `https://github.com/pytorch/vision/issues/3995#issuecomment-1013906621`

[2] `https://github.com/pytorch/vision/tree/806dba6/references/classification#vit_b_16`

begin the search, we used the default parameters of the clustering methods as defined in SCIKIT-LEARN, except for the maximum number of iterations (1 000) and the Affinity Propagation damping (0.9). For K-Means and AC, we provided the number of annotated classes within the dataset (1 000 or 10) as number of clusters to produce. Unless stated otherwise, throughout each stage of the search we took the weighted-average AMI over the three datasets, weighting ImageNet-1k twice as much as the two 10-class subsets, and selected the parameter value which yielded the highest weighted-average AMI. For AP, it was infeasible to conduct this search on IN-1k due to its compute and memory scaling w.r.t. number of samples; hence we optimized its parameters using Imagenette and Imagewoof only.

The random seed (for random, numpy, dimensionality reducer, and clusterer) was held fixed at one value (100) throughout the parameter search, and then changed to a different value (1) for the final experiments.

We used SCIKIT-LEARN (sklearn) version 1.3.1 for our search and experiments. The initial parameters used for each clusterer were as follows:

- K-Means
    - `n_clusters =` [number of GT classes]
    - `algorithm = "lloyd"`
    - `init = "k-means++"`
    - `n_init = 1`
    - `tol = 0.0001`
    - `max_iter = 1000`
    - `random_state = 100`
- Spectral Clustering
    - `n_clusters =` [number of GT classes]
    - `n_components = n_clusters`
    - `affinity = "nearest_neighbors"`
    - `assign_labels = "cluster_qr"`
    - `eigen_solver = "arpack"`
    - `eigen_tol = 0.0`
    - `n_components = None`
    - `n_neighbors = 10`
    - `random_state = 100`
- Agglomerative Clustering
    - `n_clusters =` [number of GT classes]
    - `distance_threshold = None`
    - `metric = "euclidean"`
    - `linkage = "ward"`
    - `compute_full_tree = "auto"`
- Affinity Propagation
    - `damping = 0.9`
    - `convergence_iter = 15`
    - `affinity = "euclidean"`
    - `max_iter = 1000`
    - `random_state = 100`
- HDBSCAN
    - `min_cluster_size = 5`
    - `min_samples = min_cluster_size`
    - `max_cluster_size =` [20% of the number of samples]
    - `metric = "euclidean"`
    - `cluster_selection_method = "eom"`
    - `cluster_selection_epsilon = 0.0`
    - `alpha = 1.0`
    - `algorithm = "auto"`
    - `leaf_size = 40`
    - `allow_single_cluster = False`

E.2. Dimensionality Reduction

First, as the curse of dimensionality can negatively affect the performance of the considered clustering methods (Bellman et al., 1957), we searched for an appropriate dimensionality reduction process. We compared the performance of using the original un-reduced feature embedding space (2048-d for ResNet-50, 768-d for ViT-B) against applying PCA (Pearson, 1901), UMAP (McInnes et al., 2018), or PaCMAP (Wang et al., 2021) to reduce the number of dimensions. Specifically, we considered reducing the feature embeddings to [2, 5, 10, 20, 50, 100, 200] with either PCA, UMAP, or PaCMAP. We also considered using PCA to reduce the number of dimensions to capture a target fraction of total variance of the data [0.75, 0.8, 0.85, 0.9, 0.95]; this differs from using a fixed number of dimensions as the method may select a different number of dimensions for each of the three datasets.

To perform PCA, we first took the z-score of each dimension and then used the default parameters of SCIKIT-LEARN (Pedregosa et al., 2011), without whitening the data.

To perform UMAP, we set the number of neighbours considered to 30 (increased from the default of 15) and set the minimum distance to 0.0 (decreased from the default of 0.1), following the recommendations of McInnes (2018); we otherwise used the default parameters of UMAP (McInnes et al., 2018). The dimensionality reduction distance metric was always set to euclidean (ℓ_2), irrespective of the distance metric used by the downstream clusterer.

To perform PaCMAP, used the authors' implementation, disabling the PCA-reduction preprocessing step but otherwise using the default parameters. The default number of neighbors was automatically determined from the size of the dataset as $10 + \max(0, 15\,(\log_{10}(N) - 4))$. We found the performance of PaCMAP was consistently worse than UMAP, and so we also considered setting the number of

Table 2: **Clustering parameters for raw images and ResNet-50 encoders.** We present the parameters discovered by our search on ImageNet train data, and subsequently used throughout our main experiments as presented in the paper. Some parameters are specific to particular clusterers and hence do not have a value for the other clusterers. The dimension reduction value indicates the number of reduced dimensions if larger than 1, or the target variance explained if less than 1. The parameters for AC w/ C were the same as for AC w/o C, except the distance threshold was not specified, instead being automatically determined from the target number of clusters. Continued in Table 3.

							Agg. Clustering		Spectral	Aff. Prop.
Arch.	Encoder	FT	Clusterer	Dim Reduction		Metric	Linkage	Dist. Thr.	№ Neigh.	Damping
—	Raw image		K-Means	PCA	0.90	–	–	–	–	–
			Spectral	None		–	–	–	10	–
			AC w/o C	PCA	200	cosine	average	0.71	–	–
			Affinity Prop	PCA	0.80	–	–	–	–	0.85
			HDBSCAN	UMAP	50	$L2$	–	–	–	–
RN50	Rand.		K-Means	PCA	0.95	–	–	–	–	–
			Spectral	PCA	200	–	–	–	50	–
			AC w/o C	PCA	200	$L\infty$	average	10.00	–	–
			Affinity Prop	PCA	0.90	–	–	–	–	0.90
			HDBSCAN	UMAP	50	$L1$	–	–	–	–
	X-Ent.		K-Means	UMAP	50	–	–	–	–	–
			Spectral	None		–	–	–	20	–
			AC w/o C	UMAP	50	$L2$	ward	2.00	–	–
			Affinity Prop	UMAP	50	–	–	–	–	0.90
			HDBSCAN	UMAP	50	$L2$	–	–	–	–
	MoCo-v3		K-Means	UMAP	50	–	–	–	–	–
			Spectral	None		–	–	–	30	–
			AC w/o C	UMAP	50	$L2$	ward	10.00	–	–
			Affinity Prop	UMAP	50	–	–	–	–	0.75
			HDBSCAN	UMAP	50	$L2$	–	–	–	–
	DINO		K-Means	UMAP	50	–	–	–	–	–
			Spectral	PCA	0.80	–	–	–	10	–
			AC w/o C	UMAP	50	$L2$	average	0.50	–	–
			Affinity Prop	UMAP	50	–	–	–	–	0.90
			HDBSCAN	UMAP	50	$L1$	–	–	–	–
	VICReg		K-Means	UMAP	50	–	–	–	–	–
			Spectral	None		–	–	–	10	–
			AC w/o C	UMAP	50	$L2$	average	0.50	–	–
			Affinity Prop	UMAP	50	–	–	–	–	0.80
			HDBSCAN	UMAP	50	$L1$	–	–	–	–
	MoCo-v3	✓	K-Means	UMAP	50	–	–	–	–	–
		✓	Spectral	None		–	–	–	30	–
		✓	AC w/o C	UMAP	50	$L2$	ward	2.00	–	–
		✓	Affinity Prop	UMAP	50	–	–	–	–	0.95
		✓	HDBSCAN	UMAP	50	$L\infty$	–	–	–	–
	DINO	✓	K-Means	UMAP	50	–	–	–	–	–
		✓	Spectral	PCA	0.80	–	–	–	20	–
		✓	AC w/o C	UMAP	50	$L2$	ward	2.00	–	–
		✓	Affinity Prop	UMAP	50	–	–	–	–	0.90
		✓	HDBSCAN	UMAP	50	$L1$	–	–	–	–
	VICReg	✓	K-Means	UMAP	50	–	–	–	–	–
		✓	Spectral	None		–	–	–	20	–
		✓	AC w/o C	UMAP	50	$L2$	ward	2.00	–	–
		✓	Affinity Prop	UMAP	50	–	–	–	–	0.90
		✓	HDBSCAN	UMAP	50	$L2$	–	–	–	–

Table 3: **Clustering parameters for ViT-B encoders.** Continues Table 2 to show parameters for ViT-B encoders. For MAE with Spectral Clustering, we found standardizing the data with a z-score (and not applying PCA) yielded the best performance.

						Agg. Clustering		Spectral	Aff. Prop.
Arch.	Encoder	FT	Clusterer	Dim Reduction	Metric	Linkage	Dist. Thr.	№ Neigh.	Damping
ViT-B	Rand.		K-Means	PCA 100	–	–	–	–	–
			Spectral	PCA 0.95	–	–	–	50	–
			AC w/o C	PCA 0.85	$L\infty$	average	2.00	–	–
			Affinity Prop	PCA 0.90	–	–	–	–	0.95
			HDBSCAN	UMAP 50	$L1$	–	–	–	–
	X-Ent.		K-Means	UMAP 50	–	–	–	–	–
			Spectral	PCA 0.70	–	–	–	30	–
			AC w/o C	UMAP 50	$L2$	ward	2.00	–	–
			Affinity Prop	UMAP 50	–	–	–	–	0.90
			HDBSCAN	UMAP 50	$L\infty$	–	–	–	–
	MoCo-v3		K-Means	UMAP 50	–	–	–	–	–
			Spectral	PCA 0.85	–	–	–	50	–
			AC w/o C	UMAP 50	$L\infty$	average	1.00	–	–
			Affinity Prop	UMAP 50	–	–	–	–	0.75
			HDBSCAN	UMAP 50	$L2$	–	–	–	–
	DINO		K-Means	UMAP 50	–	–	–	–	–
			Spectral	PCA 0.90	–	–	–	10	–
			AC w/o C	UMAP 50	$L2$	average	0.20	–	–
			Affinity Prop	UMAP 50	–	–	–	–	0.85
			HDBSCAN	UMAP 50	$L1$	–	–	–	–
	MAE (CLS)		K-Means	PCA 0.95	–	–	–	–	–
			Spectral	z-score only	–	–	–	10	–
			AC w/o C	PCA 0.90	cosine	average	0.71	–	–
			Affinity Prop	PCA 200	–	–	–	–	0.60
			HDBSCAN	PCA 0.95	$L2$	–	–	–	–
	MAE (avg)		K-Means	PCA 0.90	–	–	–	–	–
			Spectral	z-score only	–	–	–	30	–
			AC w/o C	PCA 0.85	cosine	average	0.71	–	–
			Affinity Prop	UMAP 50	–	–	–	–	0.60
			HDBSCAN	UMAP 50	$L1$	–	–	–	–
	MoCo-v3	✓	K-Means	UMAP 50	–	–	–	–	–
		✓	Spectral	PCA 0.95	–	–	–	50	–
		✓	AC w/o C	UMAP 50	$L2$	ward	2.00	–	–
		✓	Affinity Prop	UMAP 50	–	–	–	–	0.95
		✓	HDBSCAN	UMAP 50	$L\infty$	–	–	–	–
	DINO	✓	K-Means	UMAP 50	–	–	–	–	–
		✓	Spectral	PCA 0.90	–	–	–	50	–
		✓	AC w/o C	UMAP 50	$L2$	ward	2.00	–	–
		✓	Affinity Prop	UMAP 50	–	–	–	–	0.90
		✓	HDBSCAN	UMAP 50	$L\infty$	–	–	–	–
	MAE (avg)	✓	K-Means	UMAP 50	–	–	–	–	–
		✓	Spectral	PCA 0.75	–	–	–	50	–
		✓	AC w/o C	UMAP 50	$L2$	ward	2.00	–	–
		✓	Affinity Prop	UMAP 50	–	–	–	–	0.90
		✓	HDBSCAN	UMAP 50	$L\infty$	–	–	–	–

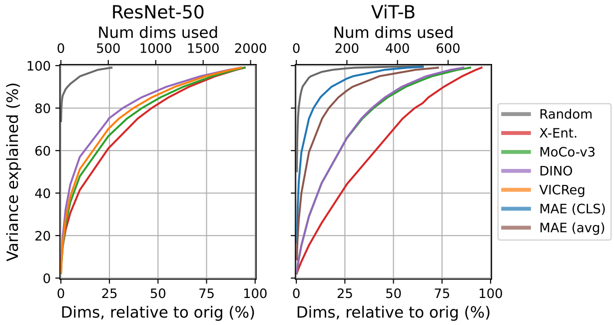


Figure 3: **Percentage of variance explained by PCA-reduced embeddings**. We show the fraction of the total variance of the data which is explained by the first N PCA dimensions. The number of dimensions included is represented both in absolute terms (upper x-axes) and relative to the number of dimensions of the original embeddings (lower x-axes).

neighbours to 30 to match UMAP; however this did not lead to a significant change in its performance.

For raw images and randomly initialized (untrained) networks, we found that PCA reduction typically performed best (see Table 2 and Table 3), and was optimal with relatively large number of dimensions (at least 100), as a large number of dimensions was needed to capture the majority of the variance of the data (shown in Fig. 3). However, the majority of trained encoders performed best with UMAP-reduced embeddings and were insensitive to the choice of dimension, with minimal change in mean AMI across the range 5 to 200. Thus for consistency, we selected a 50-dim UMAP reduction for all encoders/clusterers where UMAP performed best. The MAE-trained ViT-B encoder bucked this trend and performed poorly with UMAP reduction across all clusterers (and all three datasets), yielding better performance when using PCA instead. This was true for both the CLS token embedding (which was not connected to any loss during the training of the network), and when taking the average over all the embeddings of patch tokens (for some but not all clusterers).

For Spectral Clustering, we found using PCA-reduced or unreduced embeddings as the input to the clusterer yielded better performance than using UMAP-reduced embeddings. This is because the Spectral Clustering methodology already includes a manifold-based dimensionality reduction step (taking the eigenvalues of the neighborhood-based affinity matrix) as part of its pipeline. Performing both UMAP and Spectral dimensionality reduction reduced the performance, as UMAP is not distance-preserving.

