Aalborg Universitet

AALBORG UNIVERSITY
DENMARK

Zero-shot Clustering of Embeddings with Self-Supervised Learnt Encoders

Lowe, Scott C.; Haurum, Joakim Bruslund; Oore, Sageev; Moeslund, Thomas B.; Taylor, Graham W.

Creative Commons License
CC BY 4.0

Publication date:
2023

Document Version
Accepted author manuscript, peer reviewed version

Link to publication from Aalborg University

Citation for published version (APA):
Lowe, S. C., Haurum, J. B., Oore, S., Moeslund, T. B., & Taylor, G. W. (2023). *Zero-shot Clustering of Embeddings with Self-Supervised Learnt Encoders*. Paper presented at 4th Workshop on Self-Supervised Learning: Theory and Practice (NeurIPS 2023), New Orleans, Louisiana, United States.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

Take down policy
If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from vbn.aau.dk on: April 28, 2024

Zero-shot Clustering of Embeddings with Self-Supervised Learnt Encoders

Scott C. Lowe[*1,2], **Joakim Bruslund Haurum**[*3,4],
Sageev Oore[†1,2], **Thomas B. Moeslund**[†3,4], **and Graham W. Taylor**[†1,5]
[1] Vector Institute for AI, Canada, [2] Dalhousie University, Canada, [3] Aalborg University, Denmark,
[4] Pioneer Centre for AI, Denmark, [5] University of Guelph, Canada

Abstract

We explore whether self-supervised pretrained models can provide a useful representation space for datasets they were not trained on, and whether these representations can be used to group novel unlabelled data into meaningful clusters. To this end, we conduct experiments using image representation encoders pretrained on ImageNet using a variety of self-supervised training techniques. These encoders are deployed on image datasets that were not seen during training, without fine-tuning, and we investigate whether their embeddings can be clustered with conventional clustering algorithms. We find that it is possible to create well-defined clusters using self-supervised feature encoders, especially when using the Agglomerative Clustering method, and that it is possible to do so even for very fine-grained datasets such as NABirds. We also find indications that the Silhouette score is a good proxy of cluster quality when no ground-truth is available.

1 Introduction

Self-supervised learning (SSL) has seen a large amount of interest in recent years across almost every machine learning sub-field, due to the promise of being able to harness the large quantities of unlabelled data available and obtaining generic feature embeddings useful for a variety of downstream tasks Balestriero et al. (2023). 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). While the embeddings from an SSL-trained encoder can perform well on downstream tasks after fine-tuning, there has been little investigation into the utility of the embeddings without any retraining. Prior work by Vaze et al. (2022) and Zhou & Zhang (2022) suggests SSL feature encoders generate embeddings suitable for clustering, but nonetheless still fine-tune the encoders. Yet, widespread interest in application of large pretrained models on custom datasets, combined with prohibitive cost of compute, make this question important and increasingly urgent.

We find that there has so far been no investigation into whether SSL-trained feature encoders can generate informative clusters of embeddings on datasets that were totally unseen to the encoder. In this work, we perform a zero-shot transfer learning task, evaluating the performance of a suite of SSL-trained feature encoders across a diverse set of datasets, using different classical clustering methods. In summary, we make the following contributions:

- We conduct the first investigation into zero-shot clustering of SSL feature encoders, finding that contrastive and multi-modal SSL approaches can produce meaningful clusters across a variety of datasets without per-dataset parameter tuning.
- We find that the Agglomerative Clustering method is consistently strong across SSL encoders, backbones, and datasets—especially on representations reduced with UMAP.

[*]Joint first author. [†]Joint last author. Correspondence: scott.lowe@vectorinstitute.ai

4th Workshop on Self-Supervised Learning: Theory and Practice (NeurIPS 2023).

- We find the Silhouette score in a UMAP-reduced space is highly correlated with the AMI, and 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 self-supervised encoders. The aim of this task is to cluster the feature embeddings from various as-yet unseen datasets, in a way such that the clusters are intrinsically well-defined and, ideally, match the ground-truth label assignments. Our feature encoders and clustering methods are only tuned on data from a single dataset, the commonly used ImageNet-1k (Russakovsky et al., 2015). This methodology is then deployed on all other tested datasets without re-tuning any of the parameters.

Feature Encoders In order to capture the diverse methodologies within the self-supervised learning field, we compare methods from the major self-supervised paradigms within computer vision; see §A.1 for an overview. We choose one representative method per paradigm (below), and compare the clusterability of their features against a model pretrained with cross-entropy supervision.

- **Contrastive Learning**: MoCo-v3 (Chen et al., 2021)
- **Self-Distillation**: DINO (Caron et al., 2021)
- **Canonical Correlation Analysis**: VICReg (Bardes et al., 2022)
- **Masked Image Modelling**: MAE (He et al., 2022)
- **Multi-Modal Learning**: CLIP (Radford et al., 2021)

For each method we consider two common backbone networks: ResNet-50 (He et al., 2016) and ViT-B (Dosovitskiy et al., 2021) trained on ImageNet-1k, using publicly available checkpoints. However, note that (1) MAE only supports transformer architectures and so does not have a ResNet-50 checkpoint; (2) VICReg does not have a pretrained ViT-B checkpoint; and (3) the CLIP model makes several modifications to the backbone architectures. Furthermore, CLIP was not trained on ImageNet-1k, instead on a different, non-disclosed, large dataset of paired images and text captions. We include CLIP nonetheless since it has previously been shown to perform well on zero-shot classification tasks when supplied with text embeddings of the class names (Radford et al., 2021).

Clustering Methods In order to cluster the feature embeddings, we considered several classical clustering methods: K-Means (Lloyd, 1982) with K-Means++ initialization Arthur & Vassilvitskii (2007), Agglomerative Clustering (AC; Everitt et al., 2011), Affinity Propagation (Frey & Dueck, 2007), and HDBSCAN (McInnes & Healy, 2017). These were chosen because they have few hyperparameters to tune, cover several clustering paradigms (partition, hierarchical, graph-theory, and density), and include both parametric and non-parametric methods. Since K-Means requires the number of clusters, we assume that this is known *a priori*. In contrast, HDBSCAN, AC, and Affinity Prop. can automatically determine the number of clusters in the data.

