Coreset Sampling from Open-Set for Fine-Grained Self-Supervised Learning

Sungnyun Kim* **Sangmin Bae*** **Se-Young Yun**
Graduate School of Artificial Intelligence
Korea Advanced Institute of Science and Technology
{ksn4397, bsmn0223, yunseyoung}@kaist.ac.kr

Abstract

Despite the increased interest in applying deep learning to specific domains, developing algorithms for fine-grained datasets suffers from two challenges: expert knowledge for annotation and the necessity of a versatile model for subordinate tasks in a specific domain. We can leverage the recent self-supervised learning approach to pretrain a model with the fine-grained dataset, serving as an effective initialization for any downstream tasks. Here, we introduce a novel Open-set Self-Supervised Learning problem with the assumption that a large-scale unlabeled open-set is available during a pretraining phase. In this problem setup, it is crucial to consider the distribution mismatch between pretraining and target datasets. Hence, we propose a SimCore algorithm to sample a coreset, the subset of open-set that has a minimum distance to the target dataset in a latent space. We demonstrate that SimCore significantly improves representation learning through extensive experimental settings with eight fine-grained datasets and two open-sets.

1 Introduction

While the application of deep learning to specific domains is drawing attention [19, 54, 34], the visual recognition for fine-grained datasets poses two challenges for researchers to develop algorithms. First, it requires a number of experts for annotation, which incurs a large amount of cost [3, 12, 46]. For example, ordinary people do not have professional knowledge about aircraft types or fine-grained categories of birds. Therefore, a realistic presumption for a domain-specific fine-grained dataset is that one might have no or very few labeled samples. Second, fine-grained datasets are often re-purposed or used for various tasks according to the user's demand, which motivates us to make a versatile model. One might ask, as a target task, that aircraft images be classified by model variants or even segmented into foreground and background. A good initialization model can handle a variety of annotations for fine-grained datasets, such as multiple attributes [55, 35, 32], pixel-level annotations [55, 40], or bounding boxes [40, 35, 30].

Hence, we can leverage the recent self-supervised learning (SSL) [12, 24, 21, 10] to pretrain a model with the fine-grained dataset so that it can serve as an effective initialization for any future downstream tasks. Current SSL literature emphasizes the benefits of pretraining on large-scale benchmarks and then fine-tuning on the fine-grained dataset [12, 21], rather than SSL pretraining on the fine-grained dataset itself. For convenience, we denote the large-scale unlabeled datasets, which can be easily obtained by web crawling, as an *open-set*. However, there has been a lack of discussion about the distribution mismatch between pretraining and target datasets. Does pretraining a model with an open-set mostly of animal images improve the target task of classifying aircraft model types? We hypothesize that (1) a large distribution mismatch inhibits representation learning for the target task,

[*] Equal contribution.

NeurIPS 2022 Workshop on Self-Supervised Learning – Theory and Practice.

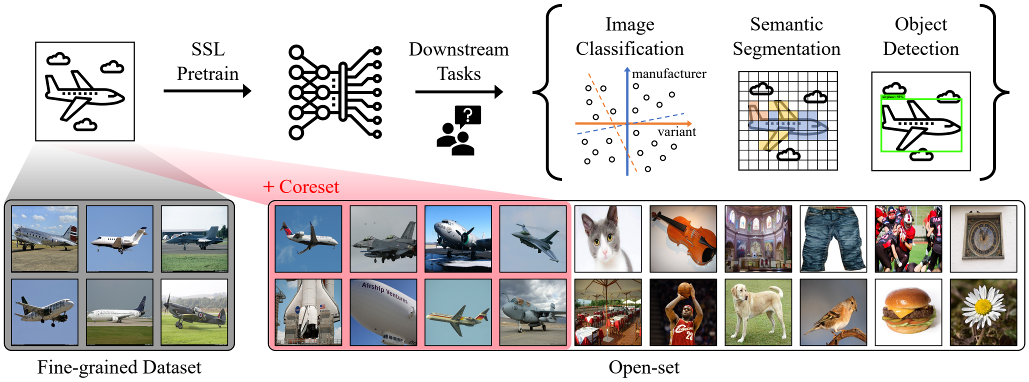
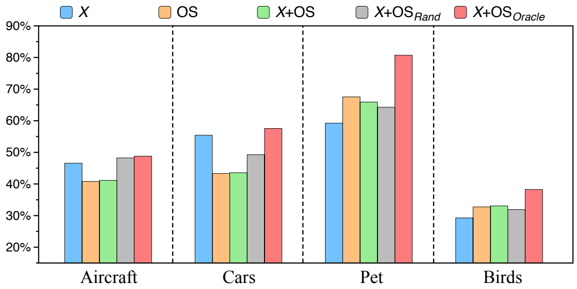


Figure 1: Overview of an OpenSSL problem. For any downstream tasks, we pretrain an effective model for the fine-grained dataset via self-supervised learning. Here, the assumption for a large-scale unlabeled open-set in a pretraining phase is well-suited for a real-world scenario. The main goal is to find a coreset, highlighted by the red box, among the open-set to enhance SSL.



X = ?	Selected Classes for OS$_{Oracle}$
Aircraft [35]	airliner, warplane, airship, ... (5 more)
Cars [30]	convertible, limousine, jeep, ... (7 more)
Pet [40]	Persian cat, Yorkshire terrier, ... (22 more)
Birds [55]	goldfinch, junco, robin, jay, ... (16 more)

Figure 2: **Left**: Linear evaluation performance on four fine-grained datasets. Each label corresponds to the pretraining dataset, while **+** means merging two datasets. **Right**: Manually selected categories from an open-set (OS), ImageNet [18] in this case, according to each target dataset (X). Selected categories and exact numbers are detailed in Appendix A. We followed the typical linear evaluation protocol [12, 21] and used the SimCLR method [12] on ResNet50 [23].

and (2) exploiting a *coreset*, a subset of open-set that shares similar semantics with the target dataset, is beneficial.