These results emphasize that the output of untrained networks are distributed amorphously, as is the case for the raw stimuli in pixel-space, whereas the output of encoders trained on a whole-stimulus task lie on a low-dimensional manifold which can be discovered by manifold-based dimensionality reduction methods such as UMAP or Spectral Clustering, but not linear methods such as PCA. Hence using manifold-based dimensionality reduction provides the best clustering results, even for clusterers which rely on distance metrics between pairs of samples and despite the fact these distance metrics are not preserved by the dimensionality reduction. Meanwhile, for encoders trained on a local-feature task—MAE (avg)—the output is somewhere in between the two, with PCA and UMAP reduced embeddings giving comparable performance.

In the subsequent stages of the parameter search, we iterated over the per-method specific parameters, whilst using the dimensionality reductions per encoder selected in this stage.

E.3. K-Means

For K-Means, we did not optimize any parameters other than the dimensionality reduction method. We used the `kmeans++` initialization (Arthur & Vassilvitskii, 2007) throughout our experiments, with 1 initialization per clustering.

E.4. Spectral Clustering

For Spectral Clustering, we optimized the number of neighbors used when building the affinity matrix over the search space [5, 10, 20, 30, 50, 100]. We found the performance was not very sensitive to the neighborhood size, with optimal values in the range 10–50 depending on the encoder.

After fixing the neighborhood size, we investigated the effect of the number of eigenvectors (components). We found the number of components which yielded the best performance varied greatly between Imagenette/Imagewoof and ImageNet-1k (10, and 100–1 000, respectively). As this was around the same as the target number of clusters, which was the default parameter, we retained the default behaviour.

E.5. Affinity Propagation

For Affinity Propagation, we optimized the damping parameter over the search space [0.5, 0.6, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95, 0.98]. Then, after freezing the amount of damping, we investigated the effect of the convergence stopping threshold over the search space [5, 8, 10, 15, 20, 25, 30]. We found the performance was insensitive to the stopping threshold, and so froze it at the default value of 15 for all encoders.

E.6. HDBSCAN

For HDBSCAN, we investigated the effect of the distance metric and cluster selection method jointly. We considered distance metrics $\{L1, L2, L\infty\}$, and both the excess of mass (eom) and leaf selection methods. We found the eom selection method universally outperformed leaf in terms of AMI, and there was minimal effect from the choice of distance metric.

We used the default minimum cluster size of 5 throughout our search and consequently also for the majority of our experiments. However, for BIOSCAN-1M, CelebA, and UTKFace (where some classes have only 1 or 2 occurrences in the test set) we reduced the minimum cluster size to 2.

E.7. Agglomerative Clustering

For AC, continuing to use the "ground-truth" number of classes as the number of clusters, we evaluated all combinations of distance metric $\{L1, L2, L\infty, \text{cosine}\}$ and linkage method {ward ($L2$ only), complete, average, single}, for 13 options in total. For each encoder, we selected the metric and linkage which yielded the best weighted-average AMI over the three datasets. This selection completed the parameter options to use for AC w/ C.

Finally, for AC w/o C, we selected the distance threshold to use for each encoder. The distance threshold provides an alternative stopping criteria for AC so it does not need to know the number of clusters *a priori*. To make the distance threshold more likely to be comparable across embeddings from different datasets, after dimensionality reduction we standardized the embeddings by subtracting the mean of each dimension and dividing by the average standard-deviation across all dimensions. This spherically rescales the distances of the space without stretching dimensions relative to each other and thus without changing the relative importance of each dimension toward the distance between samples. We also divided by the number of dimensions for encoders where the $L1$ metric was selected, or by the square-root of the number of dimensions for encoders where the $L2$ metric was selected. This process keeps the expected distance between samples similar even if the dimensionality differed between reduced embeddings.

For each encoder, we fit the clusterer on each of the 3 datasets for 21 distance thresholds sampled logarithmically from 0.001 to 5000.0. For each of the three datasets, we scaled the values across the distance thresholds relative to the maximum AMI to make them more comparable—since the AMI falls to 0 if the distance threshold is too high (only one cluster) or too low (every sample in its own cluster), rescaling the AMI in this way gives each dataset the same dynamic range. We then selected the distance threshold which yielded the highest weighted-average relative-AMI.

We found that embeddings which had been reduced with UMAP had a broad curve for the distance threshold, but PCA-reduced embeddings were highly sensitive to the distance threshold with a narrow peak across only a pair of values in our search grid. Because of this, we refined the search for the distance threshold on PCA-reduced embeddings at twice the resolution before picking the best distance threshold value.

F. Clustering Raw Images

We investigated using embeddings from randomized ResNet-50 and ViT-B networks, or using the raw image pixels, but across all datasets the performance of these was negligible and did not serve as a worthwhile baseline comparator (see Appendix H).

To cluster the raw images, we used an image size of 32×32×3 throughout our parameter search, reduced by resizing the shortest side to 32 pixels and cropping to a square. For the final experiments, we used the same process, except for MNIST and Fashion-MNIST which have smaller images than this and hence we dimensionality-reduced them starting from 28×28×3 images.

G. Computational Resource Requirements

In this section, we describe the computational requirements of our experiments. All experiments were performed on a compute cluster with the job utilizing two CPU cores (2x Intel Xeon Gold 6148 CPU @ 2.40GHz).

The amount of memory used per job varied depending on the demands of the clusterer and the size of the dataset. An upper-bound for the memory requirements of each experiment is shown in Table 4.

The total runtime of our parameter search was 4.9 years. The total runtime of the clustering results shown in the main figures (1, 2, etc) and tables (6–17) was 351 days. Including auxiliary results, preliminary experiments, and otherwise discarded experiments, the total runtime of the CPU-only clustering steps for this project was 7.6 years. Typical runtimes for each clusterer and dataset are shown in Table 5.

The fine-tuning of the SSL encoders in §3.2 was conducted on two Nvidia A40 GPUs following the MAE fine-tuning schedule from He et al. (2022). Each training run took approximately 43 hours, resulting in a total of 11 GPU compute days.

Table 4: **Memory requirements (GB).** We indicate an upper-bound on the amount of RAM required, in GB, to cluster the test set of each dataset using each clustering method.

Dataset	KMeans	Spectral	Agglom.	Aff. Prop	HDBSCAN
Imagenette	1	2	2	1	1
Imagewoof	1	2	2	1	1
ImageNet-1k	4	20	20	72	4
ImageNet-v2	2	6	6	6	2
CIFAR-10	2	6	6	6	2
CIFAR-100	2	6	6	6	2
ImageNet-9 (+vars.)	2	4	4	2	2
ImageNet-R	4	16	16	48	4
ImageNet-S	4	20	20	72	4
ImageNet-O	1	2	2	1	1
LSUN	2	6	6	6	2
Places365	4	16	16	48	4
FGVC Aircraft	1	2	2	1	1
Stanford Cars	2	6	6	6	2
Oxford Flowers 102	2	4	4	2	2
BIOSCAN-1M	4	16	16	48	4
NABirds	4	16	16	48	4
iNaturalist-2021	6	72	72	292	6
CelebA	4	12	12	12	4
UTKFace	2	4	4	2	2
BreakHis	1	2	2	1	1
DTD	1	2	2	1	1
EuroSAT	2	4	4	2	2
MNIST	2	6	6	6	2
FashionMNIST	2	6	6	6	2
SVHN	4	16	16	48	4

Table 5: **Clustering job runtime.** For each clustering method, we show the runtime of the clustering process (including dimensionality reduction, as applicable) on each dataset in seconds, minutes, or hours. We take the median value across all encoders, excluding raw pixel and randomized (untrained) networks. See Appendix H for dataset abbreviations. Background: from fastest (white) to slowest (red) per dataset.

	In-domain					Domain-shift				Near-OOD			Fine-grained						Far-OOD							
Clusterer	IN1k	INv2	C10	C100	IN9	9-FG	9-MR	IN-R	IN-S	IN-O	LSU	P365	Air	Cars	F102	Bio	Birds	iNat	CelA	UTKF	BHis	DTD	ESAT	MNST	Fash	SVHN
K-Means	9.0 h	19.2 min	20.0 min	19.4 min	3.4 min	3.2 min	4.6 min	4.1 h	11.2 h	1.0 min	1.9 min	5.3 h	2.7 min	11.3 min	9.1 min	2.9 h	2.5 h	31.6 h	1.6 h	7.1 min	2.0 min	62.7 s	3.1 min	16.3 min	18.5 min	2.4 h
Spectral	22.2 h	49.0 min	37.0 min	55.5 min	6.8 min	6.6 min	7.3 min	9.0 h	28.4 h	1.6 min	3.3 min	18.7 h	5.1 min	27.0 min	13.6 min	8.0 h	5.1 h	DNF	4.4 h	14.5 min	2.5 min	90.9 s	7.2 min	25.8 min	25.6 min	4.7 h
AC w/ C	7.8 h	18.7 min	18.8 min	18.0 min	3.7 min	3.1 min	3.4 min	3.7 h	8.3 h	1.1 min	1.7 min	5.3 h	2.4 min	9.0 min	8.7 min	2.0 h	1.9 h	18.7 h	1.3 h	6.1 min	1.8 min	54.9 s	2.8 min	14.3 min	18.3 min	2.2 h
AC w/o C	9.2 h	17.3 min	17.2 min	14.8 min	3.5 min	3.4 min	4.1 min	3.0 h	9.5 h	1.1 min	2.4 min	5.5 h	2.3 min	12.4 min	8.1 min	2.5 h	2.0 h	27.3 h	1.4 h	6.0 min	1.9 min	62.6 s	3.0 min	18.9 min	18.0 min	2.3 h
Affinity Prop	10.4 h	18.6 min	22.1 min	19.7 min	5.4 min	4.0 min	3.5 min	3.8 h	11.3 h	1.0 min	2.3 min	5.5 h	2.6 min	15.3 min	9.0 min	2.5 h	2.5 h	43.8 h	1.2 h	7.9 min	2.3 min	67.6 s	3.8 min	23.2 min	23.4 min	1.9 h
HDBSCAN	5.3 h	15.9 min	11.3 min	11.6 min	4.3 min	3.0 min	3.5 min	2.2 h	6.4 h	1.0 min	1.8 min	3.6 h	1.6 min	10.0 min	6.9 min	1.9 h	1.3 h	14.9 h	1.0 h	7.7 min	1.5 min	66.3 s	2.6 min	12.7 min	11.3 min	1.0 h

H. AMI Results for Individual Datasets

In this section, we tabulate the results for clustering the embeddings of each of the test datasets used in the main results (26 datasets) with each of the encoders (raw images, 2 random networks, 2 supervised encoders, 7 SSL encoders, 6 SSL+FT encoders; for 18 encoders total), using each of the clustering methods (6; counting both AC w/ and w/o C). This yields a total of 2 808 clustering results.

For each dataset, we clustered the images from the test partition only. In cases where there is no public test partition, but there is a public validation partition (e.g. ImageNet-1k), we evaluated the clustering on the validation partition. For BIOSCAN-1M, we use the splits from Gong et al. (2024) and evaluate on the union of their key partitions and test partitions. For some datasets, no partitioning is indicated in the dataset release, and we partitioned these as follows. For BreakHis, we used a random test split of 40% of the data, stratified on the joint distribution of tumor type and image magnification. For EuroSAT, we used a stratified random test split of 15% of the data. For UTKFace, we use a random test split of 25% of the data, stratified over age and approximately stratified over age and gender within this.

The datasets used in the experiments, described in Table 1, are abbreviated here as follows:

- In-domain
  - IN1k: ImageNet-1k (ILSVRC 2012)
  - INv2: ImageNet-v2
  - C10: CIFAR-10
  - C100: CIFAR-100
  - IN9: ImageNet-9 – Original images
- Domain-shift
  - 9-FG: ImageNet-9 – Foreground-only
  - 9-MR: ImageNet-9 – Mixed-random
  - IN-R: ImageNet-Rendition
  - IN-S: ImageNet-Sketch
- Near-OOD
  - IN-O: ImageNet-O
  - LSU: Large-scale Scene Understanding (LSUN)
  - P365: Places365
- Fine-grained
  - Air: FGVC Aircraft
  - Cars: Stanford Cars
  - F102: Oxford Flowers 102
  - Bio: BIOSCAN-1M
  - Birds: NABirds
  - iNat: iNaturalist-2021
- Far-OOD
  - CelA: CelebA
  - UTKF: UTKFace
  - BHis: BreakHis
  - DTD: Describable Textures Dataset
  - ESAT: EuroSAT
  - MNST: Modified National Institute of Standards and Technology database (MNIST)
  - Fash: Fashion-MNIST
  - SVHN: Street View House Numbers

Table 6: **Adjusted mutual information, (AMI; %) averaged over clusterers.** These results are used to create the figures of the main paper. The AMI shown is averaged over K-Means, Spectral, AC w/ C, AC w/o C, AP, and HDBSCAN results (except iNat, which excludes Spectral). See Tables 8–13 for individual clusterers. See Appendix H for dataset abbreviations. **Bold**: best encoder per dataset (or within 0.5 of best). Underlined: best encoder per architecture (or within 0.5 best). Background: from median AMI (white) to max (blue) per dataset.