Experimental Methodology For each test dataset (see Appendix B), we preprocessed the images by resizing to 224px and taking a centered 224×224 crop. Images were standardized using the mean and std. dev. used to train each encoder, then passed through the encoder to create embeddings.

To maximize the performance of the clusterers on each encoder, we conducted a sweep over the relevant clustering hyperparameters. The sweep was optimized on subsets of the training splits of ImageNet-1k, Imagenette, and Imagewoof (Howard, 2019). For more details, see §C.2. We then clustered the generated embeddings with each clusterer, using these discovered hyperparameters. When using UMAP or PCA, this was fit separately on each dataset. When computing the Silhouette score, we perform this in the dimensionality-reduced space, not the original embedding space.

3 Experimental Results

We measured the Adjusted Mutual Information (AMI; see §C.1.1) between the annotated classes and the clusters, tabulated in Table 2a and Table 2b, for ResNet-50 and ViT-B backbones, respectively. For full details, see §D.1. Across both ResNet-50 and ViT-B, the best performance on ImageNet-1k, CIFAR-10, and CIFAR-100 (datasets most similar to the training set) is obtained using the encoders

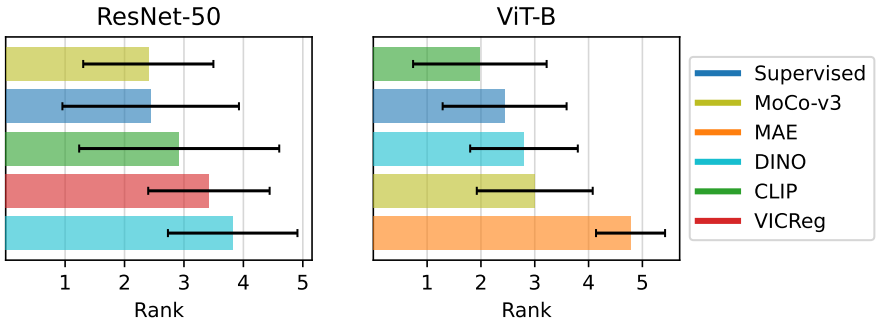

Figure 1: **Average SSL Encoder rank** (lower is better). The average rank of each tested SSL encoder plotted with ± 1 std dev. For both ResNet-50 and ViT-B, an SSL encoder results in the best clustering but the supervised method also in general produces good clusters.

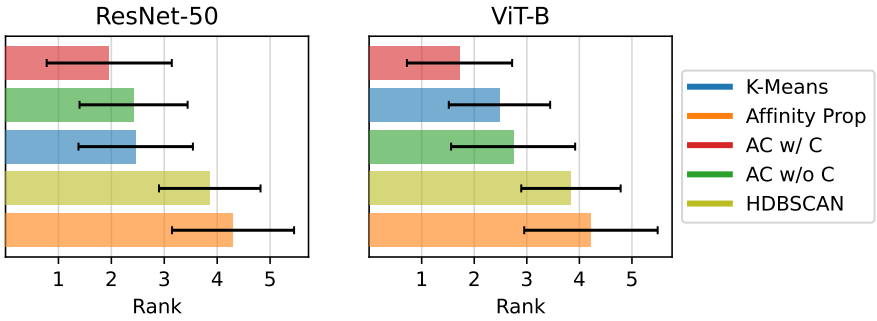

Figure 2: **Average clustering method rank** (lower is better) of each clustering method, ± 1 std dev. AC performs very well, whether the number of cluster are known *a priori* (Red) or not (Green).

trained with conventional classification supervision. However, for MNIST and Fashion-MNIST, we find the SSL encoders are much more competitive, with MoCo-v3 achieving the highest AMI in all but one case. For the smaller fine-grained datasets (Aircraft, Flowers, NABirds) and SVHN, we find the multi-modal CLIP encoder achieves the best performance, whilst the supervised network performs particularly poorly on Flowers.

Comparison of SSL Encoders To directly compare the different pretrained encoders, we rank each encoder across datasets and clustering methods (Fig. 1). We find MoCo-v3 performs best for ResNet-50, and CLIP best for ViT-B. Surprisingly, with a ResNet-50 backbone the CLIP method performs poorly, despite being trained on a much larger dataset. The supervised baseline is the second-best for both backbones. The DINO self-distillation approach performs well using a ViT-B backbone, but very poorly with ResNet-50 (the same trend as for CLIP), corroborating Vaze et al. (2022). Lastly, MAE performed poorly across all datasets, congruent with the observation that MAE models possess details about the pixel-level of stimuli, but need fine-tuning to perform well at whole-image classification (He et al., 2022).

Comparison of Clustering Methods We compared the performance of the clustering methods by ranking each method for each combination of encoder and dataset. As shown in Fig. 2, it is immediately obvious the best performing clusterer across both backbones is AC with the number of clusters known *a priori*. However, we find AC with an unknown number of clusters is competitive, outperforming K-Means when using a ResNet-50 backbone. HDBSCAN and Affinity Prop. are consistently the worst performing clusterers considered.