Motivating experiment. To this end, we introduce a novel Open-set Self-Supervised Learning (OpenSSL) problem, where we can utilize open-set as well as the fine-grained dataset in a pretraining phase. In this problem setup (see Figure 1), our main goal is to find a coreset that enhances the pretraining and therefore improves the target task performance on the fine-grained dataset. Figure 2 describes the motivating experiments, verifying our two hypotheses. First, SSL on the open-set does not always outperform SSL on the fine-grained dataset since it depends on the semantic similarity between X and OS. This is in line with the observation in [20, 19] that the performance of self-supervised representation on downstream tasks is correlated with how similar the target dataset is to the pretrained dataset. Next, to prove our insights on the coreset, we manually selected the relevant classes from ImageNet (OS$_{Oracle}$) that are supposed to be helpful according to each target dataset (X). Interestingly, merging OS$_{Oracle}$ to X showed a significant performance gain, and its comparison to merging the entire open-set (X+OS) or the randomly sampled subset (X+OS$_{Rand}$) implies the necessity of a sampling algorithm for the coreset in the OpenSSL problem.

In this work, we propose **SimCore**, a simple and effective coreset sampling algorithm from the unlabeled open-set. We formulate the data subset selection problem to obtain a coreset that has a minimum distance to the target dataset in a latent space. Since there is no label information, we exploit the feature-space distance to measure semantic similarity. Our method is reminiscent of the facility location problem [36, 53], but our goal is to find a semantically relevant subset to the target dataset, not to choose a representative for the open-set. SimCore significantly improves the performance in extensive experimental settings (eight fine-grained datasets and two open-sets) and shows consistent gains with different self-supervised losses, architectures, and target tasks.

2 SimCore: Simple Coreset Sampling from Open-set

We introduce a simple coreset sampling algorithm, coined as SimCore. Motivated by Figure 2, selecting an appropriate coreset is the most crucial factor for the OpenSSL problem. To this end, we build a set with the open-set samples that are the nearest neighbors of the target samples.

This is formulated by finding a subset S that maximizes the following objective function. If we denote X and V as the target dataset and the open-set, respectively,

$$f(S) = \sum_{x \in X} \max_{v \in S} w(x, v), \quad \text{where} \;\; S \subseteq V, \; V \cap X = \emptyset, \; |V| \gg |X|, \tag{1}$$

while $w(x, v) = \max_{i \in X, j \in V} \|\hat{u}_i - \hat{u}_j\|_2 - \|\hat{u}_x - \hat{u}_v\|_2$ estimates similarity of two samples from each set, and $\hat{u}$ is the normalized feature from an encoder E_θ pretrained on X with small epochs. From this, SimCore can find a subset that shares the most similar semantics with the target set. It is also reminiscent of the facility location problem [36, 53], $\arg\max f_{\text{fac}}(S) = \sum_{x \in V} \max_{v \in S} w(x, v)$, $S \subseteq V$. However, $f(S)$ is different from $f_{\text{fac}}(S)$, since in Eq. 1, S does not include any target sample x.

Iterative coreset sampling. $f(S)$ is a monotonically increasing submodular function. One remark is that if there is no constrained budget on S, $f(S)$ continues to increase until it converges to the maximum value. If we denote S^* as a minimal set that achieves maximum f value, S^* is obtained when including only the instances closest to each instance of X, i.e., $|S^*| \leq |X|$. Since we want to sample sufficiently large subset, we re-define Eq. 1 with the selection round t: $f_t(S)$ where the candidate set is V_t. Thus, we iterate the rounds to repeat sampling S_t^* and excluding them from the candidate set V_t, until we reach the proper budget size (see Algorithm 1 of Appendix B).

Reducing the complexity. Meanwhile, the direct calculation of pairwise distances requires the complexity of $\mathcal{O}(|X||V|)$, which might be extremely large. To reduce the computational overhead and make the algorithm scalable, we adapt k-means clustering [2] to the target dataset X. Exploiting only the centroids ($k = 100$ in practice), we show significant performance gains on various benchmarks.

3 Related Works

Self-supervised learning. After Oord *et al.* [39] proposed an InfoNCE loss, contrastive learning algorithms began to show a remarkable improvement in representation learning with a Siamese network [12, 24, 21, 9, 13] or a Vision Transformer [10, 4, 31, 14]. While a large-scale open-set enhances the generalization of learned representation [26, 48, 17], recent literature has pointed out the distribution mismatch between a pretraining dataset and a fine-tuning target dataset [47, 19]. Particularly, El *et al.* [19] have claimed that pretraining on ImageNet may not always be effective on the target task from different domains and proposed a denoising autoencoder for the robust SSL only with the fine-grained dataset. Their motivation is similar to ours, but our goal is sampling an effective coreset to profit from the open-set by augmenting the small-scale fine-grained dataset.

Coreset selection from open-set. In an OpenSSL problem, we denote an open-set as the additional unlabeled pretraining set that includes instances either from similar or unsimilar domain to target dataset. The assumption of available open-set is also common in other research fields, such as open-set recognition [44, 6, 11, 51], open-set [38, 59, 15, 43] or open-world [5, 8, 7] semi-supervised learning, and open-set annotation [37]. The main goal of our OpenSSL problem is sampling open-set instances relevant to the target dataset, and it is related to recent coreset selection approaches. Existing studies find the representative subset of the unlabeled set for active selection [53, 45], or find the subset of current task data to avoid catastrophic forgetting for continual learning [1, 58, 49]. However, our work is clearly different from them because we sample a coreset from the unlabeled open-set, and the selected coreset is for augmenting the fine-grained dataset. More details of the comparison to recent literature are summarized in Table 5 of Appendix C.

4 Experiments

Linear evaluation. We evaluated the learned representation quality according to each pretraining dataset, and the detailed experimental setups are in Appendix D. In addition to the four datasets we used in Section 1, we also included Dogs [27], Action [57], Indoor [41], and Textures [16] datasets. Table 1 summarizes the linear evaluation results, where using the coreset sampled by SimCore is

Table 1: Linear evaluation performance on eight fine-grained datasets. We used ImageNet [18] as an open-set. We sampled p-ratio of the open-set, either via random sampling (OS$_{Rand}$) or SimCore (OS$_{SimCore}$), and merged them to the target dataset (X). † denotes SimCore with $k = 1$, single centroid. **Bold** indicates the best accuracy for each target, and for each sampling ratio, the best accuracy which outperformed both X and OS pretraining is also underlined.