		In-domain					Domain-shift				Near-OOD			Fine-grained						Far-OOD							
Encoder	FT	IN1k	INv2	C10	C100	IN9	9-FG	9-MR	IN-R	IN-S	IN-O	LSU	P365	Air	Cars	F102	Bio	Birds	iNat	CelA	UTKF	BHis	DTD	ESAT	MNST	Fash	SVHN
Raw image		2	1	9	10	5	5	1	2	9	6	6	4	4	2	16	8	3	1	4	3	13	5	21	54	48	2
RN50 — Rand.		2	1	4	7	3	6	2	2	9	4	4	3	3	2	13	9	2	0	2	1	14	4	22	37	37	1
X-Ent.		70	45	57	50	69	**70**	60	34	40	60	58	37	15	22	62	21	40	12	8	10	26	47	66	71	63	6
MoCo-v3		47	26	57	48	70	61	48	26	36	37	52	32	19	11	76	24	29	8	**11**	**11**	30	49	68	**82**	**64**	**12**
DINO		44	24	43	40	70	64	43	18	28	36	50	34	16	13	78	29	21	7	11	11	**43**	51	74	70	61	3
VICReg		45	26	46	43	69	63	40	20	31	38	50	32	14	12	78	28	21	8	**11**	**11**	36	52	**76**	75	64	5
MoCo-v3	✓	69	46	59	50	**77**	**70**	**64**	35	42	67	60	37	17	22	59	17	40	11	8	10	22	49	60	58	63	7
DINO	✓	69	46	57	50	75	68	62	34	41	**68**	56	36	16	21	59	17	40	11	8	9	22	47	62	67	62	5
VICReg	✓	68	45	57	49	75	67	**64**	33	39	66	57	36	15	22	62	18	40	11	8	10	22	50	61	60	**64**	6
ViT-B — Rand.		3	1	9	9	6	4	1	3	8	6	8	5	3	2	18	12	3	1	3	2	15	8	25	14	20	0
X-Ent.		75	54	76	63	61	61	51	38	43	**67**	62	39	18	24	67	21	38	12	8	9	25	45	61	73	63	3
MoCo-v3		57	35	72	60	62	62	44	26	36	36	60	37	14	11	78	29	29	10	10	**11**	37	55	70	75	58	3
DINO		64	42	66	60	72	68	61	33	42	44	63	**40**	**20**	13	**88**	**32**	44	14	**11**	10	40	**58**	74	74	63	2
MAE (CLS)		21	11	26	24	38	39	18	10	23	17	29	18	9	5	45	15	12	4	7	6	28	29	49	56	50	2
MAE (avg)		23	11	29	27	44	41	15	10	23	19	36	22	8	6	54	13	11	3	8	8	30	37	52	56	48	1
MoCo-v3	✓	77	57	**77**	**67**	64	53	52	**44**	48	61	**64**	**40**	19	27	70	21	44	14	9	10	24	48	57	70	**64**	2
DINO	✓	**79**	**58**	71	61	65	53	53	43	47	58	63	39	19	26	66	19	42	14	9	9	24	45	57	67	63	2
MAE (avg)	✓	**78**	**58**	73	61	66	57	54	**44**	**48**	64	63	**40**	18	**28**	75	22	**46**	**15**	10	10	26	49	63	71	**64**	3

Table 7: **Adjusted Rand index (ARI; %), averaged over clusterers.** The ARI reported is averaged over K-Means, Spectral, AC w/ C, AC w/o C, AP, and HDBSCAN results (except iNat, which excludes Spectral). The magnitude of the values differs from the AMI reported in Table 6, but the trends across encoders is the same.

		In-domain					Domain-shift				Near-OOD			Fine-grained						Far-OOD							
Encoder	FT	IN1k	INv2	C10	C100	IN9	9-FG	9-MR	IN-R	IN-S	IN-O	LSU	P365	Air	Cars	F102	Bio	Birds	iNat	CelA	UTKF	BHis	DTD	ESAT	MNST	Fash	SVHN
Raw image		0	0	3	1	1	2	0	0	1	1	2	0	1	0	3	1	0	0	0	1	3	1	9	35	26	0
RN50 — Rand.		0	0	1	1	1	2	0	0	1	1	1	0	0	0	2	1	0	0	0	0	3	1	8	18	16	−0
X-Ent.		33	19	39	21	42	46	35	12	11	35	41	11	4	6	36	5	12	**2**	2	3	6	23	49	54	43	2
MoCo-v3		15	8	39	18	50	38	23	6	8	16	34	7	**5**	2	51	6	6	1	**2**	3	9	23	58	**74**	**48**	**4**
DINO		11	7	25	13	49	41	19	3	4	16	31	7	4	2	51	9	3	1	**2**	5	**16**	26	65	56	43	1
VICReg		11	7	28	15	47	40	19	4	5	16	33	7	3	2	53	9	3	1	**2**	**5**	12	26	**67**	65	**48**	1
MoCo-v3	✓	32	19	41	21	**55**	**49**	**44**	12	12	41	43	10	**5**	6	33	4	12	1	1	3	5	26	41	39	44	2
DINO	✓	31	18	38	21	53	45	39	11	11	43	36	10	4	6	34	4	11	1	2	3	5	24	44	51	41	2
VICReg	✓	30	18	37	21	53	43	42	10	10	40	40	10	4	6	37	4	11	1	2	3	5	27	44	41	47	2
ViT-B — Rand.		0	0	3	1	2	1	0	0	0	1	3	0	0	0	4	2	0	0	0	1	2	2	10	4	9	0
X-Ent.		41	25	65	33	27	33	24	13	12	**46**	48	10	**5**	6	42	5	11	**1**	1	3	6	23	43	58	45	0
MoCo-v3		19	13	58	29	37	36	18	6	8	15	45	10	3	2	54	9	6	1	**2**	**5**	11	31	55	61	44	1
DINO		23	15	46	28	47	42	31	8	10	22	49	11	**5**	3	**72**	**11**	11	**1**	**2**	4	12	**34**	61	57	44	0
MAE (CLS)		4	3	13	6	19	18	8	1	3	6	14	3	2	1	20	4	2	0	1	2	8	11	29	33	27	0
MAE (avg)		5	3	14	7	22	19	6	1	3	6	20	3	1	1	29	2	1	0	1	3	9	14	33	39	31	0
MoCo-v3	✓	42	27	**67**	**39**	30	17	25	17	**15**	39	**51**	**12**	**5**	**7**	45	5	**14**	**2**	**2**	3	6	25	38	53	45	1
DINO	✓	**45**	**28**	59	32	32	19	27	17	**16**	35	48	**11**	**5**	7	41	5	13	**2**	**2**	3	6	23	39	48	45	1
MAE (avg)	✓	44	**28**	62	32	35	21	26	**18**	**16**	42	49	**12**	4	**8**	50	6	**14**	**2**	**2**	3	7	25	45	56	45	1

Table 8: **AMI score (%) using K-Means.**

		In-domain					Domain-shift				Near-OOD			Fine-grained						Far-OOD								
Encoder	FT	IN1k	INv2	C10	C100	IN9	9-FG	9-MR	IN-R	IN-S	IN-O	LSU	P365	Air	Cars	F102	Bio	Birds	iNat	CelA	UTKF	BHis	DTD	ESAT	MNST	Fash	SVHN
Raw image		2	1	8	13	5	5	1	4	11	5	5	4	4	3	19	7	3	0	4	3	16	6	25	44	50	0
RN50 — Rand.		2	1	2	9	2	6	2	3	11	4	3	3	3	1	15	9	2	0	2	2	16	5	23	34	41	0
X-Ent.		73	48	68	51	84	80	72	34	42	63	63	39	15	23	64	20	39	9	8	11	28	48	76	81	69	5
MoCo-v3		48	25	64	51	78	68	49	27	38	41	57	33	**21**	12	80	23	28	4	**10**	12	33	52	70	**86**	**71**	**11**
DINO		44	22	49	42	78	70	47	18	29	37	57	35	18	13	82	27	18	4	9	**12**	**44**	52	79	74	64	1
VICReg		46	24	53	45	81	71	45	21	33	38	55	33	16	12	81	26	18	4	**10**	**12**	38	52	80	80	70	3
MoCo-v3	✓	73	49	66	51	**88**	80	**77**	36	44	72	65	38	17	22	61	17	39	8	7	11	23	50	69	65	71	7
DINO	✓	72	49	67	52	86	75	75	34	42	**74**	63	38	15	22	60	16	39	8	7	11	24	49	71	76	70	5
VICReg	✓	71	48	66	51	86	73	73	33	41	71	62	38	15	22	63	17	38	8	7	11	23	51	68	71	71	5
ViT-B — Rand.		2	1	9	10	6	3	1	3	11	5	8	5	3	2	19	14	3	0	2	3	17	8	25	13	22	0
X-Ent.		79	59	**83**	65	64	73	61	39	45	71	67	39	18	25	68	21	38	8	7	10	26	45	65	80	70	1
MoCo-v3		60	36	79	62	79	69	45	26	38	37	64	39	15	12	81	28	27	6	9	**12**	40	55	79	83	71	1
DINO		67	44	77	62	**88**	**81**	68	34	43	45	66	**41**	**21**	13	**89**	**31**	44	9	**10**	11	43	**60**	**84**	81	69	1
MAE (CLS)		21	9	29	29	42	38	17	11	25	17	35	21	10	5	47	15	10	1	6	7	31	31	53	45	55	1
MAE (avg)		22	9	32	29	41	31	1	12	23	19	44	23	8	6	53	12	9	1	7	8	30	37	49	42	55	1
MoCo-v3	✓	81	61	81	**69**	60	48	47	44	49	66	67	41	19	26	72	20	43	10	8	11	24	49	63	78	71	1
DINO	✓	**82**	**63**	80	62	71	47	53	43	49	62	**68**	40	18	26	67	18	41	10	8	11	25	45	63	78	70	2
MAE (avg)	✓	**82**	**63**	79	63	76	57	55	**45**	**51**	68	67	41	18	**28**	77	21	**44**	**10**	9	11	27	48	68	82	**71**	2

Table 9: **AMI score (%) using Spectral Clustering.**

		In-domain					Domain-shift				Near-OOD			Fine-grained						Far-OOD								
Encoder	FT	IN1k	INv2	C10	C100	IN9	9-FG	9-MR	IN-R	IN-S	IN-O	LSU	P365	Air	Cars	F102	Bio	Birds	iNat	CelA	UTKF	BHis	DTD	ESAT	MNST	Fash	SVHN
Raw image		2	1	9	12	6	6	1	4	11	4	8	5	5	2	18	6	3	–	4	3	15	5	27	70	60	1
RN50 — Rand.		2	0	3	10	3	7	1	3	10	4	3	4	3	2	16	7	2	–	2	2	17	6	25	53	50	0
X-Ent.		73	48	64	52	59	75	53	37	42	64	63	39	16	21	62	18	40	–	9	11	25	48	66	70	67	5
MoCo-v3		51	28	61	51	65	63	48	30	39	40	55	35	20	13	75	21	29	–	11	**11**	30	51	66	**83**	69	**12**
DINO		47	25	47	45	71	68	47	23	33	39	54	37	20	14	76	24	23	–	10	10	**39**	52	72	74	66	2
VICReg		48	26	53	46	63	59	40	24	35	41	54	35	17	12	77	23	22	–	**12**	**11**	34	53	73	75	68	5
MoCo-v3	✓	72	49	67	52	**84**	**75**	62	37	43	71	62	38	15	21	58	15	40	–	8	11	23	50	59	54	69	7
DINO	✓	72	49	65	54	76	69	60	37	42	**72**	58	38	14	20	58	15	38	–	7	9	22	47	61	67	69	6
VICReg	✓	71	48	64	51	79	71	**67**	35	41	70	59	38	16	22	61	16	39	–	8	11	22	51	59	57	68	5
ViT-B — Rand.		2	1	10	10	6	4	1	4	10	5	9	5	3	2	20	11	2	–	2	3	19	9	30	19	25	0
X-Ent.		79	56	**79**	64	44	47	36	39	42	70	63	39	18	22	65	17	38	–	8	10	24	49	70	68	65	2
MoCo-v3		61	38	77	62	67	72	51	29	39	38	61	40	14	11	73	24	27	–	10	**12**	33	55	**75**	70	69	3
DINO		67	46	69	62	59	61	54	38	44	48	63	**42**	**23**	16	**88**	**28**	**46**	–	**13**	11	37	**57**	74	76	68	1
MAE (CLS)		27	12	36	34	51	56	28	14	28	19	36	24	11	5	55	15	12	–	7	7	34	36	57	81	66	0
MAE (avg)		27	12	34	34	54	49	25	14	25	20	45	25	8	6	59	12	9	–	8	8	31	40	58	68	64	1
MoCo-v3	✓	81	59	**79**	**68**	66	47	45	**44**	**51**	60	64	40	18	25	67	19	45	–	9	11	23	50	56	66	**70**	2
DINO	✓	**82**	**60**	73	63	63	47	48	44	50	58	65	40	17	25	65	17	43	–	9	11	24	48	57	60	65	1
MAE (avg)	✓	**82**	58	77	63	65	49	43	43	48	62	**68**	41	18	**26**	73	18	44	–	10	**11**	25	52	69	58	69	3

Table 10: **AMI score (%) using Agglomerative Clustering with number of clusters given (AC w/ C).** In this configuration, the target number of clusters is provided to AC (set to the number of classes in the GT annotations) and the distance threshold is automatically selected to split the hierarchy into the target number of clusters.