Effect of Dataset Granularity The clustering performance varies on the fine-grained datasets. While seemingly arbitrary, we find the performance correlates with how fine-grained the datasets are when considering the proposed granularity measure from Cui et al. (2019). Specifically, we

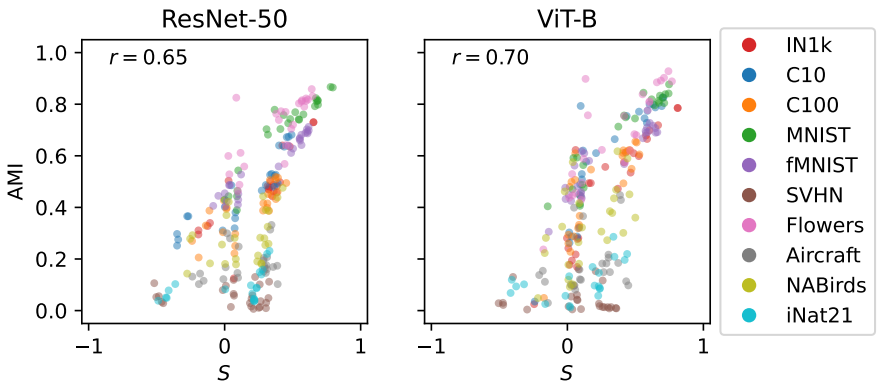

Figure 3: **AMI–Silhouette scatter plots.** The AMI and Silhouette score, S, of each encoder and clustering method combination are plotted against each other across all datasets, per backbone.

find Aircraft is the most challenging to cluster, while NABirds and Flowers are easier, matching their coarseness rankings by Cui et al. (2019). These observations echo recent results from Cole et al. (2022), where it was determined that current SSL methods are unsuitable for fine-grained tasks. The AMI for SVHN is dramatically lower than all other datasets. We believe this is due to the large intra-class diversity for each digit and small inter-class diversity, originating from the different coloured house walls and several digits being visible in each image. In comparison, images in Flowers have perceptually less variability within classes, and clustering has much higher agreement with the annotations. Lastly, we find most combinations of encoders and clustering methods perform poorly on iNat21, due to the large number of species, spanning the entire tree of life (Van Horn et al., 2021). The exception is AC, where performance is dramatically higher, reaching an AMI of 28.6%.

Correlation between AMI and Silhouette Score In the preceding analysis we focused on AMI, which measures performance by comparing the predicted clusters with ground-truth labels. However, in the context of SSL this is problematic since there is no ground-truth available. Therefore, the intrinsic Silhouette score metric (see § C.1.2), S, calculated from just the predicted clusters is potentially valuable for evaluation of SSL encoders. We investigated the relationship between AMI and S by computing the Pearson correlation coefficient between them across all encoder-clusterer combinations (Fig. 3). We find AMI and S are strongly correlated: high AMI scores have high Silhouette scores. For a per-dataset breakdown, see § D.4. We find S can be a good proxy when ground-truth labels are not available, but its effectiveness diminishes when data becomes more fine-grained and further from the training domain.

4 Conclusion

We empirically investigated how well the feature embeddings produced by pretrained networks can be clustered in a zero-shot setting. We considered two architectures trained using one of six methodologies (one supervised, five SSL), on 10 datasets, using five classic clustering methods. We find it's possible to create well-defined clusters across nearly all tested datasets, even notoriously hard fine-grained datasets such as NABirds. In many cases, the performance on novel datasets was comparable to that on the in-domain ImageNet-1k test set. AC is consistently the strongest clusterer when the number of clusters is known *a priori*, and when the number of classes are not known (using a distance threshold tuned on ImageNet-1k). In contrast, there is not a single overall best SSL paradigm. Instead, we find the contrastive MoCo-v3 method is the best with ResNet-50, whereas the multi-modal CLIP approach is the strongest when using a ViT-B backbone.

To cluster embeddings of a novel dataset, we recommend reducing the dimensionality with UMAP (use 5–100 dims), then applying Agglomerative Clustering. We also show promising results that Silhouette score can be used to evaluate SSL methods for clustering when no ground-truth is available.

We believe these results shed an important light on the capabilities of SSL trained encoders, highlighting they in many cases produce meaningful clusters on new datasets without additional tuning.

References

Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. OPTICS: Ordering points to identify the clustering structure. *SIGMOD Rec.*, 28(2):49–60, jun 1999. ISSN 0163-5808. doi: 10.1145/304181.304187. URL https://doi.org/10.1145/304181.304187.

David Arthur and Sergei Vassilvitskii. 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.

Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, Avi Schwarzschild, Andrew Gordon Wilson, Jonas Geiping, Quentin Garrido, Pierre Fernandez, Amir Bar, Hamed Pirsiavash, Yann LeCun, and Micah Goldblum. A cookbook of self-supervised learning, 2023. arXiv: 2304.12210.

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BERT pre-training of image transformers. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=p-BhZSz59o4.

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *ICLR*, 2022.

Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

James C. Bezdek, Robert Ehrlich, and William Full. FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2):191–203, 1984. ISSN 0098-3004. doi: https://doi.org/10.1016/0098-3004(84)90020-7. URL https://www.sciencedirect.com/science/article/pii/0098300484900207.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (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.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *European Conference on Computer Vision*, 2018.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh (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.

Xinlei Chen and Kaiming He. 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.

Xinlei Chen, Saining Xie, and Kaiming He. 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.

Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pp. 539–546. IEEE, 2005.

Elijah Cole, Xuan Yang, Kimberly Wilber, Oisin Mac Aodha, and Serge Belongie. 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.

Yin Cui, Zeqi Gu, Dhruv Mahajan, Laurens van der Maaten, Serge Belongie, and Ser-Nam Lim. Measuring dataset granularity, 2019. arXiv: 1912.10154.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

Aleksandr Ermolov, Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening for self-supervised representation learning. In Marina Meila and Tong Zhang (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`.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 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.

Brian S. Everitt, Sabine Landau, Morven Leese, and Daniel Stahl. *Cluster Analysis*. Wiley, January 2011. doi: 10.1002/9780470977811. URL `https://doi.org/10.1002/9780470977811`.

Brendan J. Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007. doi: 10.1126/science.1136800. URL `https://www.science.org/doi/abs/10.1126/science.1136800`.

Priya Goyal, Mathilde Caron, Benjamin Lefaudeux, Min Xu, Pengchao Wang, Vivek Pai, Mannat Singh, Vitaliy Liptchinsky, Ishan Misra, Armand Joulin, and Piotr Bojanowski. Self-supervised pretraining of visual features in the wild, 2021. arXiv: 2103.01988.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (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`.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 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.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 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.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 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.

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

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th*

International Conference on Machine Learning, volume 139 of *Proceedings of Machine Learning Research*, pp. 4904–4916. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/jia21b.html.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. URL https://api.semanticscholar.org/CorpusID:18268744.

Yann Lecun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

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

Subhransu Maji, Juho Kannala, Esa Rahtu, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft, 2013. arXiv: 1306.5151.

