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

Sungnyun Kim* Sangmin Bae* Se-Young Yun
KAIST AI
{ksn4397, bsmn0223, yunseyoung}@kaist.ac.kr

Abstract

Deep learning in general domains has constantly been extended to domain-specific tasks requiring the recognition of fine-grained characteristics. However, real-world applications for fine-grained tasks suffer from two challenges: a high reliance on expert knowledge for annotation and necessity of a versatile model for various downstream tasks in a specific domain (e.g., prediction of categories, bounding boxes, or pixel-wise annotations). Fortunately, the recent self-supervised learning (SSL) is a promising approach to pretrain a model without annotations, serving as an effective initialization for any downstream tasks. Since SSL does not rely on the presence of annotation, in general, it utilizes the large-scale unlabeled dataset, referred to as an open-set. In this sense, we introduce a novel Open-Set Self-Supervised Learning problem under the assumption that a large-scale unlabeled open-set is available, as well as the fine-grained target dataset, during a pretraining phase. In our problem setup, it is crucial to consider the distribution mismatch between the open-set and target dataset. Hence, we propose SimCore algorithm to sample a coreset, the subset of an open-set that has a minimum distance to the target dataset in the latent space. We demonstrate that SimCore significantly improves representation learning performance through extensive experimental settings, including eleven fine-grained datasets and seven open-sets in various downstream tasks.

1. Introduction

The success of deep learning in general computer vision tasks has encouraged its widespread applications to specific domains of industry and research [21, 46, 73], such as facial recognition or vehicle identification. We particularly focus on the visual recognition of fine-grained datasets, where the goal is to differentiate between hard-to-distinguish images. However, real-world application for fine-grained tasks poses two challenges for practitioners and researchers developing algorithms. First, it requires a number of experts for annotation, which incurs a large cost [3, 14, 60]. 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 there may be 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 development of a versatile pretrained model. One might ask, as a target task, that bird images be classified by species 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 [43, 47, 74], pixel-level annotations [53, 74], or bounding boxes [35, 47, 53].

Recently, self-supervised learning (SSL) [11, 14, 24, 28] has enabled learning how to represent the data even without annotations, such that the representations serve as an effective initialization for any future downstream tasks. Since labeling is not necessary, SSL generally utilizes an *open-set*, or large-scale unlabeled dataset, which can be easily obtained by web crawling [23, 64], for the pretraining. In this paper, we introduce a novel Open-Set Self-Supervised Learning (OpenSSL) problem, where we can leverage the open-set as well as the training set of fine-grained target dataset. Refer to Figure 1 for the overview of OpenSSL.

In the OpenSSL setup, since the open-set may contain instances irrelevant to the fine-grained dataset, we should consider the distribution mismatch. A large distribution mismatch might inhibit representation learning for the target task. For instance, in Figure 2, SSL on the open-set (*OS*) does not always outperform SSL on the fine-grained dataset (X) because it depends on the semantic similarity between X and *OS*. This is in line with the previous observations [21, 22, 64] that the performance of self-supervised representation on downstream tasks is correlated with similarity of pretraining and fine-tuning datasets.

To alleviate this mismatch issue, we could exploit a *coreset*, a subset of an open-set, which shares similar semantics with the target dataset. As a motivating experiment, we manually selected the relevant classes from ImageNet (OS_{oracle}) that are supposed to be helpful according to each target

*equal contribution

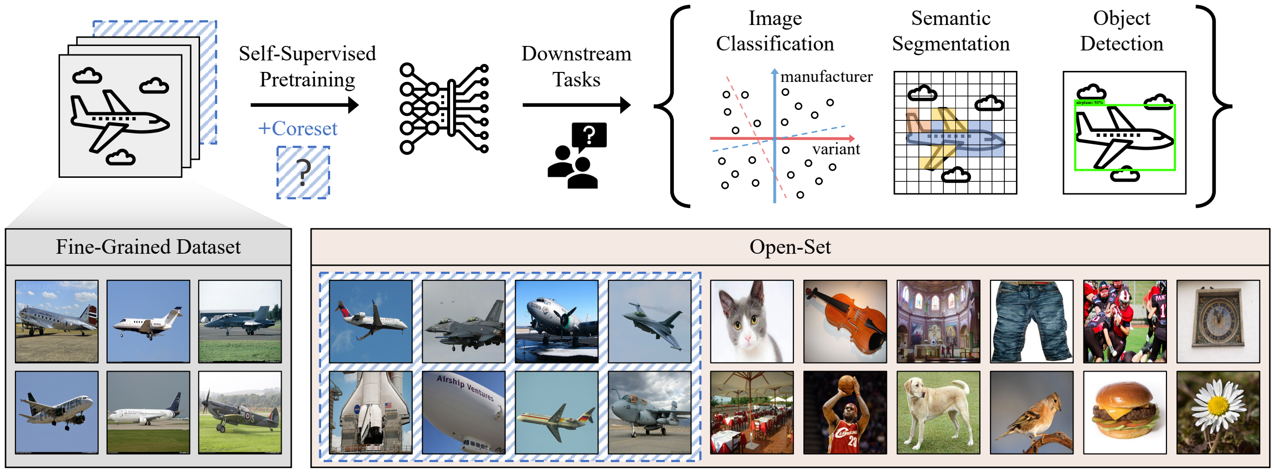
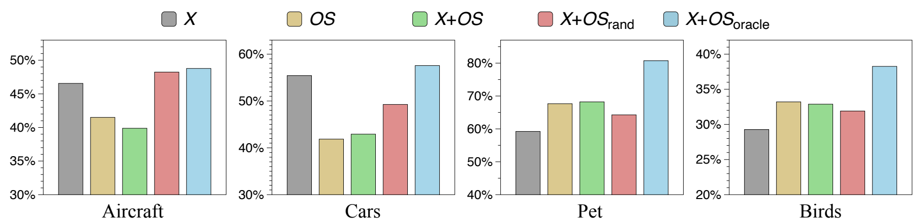


Figure 1. Overview of an OpenSSL problem. For any downstream tasks, we pretrain an effective model with the fine-grained dataset via self-supervised learning (SSL). Here, the assumption for a large-scale unlabeled open-set in the pretraining phase is well-suited for a real-world scenario. The main goal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



Target (X)	Classes for OS_{oracle} (#)
Aircraft	airliner, warplane, ... (8)
Cars	convertible, jeep, ... (10)
Pet	Persian cat, beagle, ... (24)
Birds	goldfinch, junco, ... (20)

Figure 2. Linear evaluation performance on the fine-grained target dataset. Each color corresponds to a pretraining dataset, while "+" means merging two datasets. The table on the right side shows the manually selected categories from an open-set (OS), ImageNet-1k [20] 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 [14, 24] and used the SimCLR method [14] on ResNet50 encoder [29].

dataset. Interestingly, in Figure 2, merging OS_{oracle} to X shows a significant performance gain, and its superiority over 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.