	X = ? (# of samples)							
pretraining	Aircraft 6,667	Cars 8,144	Pet 3,680	Birds 5,990	Dogs 12,000	Action 4,000	Indoor 5,360	Textures 3,760
X	46.56%	55.42%	59.23%	29.27%	49.88%	43.76%	54.10%	58.78%
OS (1.3M)	41.50%	41.86%	67.66%	33.21%	49.94%	60.65%	64.46%	67.23%
$p = 0.01$ or 1% of |OS| (12.8K)								
X+OS$_{Rand}$	48.24%	49.26%	64.27%	31.90%	49.62%	47.25%	55.37%	61.33%
X+OS$_{SimCore†}$	48.06%	58.56%	74.82%	33.37%	57.42%	51.37%	57.84%	61.76%
X+OS$_{SimCore}$	**48.45%**	**59.00%**	77.13%	36.56%	59.83%	52.98%	59.18%	63.40%
$p = 0.05$ or 5% of |OS| (64.1K)								
X+OS$_{rand}$	45.75%	46.03%	68.38%	33.63%	50.24%	57.27%	60.71%	65.80%
X+OS$_{SimCore†}$	45.57%	50.75%	80.20%	35.56%	64.62%	64.53%	68.13%	66.22%
X+OS$_{SimCore}$	47.14%	52.22%	**81.75%**	**39.21%**	**66.82%**	**66.38%**	**70.96%**	**68.13%**

Table 2: **Left**: Linear evaluation performance with different architecture and different SSL method. We followed the best sampling ratio in Table 1 for each fine-grained dataset. **Right**: iNaturalist 2021-mini [50] as an open-set, with sampling ratio of $p = 0.05$ (25K). We used Pets and Birds for natural image datasets and Action and Indoor for unnatural image datasets.

method	architecture	pretraining	Aircarft	Cars	Pet	Birds
SimCLR	ResNet18	X	43.44%	51.85%	58.19%	25.94%
SimCLR	ResNet18	OS	33.92%	33.07%	62.54%	27.72%
SimCLR	ResNet18	X+**OS$_{SimCore}$**	**45.81%**	**53.70%**	**76.62%**	**32.83%**
BYOL	ResNet50	X	40.56%	49.42%	56.47%	27.55%
BYOL	ResNet50	OS	46.06%	49.60%	78.37%	44.72%
BYOL	ResNet50	X+**OS$_{SimCore}$**	**46.54%**	**51.70%**	**81.68%**	**44.96%**

pretraining	Pet	Birds	Action	Indoor
X	59.23%	29.27%	43.76%	54.10%
OS (0.5M)	62.60%	34.09%	**49.33%**	54.05%
X+**OS$_{SimCore}$**	**65.90%**	**37.36%**	48.81%	**57.21%**

effective in most cases. We sampled p-ratio of the open-set to compare the performance of pretraining datasets within the limited budget size. The different trend across target datasets gives us a hint about the optimal coreset size based on the level of distribution mismatch to the open-set. We have also confirmed that using a number of centroids is more advantageous than a single centroid, although SimCore with $k = 1$ even outperforms the random sampling. We visualized the coreset examples in Appendix E. We leave the adaptive sampling strategy and the sensitivity study for hyperparameter k as future works. The results of the nearest neighbor classifier [56, 9] and semi-supervised learning [12, 21] are summarized in Appendix F.

ResNet18 and BYOL. In Table 2 (left), we have applied our method with different architecture, ResNet18 [23], and different SSL method, BYOL [21]. SimCore consistently proves the effect of merging the coreset samples.

iNaturalist open-set. Thus far, we have used ImageNet benchmark [18] as the open-set. In practice, however, an open-set is not what we know about; it is rather a bunch of data randomly drawn from the web or database. Thus, we experimented with other open-set, iNaturalist 2021-mini [50] that contains 0.5M samples in total. Table 2 (right) shows that in unnatural fine-grained datasets, iNaturalist was not as effective as ImageNet. Still, SimCore outperformed the pretraining on X in every dataset.

5 Conclusion

In this study, we introduce an OpenSSL problem, where a large-scale open-set can be utilized to help SSL on the fine-grained dataset. Besides, we propose a SimCore algorithm, sampling the effective coreset from the open-set, and demonstrate that these coreset samples help better representation learning. We believe that our work will encourage future research on SSL with the open-set. Future work will address the sensitivity study on the number of k, the stopping criterion that replaces the choice of sampling portion, and additional downstream tasks like semantic segmentation.

Acknowledgments and Disclosure of Funding

This work was supported by Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No.2019-0-00075, Artificial Intelligence Graduate School Program(KAIST), 10%) and Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. 2022-0-00871, Development of AI Autonomy and Knowledge Enhancement for AI Agent Collaboration, 90%)

References

[1] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio. Gradient based sample selection for online continual learning. *Advances in neural information processing systems*, 32, 2019.

[2] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006.

[3] J. T. Ash, C. Zhang, A. Krishnamurthy, J. Langford, and A. Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671*, 2019.

[4] S. Atito, M. Awais, and J. Kittler. Sit: Self-supervised vision transformer. *arXiv preprint arXiv:2104.03602*, 2021.

[5] A. Bendale and T. Boult. Towards open world recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1893–1902, 2015.

[6] A. Bendale and T. E. Boult. Towards open set deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1563–1572, 2016.

[7] T. E. Boult, S. Cruz, A. R. Dhamija, M. Gunther, J. Henrydoss, and W. J. Scheirer. Learning and the unknown: Surveying steps toward open world recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 9801–9807, 2019.

[8] K. Cao, M. Brbic, and J. Leskovec. Open-world semi-supervised learning. *arXiv preprint arXiv:2102.03526*, 2021.

[9] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020.

[10] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

[11] G. Chen, P. Peng, X. Wang, and Y. Tian. Adversarial reciprocal points learning for open set recognition. *arXiv preprint arXiv:2103.00953*, 2021.

[12] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[13] X. Chen and K. He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

[14] X. Chen, S. Xie, and K. He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649, 2021.

[15] Y. Chen, X. Zhu, W. Li, and S. Gong. Semi-supervised learning under class distribution mismatch. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3569–3576, 2020.

[16] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014.

[17] E. Cole, X. Yang, K. Wilber, O. Mac Aodha, and S. Belongie. When does contrastive visual representation learning work? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14755–14764, 2022.

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[19] A. El-Nouby, G. Izacard, H. Touvron, I. Laptev, H. Jegou, and E. Grave. Are large-scale datasets necessary for self-supervised pre-training? *arXiv preprint arXiv:2112.10740*, 2021.