Leland McInnes. 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.

Leland McInnes and John Healy. Accelerated hierarchical density based clustering. In *Data Mining Workshops (ICDMW), 2017 IEEE International Conference on*, pp. 33–42. IEEE, 2017.

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

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. UMAP: Uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861, 2018.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. 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.

Maria-Elena Nilsback and Andrew Zisserman. 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.

Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4004–4012, 2016.

Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision, 2023. arXiv: 2304.07193.

Ari Pakman, Yueqi Wang, Catalin Mitelut, Jinhyung Lee, and Liam Paninski. Neural clustering processes. In Hal Daumé III and Aarti Singh (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.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2536–2544, 2016.

Karl Pearson. 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.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, 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.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Meitar Ronen, Shahaf E. Finder, and Oren Freifeld. 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.

Peter J. Rousseeuw. 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: https://doi.org/10.1016/0377-0427(87)90125-7. URL https://www.sciencedirect.com/science/article/pii/0377042787901257.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115 (3):211–252, April 2015. doi: 10.1007/s11263-015-0816-y. URL https://doi.org/10.1007/s11263-015-0816-y.

Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29, 2016.

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

Makarand Tapaswi, Marc Law, and Sanja Fidler. 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.

Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. SCAN: Learning to classify images without labels. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision – ECCV 2020*, pp. 268–285, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58607-2.

Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. 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.

Grant Van Horn, Elijah Cole, Sara Beery, Kimberly Wilber, Serge Belongie, and Oisin Mac Aodha. Benchmarking representation learning for natural world image collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12884–12893, June 2021.

Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Generalized category discovery. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Is a correction for chance necessary? In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pp. 1073–1080, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585161. doi: 10.1145/1553374.1553511. URL https://doi.org/10.1145/1553374.1553511.

Joe H. Ward. 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.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms, 2017. arXiv: 1708.07747.

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

Bo Yang, Xiao Fu, Nicholas D. Sidiropoulos, and Mingyi Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In Doina Precup and Yee Whye Teh (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.

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

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stephane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In Marina Meila and Tong Zhang (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.

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. iBOT: Image BERT pre-training with online tokenizer, 2022.

Xingzhi Zhou and Nevin L. Zhang. Deep clustering with features from self-supervised pretraining, 2022.

A Related Work

Our work builds upon two broad fields of research: self-supervised learning for computer vision applications, and unsupervised clustering. We give a general overview of each field.

A.1 Self-supervised learning

Self-Supervised Learning (SSL) has recently received an increasing amount of interest within 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, the currently dominant methods can be divided into five general categories as follows (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); Oh 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 the embeddings of the teacher (Grill et al., 2020; Chen & He, 2021; Caron et al., 2021; Zhou et al., 2022; Oquab et al., 2023). (3) Canonical Correlation Analysis approaches, where the 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 (Zbontar et al., 2021; Bardes et al., 2022; Caron et al., 2020; Ermolov et al., 2021). (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). (5) Multi-Modal Learning approaches, where the utilized data consists of different modalities, such as image-text pairs, which are separately embedded and must be aligned (Radford et al., 2021; Jia et al., 2021).

A.2 Clustering

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 (Frey & Dueck, 2007; Yu & Shi, 2003), density (Ankerst et al., 1999; Ester et al., 1996; McInnes & Healy, 2017), hierarchies (Ward, 1963; Sokal & Michener, 1958), 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 (Pakman et al., 2020; Caron et al., 2018; Tapaswi et al., 2019; Ronen et al., 2022; Yang et al., 2017; Van Gansbeke et al., 2020).

B Datasets

We evaluate 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 ImageNet training split has previously been observed during training of the feature encoders* as well as setting the hyperparameters of the clustering method. All other datasets have not previously been observed by the model and the considered methods are not tuned in any way on these.

Table 1: **Dataset overview.** For our zero-shot clustering protocol we consider a diverse set of experiments of differing levels of task granularity, number of classes and samples, domain shift, and class imbalance. The reported numbers are on the publicly available test splits. If the test labels are not publicly available the public validation split is used instead. The class imbalance, ρ, is measured with the ratio between the number of samples in the largest and smallest classes in the dataset.

Dataset	Reference	№ Samples	№ Classes	ρ	Description
ImageNet-1k	(Russakovsky et al., 2015)	50,000	1,000	1.00	Diverse general objects
CIFAR10	(Krizhevsky, 2009)	10,000	10	1.00	Diverse general objects
CIFAR100	(Krizhevsky, 2009)	10,000	100	1.00	Diverse general objects
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
Oxford Flowers	(Nilsback & Zisserman, 2008)	6,149	102	11.90	Flower variants
FGVC Aircraft	(Maji et al., 2013)	3,333	100	1.03	Aircraft variants
NABirds	(Van Horn et al., 2015)	24,633	555	6.67	Bird species
iNaturalist-2021	(Van Horn et al., 2021)	100,000	10,000	1.00	Plant & animal species

C Additional Methodology Details

C.1 Evaluation Metrics

We evaluate the performance of a clustering using two metrics: Adjusted Mutual Information (AMI) (Vinh et al., 2009) and the Silhouette score (Rousseeuw, 1987). AMI measures the agreement between the constructed clusters and the ground-truth clustering, while the Silhouette score measures how well-defined the clusters are irrespective of whether the cluster elements are correctly assigned.

C.1.1 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

*Except potentially the CLIP models, for which we don't know whether or not it was trained on these datasets.

Normalized Mutual Information (NMI) between two label assignments V an U is defined as

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

where $\text{MI}(U, V)$ is the mutual information between label assignments V an U, and $H(\dot{})$ is the Shannon entropy of the considered label assignment. 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.

However, NMI is not corrected for chance so its value can increase merely by increasing the number of clusters used (Vinh et al., 2009). In order to account for this, we use the Adjusted Mutual Information metric proposed by Vinh et al. (2009), defined as

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

where $\mathbb{E}[\text{MI}(U, V)]$ 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).