Therefore, we propose **SimCore**, a simple yet effective coreset sampling algorithm from an unlabeled open-set. Our main goal is to find a subset semantically similar to the target dataset. We formulate the data subset selection problem to obtain a coreset that has a minimum distance to the target dataset in the latent space. SimCore significantly improves performance in extensive experimental settings (eleven fine-grained datasets and seven open-sets), and shows consistent gains with different architectures, SSL losses, and downstream tasks. Our contributions are outlined as follows:

- We first propose a realistic OpenSSL task, assuming an unlabeled open-set available during the pretraining phase on the fine-grained dataset.
- We propose a coreset selection algorithm, SimCore, to leverage a subset semantically similar to the target dataset.
- Our extensive experiments with eleven fine-grained datasets and seven open-sets substantiate the significance of data selection in our OpenSSL problem.

2. Related Works

2.1. Self-Supervised Learning

After Oord *et al.* [52] proposed an InfoNCE loss, contrastive learning algorithms began to show remarkable improvements in representation learning [10, 11, 14, 16, 24, 28, 37]. While a large-scale open-set enhances the generalization of learned representation [19, 31, 64], recent literature has pointed out the distribution mismatch between pretraining and fine-tuning datasets [21, 23, 64]. Particularly, El *et al.* [21] claimed that pretraining on ImageNet may not always be effective on the target task from different domains. Tian *et al.* [64] found that pretraining with uncurated data, a more realistic scenario, deteriorates the target task performance. Although the motivations coincide with ours, their proposed methods are focused on a single data scheme: a denoising autoencoder [21] only for fine-grained dataset, and distillation from expert models [64] or a novel architecture [23] for uncurated open-sets. In contrast, we propose an explicit sampling strategy from an open-set, which becomes more effective by augmenting fine-grained dataset with well-matched samples, as well as achieving robustness to the distribution discrepancy or curation of the open-set.

Task [ref.]	Problem Setting	Train (Labeled)	Train (Unlabeled)	Test	Definition of OS / CS	Main Goal
Novel Class Discovery [26, 30, 81]	test data consist of only novel classes	seen	-	novel	-	cluster novel classes in test dataset
Open-Set Recognition [5, 12, 57, 68]	test set contains seen and novel classes	seen	-	seen + novel	[OS] test dataset containing seen and novel classes	reject instances from novel classes at test time
Webly Sup. [15, 39, 62]	train data contains web-crawled noisy samples	partially noisy	-	seen	[OS] web-crawled train dataset containing noisy samples	robustly train instances with corrupted labels
Open-Set Semi-Sup. [17, 33, 51, 56, 61]	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-Sup. [4, 8, 9]	train and test data contain novel classes	seen	seen + novel	seen + novel	[OS] dataset containing seen and novel classes	discover novel classes and assign samples at test time
Open-Set Annotation [50]	unlabeled data pool contains novel classes	seen	seen* + novel*	seen	[OS] unlabeled data pool with seen and novel classes	aim to query seen classes from unlabeled data pool
Coreset Selection in AL [58, 72]	query instances to be annotated	seen	seen*	seen	[CS] the most representative subset of unlabeled set	find a small subset competitive to whole dataset
Coreset Selection in CL [1, 65, 78]	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 in Self-Sup. [55, 70]	assume that hard negatives are helpful	-	target	target	[CS] the hardest contrastive pair instances for SSL	improve SSL performance using core-negative instances
Open-Set Self-Sup. [ours]	utilize open-set in pretraining, which may have irrelevant data	-	target + irrelevant	target	[OS] large-scale unlabeled set [CS] subset of OS sharing the same semantics with target set	improve SSL performance on fine-grained datset via coreset sampling method

Table 1. Comparisons of the OpenSSL problem with relevant literature. We focus on the problem setting and the definition of open-set (OS) or coreset (CS) in each field. AL and CL refer to active learning and continual learning, respectively, and * indicates the instances in the unlabeled data pool that are supposed to be annotated after the active selection.

2.2. 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 relevant or irrelevant domain to target dataset. The assumption of available open-set is also common in other research fields, such as open-set recognition [5, 12, 57, 68], webly supervised learning [15, 39, 62], open-set [17, 33, 51, 56] or open-world [4, 8, 9] semi-supervised learning, and open-set annotation [50], although its detailed meaning varies in each field. We summarize the details in Table 1. Especially, our OpenSSL task is to recognize coreset from a large-scale open-set, without exploiting any label information. From this perspective, recent coreset selection approaches give us a good intuition. Existing studies find the representative subset of the unlabeled set for active selection [58, 72], or find the subset of current task data to avoid catastrophic forgetting for continual learning [1, 65, 78]. In the meantime, several works on self-supervised learning have developed a novel loss function that leverages hard negative samples, *i.e.*, hard negative mining [55, 70], which shares similar concepts to our coreset. Our problem setup further requires an effective algorithm that takes into consideration of the distribution discrepancy in the open-set.

3. Method

3.1. OpenSSL Problem Formulation

SSL is a groundbreaking paradigm for learning inherent properties of data, while discarding irrelevant signals by discriminating perturbed samples. Recent literature has proposed a contrastive loss function to encourage making representations from the same image similar and representations from different images dissimilar [14, 28, 79].

Given an input data $X = \{x_i\}_{i=1}^N$, we generate two copies of random augmented image, $\mathcal{A}(X) = \{\tilde{x}_i\}_{i=1}^{2N}$, where $\tilde{x}_i$ and $\tilde{x}_{N+i}$ are an augmented pair of each other. Let E_θ be an encoder network. Then, with the augmented pairs, we generally formulate a contrastive loss as follows:

$$\mathcal{L}(X; E_\theta) = \frac{1}{2|X|} \sum_{\tilde{x}_i \in \mathcal{A}(X)} \ell_{\mathsf{ssl}}(z_i, z_i^+; \{z_i^-\}) \tag{1}$$

where $z_i = E_\theta(\tilde{x}_i)$, z_i^+ denotes the representation from the augmented pair of $\tilde{x}_i$, and $\{z_i^-\}$ is the set of features from all the other samples. Eq. 1 forces z_i to be closer with z_i^+ and farther from $\{z_i^-\}$. Projection heads [14] or predictors [24] are often used in measuring the similarities of representations. In addition, non-contrastive methods

[10, 11, 16, 24] have also been proposed by not utilizing any negative pair set, *i.e.*, $\{z_i^-\} = \emptyset$.