[20] L. Ericsson, H. Gouk, and T. M. Hospedales. How well do self-supervised models transfer? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5414–5423, 2021.

[21] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33: 21271–21284, 2020.

[22] K. Han, S.-A. Rebuffi, S. Ehrhardt, A. Vedaldi, and A. Zisserman. Automatically discovering and learning new visual categories with ranking statistics. *arXiv preprint arXiv:2002.05714*, 2020.

[23] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[24] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[25] Y.-C. Hsu, Z. Lv, and Z. Kira. Learning to cluster in order to transfer across domains and tasks. *arXiv preprint arXiv:1711.10125*, 2017.

[26] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.

[27] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR workshop on fine-grained visual categorization (FGVC)*, volume 2. Citeseer, 2011.

[28] K. Killamsetty, X. Zhao, F. Chen, and R. Iyer. Retrieve: Coreset selection for efficient and robust semi-supervised learning. *Advances in Neural Information Processing Systems*, 34: 14488–14501, 2021.

[29] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[30] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.

[31] C. Li, J. Yang, P. Zhang, M. Gao, B. Xiao, X. Dai, L. Yuan, and J. Gao. Efficient self-supervised vision transformers for representation learning. *arXiv preprint arXiv:2106.09785*, 2021.

[32] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[33] I. Loshchilov and F. Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[34] W. Luo, X. Yang, X. Mo, Y. Lu, L. S. Davis, J. Li, J. Yang, and S.-N. Lim. Cross-x learning for fine-grained visual categorization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8242–8251, 2019.

[35] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[36] P. B. Mirchandani and R. L. Francis. *Discrete location theory*. 1990.

[37] K.-P. Ning, X. Zhao, Y. Li, and S.-J. Huang. Active learning for open-set annotation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 41–49, 2022.

[38] A. Oliver, A. Odena, C. A. Raffel, E. D. Cubuk, and I. Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. *Advances in neural information processing systems*, 31, 2018.

[39] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[40] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012.

[41] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *2009 IEEE conference on computer vision and pattern recognition*, pages 413–420. IEEE, 2009.

[42] J. Robinson, C.-Y. Chuang, S. Sra, and S. Jegelka. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592*, 2020.

[43] K. Saito, D. Kim, and K. Saenko. Openmatch: Open-set consistency regularization for semi-supervised learning with outliers. *arXiv preprint arXiv:2105.14148*, 2021.

[44] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult. Toward open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1757–1772, 2012.

[45] O. Sener and S. Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

[46] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020.

[47] J.-C. Su, Z. Cheng, and S. Maji. A realistic evaluation of semi-supervised learning for fine-grained classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12966–12975, 2021.

[48] Y. Tian, O. J. Henaff, and A. van den Oord. Divide and contrast: Self-supervised learning from uncurated data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10063–10074, 2021.

[49] R. Tiwari, K. Killamsetty, R. Iyer, and P. Shenoy. Gcr: Gradient coreset based replay buffer selection for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 99–108, 2022.

[50] G. Van Horn, O. Mac Aodha, Y. Song, Y. Cui, C. Sun, A. Shepard, H. Adam, P. Perona, and S. Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018.

[51] S. Vaze, K. Han, A. Vedaldi, and A. Zisserman. Open-set recognition: A good closed-set classifier is all you need. *arXiv preprint arXiv:2110.06207*, 2021.

[52] F. Wang and H. Liu. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2495–2504, 2021.

[53] K. Wei, R. Iyer, and J. Bilmes. Submodularity in data subset selection and active learning. In *International conference on machine learning*, pages 1954–1963. PMLR, 2015.

[54] X.-S. Wei, Y.-Z. Song, O. Mac Aodha, J. Wu, Y. Peng, J. Tang, J. Yang, and S. Belongie. Fine-grained image analysis with deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[55] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-ucsd birds 200. 2010.

[56] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018.

[57] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei. Human action recognition by learning bases of action attributes and parts. In *2011 International conference on computer vision*, pages 1331–1338. IEEE, 2011.

[58] J. Yoon, D. Madaan, E. Yang, and S. J. Hwang. Online coreset selection for rehearsal-based continual learning. *arXiv preprint arXiv:2106.01085*, 2021.

[59] Q. Yu, D. Ikami, G. Irie, and K. Aizawa. Multi-task curriculum framework for open-set semi-supervised learning. In *European Conference on Computer Vision*, pages 438–454. Springer, 2020.

[60] Z. Zhong, L. Zhu, Z. Luo, S. Li, Y. Yang, and N. Sebe. Openmix: Reviving known knowledge for discovering novel visual categories in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9462–9470, 2021.

A Details of Motivating Experiment in Section 1

In Table 3, we described all the selected classes from ImageNet to construct OS_{Oracle} for each target dataset. For clarity, we also visualized examples for each selected category in Figures 4–6. Also, Table 4 summarizes the exact numbers of the motivating experiment results.

Table 3: The class list of OS_{Oracle} according to each target dataset (X).

X = ?	Selected Classes for OS_{Oracle}
Aircraft	airliner, airship, American egret, crane, space shuttle, spoonbill, warplane, white stork
Cars	ambulance, beach wagon, cab, convertible, jeep, limousine, minivan, Model T, racer, sports car
Pet	basset, beagle, boxer, cocker spaniel, Chihuahua, Egyptian cat, English setter, German short-haired pointer, Great Pyrenees, keeshond, Leonberg, miniature pinscher, Newfoundland, Persian cat, Pomeranian, pug, Saint Bernard, Samoyed, Scotch terrier, Siamese cat, soft-coated wheaten terrier, Staffordshire bullterrier, tiger cat, Yorkshire terrier
Birds	albatross, bee eater, brambling, bulbul, chickadee, goldfinch, house finch, hummingbird, indigo bunting, jacamar, jay, junco, kite, magpie, pelican, quail, red-backed sandpiper, red-breasted merganser, redshank, robin