Among the clusterers we considered, HDBSCAN can identify samples which belong to *no* cluster (noise samples). Unless stated otherwise, we consider the noise class to be its own class when computing the AMI. This unfortunately sets HDBSCAN at a disadvantage, since the samples it identifies as noise are typically distributed across all ground-truth classes, but is fairer than ignoring samples it identifies as noise since that would evaluate it only on easier samples.

C.1.2 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.

C.2 Hyperparameter Search

In order to maximize the performance of each permutation of the feature encoder and clustering methods, we conducted a staggered sweep over the relevant clustering hyperparameters. The sweep was conducted using subsets of the training splits of ImageNet-1k, Imagenette, and Imagewoof (Howard, 2019). Imagenette and Imagewoof are coarse- and fine-grained subsets of ImageNet-1k, respectively, with 10 classes each. These datasets were selected to find hyperparameters which were 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 the three datasets, we created a validation set by taking a class-stratified random subset of the training set, using the same number of samples as appeared in the datasets' test set (50000, 3925, and 3929 respectively). The same split was used across all encoders, clusterers, and stages of the hyperparameter search. For Affinity Propagation, it was not feasible to conduct this search on ImageNet due to compute and memory scaling w.r.t. number of samples; hence we optimized Affinity Propagation hyperparameters using Imagenette and Imagewoof only.

First, as the curse of dimensionality can negatively affect the performance of the considered clustering methods (Bellman, 1966), we searched for an appropriate dimensionality reduction process. We

compared the performance of using the original un-reduced feature embedding space (up to 2048-d) against applying PCA (Pearson, 1901) or UMAP (McInnes et al., 2018) to reduce the number of dimensions. Specifically, we considered reducing the feature embeddings to $[2, 5, 10, 20, 50, 100, 200, 500]$ with either PCA or UMAP, and considered reducing the number of dimensions to capture a target fraction of total variance of the data $[0.75, 0.8, 0.85, 0.9, 0.95, 0.98, 0.99]$. To perform PCA, we first took the z-score of each dimension and then used the default hyperparameters of SCIKIT-LEARN (Pedregosa et al., 2011), without whitening the data. To perform UMAP, we increased the number of neighbours considered to 30 and decreased the minimum distance to 0, following the recommendations of (McInnes, 2018); we otherwise used the default hyperparameters of UMAP (McInnes et al., 2018). In this first stage, we used the default hyperparameters of the clustering methods as defined in SCIKIT-LEARN. For K-Means and AC, we provided the number of annotated classes within the dataset (1000 or 10) as number of clusters to produce. For each encoder and clusterer, we took the average AMI over the three datasets and selected the method which yielded the highest average AMI (a particular PCA dim, PCA variance, UMAP dim, or no reduction).

We observed that for K-Means, AC, and HDBSCAN, the majority of encoders all 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 500. 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). For Affinity Propagation, PCA outperformed UMAP (as it failed to converge on UMAP-reduced embeddings); most encoders worked best with a 10-dim PCA reduction.

In the second stage, using the dimensionality reductions per encoder from the first stage, we iterated over the per-method specific hyperparameters for AC. Continuing to use the "ground-truth" number of classes as the number of clusters, we evaluated all combinations of distance metric (ℓ_1, ℓ_2, ℓ_∞, cosine) and linkage method (ward [ℓ_2 only], complete, average, single), for 13 options in total. For each encoder, we selected the metric and linkage which yielded the best average AMI over the three datasets. The selected options were ℓ_2 + ward (5 encoders), ℓ_2 + avg (3 encoders), or ℓ_∞ + avg (2 encoders).

Thirdly, we tuned 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*. 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, and then selected the distance threshold which yielded the highest average AMI.

For Affinity Propagation, we explored the impact of the convergence threshold and damping parameters, and confirmed the performance on Imagenette and Imagewoof were insensitive to these parameters.

For HDBSCAN, we noticed that for some encoders it would select very few clusters for Imagenette and Imagewoof, reducing 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 experiments, so as to ensure HDBSCAN produced more than a couple of clusters but without forcing it to produce a certain number of clusters.

D Supplementary Results

D.1 Adjusted Mutual Information

We measured the Adjusted Mutual Information (AMI; see §C.1.1) between the annotated classes and the clusters, shown in Table 2a and Table 2b, for ResNet-50 and ViT-B backbones, respectively*.

Across both the ResNet-50 and ViT-B backbones, the best performance on ImageNet-1k (the dataset used for training) and CIFAR-10 and CIFAR-100 (the datasets most similar in their domain to ImageNet) is obtained using the encoders trained with conventional classification supervision. For ImageNet, the gap between the supervised and self-supervised methods is especially noticeable, with a difference of nearly 14 percentage points with the ResNet-50 between the supervised method and

*Some Affinity Prop. results couldn't be obtained due poor memory and compute scaling with N samples.

Table 2: **AMI scores of SSL encoders and clustering methods.** We report AMI score, as a percentage, on each dataset (see Appendix B) for each encoder and clusterer. The performance of Agglomerative Clustering is shown twice: once using the ground-truth num. of classes as the num. of clusters (AC w/ C), once predicting num. of clusters (AC w/o C). The best combination of encoder/clusterer per dataset and backbone is **bolded**; the best encoder per clusterer is underlined.