		In-domain					Domain-shift				Near-OOD			Fine-grained						Far-OOD							
Encoder	FT	IN1k	INv2	C10	C100	IN9	9-FG	9-MR	IN-R	IN-S	IN-O	LSU	P365	Air	Cars	F102	Bio	Birds	iNat	CelA	UTKF	BHis	DTD	ESAT	MNST	Fash	SVHN
Raw image		4	3	7	8	3	4	0	1	7	8	6	5	4	2	15	11	3	2	7	3	9	3	15	51	45	0
RN50 — Rand.		2	1	2	6	3	6	1	2	9	4	3	4	3	2	13	11	3	0	3	1	9	3	21	23	24	0
X-Ent.		73	49	67	52	83	**81**	70	34	43	65	64	39	15	23	64	20	39	9	8	11	28	48	75	82	69	4
MoCo-v3		49	26	64	51	78	66	54	27	39	41	57	33	20	12	81	24	28	5	10	12	33	52	72	**87**	70	**10**
DINO		48	27	48	42	76	67	47	18	31	38	57	38	19	15	82	34	21	7	12	**13**	**46**	52	78	74	67	1
VICReg		49	28	53	46	79	69	38	20	34	40	55	35	16	13	82	33	23	7	13	**13**	40	52	81	81	69	2
MoCo-v3	✓	73	50	67	52	**88**	80	**79**	35	45	72	63	38	16	23	61	17	40	8	7	11	23	50	66	66	70	7
DINO	✓	72	49	66	52	86	80	73	34	43	**74**	63	38	16	23	61	16	39	8	7	11	23	48	68	76	69	5
VICReg	✓	71	48	65	50	83	74	76	33	42	72	63	38	16	22	63	17	39	8	8	12	23	51	68	69	71	5
ViT-B — Rand.		3	1	9	9	5	2	1	2	7	6	8	6	3	3	19	13	4	0	3	2	13	8	18	9	20	0
X-Ent.		79	59	83	66	73	70	58	39	46	72	67	39	18	25	68	21	39	9	8	11	26	46	67	84	71	2
MoCo-v3		61	37	80	62	30	53	41	25	39	38	62	40	15	12	81	35	31	10	12	**14**	40	56	79	84	**73**	1
DINO		68	46	75	62	78	78	72	33	45	46	67	**44**	**22**	14	**90**	**38**	**47**	**15**	**14**	12	44	**60**	**84**	83	69	1
MAE (CLS)		28	15	29	26	38	44	20	10	28	23	37	24	11	6	49	23	18	5	9	8	27	34	52	60	51	1
MAE (avg)		29	14	30	29	42	42	6	10	26	23	36	26	8	6	55	16	16	4	11	9	25	39	47	44	55	1
MoCo-v3	✓	81	62	**84**	**69**	70	53	61	44	50	65	67	41	20	27	72	20	43	10	9	11	25	48	65	79	71	1
DINO	✓	**82**	**63**	80	62	69	54	60	43	50	63	**69**	40	18	26	67	18	41	10	8	11	25	46	64	80	70	2
MAE (avg)	✓	**82**	**63**	78	63	68	59	63	**45**	**52**	69	**69**	41	19	**28**	77	22	45	11	9	11	26	50	71	82	71	2

Table 11: **AMI score (%) using Agglomerative Clustering with number of clusters unknown (AC w/o C).** In this configuration, the number of clusters is determined automatically using a distance threshold tuned on a subset of IN-1k training data (see Appendix E for details).

		In-domain					Domain-shift				Near-OOD			Fine-grained						Far-OOD							
Encoder	FT	IN1k	INv2	C10	C100	IN9	9-FG	9-MR	IN-R	IN-S	IN-O	LSU	P365	Air	Cars	F102	Bio	Birds	iNat	CelA	UTKF	BHis	DTD	ESAT	MNST	Fash	SVHN
Raw image		4	3	11	10	5	6	1	2	7	8	6	5	4	3	17	12	3	3	6	3	11	4	16	54	54	2
RN50 — Rand.		3	2	4	6	4	6	2	2	7	5	4	4	3	2	12	13	3	1	3	1	15	4	21	37	42	1
X-Ent.		68	44	54	52	68	67	59	35	42	58	62	40	17	26	64	29	46	17	10	12	29	48	60	59	58	8
MoCo-v3		47	30	64	47	**83**	67	55	24	35	30	52	37	14	12	65	35	33	14	**18**	**14**	31	41	73	**85**	**70**	**14**
DINO		47	30	45	42	75	66	43	18	28	37	46	39	9	15	72	41	25	14	15	**14**	**47**	52	76	70	65	5
VICReg		45	30	47	46	76	**71**	43	20	31	39	45	37	8	13	73	39	26	14	15	13	38	52	**78**	77	**69**	7
MoCo-v3	✓	68	45	54	52	73	67	**63**	35	45	62	60	39	19	26	61	23	47	16	10	12	23	50	56	51	58	8
DINO	✓	68	45	54	51	72	68	63	34	43	63	57	39	17	25	61	23	46	15	10	11	23	48	57	57	56	6
VICReg	✓	67	44	52	50	72	67	63	33	42	61	59	39	18	24	64	24	45	15	11	12	23	51	56	53	58	7
ViT-B — Rand.		5	3	9	9	6	3	1	3	3	8	9	7	3	3	18	13	5	3	4	3	7	7	18	9	17	0
X-Ent.		72	50	68	66	66	64	52	39	45	**64**	62	41	20	28	69	29	44	17	11	11	27	46	57	66	60	4
MoCo-v3		55	37	67	62	73	65	47	26	38	37	62	38	12	10	78	39	33	15	11	13	36	56	63	67	68	4
DINO		63	43	56	59	74	65	59	34	45	46	63	**42**	19	13	**90**	**43**	47	20	12	10	37	**58**	65	60	61	3
MAE (CLS)		28	16	30	27	49	48	25	11	27	22	33	24	10	6	51	21	19	9	9	7	32	34	53	62	53	2
MAE (avg)		28	15	30	30	44	43	23	12	24	23	34	24	8	6	59	17	17	8	12	9	30	42	53	52	56	2
MoCo-v3	✓	73	53	**70**	**69**	67	61	55	44	48	59	61	**42**	**21**	29	71	29	49	19	12	12	26	48	55	61	59	3
DINO	✓	**75**	53	63	62	68	62	56	43	49	56	63	41	**21**	29	67	26	47	19	11	11	26	45	54	59	57	3
MAE (avg)	✓	**74**	**54**	64	63	68	63	57	**45**	**50**	62	**65**	**42**	20	**31**	76	31	**51**	**20**	13	12	29	49	58	63	61	5

Table 12: **AMI score (%) using Affinity Propagation.**

		In-domain					Domain-shift				Near-OOD			Fine-grained						Far-OOD							
Encoder	FT	IN1k	INv2	C10	C100	IN9	9-FG	9-MR	IN-R	IN-S	IN-O	LSU	P365	Air	Cars	F102	Bio	Birds	iNat	CelA	UTKF	BHis	DTD	ESAT	MNST	Fash	SVHN
Raw image		2	1	11	10	6	6	1	3	10	6	7	3	3	2	16	8	2	0	4	3	16	5	26	43	43	4
RN50 — Rand.		2	1	7	7	5	7	2	2	10	5	5	3	2	1	14	11	2	0	3	2	18	5	24	40	40	1
X-Ent.		67	44	54	52	70	71	60	34	42	56	57	42	18	27	64	31	48	20	13	12	28	49	63	63	65	7
MoCo-v3		50	30	54	51	68	60	46	27	38	39	53	35	**22**	12	80	33	33	11	14	13	32	52	64	**76**	61	**15**
DINO		48	28	43	42	68	63	41	19	29	36	53	37	19	14	80	40	24	10	14	**13**	**45**	52	70	62	59	4
VICReg		49	29	45	45	70	64	42	21	32	37	54	35	16	13	80	37	25	10	14	**13**	38	52	70	66	55	6
MoCo-v3	✓	65	44	57	52	79	70	66	35	44	61	58	42	19	27	61	26	48	19	12	12	23	50	60	51	65	8
DINO	✓	66	44	55	52	79	73	64	34	43	61	58	41	18	26	61	26	47	18	13	12	24	48	60	61	60	6
VICReg	✓	65	44	54	51	79	71	64	33	41	61	57	41	18	27	63	27	47	18	13	13	24	51	60	58	64	7
ViT-B — Rand.		3	1	9	9	6	5	1	2	10	7	9	4	3	2	18	17	3	0	3	3	18	8	31	20	20	0
X-Ent.		71	50	72	65	66	64	53	39	44	**64**	62	42	20	30	68	31	45	19	13	11	26	45	59	71	**68**	4
MoCo-v3		57	36	68	63	75	68	46	26	37	36	63	42	15	12	78	41	34	14	15	**13**	41	55	69	72	20	4
DINO		62	41	63	62	**84**	**75**	**66**	33	43	44	62	**44**	**22**	15	**87**	**44**	45	19	**16**	12	44	**60**	**74**	67	65	2
MAE (CLS)		18	10	26	23	40	38	19	9	24	18	28	15	10	4	45	17	9	2	6	5	33	30	48	47	43	6
MAE (avg)		23	12	30	25	47	45	17	9	23	18	35	23	8	6	52	18	13	3	8	9	36	35	58	64	11	1
MoCo-v3	✓	73	52	**74**	**68**	68	61	55	44	48	59	61	44	21	32	70	32	49	22	14	12	26	50	57	64	67	3
DINO	✓	**74**	52	68	62	70	61	56	43	48	56	**63**	43	21	33	67	29	48	21	14	11	26	45	57	63	65	3
MAE (avg)	✓	**73**	**53**	68	62	70	63	57	**45**	**50**	62	**63**	44	19	**34**	76	33	**51**	**23**	15	12	28	49	63	68	64	4

Table 13: **AMI score (%) using HDBSCAN.** In this analysis, we evaluate the HDBSCAN output by counting all the samples labelled as "noise" as being combined together into their own cluster. Such analysis is contrary to the intended usage of HDBSCAN and will negatively impact the measured performance of HDBSCAN, but is the fairest comparison available. For an evaluation of HDBSCAN excluding rejected samples, see Table 14.

		In-domain					Domain-shift				Near-OOD			Fine-grained						Far-OOD							
Encoder	FT	IN1k	INv2	C10	C100	IN9	9-FG	9-MR	IN-R	IN-S	IN-O	LSU	P365	Air	Cars	F102	Bio	Birds	iNat	CelA	UTKF	BHis	DTD	ESAT	MNST	Fash	SVHN
Raw image		1	1	7	6	6	5	1	1	6	7	5	1	3	1	9	1	1	0	1	1	12	4	21	65	38	3
RN50 — Rand.		0	0	4	4	3	5	1	1	5	2	3	1	2	0	8	2	1	0	0	0	11	3	17	36	27	1
X-Ent.		64	38	37	43	50	48	47	32	30	52	39	25	10	13	56	6	28	8	3	**3**	20	42	58	70	49	6
MoCo-v3		34	18	37	38	46	42	35	22	25	30	39	20	14	7	76	8	26	5	4	**2**	23	47	64	**77**	45	**11**
DINO		29	15	28	28	49	47	33	13	19	26	34	20	13	7	78	9	16	4	**5**	**2**	**37**	43	**72**	68	44	4
VICReg		32	18	27	32	42	45	32	16	22	29	37	19	12	8	77	9	13	4	**4**	**3**	28	48	**72**	72	51	5
MoCo-v3	✓	61	39	43	42	50	49	39	31	30	62	51	24	15	14	51	5	27	7	2	**3**	16	45	52	59	44	5
DINO	✓	61	38	35	39	49	46	38	28	30	63	36	23	16	12	54	5	29	6	3	**3**	17	45	53	66	47	4
VICReg	✓	60	37	38	40	50	45	44	28	29	61	43	22	10	13	58	5	31	6	3	2	15	45	57	54	**52**	5
ViT-B — Rand.		1	0	7	5	5	4	1	1	6	5	7	1	2	1	11	4	1	0	1	1	15	6	26	15	18	0
X-Ent.		72	51	70	54	51	49	43	37	34	**64**	53	30	14	13	64	7	28	9	2	**2**	19	41	46	71	46	3
MoCo-v3		49	27	62	50	49	48	35	23	27	28	46	26	11	6	75	10	22	7	**4**	**3**	32	52	56	76	48	3
DINO		56	34	56	51	**52**	**50**	**49**	28	31	36	59	26	15	8	**84**	**11**	34	9	**4**	**3**	36	**53**	67	74	45	2
MAE (CLS)		2	1	4	5	9	7	1	1	9	4	6	2	3	2	24	2	5	3	4	1	11	11	32	40	31	0
MAE (avg)		11	6	18	16	33	38	16	6	16	12	26	10	6	4	44	5	5	1	3	**2**	29	29	45	68	46	1
MoCo-v3	✓	75	55	**76**	**61**	**52**	49	47	**43**	**39**	56	**63**	**31**	15	**22**	66	6	36	10	3	**2**	17	44	44	71	45	2
DINO	✓	**77**	55	64	52	51	49	45	41	39	53	48	30	**18**	17	64	5	32	10	3	2	18	42	46	61	51	2
MAE (avg)	✓	76	**56**	75	53	**52**	49	48	**43**	**39**	61	48	30	16	21	71	7	**38**	**11**	4	**3**	21	43	47	75	49	3

Table 14: **AMI score (%) using HDBSCAN, excluding samples rejected by the clusterer as background noise.** These scores are highly inflated because HDBSCAN will reject the samples which are hardest to cluster, and is only being evaluated here on samples it was confident in clustering—HDBSCAN frequently rejected half the samples in a dataset (see Table 15).