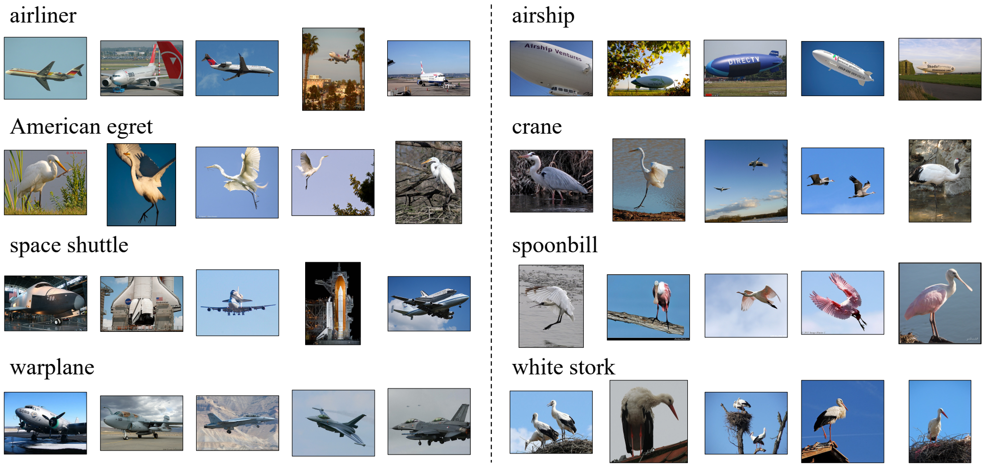


Figure 3: Visualization of examples whose classes belong to OS_{Oracle} (X = FGVC-Aircraft).

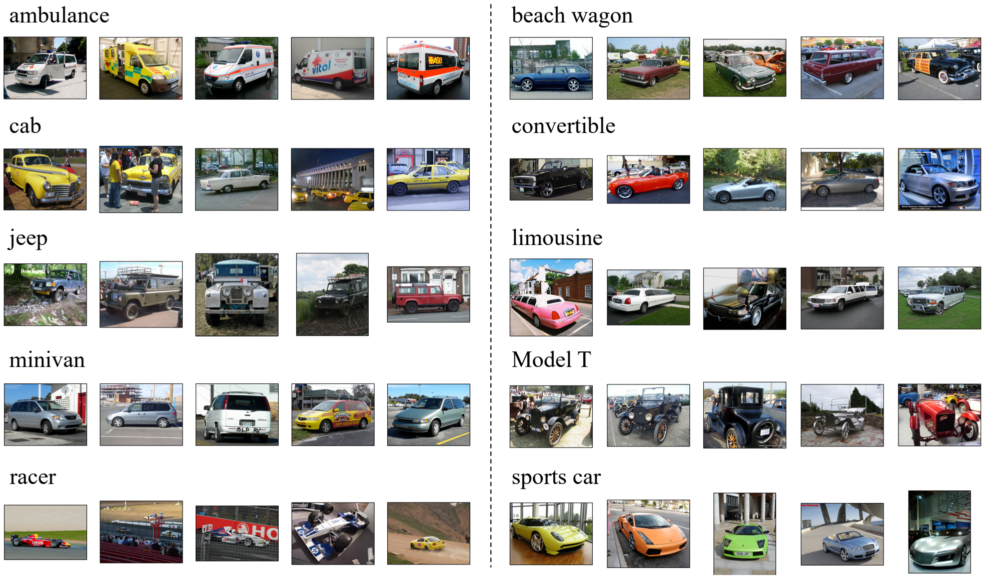


Figure 4: Visualization of examples whose classes belong to OS_{Oracle} (X = Stanford Cars).

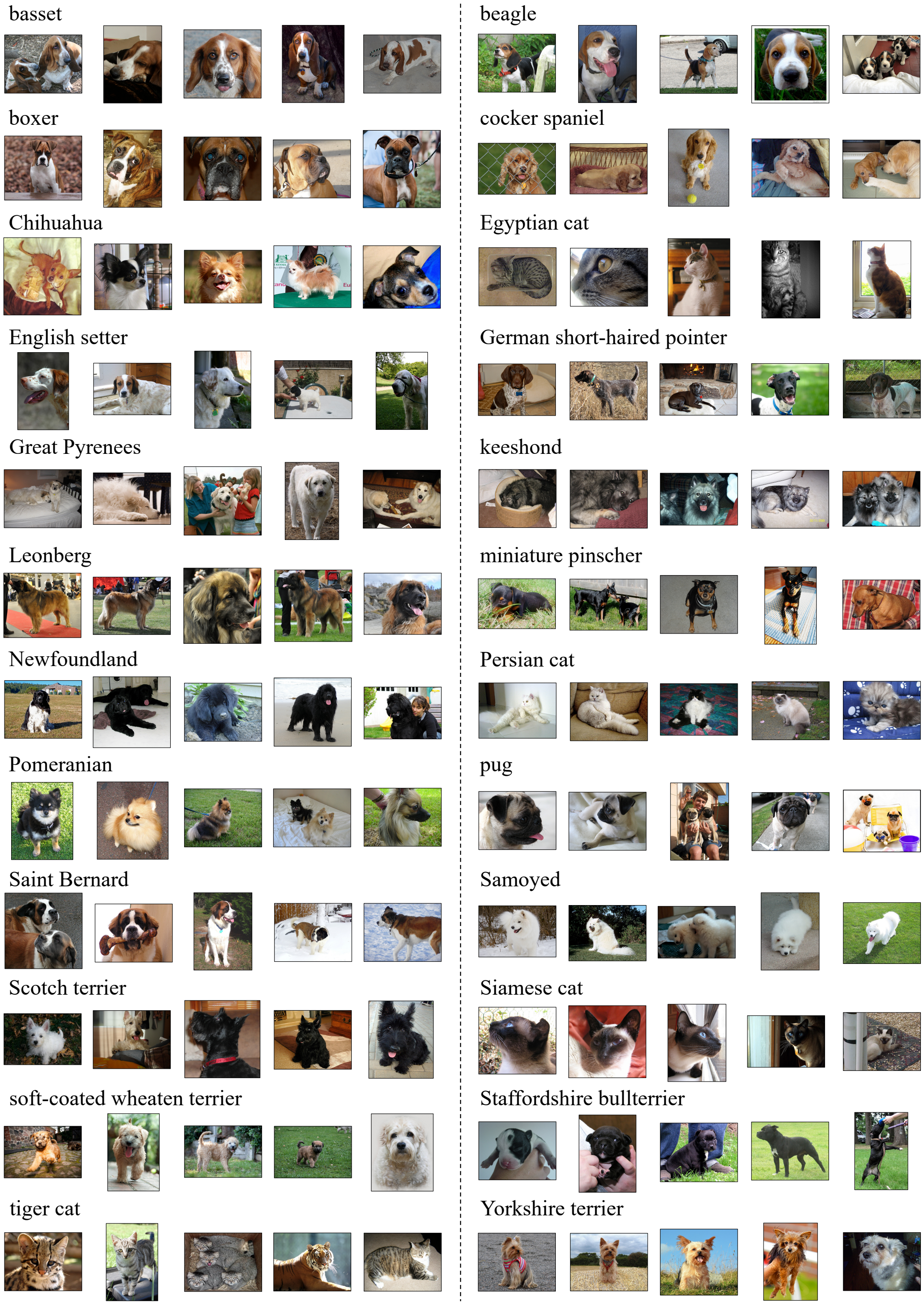


Figure 5: Visualization of examples whose classes belong to OS_{Oracle} (X = Oxford-IIIT Pet).