In the OpenSSL problem, X corresponds to the training set of the fine-grained target dataset. Using Eq. 1, we can pretrain an encoder without any annotation of X. Furthermore, we have an unlabeled open-set $\mathcal{U}$, which can be jointly used with X. Rather than pretraining on a simple fusion of X and $\mathcal{U}$, we are motivated to sample a relevant subset $\mathcal{S}$ from the open-set. We then pretrain the encoder with $\mathcal{L}(X \cup \mathcal{S}; E_\theta)$.

3.2. 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 a key 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 $\mathcal{S}$ that maximizes the following objective function:

$$f(\mathcal{S}) = \sum_{x \in X} \max_{u \in \mathcal{S}} w(x, u), \text{ where } \mathcal{S} \subseteq \mathcal{U},\ \mathcal{U} \cap X = \emptyset \quad (2)$$

while $w(x, u) = z_x^\top z_u$ estimates similarity of two representations, and z is the normalized feature from the encoder E_θ pretrained on X with small epochs. From Eq. 2, SimCore finds a subset that shares the most similar semantics with the target set. This is reminiscent of the facility location function [48, 72], $f_{\text{fac}}(\mathcal{S}) = \sum_{x \in \mathcal{U}} \max_{u \in \mathcal{S}} w(x, u), \mathcal{S} \subseteq \mathcal{U}$. However, $f(\mathcal{S})$ is different from $f_{\text{fac}}(\mathcal{S})$, since target samples are the elements of X, which is disjoint with $\mathcal{S}$.

Meanwhile, the direct calculation of pairwise similarities requires the complexity of $\mathcal{O}(|X||\mathcal{U}|)$, which might be extremely large. To reduce the computational overhead and make the algorithm scalable, we adapt k-means clustering [2] for the target dataset X, and replace it with the centroid set $\hat{X}$. Its associated objective function is $\hat{f}(\mathcal{S})$. Even when exploiting only the centroids ($k = 100$ in practice), we show significant performance gains on various benchmarks.

Iterative coreset sampling: $\hat{f}(\mathcal{S})$ is a monotonically increasing submodular function. One remark is that if there is no constrained budget on $\mathcal{S}$, a subset achieving the maximum value of $\hat{f}(\mathcal{S})$ is not unique. If we denote $\mathcal{S}^*$ as a minimal set with the maximum value, $\mathcal{S}^*$ is obtained when including only the instances closest to each instance of $\hat{X}$ (*i.e.*, $|\mathcal{S}^*| \leq |\hat{X}|$). Since we want to sample sufficiently large subset, we re-define Eq. 2 with the selection round t: $\hat{f}(\mathcal{S}_t)$ where the candidate set is $\mathcal{U}_t$. Thus, we iterate the rounds to repeat sampling $\mathcal{S}_t^*$ and excluding them from the candidate set, until we reach the proper budget size (see Algorithm 1). We collect all the coreset samples into a set $\mathcal{I} = \mathcal{S}_1^* \cup \mathcal{S}_2^* \cup \dots \cup \mathcal{S}_T^*$ to merge with X.

Stopping criterion: However, we have no knowledge in practice if every sampled instance within the budget is sufficiently close to the target set. This necessitates stopping criterion that blocks the sampling process from continuing when the samples are no longer close to the target set. To this end, we calculate the ratio $\hat{f}(\mathcal{S}_t^*)/\hat{f}(\mathcal{S}_1^*)$ at each iteration and stop the sampling process if its value is less than the threshold. This implies that we stop at iteration t if the sampled subset is not as similar as the first subset is to the target set. We filtered-out by using the threshold $\tau = 0.95$ throughout experiments.

Algorithm 1: Simple coreset sampling from open-set

Require: E_θ: encoder pretrained on X;
Require: $\mathcal{U}_0$: initial candidate set (open-set);
Require: $\mathcal{B}, \tau$: coreset budget, threshold;
initialize $\mathcal{I} \leftarrow \emptyset, t \leftarrow 0$;
replace $\hat{X} \leftarrow$ cluster centroids of X;
calculate $z_x, z_u \leftarrow E_\theta(x), E_\theta(u)$ for $\forall x, u \in \hat{X} \times \mathcal{U}_0$;
while $|\mathcal{I}| < \mathcal{B}$ **do**
    set $\mathcal{S}_t^*$ as the elements in $\mathcal{U}_t$ that are closest to each element in $\hat{X}$ (Eq. 2);
    $\mathcal{I} \leftarrow \mathcal{I} \cup \mathcal{S}_t^*,\ \mathcal{U}_{t+1} \leftarrow \mathcal{U}_t \setminus \mathcal{S}_t^*$
    $t \leftarrow t + 1$
    // stopping criterion
    if $\hat{f}(\mathcal{S}_t^*) < \tau \cdot \hat{f}(\mathcal{S}_1^*)$ **then**
        stop sampling;
    end
end
re-initialize θ and pretrain E_θ with $X \cup \mathcal{I}$;

4. Experiments

We evaluate the learned representation quality according to each pretraining dataset. We thus demonstrate, through various fine-grained target tasks, that SimCore samples an effective subset that enhances pretraining. Here, we used eleven target datasets: Aircraft [47], Cars [35], Pet [53], Birds [74], Dogs [32], Flowers [49], Action [77], Indoor [54], Textures [18], Faces [36], and Food [59]. For the open-set, we mainly used ImageNet-1k [20], while we extended to other open-sets in Section 4.2. By default, we used SimCLR [14] with ResNet50 architecture as the encoder. The detailed experimental setups and dataset configurations are in Appendix B.