(a) **AMI score (%) with a ResNet-50 backbone.**

	Encoder	IN1k	C10	C100	MNIST	fMNIST	SVHN	Flowers	Aircraft	NABirds	iNat21
K-Means	Supervised	**73**	**68**	**51**	81	69	5	64	15	39	9
	MoCo-v3	48	64	51	**86**	**71**	11	80	21	28	4
	VICReg	46	53	45	80	70	3	81	16	18	4
	DINO	44	49	42	74	64	1	82	18	18	4
	CLIP	50	49	40	54	53	1	83	30	42	8
AC w/ C	Supervised	**73**	67	**52**	82	69	4	64	15	39	9
	MoCo-v3	49	64	51	**87**	70	10	81	20	28	5
	VICReg	46	53	45	79	69	1	82	16	19	4
	DINO	48	48	42	74	67	1	82	19	21	7
	CLIP	50	52	39	81	61	1	**86**	31	44	10
AC w/o C	Supervised	64	67	49	74	66	6	57	17	**48**	22
	MoCo-v3	48	64	46	82	68	**13**	70	17	32	15
	VICReg	47	53	43	76	67	5	72	10	26	14
	DINO	47	47	40	70	63	3	79	17	25	16
	CLIP	50	49	39	76	56	2	83	**33**	44	**23**
Affinity Prop	Supervised	69	40	39	44	42	10	53	12	37	–
	MoCo-v3	18	38	30	46	45	12	46	15	20	–
	VICReg	12	32	25	39	43	6	49	13	18	–
	DINO	–	31	30	43	41	8	60	17	18	–
	CLIP	–	36	22	44	41	6	61	25	28	–
HDBSCAN	Supervised	64	37	43	70	49	6	56	10	28	8
	MoCo-v3	34	37	38	77	45	11	76	14	26	5
	VICReg	33	30	33	73	49	5	78	12	14	5
	DINO	29	28	28	68	44	3	77	13	18	4
	CLIP	31	25	22	78	41	3	78	28	29	10

(b) **AMI score (%) with a ViT-B backbone.**

	Encoder	IN1k	C10	C100	MNIST	fMNIST	SVHN	Flowers	Aircraft	NABirds	iNat21
K-Means	Supervised	**79**	**83**	**65**	80	70	1	68	18	38	8
	MoCo-v3	60	79	62	83	71	1	81	15	27	6
	MAE	19	29	29	48	58	1	46	10	10	1
	DINO	67	77	62	81	69	1	89	21	44	9
	CLIP	62	79	61	56	61	10	90	40	58	13
AC w/ C	Supervised	**79**	**83**	**66**	84	71	2	68	18	39	9
	MoCo-v3	61	80	62	84	**73**	1	81	15	31	10
	MAE	24	29	29	59	62	0	53	10	12	2
	DINO	68	75	62	83	69	1	90	22	47	15
	CLIP	61	80	61	**88**	69	12	**93**	43	**62**	17
AC w/o C	Supervised	70	79	61	80	67	3	58	18	45	21
	MoCo-v3	47	77	55	82	72	1	62	10	30	20
	MAE	28	30	26	59	56	2	44	8	18	6
	DINO	53	73	49	81	69	1	78	11	36	22
	CLIP	57	75	60	75	57	15	89	**45**	59	**29**
Affinity Prop	Supervised	22	53	37	45	44	4	43	12	32	–
	MoCo-v3	17	49	33	41	46	6	49	12	18	–
	MAE	18	26	23	47	43	5	45	10	9	–
	DINO	28	45	32	41	45	4	59	16	27	–
	CLIP	28	51	32	49	44	**16**	76	37	42	–
HDBSCAN	Supervised	72	66	55	71	48	3	62	14	28	10
	MoCo-v3	49	62	50	76	48	3	75	11	22	7
	MAE	2	4	5	40	31	0	24	3	5	3
	DINO	56	58	51	74	45	2	84	15	33	9
	CLIP	49	61	48	84	43	13	88	32	46	12

the best self-supervised method (CLIP). However, for MNIST and Fashion-MNIST, we find the SSL encoders are much more competitive, with the contrastive MoCo-v3 encoder achieving the highest AMI in all but one case (MNIST with ViT-B, where CLIP is the best encoder), and the supervised network outperformed by multiple SSL encoders.

For the smaller fine-grained datasets (FGVC Aircraft, Oxford Flowers, and NABirds) as well as SVHN we find that the multi-modal CLIP encoder achieves the best performance for both ResNet-50 and ViT-B. The supervised network performs particularly poorly on Oxford Flowers (around 10 percentage points worse than the SSL networks). It is worth noting also that the performance on the SVHN dataset is dramatically lower than all other datasets. We believe this is due to the very large intra-class diversity for each digit and small inter-class diversity among digits, originating from the different colored house walls and several digits being visible in each image. In comparison, the images in Oxford Flowers have perceptually less variability within classes, and the clustering has much higher agreement with the annotations for this dataset. Lastly, we find that most combinations of encoders and clustering methods perform poorly on iNaturalist-21, due to the large number of considered species (10,000) spanning the entire tree of life (Van Horn et al., 2021). The exception is AC with unknown amount of clusters where performance is dramatically higher, reaching an AMI of 28.6%.

D.2 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 Table 3, the number of clusters predicted is typically a consistent order of magnitude for a given clusterer and dataset, irrespective of the encoder used. However there is great variability between clusterers. Affinity propagation predicted a couple of hundred clusters, irrespective of the dataset. Agglomerative clustering predicted the fewest clusters, even predicting only in the order of 100 clusters for ImageNet-1k, the dataset the encoders were trained on. HDBSCAN varied more in the number of clusters it predicted, with around the right number of classes being predicted for the datasets which were comprised of at least 100 classes.

D.3 Silhouette Scores

We report the Silhouette scores for each clustering of the test datasets, shown for ResNet-50 architectures in Table 4 and ViT-B architectures in Table 5.

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.

D.4 Per-dataset and Per-clusterer Correlation between AMI and Silhouette score

When looking at per-dataset r values, shown in Table 6a, we find that strongest correlations are obtained for ImageNet, CIFAR-10, CIFAR-100, MNIST and Fashion MNIST. However, for all

Table 3: **Predicted number of clusters.** For each clusterer, we report the number of clusters generated. We report the ground-truth number of classes in the dataset (Num targets), information which the clusterer was blinded to.