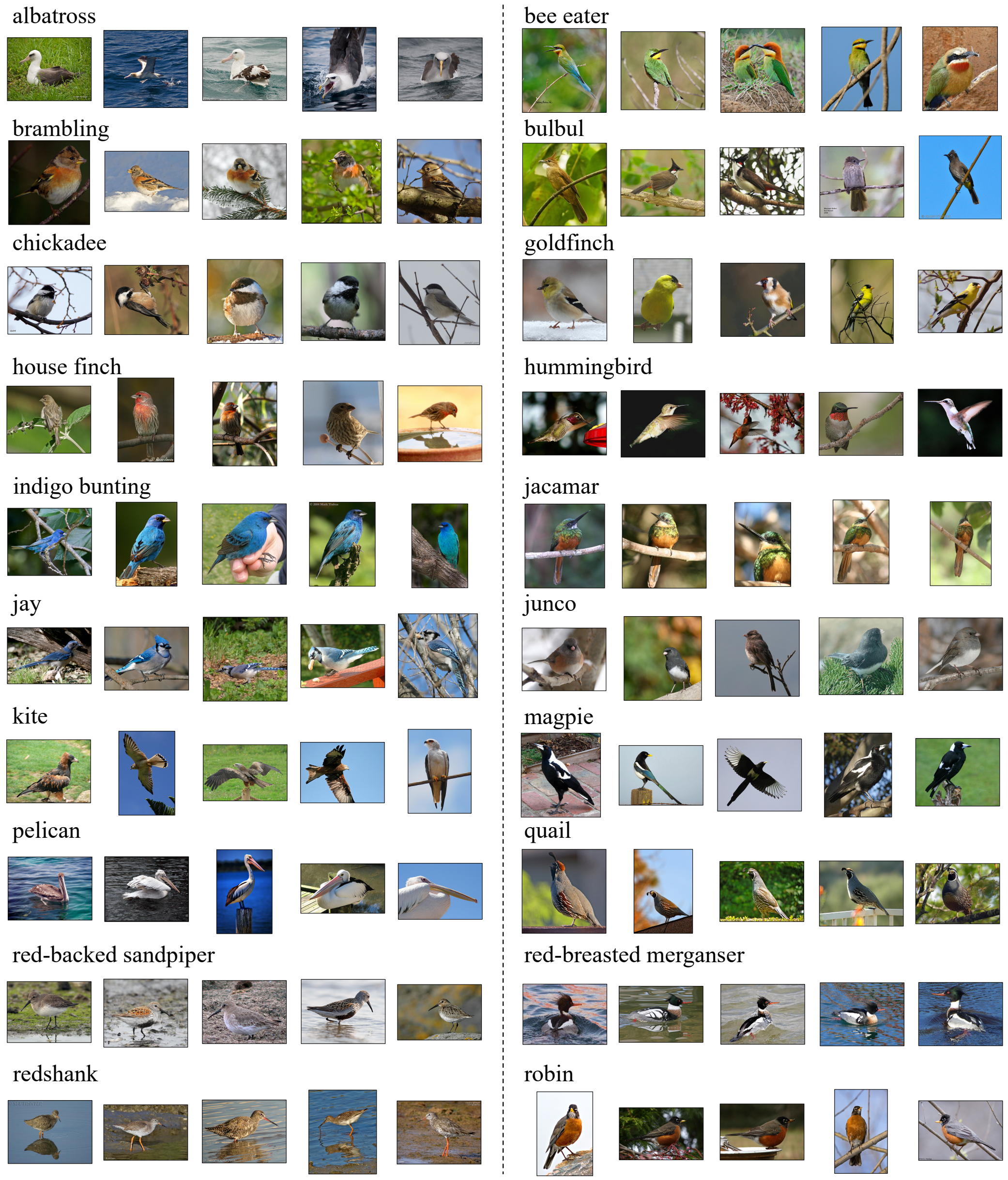


Figure 6: Visualization of examples whose classes belong to OS_{Oracle} (X = Caltech-UCSD Birds).

Table 4: Linear evaluation performance with sample size for each pretraining dataset. ImageNet is used as the open-set, and the sampling portion is 1% of $|\mathrm{OS}|$.

	X = ?			
pretraining	FGVC-Aircraft	Stanford Cars	Oxford-IIIT Pet	Caltech-UCSD Birds
X	46.56% (6.7K)	55.42% (8.1K)	59.23% (3.7K)	29.27% (6.0K)
OS	41.50% (1.3M)	41.86% (1.3M)	67.66% (1.3M)	33.21% (1.3M)
X+OS	39.88% (1.3M)	42.92% (1.3M)	68.22% (1.3M)	32.88% (1.3M)
$X+\mathrm{OS}_{Rand}$	48.24% (19.5K)	49.26% (20.9K)	64.27% (16.5K)	31.90% (18.8K)
$\boldsymbol{X+\mathrm{OS}_{Oracle}}$	**48.78%** (17.1K)	**57.56%** (21.1K)	**80.73%** (34.9K)	**38.26%** (32.0K)

B Algorithm Description

In Algorithm 1, we described the full algorithm of SimCore sampling procedure.