4.1. Performance Evaluation on Target Tasks

Linear evaluation: Conventional SSL literature evaluates linear probing performance with frozen pretrained encoder to measure the representation quality [14, 24]. Table 2 summarizes the linear evaluation results on eleven fine-grained datasets. We note two observations from this experiment. First, we evaluated if the coreset sampled by SimCore is qualitative as pretraining data. To this end, we set the budget size to $p = 1\%$ or $p = 5\%$ (*i.e.*, sampling p-ratio of the open-

pretrain	p	Target dataset (X) and its number of samples										
		Aircraft 6,667	Cars 8,144	Pet 3,680	Birds 5,990	Dogs 12,000	Flowers 2,040	Action 4,000	Indoor 5,360	Textures 3,760	Faces 4,263	Food 13,296
X	-	46.56	55.42	59.23	29.27	49.88	80.14	43.76	54.10	58.78	56.63	87.99
OS	-	41.50	41.86	67.66	33.21	49.94	85.67	60.65	64.46	67.23	52.84	86.14
X+OS	-	39.88	42.92	68.22	32.88	50.42	85.34	60.61	63.66	67.98	52.76	85.90
X+OS_{rand}	1%	48.24	49.26	64.27	31.90	49.62	83.17	47.25	55.37	61.33	57.37	88.08
X+$OS_{\text{SimCore}\dagger}$	1%	48.06	58.56	74.82	33.37	57.42	82.12	51.37	57.84	61.76	56.95	90.35
X+OS_{SimCore}	1%	**48.45**	59.00	77.13	36.56	59.83	86.70	52.98	59.18	63.40	58.85	89.78
X+OS_{rand}	5%	45.75	46.03	68.38	33.63	50.24	84.52	57.27	60.71	65.80	56.05	87.75
X+$OS_{\text{SimCore}\dagger}$	5%	45.57	50.75	80.20	35.56	64.62	85.11	64.53	68.13	66.22	58.93	89.87
X+OS_{SimCore}	5%	47.14	52.22	**81.75**	**39.21**	**66.82**	**87.28**	66.38	70.96	**68.13**	**59.34**	90.74
Stopping Criterion		1.03%	0.95%	14.4%	13.7%	9.72%	7.96%	15.6%	13.5%	5.89%	0.27%	3.86%
X+OS_{SimCore}	-	48.27	**60.29**	79.66	37.65	66.48	87.04	**67.46**	**71.95**	67.66	59.01	**91.31**

Table 2. Linear evaluation performance on eleven fine-grained datasets. We used ImageNet-1k [20] as an open-set. The ratio p is a fixed budget size to sample from the open-set, either via random sampling (OS_{rand}) or SimCore (OS_{SimCore}). Given that OS has 1.3M samples, $p = 1\%$ corresponds to 13K samples. † denotes SimCore with $k = 1$, a single cluster. **Bold** and underline indicate the best and the second best accuracy for each target dataset, respectively.

method	architecture	pretrain	Aircraft	Cars	Pet	Birds
SimCLR	EfficientNet	X	25.5	37.0	58.1	27.8
SimCLR	EfficientNet	OS	31.6	29.5	57.8	26.5
SimCLR	EfficientNet	**SimCore**	**41.7**	**52.8**	**69.5**	**29.6**
SimCLR	ResNet18	X	43.4	51.9	58.2	25.9
SimCLR	ResNet18	OS	33.9	33.1	62.5	27.7
SimCLR	ResNet18	**SimCore**	**44.5**	**55.1**	**72.7**	**31.3**
SimCLR	ResNeXt50	X	45.9	56.5	63.4	28.6
SimCLR	ResNeXt50	OS	39.2	39.4	68.2	32.6
SimCLR	ResNeXt50	**SimCore**	**49.5**	**59.5**	**81.0**	**37.4**
SimCLR	ResNet101	X	49.4	54.5	64.0	29.1
SimCLR	ResNet101	OS	40.4	41.9	69.5	34.2
SimCLR	ResNet101	**SimCore**	**50.9**	**58.8**	**83.0**	**39.1**

Table 3. Linear evaluation performance with different architectures. SimCore corresponds to X+OS_{SimCore} with a stopping criterion.

method	architecture	pretrain	Aircraft	Cars	Pet	Birds
BYOL	ResNet50	X	40.6	49.4	56.5	27.6
BYOL	ResNet50	OS	46.1	49.6	78.4	44.7
BYOL	ResNet50	**SimCore**	**46.5**	**50.4**	**85.1**	**47.9**
SwAV	ResNet50	X	34.5	42.4	49.4	21.6
SwAV	ResNet50	OS	33.8	30.0	64.2	27.3
SwAV	ResNet50	**SimCore**	**45.0**	**45.1**	**80.2**	**36.6**
DINO	ViT-Ti/16	X	27.3	**48.2**	42.4	28.5
DINO	ViT-Ti/16	OS	42.0	39.1	78.4	61.2
DINO	ViT-Ti/16	**SimCore**	**43.2**	47.2	**83.3**	**72.6**
MAE	ViT-B/16	X	**55.9**	44.7	56.3	32.2
MAE	ViT-B/16	OS	39.8	37.3	68.3	31.4
MAE	ViT-B/16	**SimCore**	48.1	**52.4**	**77.8**	**42.1**

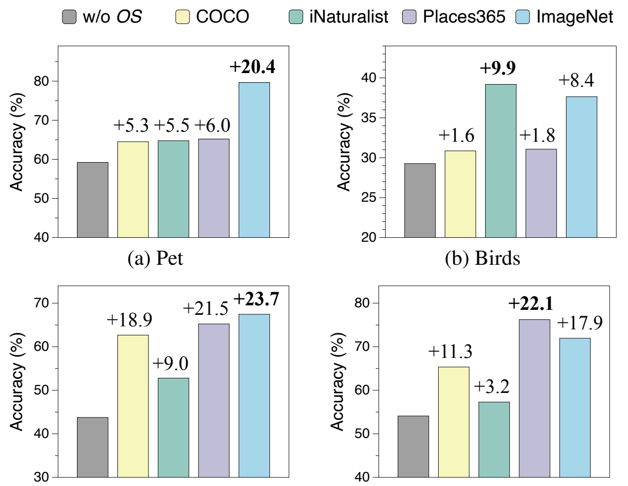
Table 4. Linear evaluation performance with different SSL methods. SimCore corresponds to X+OS_{SimCore} with a stopping criterion.

set) and compared them with $p\%$ random sampling strategy. In every case, SimCore outperformed the random sampling. This demonstrates that exploiting the coreset is actually crucial compared to naïvely using random samples. Second, using a number of cluster centroids of the target dataset is more advantageous than a single cluster, although SimCore with $k = 1$ also outperformed the random sampling.