(a) **ResNet-50 backbone.**

	Encoder	IN1k	C10	C100	MNIST	fMNIST	SVHN	Flowers	Aircraft	NABirds	iNat21
	Num targets	1000	10	100	10	10	10	102	100	555	10 000
AC w/o C	Supervised	147	12	22	16	15	33	21	9	38	76
	MoCo-v3	63	14	19	14	15	32	25	8	28	72
	VICReg	70	16	22	16	15	30	29	7	29	69
	DINO	84	27	38	26	20	54	54	21	41	64
	CLIP	70	28	41	32	27	61	57	31	47	44
Affinity Prop	Supervised	4398	392	459	401	338	1041	340	385	974	–
	MoCo-v3	1324	295	348	296	194	839	254	132	536	–
	VICReg	1394	320	346	318	231	807	234	130	529	–
	DINO	–	741	872	754	623	3051	636	371	1507	–
	CLIP	–	286	366	288	207	696	224	132	632	–
HDBSCAN	Supervised	1181	228	196	81	178	617	180	98	526	1617
	MoCo-v3	1302	222	236	81	214	544	138	114	414	1685
	VICReg	1212	225	245	83	180	594	156	115	563	1687
	DINO	1163	265	244	87	188	678	148	109	376	1683
	CLIP	1072	276	260	57	225	631	148	83	462	1328

(b) **ViT-B backbone.**

	Encoder	IN1k	C10	C100	MNIST	fMNIST	SVHN	Flowers	Aircraft	NABirds	iNat21
	Num targets	1000	10	100	10	10	10	102	100	555	10 000
AC w/o C	Supervised	226	14	25	13	16	31	20	8	36	78
	MoCo-v3	93	16	20	14	11	13	23	6	18	28
	MAE	131	26	30	35	27	68	22	10	60	218
	DINO	90	8	14	9	9	6	33	5	18	17
	CLIP	111	30	71	35	55	125	58	40	78	79
Affinity Prop	Supervised	1456	164	274	261	198	712	223	168	426	–
	MoCo-v3	1274	205	293	292	185	756	234	101	440	–
	MAE	2009	422	461	400	304	840	355	170	948	–
	DINO	1090	246	330	284	201	698	222	98	420	–
	CLIP	738	196	314	250	192	617	174	108	566	–
HDBSCAN	Supervised	1123	118	209	100	209	594	174	92	502	1325
	MoCo-v3	1145	105	235	85	171	548	162	97	456	1592
	MAE	133	19	21	17	14	51	52	10	50	40
	DINO	1142	144	226	77	215	634	152	110	477	1555
	CLIP	968	138	237	40	231	630	128	142	449	1390

fine-grained datasets (except Flowers) the strength of the correlation drops dramatically. For SVHN the metrics are not correlated at all, since AMI was very low across all models, irrespective of S. Looking at the per-clustering method results, see Table 6b, we find that the AMI and S metrics are strongly correlated for the Agglomerative Clustering and HDBSCAN methods, while Affinity Propagation is very weakly correlated when using a ResNet-50 backbone.

Table 4: **Silhouette scores, with ResNet-50 backbone.** We report the Silhouette score, on each tested dataset (see Table 1) for each combination of SSL encoder and clustering method. The performance of Agglomerative Clustering is shown twice, either using the ground-truth number of classes as the number of clusters to generate (AC w/ C), or predicting the number of clusters (AC w/o C). The hyperparameters of the clustering methods are only tuned on ImageNet-1k (IN1k). The best combination of SSL encoder and clustering method per dataset and backbone is highlighted in **bold**, while the best SSL encoder per clustering method is underlined. We also present the Silhouette scores attained for the embeddings using the "ground-truth" classes as per the dataset annotation (G.T.).

	Encoder	IN1k	C10	C100	MNIST	fMNIST	SVHN	Flowers	Aircraft	NABirds	iNat21
G.T.	Supervised	0.59	0.46	0.38	0.58	0.57	0.20	0.45	0.30	0.32	0.24
	MoCo-v3	0.36	0.44	0.37	0.78	0.60	0.24	0.55	0.31	0.29	0.22
	VICReg	0.34	0.35	0.34	0.66	0.61	0.25	0.60	0.31	0.26	0.22
	DINO	0.33	0.32	0.29	0.48	0.55	0.21	0.57	0.29	0.26	0.21
	CLIP	0.34	0.37	0.28	0.68	0.49	0.23	0.63	0.34	0.29	0.19
K-Means	Supervised	**0.65**	**0.49**	0.39	0.69	0.61	0.32	0.47	0.30	0.32	0.24
	MoCo-v3	0.38	0.47	0.37	**0.80**	**0.62**	**0.33**	0.55	0.31	0.29	0.22
	VICReg	0.38	0.38	0.34	0.69	**0.62**	0.31	0.60	0.31	0.27	0.22
	DINO	0.35	0.39	0.31	0.54	0.57	0.30	0.57	0.31	0.27	0.22
	CLIP	0.03	0.05	0.01	0.10	0.12	0.05	0.09	0.02	0.03	0.00
AC w/ C	Supervised	**0.65**	0.47	0.37	0.68	0.59	0.27	0.45	0.29	0.30	0.23
	MoCo-v3	0.36	0.44	0.33	0.78	0.60	0.26	0.57	0.29	0.26	0.21
	VICReg	0.36	0.35	0.32	0.66	0.61	0.26	0.61	0.30	0.24	0.22
	DINO	0.33	0.36	0.28	0.55	0.55	0.21	0.60	0.28	0.25	0.20
	CLIP	0.34	0.37	0.28	0.68	0.49	0.25	0.64	0.34	0.28	0.19
AC w/o C	Supervised	0.48	0.46	**0.45**	0.48	0.55	0.20	0.44	**0.39**	**0.43**	0.31
	MoCo-v3	0.32	0.40	0.39	0.67	0.56	0.22	0.49	0.36	0.32	0.26
	VICReg	0.32	0.33	0.38	0.58	0.55	0.20	0.50	**0.39**	0.32	0.27
	DINO	0.32	0.27	0.31	0.41	0.50	0.19	0.59	0.29	0.30	0.28
	CLIP	0.41	0.30	0.33	0.43	0.40	0.20	**0.65**	0.34	0.39	**0.33**
Affinity Prop	Supervised	0.11	0.00	0.01	0.03	0.01	0.01	0.03	−0.01	0.01	–
	MoCo-v3	0.07	0.08	0.08	0.09	0.10	0.07	0.10	0.09	0.08	–
	VICReg	0.07	0.07	0.08	0.09	0.10	0.07	0.11	0.10	0.08	–
	DINO	–	−0.01	−0.01	0.00	0.01	−0.01	0.03	0.01	0.00	–
	CLIP	–	0.08	0.07	0.09	0.10	0.07	0.12	0.10	0.07	–
HDBSCAN	Supervised	0.42	−0.27	0.01	0.31	0.03	−0.48	0.14	−0.19	−0.25	−0.41
	MoCo-v3	−0.11	−0.28	−0.13	0.52	0.00	−0.52	0.40	−0.18	−0.02	−0.43
	VICReg	−0.12	−0.35	−0.15	0.47	0.04	−0.50	0.46	−0.15	−0.28	−0.43
	DINO	−0.19	−0.34	−0.22	0.32	−0.05	−0.46	0.43	−0.24	−0.11	−0.49
	CLIP	−0.19	−0.35	−0.21	0.37	−0.08	−0.45	0.47	0.04	−0.24	−0.36