		In-domain					Domain-shift				Near-OOD			Fine-grained						Far-OOD							
Encoder	FT	IN1k	INv2	C10	C100	IN9	9-FG	9-MR	IN-R	IN-S	IN-O	LSU	P365	Air	Cars	F102	Bio	Birds	iNat	CelA	UTKF	BHis	DTD	ESAT	MNST	Fash	SVHN
Raw image		3	2	12	15	10	8	1	3	16	10	10	3	5	2	20	3	3	1	2	1	18	6	29	75	49	6
RN50 — Rand.		1	1	6	6	4	6	2	1	10	2	4	1	2	1	13	3	2	0	1	1	14	4	21	50	34	1
X-Ent.		82	61	55	62	55	54	55	53	57	73	57	47	19	29	76	12	54	25	5	**4**	33	59	67	75	60	10
MoCo-v3		67	41	56	64	57	55	49	45	52	53	54	43	26	16	89	15	39	16	7	**4**	38	60	74	82	58	**21**
DINO		62	38	46	54	**62**	**60**	47	27	39	51	49	46	26	12	90	17	30	13	8	3	51	64	**78**	74	57	7
VICReg		65	40	46	57	55	57	46	31	45	51	52	43	21	14	89	17	28	15	8	**4**	42	64	76	78	64	11
MoCo-v3	✓	82	62	57	63	55	55	50	51	59	**80**	64	47	25	29	72	9	55	23	4	4	26	59	67	70	57	11
DINO	✓	82	62	54	62	55	54	50	50	56	**80**	54	46	26	27	74	9	55	22	5	3	27	58	67	74	60	8
VICReg	✓	82	59	54	61	55	53	55	49	53	79	56	45	17	26	77	9	55	22	5	3	26	61	70	65	63	9
ViT-B — Rand.		2	1	10	8	7	6	1	2	14	7	9	3	3	2	17	6	2	0	1	1	18	8	31	21	24	0
X-Ent.		81	65	76	72	53	52	48	53	58	78	63	46	24	29	79	12	51	24	5	3	31	55	59	78	59	6
MoCo-v3		74	51	72	71	56	56	45	41	50	49	58	48	19	11	88	17	39	19	8	**4**	49	64	72	81	59	5
DINO		77	56	69	72	57	57	**57**	50	57	56	68	**49**	28	12	**93**	**20**	53	25	8	3	51	**65**	76	79	58	5
MAE (CLS)		38	4	66	38	29	19	6	4	57	19	41	5	6	1	92	18	18	2	**22**	3	**62**	33	71	**95**	**73**	2
MAE (avg)		34	16	32	33	48	52	24	14	32	25	39	30	11	7	66	9	13	4	5	3	42	45	57	76	58	2
MoCo-v3	✓	84	68	**79**	**75**	54	54	52	**60**	**64**	74	**69**	48	**29**	**39**	81	12	54	27	6	**4**	32	58	62	79	59	4
DINO	✓	**84**	**69**	73	71	53	53	50	57	63	72	62	46	27	33	81	10	56	26	6	3	31	57	60	71	62	4
MAE (avg)	✓	**84**	68	77	69	54	53	52	59	**64**	75	62	47	24	37	87	12	**57**	**29**	7	4	36	57	62	80	61	6

Table 15: **Fraction of samples clustered by HDBSCAN.** We indicate the fraction of samples (%) which were placed into a cluster by HDBSCAN—remaining samples were rejected and placed in the "noise" category. Since every sample in each of the datasets is labelled rejections are likely incorrect, so a larger fraction of samples clustered is likely to indicate a better clustering attempt. When dealing with curated datasets, we postulate it is only plausible that a minority of the samples can truly be outliers.

		In-domain					Domain-shift				Near-OOD			Fine-grained						Far-OOD							
Encoder	FT	IN1k	INv2	C10	C100	IN9	9-FG	9-MR	IN-R	IN-S	IN-O	LSU	P365	Air	Cars	F102	Bio	Birds	iNat	CelA	UTKF	BHis	DTD	ESAT	MNST	Fash	SVHN
Raw image		26	39	37	40	44	48	45	32	39	68	42	29	55	45	41	76	32	22	76	80	59	57	58	79	67	39
RN50 — Rand.		40	54	56	57	58	64	56	50	49	77	63	41	**70**	50	60	78	49	36	**79**	**82**	76	71	71	56	69	53
X-Ent.		80	71	52	64	90	84	81	55	57	75	58	50	53	46	68	73	51	40	74	78	53	69	79	86	70	33
MoCo-v3		55	53	51	54	73	66	59	44	53	62	65	45	54	46	84	72	59	39	74	78	54	77	81	87	64	31
DINO		51	50	44	48	71	70	58	45	53	57	58	42	51	52	84	71	50	36	74	78	67	66	86	84	65	32
VICReg		53	54	41	51	66	69	59	47	54	62	60	42	55	59	85	72	47	37	73	76	59	73	**88**	86	67	30
MoCo-v3	✓	77	71	61	62	89	86	70	55	56	81	71	49	57	48	66	73	49	39	73	77	54	74	68	76	60	29
DINO	✓	77	71	49	58	86	81	66	52	58	82	56	48	58	46	69	74	51	38	73	78	55	74	68	79	64	29
VICReg	✓	76	72	57	60	89	81	74	51	57	81	67	48	59	50	72	76	54	38	75	79	52	72	73	72	**72**	32
ViT-B — Rand.		45	58	58	60	66	66	64	52	45	77	67	45	61	57	64	**80**	53	**44**	**79**	80	**77**	73	75	57	62	**56**
X-Ent.		90	85	85	71	**95**	**92**	84	65	61	**85**	78	**62**	57	44	78	73	53	**45**	71	76	52	73	70	81	64	33
MoCo-v3		69	62	78	66	82	82	68	53	58	63	71	51	57	59	83	71	55	42	73	76	59	**80**	69	86	69	34
DINO		74	69	72	66	87	84	82	52	59	68	83	51	53	**61**	**89**	71	61	43	70	77	65	**80**	82	**87**	63	31
MAE (CLS)		6	13	3	10	21	16	7	15	18	12	11	14	28	22	20	40	20	19	33	37	14	25	30	31	35	13
MAE (avg)		36	45	40	43	57	62	53	40	52	54	57	33	59	56	62	73	42	28	75	79	64	62	68	82	68	39
MoCo-v3	✓	90	86	93	**77**	**95**	88	86	67	**64**	79	**91**	61	48	53	76	73	61	44	71	75	46	74	61	81	61	31
DINO	✓	**92**	86	81	70	**95**	90	85	68	**64**	78	70	**62**	58	51	74	73	55	**45**	71	76	50	72	67	75	70	28
MAE (avg)	✓	91	**87**	**97**	72	**95**	90	**89**	**69**	**64**	84	67	59	60	53	79	72	**63**	44	71	75	50	73	65	**87**	67	30

I. Predicted Number of Clusters

We report the predicted number of clusters for the three clusterers which do not require a number of clusters to be provided to the clusterer.

As shown in Tables 16–18, the number of clusters predicted varies greatly. We found HDBSCAN usually generated the largest number of clusters, and AC w/o C generated the fewest. The number of clusters predicted was often biased toward the average magnitude (in the order of 100), such that datasets which had fewer GT clusters (in the order of 10) were more likely to be clustered with more clusters than were annotated, and datasets which had more GT clusterers (in the order of 1000) were more likely to be clustered with fewer clusters than were annotated. However, we note that for many datasets the number of classes is ambiguous as the GT categories are hierarchical, and the clustered embeddings may correspond to a coarser granularity than the finest-grained annotations, as discussed in Appendix N. Similarly, for datasets which have few annotated classes, it may be feasible to break the data down further into subclasses.

Table 16: **Number of clusters generated using AC w/o C.** Underlined (Bold): encoder which generated clusters with numerosity closest to the GT per dataset (across all clusterers). Background colour scale: logarithmic from smallest underestimate (red) to largest overestimate (blue), centered around the GT number of clusters (white).

		In-domain					Domain-shift				Near-OOD			Fine-grained						Far-OOD							
Encoder	FT	IN1k	INv2	C10	C100	IN9	9-FG	9-MR	IN-R	IN-S	IN-O	LSU	P365	Air	Cars	F102	Bio	Birds	iNat	CelA	UTKF	BHis	DTD	ESAT	MNST	Fash	SVHN
№ GT classes		1000	1000	10	100	9	9	9	200	1000	200	10	365	100	196	102	2688	555	10000	1000	101	32	47	10	10	10	10
Raw image		1543	533	411	398	277	126	344	595	1221	137	240	1199	**101**	340	393	316	914	**3797**	324	124	250	159	179	170	47	227
RN50 — Rand.		475	213	254	228	142	198	184	342	513	98	119	**351**	119	176	156	460	290	735	274	198	111	87	161	55	144	317
X-Ent.		257	104	54	111	57	52	56	262	275	58	15	321	39	126	79	296	130	436	294	71	76	41	51	73	39	748
MoCo-v3		50	20	**14**	22	**11**	**13**	**11**	54	76	9	4	58	6	29	18	37	29	85	46	11	9	9	12	**11**	11	79
DINO		90	72	66	89	26	39	68	145	141	86	3	160	4	149	32	124	72	216	269	81	54	53	20	27	13	1227
VICReg		72	43	93	96	26	22	56	263	132	73	3	169	4	184	37	129	61	154	344	119	113	37	17	21	**10**	548
MoCo-v3	✓	245	95	74	123	45	46	44	228	306	46	20	315	48	112	81	436	131	426	271	65	106	43	58	81	38	938
DINO	✓	241	101	60	114	48	42	47	251	264	50	20	311	49	125	74	463	133	430	274	91	76	40	58	75	45	786
VICReg	✓	232	96	73	116	47	44	44	253	326	46	17	313	49	151	79	414	138	455	275	96	89	42	61	101	37	876
ViT-B — Rand.		65	56	29	35	42	62	64	57	42	36	35	108	45	95	46	28	22	45	33	21	14	34	**10**	6	3	**15**
X-Ent.		287	143	38	89	56	60	72	183	317	79	19	250	34	108	74	298	143	378	219	64	76	42	51	48	33	399
MoCo-v3		164	382	64	249	48	61	97	755	355	76	**9**	864	9	898	58	957	161	525	1652	197	454	117	85	67	15	4729
DINO		213	138	204	698	48	70	102	688	634	320	26	805	19	622	81	671	183	614	2275	447	694	173	74	103	36	3804
MAE (CLS)		670	340	244	254	175	130	236	596	375	**161**	106	495	49	161	176	286	204	548	351	141	70	125	31	52	36	57
MAE (avg)		1906	**822**	333	361	317	154	266	1235	459	268	187	1063	73	**199**	247	311	198	823	481	**105**	69	127	25	44	18	57
MoCo-v3	✓	280	147	33	88	50	63	72	192	307	68	19	241	17	121	72	321	118	315	253	60	66	45	53	63	38	737
DINO	✓	299	138	49	**100**	56	61	69	196	328	65	20	264	23	119	75	308	122	339	250	61	73	44	49	67	42	755
MAE (avg)	✓	290	152	46	98	53	60	67	**197**	319	72	15	263	11	130	71	244	122	301	227	64	80	40	52	55	35	683

Table 17: **Number of clusters generated using Affinity Prop.**

		In-domain					Domain-shift				Near-OOD			Fine-grained						Far-OOD							
Encoder	FT	IN1k	INv2	C10	C100	IN9	9-FG	9-MR	IN-R	IN-S	IN-O	LSU	P365	Air	Cars	F102	Bio	Birds	iNat	CelA	UTKF	BHis	DTD	ESAT	MNST	Fash	SVHN
№ GT classes		1000	1000	10	100	9	9	9	200	1000	200	10	365	100	196	102	2688	555	10000	1000	101	32	47	10	10	10	10
Raw image		1151	299	317	317	153	249	143	572	2116	73	111	828	120	310	220	913	632	1969	582	212	137	66	108	784	315	428
RN50 — Rand.		804	220	244	227	113	169	119	443	818	64	103	646	110	234	145	562	423	1238	424	173	80	48	68	452	198	316
X-Ent.		227	108	54	59	47	33	46	179	293	65	23	167	33	90	70	121	78	179	106	64	61	43	32	40	17	202
MoCo-v3		232	102	57	87	44	51	75	241	437	67	23	219	29	96	62	129	106	313	148	75	64	41	29	20	538	194
DINO		210	86	72	96	35	39	78	205	369	59	19	204	40	101	58	98	93	295	140	77	58	51	25	40	31	210
VICReg		205	81	68	86	33	39	73	209	364	56	13	191	30	82	64	112	96	293	147	83	60	40	22	33	977	175
MoCo-v3	✓	169	61	41	62	28	31	34	181	278	46	23	161	48	88	68	154	79	172	114	66	73	35	35	74	17	265
DINO	✓	184	64	52	62	27	27	40	184	265	49	21	164	37	87	67	155	81	179	120	67	67	39	33	43	26	238
VICReg	✓	172	72	54	60	29	31	39	188	298	51	23	164	35	91	64	145	86	182	110	73	70	38	31	53	19	251
ViT-B — Rand.		616	203	192	199	108	177	104	397	1252	68	79	520	94	174	159	439	308	963	293	132	96	65	85	184	79	254
X-Ent.		263	141	24	52	47	53	67	164	303	83	19	157	28	62	65	144	78	176	101	55	72	41	33	28	15	171
MoCo-v3		1256	99	31	60	37	43	75	187	460	46	13	221	31	76	62	107	70	208	104	66	60	36	24	24	4951	169
DINO		363	44	43	81	24	27	40	165	322	46	19	151	33	60	58	79	57	166	91	71	63	37	20	33	17	155
MAE (CLS)		2001	505	426	462	254	256	248	1220	3670	132	159	1350	172	390	357	975	943	2996	**861**	302	206	143	160	397	297	872
MAE (avg)		428	112	99	127	57	49	67	404	**943**	49	64	231	35	67	**103**	168	1676	1060	208	88	58	**47**	38	43	7124	182
MoCo-v3	✓	255	144	23	57	42	65	73	188	307	76	19	165	15	65	68	117	67	138	88	55	67	46	35	39	17	235
DINO	✓	281	129	29	52	46	63	64	184	333	69	18	158	19	58	67	126	65	158	105	58	67	44	35	40	16	259
MAE (avg)	✓	270	155	29	52	45	58	65	184	300	81	18	161	7	67	63	109	62	141	96	58	74	41	26	34	21	241