Interestingly, 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. For example, in datasets like Pet and Birds, OS pretraining was pretty effective, and in those datasets, SimCore took advantage of the large budget size. This implies that several target datasets require more coreset samples than do others. However, in practice, we cannot pre-define the optimal budget size, since we do not have much knowledge about an uncurated open-set. Therefore, we should handle SimCore with a stopping criterion, as we already have proposed in Section 3.2.

Surprisingly, SimCore with a stopping criterion highly improves the accuracy by +10.5% (averaged over 11 datasets), compared to the X pretraining. This is much larger gain compared to the large-scale OS pretraining (+2.7%) and 1% random sampling (+1.3%). This is because SimCore adaptively samples a proper amount of coreset, and this amount differs by each target dataset. For Aircraft and Cars, SimCore sampled around 1% of ImageNet. This is a reasonable number, because in the ImageNet dataset [20], there are actually 4 aircraft-related and 10 cars-related classes (refer to Appendix A), out of 1,000 classes in total.

Different encoder architectures and SSL methods: In Table 3, we have applied SimCore to different architectures: EfficientNet-B0 [63], ResNet18 [29], ResNeXt50 [76], and ResNet101 [29]. Regardless of whether the encoder architecture is much smaller or larger, SimCore greatly improves pretraining on the target dataset. Moreover, we have experimented with various SSL methods, such as BYOL [24],



(a) Pet (b) Birds

(c) Action (d) Indoor

Figure 3. SimCore performances compared to the X pretraining (w/o *OS*). In addition to ImageNet-1k, we included MS COCO, iNaturalist 2021-mini, and Places365 as an open-set. For target datasets, we used Pet and Birds for natural image datasets and Action and Indoor for unnatural image datasets.

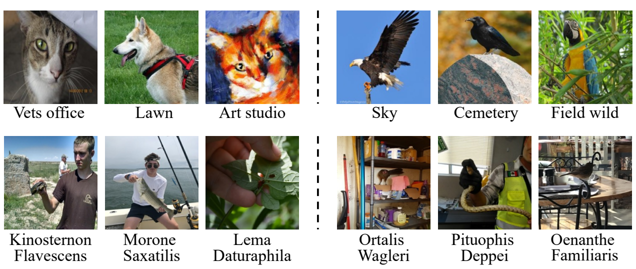


Figure 4. Selected samples of Places365 coreset (top) for Pet (left) and Birds (right) and iNaturalist coreset (bottom) for Action (left) and Indoor (right). Captions are the actual labels in each open-set.

SwAV [10], DINO [11], and MAE [27], in Table 4. SimCore consistently demonstrates the effect of merging the coreset samples, even with the recent autoencoder-based SSL.

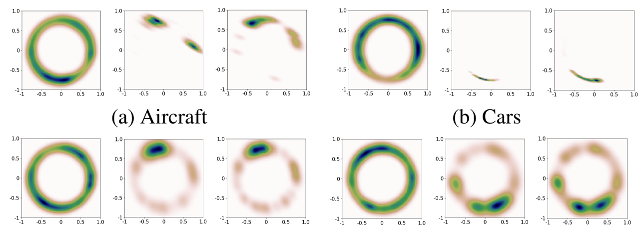
4.2. SimCore on Various Open-Sets

Thus far, we have used ImageNet-1k benchmark [20] as the open-set, which is a well-curated dataset covering general domains [23, 64]. Drawing a coreset from ImageNet has shown large performance gains in the target tasks. However, in practice, an open-set is far from what we know about; it is rather a grouping of data arbitrarily drawn from the web or database. Here, we show that our SimCore with any open-set robustly improves pretraining because it finds a well-matched coreset. We experimented with three other open-sets: MS COCO [42], iNaturalist 2021-mini [67], and Places365 [82].

Figure 3 shows that SimCore consistently outperforms the pretraining without *OS*, while the performance of SimCore depends on the open-set. In unnatural fine-grained datasets

OS	Aircraft	Cars	Birds	Pet	Action	Indoor
✗	46.6	55.4	29.3	59.2	43.8	54.1
ImageNet-1k	48.3	60.3	37.7	79.7	**67.5**	72.0
`ALL`	**58.8**	**64.8**	**38.0**	79.5	67.2	**75.1**
WebVision	48.2	59.1	36.7	**80.3**	67.1	72.6
WebFG-496	55.4	63.1	37.7	-	-	-

Table 5. Linear evaluation of SimCore with uncurated open-sets.



(a) Aircraft (b) Cars

(c) Pet (d) Birds

Figure 5. Feature distribution map of *OS* (left), X (middle), and the coreset sampled by SimCore (right).

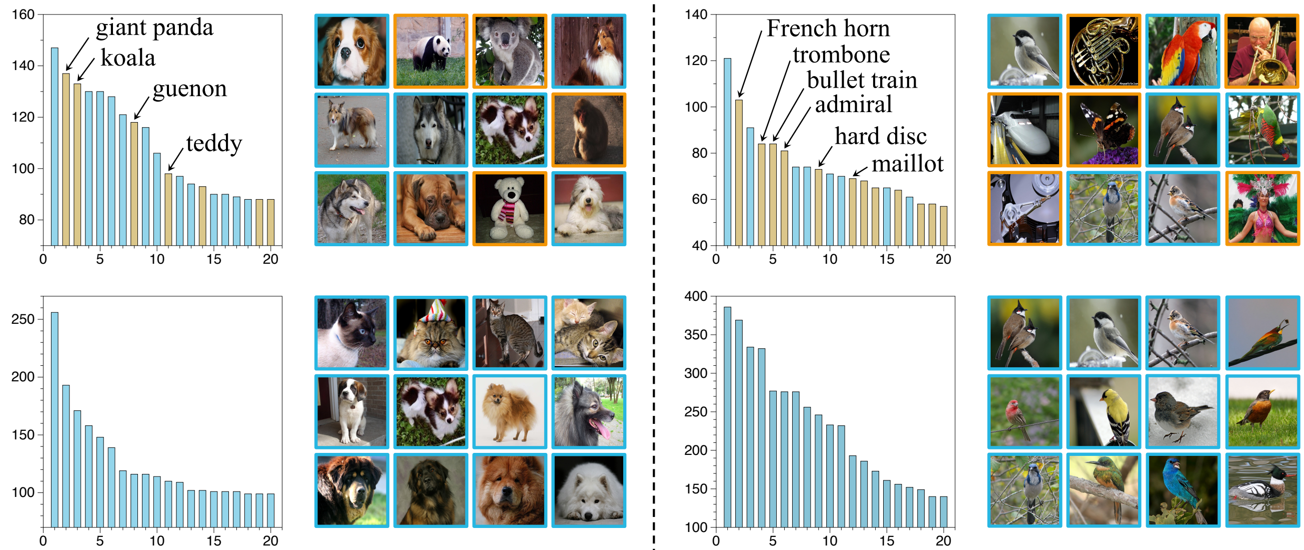
like Action and Indoor, using iNaturalist is not as effective as ImageNet, but it offers a good coreset for Birds. As expected, for Indoor, Places365 offers the best coreset among every open-sets because it contains a lot of scenery semantics.