Table 5: **Silhouette scores, with ViT-B backbone.** As for Table 4, except for encoders with ViT-B backbones instead of ResNet-50.

	Encoder	IN1k	C10	C100	MNIST	fMNIST	SVHN	Flowers	Aircraft	NABirds	iNat21
G.T.	Supervised	0.74	0.70	0.50	0.71	0.58	0.23	0.49	0.28	0.33	0.24
	MoCo-v3	0.49	0.60	0.41	0.70	0.59	0.27	0.60	0.31	0.28	0.20
	MAE	−0.22	0.02	0.02	0.04	0.03	0.03	0.04	0.03	0.02	−0.07
	DINO	0.58	0.58	0.44	0.67	0.62	0.36	0.69	0.30	0.35	0.23
	CLIP	0.43	0.63	0.41	0.75	0.58	0.24	0.72	0.39	0.39	0.19
K-Means	Supervised	**0.81**	**0.71**	0.51	0.71	0.58	0.30	0.51	0.29	0.33	0.24
	MoCo-v3	0.53	0.65	0.46	0.70	0.59	0.32	0.60	0.31	0.29	0.23
	MAE	0.03	0.07	0.05	0.12	0.16	0.10	0.06	0.05	0.04	0.02
	DINO	0.59	0.59	0.45	0.67	0.59	**0.36**	0.69	0.30	0.34	0.23
	CLIP	0.06	0.10	0.06	0.12	0.12	0.05	0.13	0.05	0.04	0.01
AC w/ C	Supervised	**0.81**	0.70	0.50	**0.74**	0.65	0.24	0.49	0.26	0.31	0.24
	MoCo-v3	0.49	0.60	0.41	0.72	0.61	0.26	0.60	0.27	0.25	0.20
	MAE	0.01	0.03	0.01	0.07	0.10	0.04	0.03	0.03	0.01	0.02
	DINO	0.58	0.57	0.44	0.68	**0.66**	0.31	0.71	0.30	0.36	0.23
	CLIP	0.43	0.63	0.41	**0.75**	0.58	0.26	0.74	0.39	0.37	0.19
AC w/o C	Supervised	0.60	0.69	**0.53**	0.64	0.56	0.23	0.42	0.37	0.46	0.35
	MoCo-v3	0.37	0.53	0.47	0.62	0.63	0.27	0.46	**0.45**	0.41	0.42
	MAE	−0.00	0.01	0.01	0.04	0.05	0.01	0.01	0.06	0.01	−0.01
	DINO	0.47	0.58	0.41	0.67	0.65	**0.36**	0.59	0.43	**0.50**	**0.44**
	CLIP	0.42	0.44	0.40	0.43	0.42	0.21	**0.77**	0.39	0.44	0.34
Affinity Prop	Supervised	0.06	0.11	0.09	0.10	0.10	0.07	0.10	0.09	0.11	–
	MoCo-v3	0.07	0.09	0.09	0.09	0.10	0.07	0.11	0.10	0.09	–
	MAE	0.02	0.02	0.02	0.04	0.03	0.03	0.04	0.03	0.02	–
	DINO	0.08	0.09	0.08	0.09	0.10	0.07	0.12	0.09	0.08	–
	CLIP	0.07	0.11	0.09	0.11	0.11	0.08	0.15	0.12	0.09	–
HDBSCAN	Supervised	0.67	0.35	0.17	0.37	−0.02	−0.47	0.25	−0.21	−0.16	−0.32
	MoCo-v3	0.17	0.12	0.03	0.42	0.07	−0.51	0.42	−0.17	−0.17	−0.42
	MAE	−0.24	−0.24	−0.17	−0.14	−0.14	−0.30	−0.18	−0.14	−0.22	−0.16
	DINO	0.28	0.14	0.06	0.50	0.01	−0.47	0.55	−0.18	−0.04	−0.38
	CLIP	0.16	0.21	0.02	0.56	−0.01	−0.40	0.66	0.09	−0.02	−0.41

Table 6: **AMI and Silhouette score correlations.** We compute the Pearson correlation between the AMI and S metrics for each dataset and each clustering method.

(a) **Per-dataset correlation coefficients.** A clear correlation is determined for the ImageNet, CIFAR, and MNIST style datasets. In contrast, the majority of the fine-grained datasets have a weaker correlation, except for Oxford Flowers, while SVHN is completely uncorrelated.

Backbone	IN1k	C10	C100	MNIST	fMNIST	SVHN	Flowers	Aircraft	NABirds	iNat21
ResNet-50	0.82	0.89	0.76	0.95	0.97	−0.10	0.75	0.28	0.40	0.40
ViT-B	0.91	0.87	0.82	0.95	0.91	−0.09	0.84	0.33	0.57	0.51

(b) **Per-clusterer correlation coefficients.** A strong correlation is determined for Agglomerative Clustering and HDBSCAN. Affinity Propagation exhibits the weakest correlation with a ResNet-50 backbone, but achieves a much stronger correlation with a ViT-B backbone.

Backbone	K-Means	Affinity Prop	AC w/ C	AC w/o C	HDBSCAN
ResNet-50	0.61	0.15	0.91	0.86	0.94
ViT-B	0.60	0.55	0.82	0.66	0.94