Algorithm 1: Simple Coreset Sampling from Open-set (SimCore)

```
Require: $X$, $E_\theta$: target dataset, encoder pretrained on $X$;
Require: $V_0$: initial candidate set (open-set);
Require: $\mathcal{B}$: coreset budget;
initialize $\mathcal{I} \leftarrow \emptyset$, $t \leftarrow 0$;
calculate $u_x \leftarrow E_\theta(x)$ for $\forall x \in X$;
calculate $u_v \leftarrow E_\theta(v)$ for $\forall v \in V_0$;
while $|\mathcal{I}| < \mathcal{B}$ do
    set $S_t^*$ as the elements in $V_t$ that are closest to each element in $X$;
    $\mathcal{I} \leftarrow \mathcal{I} \cup S_t^*$;
    $V_{t+1} \leftarrow V_t \setminus S_t^*$;
    $t \leftarrow t + 1$;
end
re-initialize $\theta$;
start pretraining $E_\theta$ with $X \cup \mathcal{I}$;
```

C Comparisons with Relevant Literature

To clearly compare the OpenSSL problem with existing research fields, we summarized the comparison table as follows.

Table 5: Comparisons of the OpenSSL problem with relevant literature. Especially, we focus on each problem setting and the definition of open-set (OS) or coreset (CS). * indicates the instances in unlabeled data pool supposed to be annotated after the active selection.

Task	Problem Setting	Train (Labeled)	Train (Unlabeled)	Test	Definition of OS / CS	Main Goal
Novel Class Discovery [25, 22, 60]	unlabeled data consist of only novel classes	seen	-	novel	-	cluster novel classes in unlabeled dataset
Open-set Recognition [44, 6, 11, 51]	test set contains seen and novel classes	seen	-	seen + novel	[OS] test dataset containing seen and novel classes	reject instances from novel classes in test time
Open-set Semi-Supervised Learning [43, 38, 59, 15, 28]	unlabeled train data contain novel classes	seen	seen + novel	seen	[OS] training dataset containing seen and novel classes	train a robust model while regularizing novel classes
Open-World Semi-Supervised Learning [8, 7, 5]	test set contains seen and novel classes	seen	seen + novel	seen + novel	[OS] dataset containing seen and novel classes	discover novel classes and assign samples in test time
Open-set Annotation [37]	unlabeled data pool contains novel classes	seen	seen* + novel*	seen	[OS] unlabeled data pool with seen and novel classes	aims to query seen classes from unlabeled data pool
Coreset Selection in Active Learning [45, 53]	query instances to be annotated given fixed labeling budget	seen	seen*	seen	[CS] the most representative subset of unlabeled set	find a small subset competitive to whole dataset
Coreset Selection in Continual Learning [1, 58, 49]	continuously learn a sequence of tasks	partially novel	-	seen	[CS] the most representative instances at each task	promote task adaptation with less catastrophic forgetting
Hard Negative Mining for SSL [42, 52]	assumes that hard negatives are useful	-	target	target	[CS] the hardest contrastive pair instances for SSL	improve SSL performance using core-negative instances
OpenSSL (ours)	open-set may have data irrelevant to target dataset	-	target + irrelevant	target	[OS] large-scale unlabeled set [CS] subset of OS sharing the same semantic with target set	improve SSL performance on fine-grained datset via coreset sampling method

D Experimental Settings

We used eight fine-grained datasets and two open-sets in the main experiments. We summarized the dataset configurations in the following Table 6. Also, we followed the linear evaluation protocol used in recent SSL literature [12, 21]: pretraining the encoder and fine-tuning only the classifier with the frozen encoder part.

Pretraining the encoder. We set the maximum cost of pretraining to 8 GPU days since the size of train sets are different depending on the pretraining dataset. For example, we trained 200 epochs for ImageNet open-set (OS) and X+OS experiments. Meanwhile, on small-scale fine-grained datasets, we pretrain the model for much longer epochs of 5K. In the motivating experiment, we also used 5K epochs pretraining by default, but for the experiments that exceeded 8 GPU days (X+OS$_{Oracle}$ for Pet and Birds), we reduced the epochs to 3K. For X+OS$_{SimCore}$ pretraining, every 1% sampling used 5K epochs, but every 5% sampling used 2K epochs. Before sampling the coreset, we should have a model pretrained on the target dataset for small epochs. For this, we trained for 1K epochs on each dataset.

When pretraining the encoder with SimCLR method, we used an SGD optimizer with initial learning rate of 0.1 and ℓ_2 regularization parameter 1e-4, and with 512 batch size. The learning rate is scheduled by cosine annealing [33], where the minimum learning rate is zero on the last epoch. We followed the same hyperparameter of 0.07 temperature and 128 projection dimension as in the original paper [12]. For BYOL experiments in Table 2, we used an Adam optimizer [29] with a learning rate of 1e-3 and ℓ_2 regularization parameter 1e-6. BYOL method is highly sensitive to the exponential moving average (EMA) value of a momentum encoder [24]. Therefore, we followed the same EMA scheduler as in the original implementation [21], starting from 0.996 to 1. We should also note that we did not utilize the EMA scheduler for some experiments with lower epochs.

Fine-tuning for linear evaluation. We fine-tuned a linear classifier for 100 epochs and searched the optimal learning rate among five logarithmically spaced values from 1 to 10^2. We decayed the learning rate by 0.1 at 60 and 80 epochs, and any regularization techniques, such as weight decay, were not used.

Table 6: Datasets we used and their configurations. For the fine-grained dataset that has a separate validation set, we incorporated the validation set to the train set because we only need unlabeled data for self-supervised pretraining.