Nevertheless, we have observed the *unexpected* gains, such as Places365 open-set for Pet target dataset. Figure 4 illustrates a few selected samples of those coresets. It is interesting to see that SimCore has found the animal images, although the actual labels correspond to the locations. Also, the iNaturalist coreset contains natural creatures that are held by humans or located in indoor. This might help in Action target, part of which are humans taking photos, fishing, or gardening, as well as in Indoor scenery target.

Uncurated open-sets: To demonstrate the effect of SimCore in the more realistic scenario, we have tested SimCore with uncurated open-sets. First, we used a combined dataset of all four pre-mentioned open-sets (`ALL`), to simulate the more large-scaled and heterogeneous open-set case. In addition, we further used web-crawled image dataset queried by ImageNet classes (WebVision [40]), and the images queried by Aircraft+Cars+Birds classes (WebFG-496 [62]). Interestingly, in the case of WebFG-496 that includes noisy instances, SimCore sampled slightly less of the actual crawled set for each target. For example, while WebFG-496 contains 13,508 number of queried instances for Aircraft, SimCore sampled a coreset of 8,089 instances. Table 5 demonstrates that these uncurated open-sets are comparable to the curated ones and significantly outperform the pretraining without open-sets. Indeed, SimCore could sample a useful coreset regardless of how uncurated an open-set is.

4.3. Qualitative Evaluation

Feature distribution analysis: To analyze how the SimCore algorithm samples the coreset in the latent space, we



(a) OS_{SimCore} with X = Pet and $\{k = 1$ (top), $k = 100$ (bottom)$\}$

(b) OS_{SimCore} with X = Birds and $\{k = 1$ (top), $k = 100$ (bottom)$\}$

Figure 6. Visualization of the 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-12 classes. We highlighted with orange the coreset classes that look dissimilar to the target data. 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.

visualized the feature distribution on a unit ring by Gaussian kernel density estimation in $\mathbb{R}^2$ [71] (implementation details in Appendix D). The results in Figure 5 are interesting, as SimCore actually samples the instances that are closely embedded to the target data. In addition, through a comparison of the occupied areas of OS and X, we confirmed the distribution similarity of the open-set to each target dataset, indicating the sampling ratios by SimCore in Table 2 are reasonable.

Coreset visualization: We visualized which instances from the open-set are actually sampled by our SimCore algorithm. To this end, we displayed in Figure 6 the ground-truth labels of the coreset samples when the target dataset is Pet or Birds. For comparison, we also displayed the coreset by SimCore with $k = 1$, using a single centroid. We used the open-set as ImageNet, so the ground-truth labels of coreset samples correspond to the ImageNet classes. Note that Pet contains 12 cat breeds and 25 dog breeds [53], and Birds contains 200 bird species [74].

For the Pet dataset (Figure 6a), SimCore with $k = 1$ sampled mostly animal images, but included data somewhat irrelevant to cats and dogs. 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 $k = 100$ sampled mostly cat or dog images, with up to top-20 classes, each being a breed of either cat or dog. Interestingly, eight out of the top-10 classes were those that overlap with the Pet class labels, such as Persian cat, Saint Bernard, etc.

For the Birds dataset (Figure 6b), SimCore with $k = 1$ sampled a lot of irrelevant images, such as French horn, trombone, bullet train, admiral, hard disc, maillot, etc. On the contrary, SimCore with $k = 100$ sampled only the bird species up to the top-20 classes, including bulbul, chickadee, brambling, bee eater, house finch, goldfinch, etc.

	framework: *Open-Set Semi-Sup.*				framework: *Webly Sup.*			
pretrain	method	Aircraft	Cars	Birds	method	Aircraft	Cars	Birds
SimCore	FT (50%)	73.5	80.1	57.4	FT (100%)	84.3	**89.3**	70.6
✗	SelfTrain	51.9	55.5	35.7	CoTeach	79.3	51.7	70.4
SimCore	SelfTrain	**78.1**	**81.3**	**59.1**	CoTeach	**89.8**	57.0	**78.9**
✗	OpenMatch	70.1	70.2	52.3	DivideMix	82.2	54.4	74.5
SimCore	OpenMatch	**83.5**	**89.5**	**66.4**	DivideMix	**86.5**	56.5	**80.0**

Table 6. Comparisons with Self-Training [61] and OpenMatch [56] in the OpenSemi framework, and Co-teaching [25] and DivideMix [38] in the WeblySup framework.

4.4. Comparisons with Open-Set Semi-Supervised and Webly Supervised Learning

Prior literature has similarly utilized unlabeled or noisy-labeled open-sets, such as open-set semi-supervised learning (OpenSemi) [56, 61] and webly supervised learning (WeblySup) [15, 62]. OpenSemi and WeblySup work with predefined labels and co-train with the entire huge-scale open-set. Our OpenSSL, on the other hand, is a valuable problem itself in that a model can be pretrained without any label information and with efficient subset sampling. We thus design an experiment under each framework, setting (X / OS) as the following example: (Birds$^{50\%}$ / Birds$^{50\%}$+WebFG) and (Birds$^{100\%}$ / WebFG$^{\text{Birds}}$), respectively.

pretrain	Aircraft		Cars		Pet		Birds	
	20	200	20	200	20	200	20	200
X	36.1	36.7	33.1	34.3	52.0	51.8	**20.7**	**21.8**
OS	19.3	17.7	11.4	10.9	50.4	49.0	13.9	15.1
SimCore	**40.7**	**41.4**	**33.8**	**34.6**	**61.4**	**61.4**	18.3	19.2