Table 18: **Number of clusters generated using HDBSCAN.**

		In-domain					Domain-shift				Near-OOD			Fine-grained						Far-OOD							
Encoder	FT	IN1k	INv2	C10	C100	IN9	9-FG	9-MR	IN-R	IN-S	IN-O	LSU	P365	Air	Cars	F102	Bio	Birds	iNat	CelA	UTKF	BHis	DTD	ESAT	MNST	Fash	SVHN
№ GT classes		1000	1000	10	100	9	9	9	200	1000	200	10	365	100	196	102	2688	555	10000	1000	101	32	47	10	10	10	10
Raw image		882	226	243	245	107	118	98	572	1237	43	88	694	96	213	156	6154	483	1490	5137	1569	82	79	95	168	284	602
RN50 — Rand.		1310	333	322	325	166	148	150	929	1586	69	121	1066	123	274	207	6374	735	2529	5176	1547	120	90	168	274	448	809
X-Ent.		1181	481	228	196	227	214	167	533	1714	119	77	740	98	230	180	6007	526	1617	4939	1503	134	76	80	81	178	617
MoCo-v3		1302	337	222	236	172	165	141	605	2002	104	59	728	114	242	138	5905	414	1685	4954	1515	121	57	56	81	214	544
DINO		1160	329	241	251	138	130	118	578	1967	81	76	697	111	167	143	5791	456	1616	4873	1540	140	72	49	97	183	618
VICReg		1224	344	256	240	179	150	145	582	2020	86	71	724	115	190	158	5822	594	1659	4901	1493	128	69	33	90	170	575
MoCo-v3	✓	1174	474	177	190	225	193	188	517	1592	132	60	728	86	209	182	6062	**561**	1653	4830	1483	119	62	81	127	215	545
DINO	✓	1230	482	234	236	229	198	193	608	1615	133	95	765	94	231	161	6067	575	1736	4880	1544	118	71	92	128	188	533
VICReg	✓	1247	483	224	232	217	211	194	599	1621	133	75	805	114	235	166	6305	493	1842	4980	1519	115	67	79	190	170	599
ViT-B — Rand.		1454	368	383	401	159	161	172	946	1570	77	118	1116	136	264	256	6498	835	3061	5184	1511	141	74	208	356	394	956
X-Ent.		1102	640	88	210	254	232	211	463	1571	145	51	569	86	233	162	6026	519	1333	4778	1489	104	65	96	104	207	573
MoCo-v3		1145	416	105	235	196	180	179	567	1873	99	60	685	97	164	162	5927	456	1592	4884	1469	139	68	93	85	171	548
DINO		1131	416	154	224	191	173	161	583	1912	89	40	726	114	167	150	5786	452	1548	4681	1484	141	60	57	78	205	576
MAE (CLS)		133	19	19	21	15	14	5	48	1095	6	6	31	10	3	52	**3671**	50	40	2712	845	**50**	12	20	17	14	51
MAE (avg)		**1029**	249	274	285	137	136	130	584	1947	70	76	698	91	154	195	5898	614	1984	5000	1476	137	78	130	131	212	720
MoCo-v3	✓	1103	651	54	181	250	239	208	464	1446	130	26	555	89	190	172	6006	398	1271	4750	1502	96	61	91	104	204	458
DINO	✓	1102	677	99	202	253	252	215	436	1478	128	65	579	70	212	155	6054	464	1323	4670	1507	110	64	93	147	177	514
MAE (avg)	✓	1092	649	34	197	244	237	198	423	1514	129	69	621	70	221	157	6074	395	1357	4728	1461	124	66	85	81	183	490

J. Silhouette Scores

Our results on the silhouette score are broadly in line with our main finding on the AMI between clusterings and annotation targets, reported in §3. For both the ResNet-50 and ViT-B encoders, the supervised model has the highest silhouette score by a large margin of 0.25–0.3, but otherwise the clustering quality across the encoders is very similar, achieving similar silhouette scores to each other. There are some exceptions to this, such as the silhouette scores for MAE which are near 0, illustrating the intrinsically-poor quality of the clusters it exhibited and hence it is not well-suited to this task.

Despite the very low AMI scores, we observe the silhouette scores for SVHN are generally comparable to the silhouette scores of the other datasets. We believe this is due to the heterogeneity within the classes in SVHN, where house-numbers can be written in different formats, colours, etc., and thus the encoded images can be appropriately grouped together, even if the semantic meaning of the clusters does not correspond to the identity of the digit in the center of the image.

Between the clusterers, K-Means and AC typically achieve the highest silhouette scores. For HDBSCAN, the silhouette scores were often significantly negative. This is because HDBSCAN builds clusters based on transitions in density, and the non-convex clusters that result from this can score poor silhouette scores (a known caveat to this evaluation metric). For Affinity Propagation, we observe silhouette scores near 0, indicating the clusters it discovered have high overlap with each other and are of low quality, corresponding to its poor AMI performance.

K. Detailed Comparison of Performances Across Clustering Methods

We sought to evaluate the clusterers to see which clustering methodology produced the best results when using a pretrained encoder. For each set of embeddings, created by passing one of the datasets listed in Appendix H through one of the pretrained ResNet-50 or ViT-B encoders (X-Ent., MoCo-v3, DINO, VICReg, or MAE), we compared the results of clustering that set of embeddings with each of the clusterers (tabulated in Table 8–13).

We compared the performance of the clustering methods by ranking each clusterer for each combination of pretrained encoder and dataset, shown in Fig. 4. The results show that AC w/ C performs best most often ($p < 0.05$; Wilcoxon signed-rank test versus each other clusterer). Spectral, K-Means, AC w/o C, and AP all perform similarly. HDBSCAN frequently performed worst ($p < 10^{-33}$).

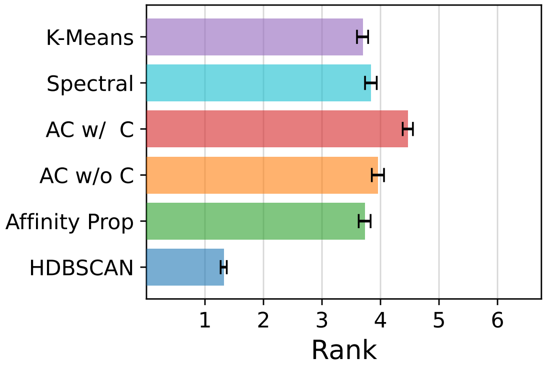


Figure 4: **Average clusterer rank** (higher is better). For each set of embeddings we apply each clusterer, compare the AMI of their clusters, and rank them against each other (lowest AMI → rank 1, highest AMI → rank 6). Error bars: ±1 stderr; $N = 225$.

We note that HDBSCAN identifies samples which belong to *no* cluster (noise/background samples). Unless stated otherwise, we consider the noise class to be its own class when computing the AMI. This sets HDBSCAN at a disadvantage, since the samples it identifies as noise are typically distributed across all GT classes, but is fairer than ignoring samples it identifies as noise since that would evaluate it only on easier samples. If we instead exclude the noise class and only evaluate the AMI on samples which HDBSCAN placed in a real cluster, we find it yields the best performance of all clusterers, shown in Table 14, suggesting that HDBSCAN can provide value depending on the goals of the user performing the clustering. However, we note that on the well-labelled datasets we considered, HDBSCAN frequently rejected half of the samples (see Table 15).

We investigated the correlation between the AMI for each pair of clustering methods, shown in Table 19 and illustrated in Fig. 5. We found the correlation between clusterers was generally high ($0.931 \le r \le 0.990$). The performance of HDBSCAN was less correlated with the other clusterers ($r \le 0.948$ vs $r \ge 0.960$).

Table 19: **Pearson correlation coefficient between clusterers.** For each pair of clustering methods, we measure the Pearson correlation coefficient (%) between the AMI each attained when clustering the embeddings of a given dataset with a given encoder. We utilize datapoints across all datasets and all encoders, including fine-tuned, randomized (untrained), and raw pixels. **Bold**: for a given clustering method (column), the clustering method (row) that it is most correlated with.

	K-Means	Spectral	AC w/ C	AC w/o C	Affinity Prop	HDBSCAN
K-Means	–	**97.6**	**99.0**	97.4	97.4	**94.8**
Spectral	97.6	–	97.1	97.2	96.0	94.3
AC w/ C	**99.0**	97.1	–	97.5	96.9	94.5
AC w/o C	97.4	97.2	97.5	–	**97.7**	93.1
Affinity Prop	97.4	96.0	96.9	**97.7**	–	93.6
HDBSCAN	94.8	94.3	94.5	93.1	93.6	–

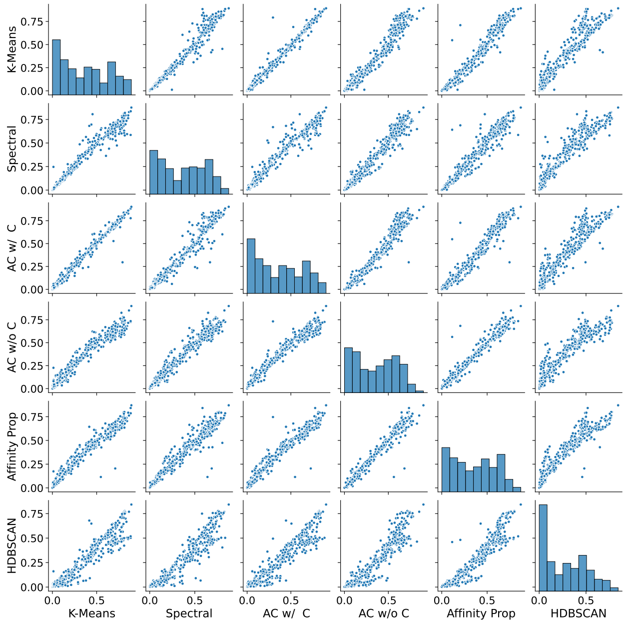


Figure 5: **Correlation of AMI between clustering methods.** For each pair of clustering methods, we show a scatter plot of the AMI each attained when clustering the embeddings of a given dataset with a given encoder. We show all datasets and all encoders, including fine-tuned, randomized (untrained), and raw pixels. Along the diagonal, the distribution of AMI values is shown for each clusterer.

L. Detailed Comparison Between Encoders

We computed and evaluated the Pearson correlation coefficient between the clusterings of pairs of encoders.

Looking across model architectures (Table 20) we find the performance of SSL encoders are typically more correlated with other SSL models of the same architecture than with the same pretraining loss but a different architecture.

As shown in Table 21 for ResNet-50 models, and Table 22 for ViT-B models, the performance of the fine-tuned models ([FT]) were well correlated with each other ($r \geq 0.989$) and with the supervised trained model (X-Ent.; $r \geq 0.978$). We also observed the performance of the whole-image SSL models were highly correlated with each other ($r \geq 0.946$, and the two read-outs of the MAE model were strongly correlated with each other ($r = 0.912$). Outside of these blocks, correlation scores were lower. In particular, we note the performance of FT encoders was much more correlated with the X-Entropy models than that of their original SSL-only pretrained encoder.

Table 20: **Pearson correlation coefficient between initial encoders.** For each pair of pretrained encoders (without fine-tuning), we measure the Pearson correlation coefficient (%) between the AMI each attained when clustering the embeddings of a given dataset with a given clusterer. **Bold**: for a given encoder (column), the other encoder (row) that it is most correlated with.

		ResNet-50					**ViT-B**					
	Raw image	Rand.	X-Ent.	MoCo-v3	DINO	VICReg	Rand.	X-Ent.	MoCo-v3	DINO	MAE (CLS)	MAE (avg)
Raw image	–	**96.1**	39.8	58.6	54.8	57.7	71.0	39.0	48.8	43.0	69.6	66.8
RN50 — Rand.	**96.1**	–	39.3	58.4	58.0	59.6	**78.5**	36.5	48.1	44.4	72.7	68.1
X-Ent.	39.8	39.3	–	89.8	86.2	87.8	41.2	**95.2**	87.5	93.6	74.5	73.2
MoCo-v3	58.6	58.4	89.8	–	96.1	97.6	60.4	84.5	93.1	94.2	88.5	88.6
DINO	54.8	58.0	86.2	96.1	–	**99.1**	67.3	77.7	91.1	92.9	90.1	91.6
VICReg	57.7	59.6	87.8	**97.6**	**99.1**	–	67.0	80.9	92.5	93.5	90.3	**92.1**
ViT-B — Rand.	71.0	78.5	41.2	60.4	67.3	67.0	–	40.0	59.5	55.7	71.8	74.7
X-Ent.	39.0	36.5	**95.2**	84.5	77.7	80.9	40.0	–	86.7	90.4	65.7	67.3
MoCo-v3	48.8	48.1	87.5	93.1	91.1	92.5	59.5	86.7	–	**94.6**	81.8	87.1
DINO	43.0	44.4	93.6	94.2	92.9	93.5	55.7	90.4	**94.6**	–	79.7	81.5
MAE (CLS)	69.6	72.7	74.5	88.5	90.1	90.3	71.8	65.7	81.8	79.7	–	91.2
MAE (avg)	66.8	68.1	73.2	88.6	91.6	92.1	74.7	67.3	87.1	81.5	**91.2**	–

Table 21: **Pearson correlation coefficient between ResNet-50 encoders.** For each pair of pretrained encoders, we measure the Pearson correlation coefficient (%) between the AMI each attained when clustering the embeddings of a given dataset with a given clusterer. [FT]: fine-tuned with cross-entropy on IN-1k. **Bold**: for a given encoder (column), the other encoder (row) that it is most correlated with.

	Rand.	X-Ent.	MoCo-v3	DINO	VICReg	MoCo-v3 [FT]	DINO [FT]	VICReg [FT]
Rand.	–	39.3	58.4	58.0	59.6	28.5	34.1	32.0
X-Ent.	39.3	–	89.8	86.2	87.8	97.8	98.9	98.4
MoCo-v3	58.4	89.8	–	96.1	97.6	84.7	87.0	86.9
DINO	58.0	86.2	96.1	–	**99.1**	80.8	82.7	83.3
VICReg	**59.6**	87.8	**97.6**	**99.1**	–	82.0	84.4	84.5
MoCo-v3 [FT]	28.5	97.8	84.7	80.8	82.0	–	99.2	**99.5**
DINO [FT]	34.1	**98.9**	87.0	82.7	84.4	99.2	–	99.5
VICReg [FT]	32.0	98.4	86.9	83.3	84.5	**99.5**	**99.5**	–

Table 22: **Pearson correlation coefficient between ViT-B encoders.** For each pair of pretrained encoders, we measure the Pearson correlation coefficient (%) between the AMI each attained when clustering the embeddings of a given dataset with a given clusterer. [FT]: fine-tuned with cross-entropy on IN-1k. **Bold**: for a given encoder (column), the other encoder (row) that it is most correlated with.