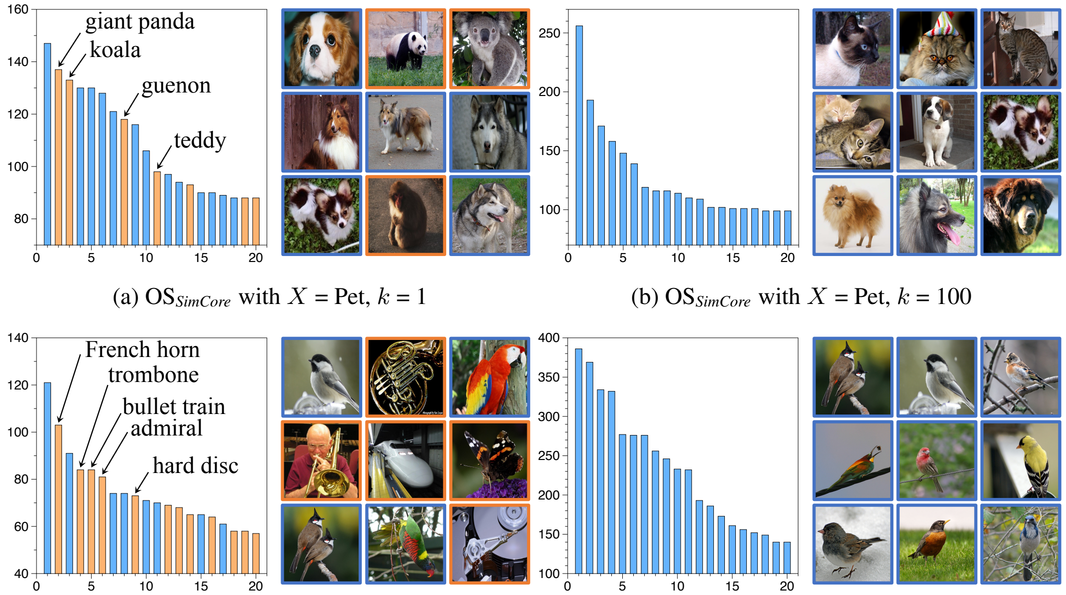
Dataset	Train #	Test #	Class #
ImageNet [18]	1,281,167	50,000	1,000
iNaturalist 2021-mini [50]	500,000	100,000	10,000
FGVC-Aircraft [35]	6,667	3,333	100
Stanford Cars [30]	8,144	8,041	196
Oxford-IIIT Pet [40]	3,680	3,669	37
Caltech-UCSD Birds [55]	5,990	5,790	200
Stanford Dogs [27]	12,000	8,580	120
Stanford 40 Actions [57]	4,000	5,532	40
MIT-67 Indoor Scene Recognition [41]	5,360	1,340	67
Describable Textures (DTD) [16]	3,760	1,880	47

E Coreset Visualization

We visualized which instances from the open-set are actually sampled by our SimCore algorithm. To this end, in Figure 7, we displayed the ground-truth labels of the coreset samples when the target dataset is Pet [40] or Birds [55]. For comparison, we also displayed the coreset by SimCore with $k = 1$, using a single centroid. Note that the Pet dataset contains 12 cat breeds and 25 dog breeds, and the Birds dataset contains 200 bird species. The open-set dataset is ImageNet [18], so the ground-truth labels of coreset samples correspond to the ImageNet classes.

For the Pet dataset, SimCore with $k = 1$ has sampled mostly animal images but included data somewhat irrelevant to cats and dogs (Figure 7a). The second most class is giant panda, the third is koala, and the eighth is guenon, a kind of monkey. On the contrary, SimCore with $k = 100$ has sampled mostly cat or dog images; up to top-20 classes, every class was from the breed of either cat or dog (Figure 7b). Interestingly, eight out of the top-10 classes were those that overlap with the Pet class labels, such as Siamese cat, Persian cat, Saint Bernard, etc.

For the Birds dataset, SimCore with $k = 1$ has sampled a lot of irrelevant images, such as French horn, trombone, bullet train, admiral, hard disc, etc. (Figure 7c). On the contrary, SimCore with $k = 100$ has sampled only the bird species up to the top-20 classes, including bulbul, chickadee, brambling, bee eater, house finch, goldfinch, junco, robin, jay, etc. (Figure 7d).



(a) $\text{OS}_{SimCore}$ with X = Pet, k = 1

(b) $\text{OS}_{SimCore}$ with X = Pet, k = 100

(c) $\text{OS}_{SimCore}$ with X = Birds, k = 1

(d) $\text{OS}_{SimCore}$ with X = Birds, k = 100

Figure 7: Visualization of sampled coreset by SimCore method. We plotted histograms for the top-20 classes with the largest number of samples and visualized one example image per top-9 classes. We highlighted the coreset classes with orange, which do not look similar to the target data. X denotes the target dataset, and k is the number of centroids in k-means clustering to reduce the complexity of SimCore. The x-axis and y-axis of histograms denote the class index and the number of samples, respectively.

F kNN Classifier and Semi-Supervised Learning Results

We have shown the linear evaluation performance on each fine-grained dataset to evaluate the quality of learned representation. Here, we also evaluate other downstream tasks: nearest neighbor (NN) classification and semi-supervised learning. The NN classification is a nonparametric way to classify the test images, which makes the prediction via weighted voting of nearest neighbors [56, 9]. Table 7 summarizes the 20 NN and 200 NN classification results, showing that SimCore consistently shows the best accuracy.

In addition, when part of the datasets becomes labeled by expert annotators, we can use those labels to further fine-tune the entire network. Here, we followed the semi-supervised learning protocol in [12, 21]. In Table 7, we show the results with three label ratios, each trained with 100 epochs. We used an SGD optimizer with ℓ_2 regularization parameter 1e-4, and tuned the learning rate for five logarithmically spaced values from 0.01 to 1. SimCore again showed the best results overall, especially by a large margin in Aircraft and Pet datasets.

Table 7: Nearest neighbor classifier and semi-supervised learning performances. We experimented with 20-nearest and 200-nearest neighbors for kNN classifier, and the label ratio of 10%, 20%, and 50% for semi-supervised learning setup. We used the pretrained model by SimCore with the sampling ratio $p = 0.01$.

	Aircraft: kNN		Aircraft: semi-sup.			Cars: kNN		Cars: semi-sup.		
pretraining	20 NN	200 NN	10%	20%	50%	20 NN	200 NN	10%	20%	50%
X	36.12	36.69	28.97	47.59	64.61	**33.11**	**34.32**	**25.07**	53.46	80.18
OS	19.32	17.67	19.59	34.05	43.93	11.38	10.86	10.84	35.70	74.11
X+OS$_{\textit{SimCore}}$	**40.44**	**40.41**	**34.82**	**51.98**	**66.97**	32.89	34.22	24.57	**54.55**	**81.73**

	Pet: kNN		Pet: semi-sup.			Birds: kNN		Birds: semi-sup.		
pretraining	20 NN	200 NN	10%	20%	50%	20 NN	200 NN	10%	20%	50%
X	52.03	51.84	47.16	58.73	71.37	20.74	21.76	13.26	25.22	51.24
OS	50.40	48.95	35.74	62.26	76.94	13.90	15.11	10.07	21.02	51.21
X+OS$_{\textit{SimCore}}$	**67.46**	**67.40**	**61.89**	**71.52**	**81.18**	**23.71**	**24.47**	**14.61**	**27.43**	**56.27**