(a) kNN classification

pretrain	Aircraft			Cars			Pet			Birds		
	10%	20%	50%	10%	20%	50%	10%	20%	50%	10%	20%	50%
X	29.0	47.6	64.6	25.1	53.5	80.2	47.2	58.7	71.4	**13.3**	25.2	51.2
OS	19.6	34.1	43.9	10.8	35.7	74.1	35.7	62.3	76.9	10.1	21.0	51.2
SimCore	**33.9**	**51.3**	**65.6**	**25.4**	**55.3**	**81.6**	**50.9**	**70.4**	**79.7**	11.9	**25.5**	**55.7**

(b) Semi-supervised learning

pretrain	Aircraft		Cars		Pet		Birds	
	mAP	mAP_{50}	mAP	mAP_{50}	IoU_{fg}	IoU_{bg}	IoU_{fg}	IoU_{bg}
X	10.8	12.7	34.7	40.0	79.1	82.0	65.3	92.6
OS	23.7	27.0	20.8	23.6	79.8	82.8	67.9	93.3
SimCore	**29.6**	**36.8**	**37.6**	**43.2**	**80.0**	**83.1**	**68.4**	**93.4**

(c) Object detection and pixel-wise segmentation

pretrain	Aircraft		Cars		Faces			
	mfr.	family	brand	type	pointy	oval	young	smiling
X	21.1	40.4	67.3	78.0	66.8	83.4	93.1	93.4
OS	17.9	35.2	49.3	61.3	64.9	81.9	92.9	86.3
SimCore	**21.9**	**41.9**	**70.7**	**80.1**	**67.5**	**83.9**	**93.6**	**93.7**

(d) Multi-attribute classification

Table 7. Various downstream task performances. For the kNN classifier, we experimented with 20-nearest and 200-nearest neighbors to classify the query images. In the semi-supervised learning setup, we used the label ratio of 10%, 20%, and 50%. mAP is evaluated in the object detection, and IoU of foreground and background are evaluated in the semantic segmentation. In the multi-attribute classification, mfr., pointy, and oval denote manufacturer, pointy nose, and oval face, respectively.

Table 6 summarizes the comparisons with the representative methods of each learning framework. We observed two findings from Table 6. (1) When the SimCore model, using WebFG-496 as an open-set, is simply fine-tuned on each target, it outperforms OpenSemi and WeblySup methods. (2) SimCore can synergize with both frameworks, serving as an effective initialization. These results are in line with Su *et al*. [61], where Self-Training with SSL pretrained model was most preferred.

4.5. Different Downstream Tasks

kNN classifier: We have shown the linear evaluation performance on each fine-grained dataset to evaluate the quality of learned representation. Here, we evaluate on another downstream task, nearest neighbor (NN) classification. The NN classification is a nonparametric way to classify test images, making the prediction via weighted voting of nearest neighbors [10, 75]. Table 7a summarizes the 20 NN and 200 NN classifier results, presenting that SimCore shows the best accuracy, except for Birds where every accuracy was low.

Semi-supervised learning: 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 [14, 24], where the randomly selected samples are annotated and used in the fine-tuning. In Table 7b, we show the results with three label ratios, each trained with 100 epochs. SimCore again showed the best results overall, by particularly large margins in Aircraft and Pet datasets. For other semi-supervised learning algorithms, refer to Appendix F.3.

Object detection and pixel-wise segmentation: For object detection, we used RetinaNet detector [41] with ResNet50 encoder and trained for 100 epochs. The metric is mAP evaluated as in [42], averaging over 10 IoU thresholds from 0.5 to 0.95 with a step size 0.05. mAP_{50} is the mAP when IoU threshold is 0.5. For pixel-wise segmentation, we used DeepLabV3+ segmentation model [13] and trained for 30 epochs. In Table 7c, our SimCore-pretrained encoder could more effectively identify the vehicle objects and segregate the foreground part of pet and bird images.

Multi-attribute classification: Fine-grained images are often distinguished by multiple attributes. For example, one might ask if the aircraft images could be classified by manufacturers (*e.g*., Boeing), families (*e.g*., Boeing 737), or variants (*e.g*., Boeing 737-700). The Cars dataset can also be identified by brands (*e.g*., Audi) or types (*e.g*., Coupe), in addition to the models (*e.g*., Audi TTS Coupe 2012) we used in linear evaluation. To this end, we evaluated our pretrained models on multi-attribute classification tasks of three fine-grained datasets, and SimCore excelled the baselines in every task (see Table 7d).

5. Conclusion

In this work, we present a novel OpenSSL problem, where a large-scale open-set can be utilized in self-supervised learning on a fine-grained target dataset. Thereafter, we propose SimCore algorithm, which samples the effective coreset from an open-set with semantics similar to the target dataset. We demonstrate that the coreset samples enhance representation learning for any fine-grained downstream tasks. We believe that our approach will stimulate more investigation into fine-grained SSL with the open-set in the future.

Acknowledgement. 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) and No. 2022-0-00871, Development of AI Autonomy and Knowledge Enhancement for AI Agent Collaboration).

References

[1] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. *Advances in neural information processing systems*, 32, 2019. 3

[2] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006. 4

[3] Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671*, 2019. 1, 20

[4] Abhijit Bendale and Terrance Boult. Towards open world recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1893–1902, 2015. 3

[5] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1563–1572, 2016. 3

[6] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *International Conference on Learning Representations*, 2020. 20

[7] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019. 16, 20

[8] Terrance E Boult, Steve Cruz, Akshay Raj Dhamija, Manuel Gunther, James Henrydoss, and Walter 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. 3

[9] Kaidi Cao, Maria Brbic, and Jure Leskovec. Open-world semi-supervised learning. *arXiv preprint arXiv:2102.03526*, 2021. 3

[10] 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. 2, 4, 6, 8, 16

[11] 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 IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021. 1, 2, 4, 6, 16

[12] Guangyao Chen, Peixi Peng, Xiangqian Wang, and Yonghong Tian. Adversarial reciprocal points learning for open set recognition. *arXiv preprint arXiv:2103.00953*, 2021. 3

[13] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 8

[14] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 1, 2, 3, 4, 8, 15, 19, 20, 21

[15] Xinlei Chen and Abhinav Gupta. Webly supervised learning of convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1431–1439, 2015. 3, 7