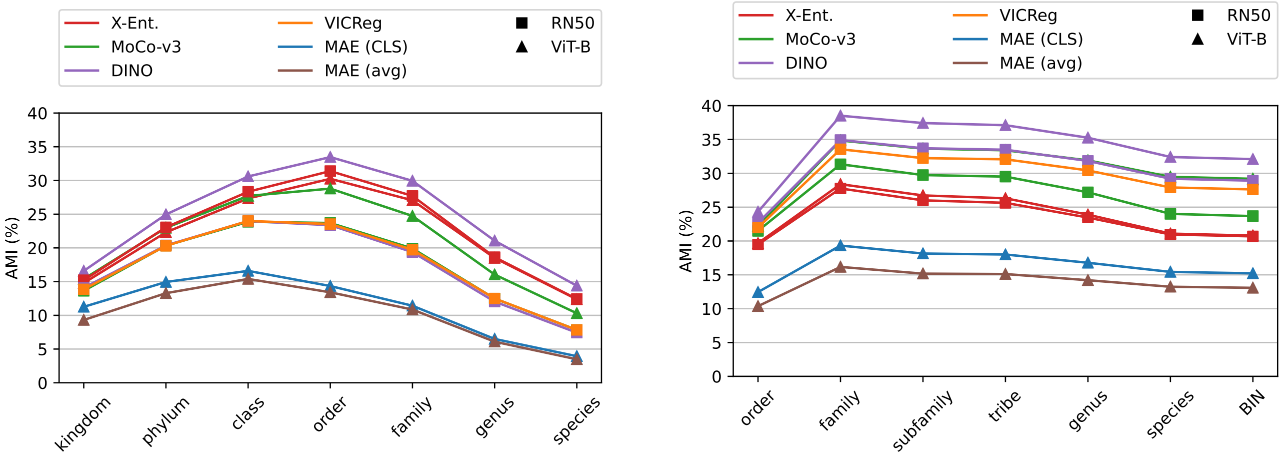
	Rand.	X-Ent.	MoCo-v3	DINO	MAE (CLS)	MAE (avg)	MoCo-v3 [FT]	DINO [FT]	MAE (avg) [FT]
Rand.	–	40.0	59.5	55.7	71.8	74.7	37.2	35.5	39.9
X-Ent.	40.0	–	86.7	90.4	65.7	67.3	97.9	97.9	98.4
MoCo-v3	59.5	86.7	–	**94.6**	81.8	87.1	85.8	85.5	87.2
DINO	55.7	90.4	**94.6**	–	79.7	81.5	88.8	89.1	90.8
MAE (CLS)	71.8	65.7	81.8	79.7	–	**91.2**	62.4	63.1	65.2
MAE (avg)	**74.7**	67.3	87.1	81.5	**91.2**	–	65.4	65.0	67.9
MoCo-v3 [FT]	37.2	97.9	85.8	88.8	62.4	65.4	–	99.2	98.9
DINO [FT]	35.5	97.9	85.5	89.1	63.1	65.0	**99.2**	–	**99.2**
MAE (avg) [FT]	39.9	**98.4**	87.2	90.8	65.2	67.9	98.9	**99.2**	–

M. Comparison of Clustering Methods

We compared the performance of the clustering methods by ranking each clusterer for each combination of pretrained encoder and dataset, shown in Fig. 4. The results show that AC w/ C performs best ($p < 0.05$; Wilcoxon signed-rank test versus each other clusterer). Spectral, K-Means, AC w/o C, and AP all perform similarly. HDBSCAN performed worst ($p < 10^{-33}$), due to its use of a noise class instead of trying to place every sample in a cluster. Although this is a legitimate and principled methodology (McInnes, 2016), it puts HDBSCAN at a disadvantage here; we found HDBSCAN often placed half the samples in the noise class (see Table 15). The trends across encoders and datasets were similar, irrespective of the clusterer used (see Appendix K). For subsequent analysis, we thus present the average over clusterers.

N. Effect of Dataset Granularity

Furthermore, we observe that the overall level of performance on FG datasets varies greatly. While seemingly arbitrary, we find that the performance correlates with how fine-grained the datasets are when considering the proposed granularity measure from Cui et al. (2019). Specifically we find that FGVC Aircraft is the most challenging dataset, matching the finding by Cui et al. (2019) that it is the most fine-grained dataset of the ones considered, while NABirds and Oxford Flowers gradually become more coarse-grained, and easier to correctly cluster. Similarly, we find that the large scale iNaturalist-21 dataset is in general a very hard dataset. These observations echo the recent results from Cole et al. (2022), where it was determined that current SSL methods are not suitable for fine-grained tasks. Using the iNaturalist-21 and BIOSCAN-1M datasets we can vary the labels from coarse to fine-grained using the 7 taxonomic levels available for each data point, see Fig. 6. On the iNaturalist-21 dataset, we find that for all methods the AMI score peaks at a medium-grained level at either the class or order taxonomic level, while drastically decreasing when the labels are too coarse or fine-grained. Similarly, we find that there is a peak at the family level when using the BIOSCAN-1M dataset, and suffers a less drastic performance drop when using species and BIN labels. This contrasts the findings of Cole et al. (2022), who found that the accuracy of SSL encoders decreases monotonically as one moves down the label hierarchy.



(a) **iNaturalist-21 AMI scores.** (b) **BIOSCAN-1M AMI scores.**

Figure 6: **AMI scores across taxonomic levels.** We measure the AMI score at each of the 7 taxonomic levels of the iNaturalist-21 dataset and from order to species level as well as when using the Barcode Index Number (BIN) as a proxy for subspecies labels for the BIOSCAN-1M dataset. The scores are reported for each encoder, averaged over the tested clustering methods.

O. ImageNet-9 Background Challenge

To investigate whether SSL encoders natively focus more on foreground or background contents of images, we analyzed the amount of information about ImageNet-9 variants (Xiao et al., 2020), tabulated in Table 23. We present the AMI when clustering the original images (OG), foreground only (FG), foreground replaced with black (FG^{C}), background only (bounding box replaced with bg texture; BG), mixed-same (fg overlaid on the bg of a sample of the same class; MS), and mixed-random (fg overlaid on the bg of a random sample; MR). Illustrative examples of these are shown in Fig. 7. We also show the difference between MS and MR performance (Gap; Xiao et al., 2020).

SSL and supervised encoders yielded similar quality to each other when clustering the original images (OG), or when clustering the foreground-only images (FG). Supervised and fine-tuned networks consistently had more information about the background of the images (FG^{C} and BG), congruent with the widely held belief that supervised networks learn to exploit information in the background of images. Surprisingly then, we find SSL-encoders have nearly twice as large a BG-gap than their supervised counterparts. Despite the fact that SSL embeddings possess less information about the image backgrounds, using a background that is incongruent with the foreground induces much more “confusion” in the SSL-encoders. We hypothesize that this is because supervised networks are better able to prioritize foreground over background information when creating their embeddings, whereas SSL-encoders are typically unable to distinguish foreground from background and thus their embeddings are always a combination of the two. This is in keeping with their training task, which is to give every unique stimulus its own unique embedding (instance learning), and the stimulus is comprised of both its foreground *and* its background.

Table 23: **ImageNet-9 breakdown.** We show the AMI (%) when clustering variants of the ImageNet-9 dataset, averaged over 6 clusterers. See Appendix O for descriptions of the variants. **Bold**: highest scoring encoder per dataset. Underlined: highest scoring encoder per backbone. Background: ranges from the median value (white) to maximum (blue/red) per dataset. FT: fine-tuned with x-ent. on IN-1k.

Encoder	FT	OG	FG	FG^{C}	BG	MS	MR	Gap
RN50 — X-Ent.		69	**70**	47	26	71	60	11
MoCo-v3		70	61	35	17	60	48	12
DINO		70	64	32	22	59	43	16
VICReg		69	63	30	19	58	40	18
MoCo-v3	✓	**77**	**70**	48	25	**76**	**64**	12
DINO	✓	75	68	49	25	**76**	62	14
VICReg	✓	75	67	47	24	**76**	**64**	12
ViT-B — X-Ent.		61	61	52	27	66	51	15
MoCo-v3		62	62	41	23	65	44	**21**
DINO		72	68	43	25	73	61	11
MAE (CLS)		38	39	21	10	29	18	11
MAE (avg)		44	41	22	9	25	15	10
MoCo-v3	✓	64	53	52	27	65	52	14
DINO	✓	65	53	**55**	**28**	71	53	18
MAE (avg)	✓	66	57	52	25	69	54	16

The only exception to this pattern was the DINO ViT-B encoder, which had the lowest BG-gap of all ViT-B encoders, with the exception of MAE. MAE’s BG-Gap is lower only

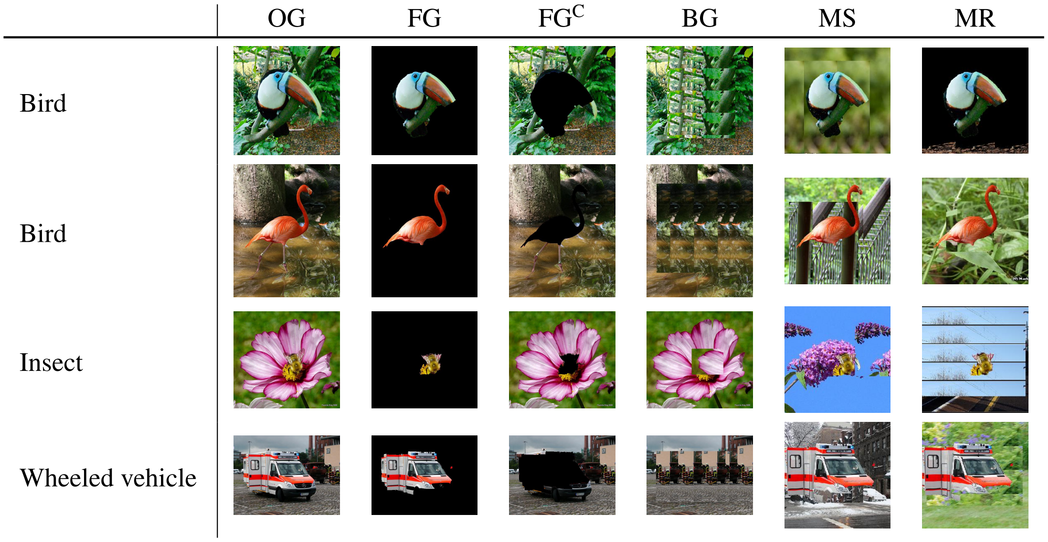


Figure 7: **Example images from the ImageNet-9 dataset.** For three classes (bird, insect, and wheeled vehicle) we show a sample from each of the variant datasets: original images (OG), foreground only (FG), foreground removed and replaced with black (FG^C), background only (bounding box replaced with background texture; BG), mixed-same (foreground overlaid on the background of a sample of the same class; MS), and mixed-random (foreground overlaid on the background of a random sample; MR). We note that MS places the foreground object on an appropriate background, whereas MR places the foreground on a background which may be out-of-context for the foreground. ImageNet-9 labels are coarse-grained superclasses, each spanning multiple IN-1k classes, hence images of toucan and flamingo are both labelled "bird".

because it performs so poorly on the IN9-MS task to begin with and it still has a large relative reduction. We speculate that DINO has such a low BG-gap because it learnt to attend to foreground objects as an emergent outcome of its training process (Caron et al., 2021). This is possible with the ViT backbone, but not the ResNet as it lacks attention mechanisms and must attend to the whole stimulus, hence the DINO ResNet-50 encoder performs the same as MoCo-v3 and VICReg. The behaviour is not replicated for MoCo-v3 ViT-B since its training loss incentives it to attend to all features that are common across multiple views of the same sample and differ between samples, including background features.

P. ImageNet-Rendition Information Breakdown

We sought to better understand what information about the stimulus is being captured in the clusters. Using a dataset which possesses multiple annotations per image, we can investigate the agreement between the clusterings and each annotation type. The ImageNet-Rendition dataset in particular has primary annotations for the object class represented in the image (goldfish, great white shark, cowboy hat, volcano, etc.), but also annotations for the style of rendition (cartoon, graffiti, embroidery, origami, etc.), see Fig. 8. We compute the AMI for each annotation stream, see Table 24.

Our results indicate there is generally a trade-off between the two: embeddings grouped according to object class identities are not grouped according to the artform, and vice versa.

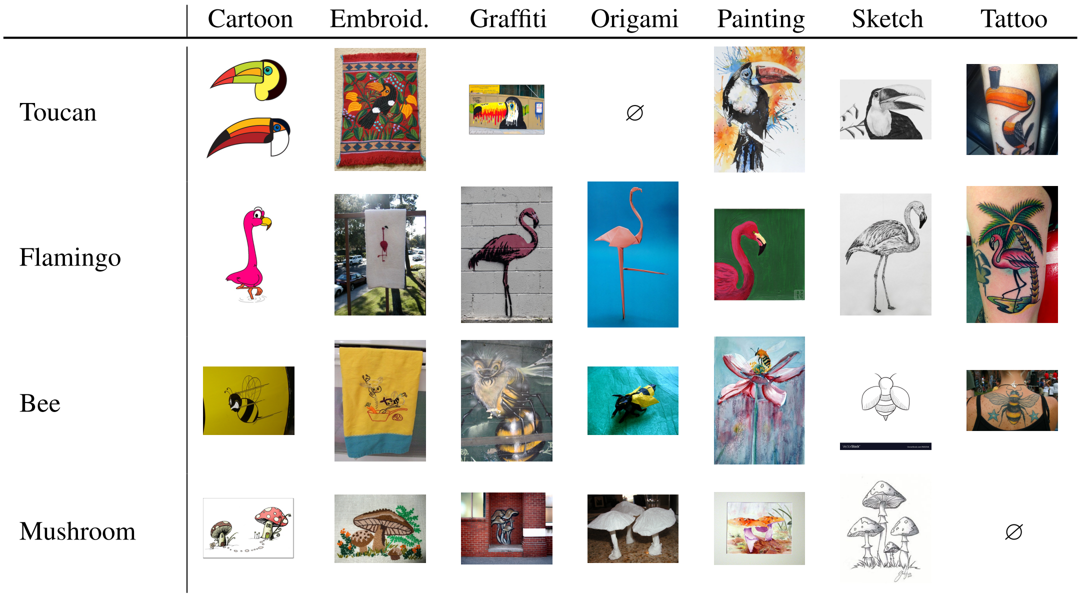


Figure 8: **Example images from ImageNet-R by both class and artform style.** ∅ indicates no images in that artform for this class. In our experiments, we measure the AMI between the clusterings and the labels pooled across each row (Object), each column (Artform), or using only the labels per row–column combination/cell (Both).

Table 24: **ImageNet-Rendition Breakdown.** Information (AMI, %) about different aspects of images in ImageNet-R: IN-1k object class, style of rendition, and their combinations. **Bold**: best per aspect. Underlined: best per arch. Background: from median AMI (white) to max (blue) per aspect. FT: fine-tuned with x-ent. on IN-1k.

Arch.	Encoder	FT	Class	Artform	Both
RN50	X-Ent.		34	19	29
	MoCo-v3		26	19	23
	DINO		18	**24**	20
	VICReg		20	23	21
	MoCo-v3	✓	35	18	29
	DINO	✓	34	18	28
	VICReg	✓	33	19	28
ViT-B	X-Ent.		38	19	32
	MoCo-v3		26	**25**	26
	DINO		33	23	30
	MAE (CLS)		10	16	11
	MAE (avg)		10	19	13
	MoCo-v3	✓	**44**	18	36
	DINO	✓	43	18	35
	MAE (avg)	✓	**44**	18	**36**

Q. BreakHis Information Breakdown

BreakHis (Spanhol et al., 2016) is a medical dataset containing images of microscopic images of breast tumor tissue collected from 81 patients. At a coarse level, the tumor can be malignant (cancerous) or benign (normal cells). Within each of these categories, the dataset contains samples for four distinct types of benign tumor and four types of malignant tumor. Images were taken for each slide (one slide per subject) at varying zoom levels (40x, 100x, 200x, 400x).

We investigated how much information the clustered embeddings contained about each of these labels, shown in Table 25. We found that SSL pretrained encoders were much better at encoding the medically relevant information about the tumor's malignancy and specific type, with up to twice as much AMI than the supervised and fine-tuned models. The embeddings from the SSL encoders were generally also superior for encoding the magnification level, and the slide ID. However, the MAE model's clusters were worst at encoding the magnification, and the MoCo-v3 model worst at encoding the slide ID. We hypothesize that MoCo-v3's poor performance on slide ID may be because the types of differences between subjects may be comparable to the augmentations it is tasked with being *robust* to during training.

Across all label types for this dataset, SSL pretrained models produced the best clusters. Within these, DINO was the best performing model with either ResNet-50 or ViT-B architecture. The DINO training paradigm features multi-crop training, which may have helped the encoder to produce encodings which work well on this dataset which includes images at a variety of zoom levels and hence features at varying apparent scales.

Table 25: **BreakHis Breakdown.** Information (AMI, %) about different aspects of images in BreakHis: **Bold**: best encoder per aspect. Underlined: best encoder per arch. Background: from median AMI (white) to max (blue) per aspect. FT: fine-tuned with x-ent. on IN-1k.

Arch.	Encoder	FT	Malignancy	Tumor type	Magnification	Tumor type x Magnifn.	Slide ID
RN50	X-Ent.		7	12	23	26	23
	MoCo-v3		5	10	31	30	18
	DINO		**14**	**22**	**35**	**43**	35
	VICReg		9	15	33	36	25
	MoCo-v3	✓	6	10	19	22	18
	DINO	✓	7	11	18	22	20
	VICReg	✓	6	10	19	22	19
ViT-B	X-Ent.		7	13	20	25	23
	MoCo-v3		12	19	30	37	31
	DINO		**14**	**22**	32	40	**36**
	MAE (CLS)		**14**	19	19	28	32
	MAE (avg)		11	16	25	30	26
	MoCo-v3	✓	6	10	22	24	19
	DINO	✓	7	12	21	24	21
	MAE (avg)	✓	8	13	22	26	22

R. Correlation Between AMI and Silhouette Score

As described in §3.3, we evaluated the correlation between the AMI and silhouette scores across all datasets and encoders, for each clusterer, which we plot for the raw values (Fig. 9) and the ranking of the each metric (Fig. 10). We observe that in the UMAP-reduced embedding space a larger extent of the silhouette score's range is used, making the correlation between AMI and S more clear. This increases the usability of the silhouette score as a proxy.

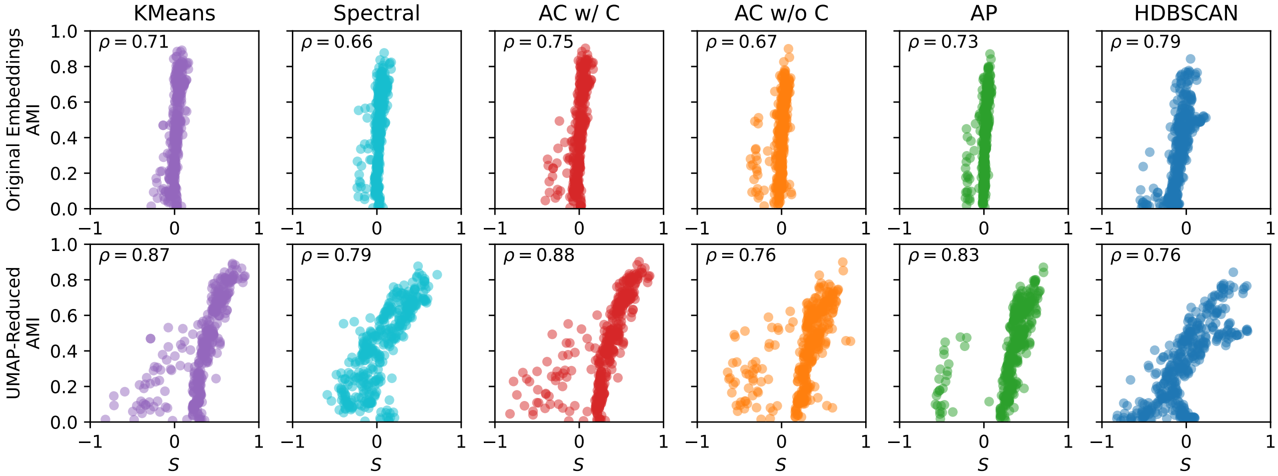


Figure 9: **AMI–Silhouette scatter plots.** The AMI and silhouette score (S) per clusterer, across datasets and encoders. The silhouette scores are measured in the original (top) and UMAP-reduced 50-d (bottom) feature spaces. We indicate the per-clustering-method Spearman's rank correlation (ρ).

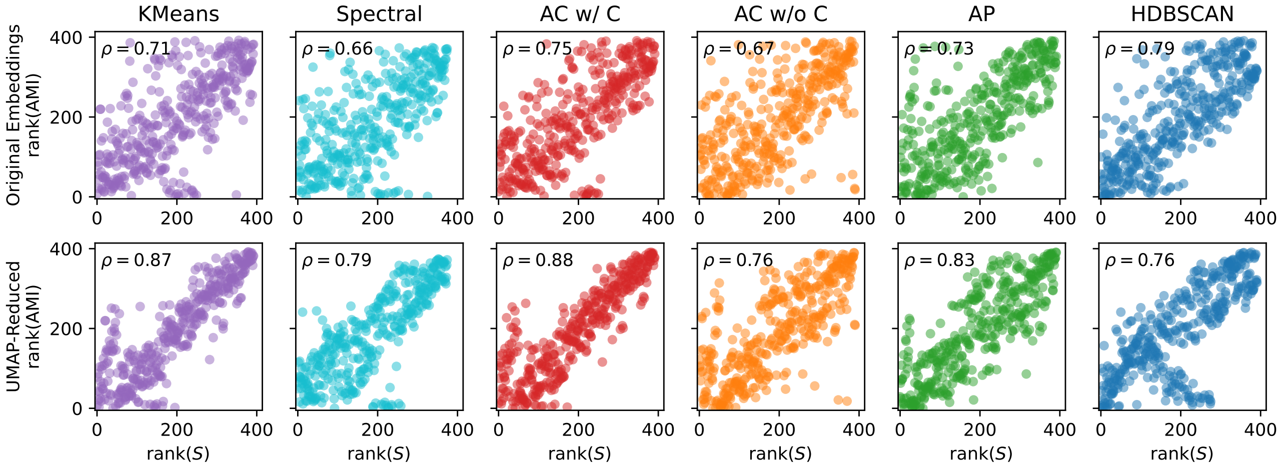


Figure 10: **Ranked AMI–Silhouette scatter plots.** The ranked AMI and silhouette score (S) per clusterer, across datasets and encoders (higher is better). The silhouette scores are measured in the original (top) and UMAP-reduced 50-d (bottom) feature spaces. We indicate the per-clustering-method Spearman's rank correlation (ρ).

S. Correlation Between Clustering and kNN

Classically, SSL encoders have been evaluated by determining their classification performance through *e.g.* kNN-probing (Balestriero et al., 2023). We propose that the quality of the clusters measured using AMI can function as an orthogonal measure. Therefore, we compare the AMI score of the different clustering methods with the accuracy obtained using kNN-probing with $k = \{1, 10, 20, 100, 200\}$ aggregating across all encoders and datasets. kNN-probing was chosen due to the computational restriction from running linear probing across all datasets for all methods. Following Caron et al. (2021) we use a weighted kNN-probing approach. Using the Spearman's rank correlation coefficient we find that there is a moderate positive correlation between kNN-probing accuracy and AMI ($0.33 \leq \rho \leq 0.54$), see Table 26. Specifically, we find that HDBSCAN and Spectral Clustering correlates the most with kNN-probing, while AP and AC w/o C correlates the least. From this we can conclude that measuring the clustering performance of the SSL encoders is not redundant, but instead is an orthogonal manner of measuring the performance of the encoders.

Table 26: **Correlation between clustering AMI and kNN accuracy.** We measure the Spearman's rank correlation coefficient (%) between the AMI score of the clustered embeddings and the accuracy of the classes predicted using kNN-probing, aggregated across datasets and encoders. The two measures are consistently less correlated when using the ViT-B embeddings, and for all clustering methods we find that the two evaluation methods are only moderately correlated.

		Number of neighbours (k)				
Arch.	Clusterer	1	10	20	100	200
RN50	K-Means	51	48	48	48	49
	Spectral	54	52	52	53	53
	AC w/ C	49	47	47	47	47
	AC w/o C	45	44	44	44	44
	Affinity Prop.	49	47	47	47	47
	HDBSCAN	51	49	49	49	49
ViT-B	K-Means	37	37	37	39	39
	Spectral	37	38	38	40	40
	AC w/ C	33	34	34	35	36
	AC w/o C	39	40	40	41	41
	Affinity Prop.	36	37	36	38	38
	HDBSCAN	38	39	38	40	40

T. Impact Statement

In this paper we analyze self-supervised encoders from the perspective of clustering. While the main goal of the paper is to advance our collective understanding of self-supervised learning, we acknowledge that the clustering process may lead to the construction of clusters which amplify stereotypical or biased groupings.

U. Limitations

While our evaluation has spanned a broad range of test datasets, we have only considered models pretrained on ImageNet-1k. Consequently, we have only studied the ability of models trained on ImageNet-1k to generalize to other datasets. While we anticipate that our findings would generalize to models trained on other datasets (with the unseen datasets being in- and out-domain changed to reflect the new training domain), this assumption has not been verified.

An aspect of changing the training data that is more likely to impact our findings is the diversity of the training data. Whilst models which are trained on a larger dataset will have a larger in-domain space, some data will still be out-of-domain and thus our considerations will be meaningful. However, the ability of models to generalize from larger datasets could be impacted differently depending on the pretraining paradigm.

Our work was constrained to only one data modality: vision. While we anticipate that our findings would generalize to other modalities provided the pretraining paradigms are comparable, this is yet to be verified.

The clusterings we have performed were on the embeddings of fully-trained networks. The behaviour of untrained networks (and to a lesser extent, MAE-trained networks without a whole-stimulus target) was not consistent with that of the trained networks in some regards, as we note in §E.2 and as previously observed by (Xu et al., 2022). Consequently, our finding that the intrinsic silhouette score of clustered UMAP-reduced embeddings is correlated with the performance of the encoder on the target dataset may not be applicable to measuring performance in the middle of training, while the feature space is still transitioning from an amorphous distribution to a structured manifold subspace.

We considered only one type of supervised pretraining, cross-entropy. We assume that other supervised loss functions would produce a similar outcome.

In our work, we explored the effect of fine-tuning the self-supervised pretrained encoders on ImageNet-1k and found their behaviour was similar to models trained from scratch with cross-entropy. However, we did not investigate the behaviour of the encoder during training. Consequently, it is unclear when the transition from the SSL-pretrained encoder behaviour to supervised training occurs.

While we considered two architectures in this paper (ResNet-50 and ViT-B), they are not of similar capacities and so it is not possible for us to draw conclusions about which architecture generalizes best outside its training domain. Consequently, we make no claims in this regard.