[16] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021. 2, 4

[17] Yanbei Chen, Xiatian Zhu, Wei Li, and Shaogang Gong. Semi-supervised learning under class distribution mismatch. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3569–3576, 2020. 3

[18] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014. 4, 15

[19] Elijah Cole, Xuan Yang, Kimberly Wilber, Oisin Mac Aodha, and Serge 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. 2

[20] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2, 4, 5, 6, 15

[21] Alaaeldin El-Nouby, Gautier Izacard, Hugo Touvron, Ivan Laptev, Hervé Jegou, and Edouard Grave. Are large-scale datasets necessary for self-supervised pre-training? *arXiv preprint arXiv:2112.10740*, 2021. 1, 2

[22] Linus Ericsson, Henry Gouk, and Timothy 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. 1

[23] Priya Goyal, Mathilde Caron, Benjamin Lefaudeux, Min Xu, Pengchao Wang, Vivek Pai, Mannat Singh, Vitaliy Liptchinsky, Ishan Misra, Armand Joulin, et al. Self-supervised pretraining of visual features in the wild. *arXiv preprint arXiv:2103.01988*, 2021. 1, 2, 6

[24] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad 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. 1, 2, 3, 4, 5, 8, 15, 20

[25] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31, 2018. 7, 16

[26] Kai Han, Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Andrea Vedaldi, and Andrew Zisserman. Automatically discovering and learning new visual categories with ranking statistics. *arXiv preprint arXiv:2002.05714*, 2020. 3

[27] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 6, 16

[28] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross 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. 1, 2, 3, 15, 21

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2, 5

[30] Yen-Chang Hsu, Zhaoyang Lv, and Zsolt Kira. Learning to cluster in order to transfer across domains and tasks. *arXiv preprint arXiv:1711.10125*, 2017. 3

[31] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020. 2

[32] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR workshop on fine-grained visual categorization (FGVC)*, volume 2. Citeseer, 2011. 4, 15

[33] Krishnateja Killamsetty, Xujiang Zhao, Feng Chen, and Rishabh Iyer. Retrieve: Coreset selection for efficient and robust semi-supervised learning. *Advances in Neural Information Processing Systems*, 34:14488–14501, 2021. 3

[34] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 15

[35] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 1, 4, 15

[36] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5549–5558, 2020. 4, 15

[37] Chunyuan Li, Jianwei Yang, Pengchuan Zhang, Mei Gao, Bin Xiao, Xiyang Dai, Lu Yuan, and Jianfeng Gao. Efficient self-supervised vision transformers for representation learning. *arXiv preprint arXiv:2106.09785*, 2021. 2

[38] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*, 2020. 7, 16

[39] Junnan Li, Caiming Xiong, and Steven CH Hoi. Mopro: Webly supervised learning with momentum prototypes. *arXiv preprint arXiv:2009.07995*, 2020. 3

[40] Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*, 2017. 6

[41] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 8

[42] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 6, 8, 15

[43] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 1

[44] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 15

[45] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 16

[46] Wei Luo, Xitong Yang, Xianjie Mo, Yuheng Lu, Larry S Davis, Jun Li, Jian Yang, and Ser-Nam Lim. Cross-x learning for fine-grained visual categorization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8242–8251, 2019. 1

[47] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 1, 4, 15

[48] Pitu B Mirchandani and Richard L Francis. *Discrete location theory*. Wiley, 1990. 4

[49] 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*, pages 722–729. IEEE, 2008. 4, 15

[50] Kun-Peng Ning, Xun Zhao, Yu Li, and Sheng-Jun Huang. Active learning for open-set annotation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 41–49, 2022. 3

[51] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. *Advances in neural information processing systems*, 31, 2018. 3

[52] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2

[53] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012. 1, 4, 7, 15

[54] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *2009 IEEE conference on computer vision and pattern recognition*, pages 413–420. IEEE, 2009. 4, 15

[55] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592*, 2020. 3, 21

[56] Kuniaki Saito, Donghyun Kim, and Kate Saenko. Openmatch: Open-set consistency regularization for semi-supervised learning with outliers. *arXiv preprint arXiv:2105.14148*, 2021. 3, 7, 16

[57] Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult. Toward open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35:1757–1772, 2012. 3

[58] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017. 3, 20

[59] Ashutosh Singla, Lin Yuan, and Touradj Ebrahimi. Food/non-food image classification and food categorization using pre-trained googlenet model. In *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management*, pages 3–11, 2016. 4, 15

[60] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020. 1, 20

[61] Jong-Chyi Su, Zezhou Cheng, and Subhransu 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. 3, 7, 8, 16

[62] Zeren Sun, Yazhou Yao, Xiu-Shen Wei, Yongshun Zhang, Fumin Shen, Jianxin Wu, Jian Zhang, and Heng Tao Shen. Webly supervised fine-grained recognition: Benchmark datasets and an approach. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10602–10611, 2021. 3, 6, 7

[63] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 5

[64] Yonglong Tian, Olivier J Henaff, and Aäron 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. 1, 2, 6

[65] Rishabh Tiwari, Krishnateja Killamsetty, Rishabh Iyer, and Pradeep 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. 3

[66] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 17

[67] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018. 6, 15

[68] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Open-set recognition: A good closed-set classifier is all you need. *arXiv preprint arXiv:2110.06207*, 2021. 3

[69] Dan Wang and Yi Shang. A new active labeling method for deep learning. In *2014 International Joint Conference on Neural Networks)*, pages 112–119, 2014. 20

[70] Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2495–2504, 2021. 3, 21

[71] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020. 7, 17

[72] Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in data subset selection and active learning. In *International conference on machine learning*, pages 1954–1963. PMLR, 2015. 3, 4

[73] Xiu-Shen Wei, Yi-Zhe Song, Oisin Mac Aodha, Jianxin Wu, Yuxin Peng, Jinhui Tang, Jian Yang, and Serge Belongie. Fine-grained image analysis with deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 1

[74] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. 2010. 1, 4, 7, 15

[75] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua 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. 8

[76] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 5

[77] Bangpeng Yao, Xiaoye Jiang, Aditya Khosla, Andy Lai Lin, Leonidas Guibas, and Li 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. 4, 15

[78] Jaehong Yoon, Divyam Madaan, Eunho Yang, and Sung Ju Hwang. Online coreset selection for rehearsal-based continual learning. *arXiv preprint arXiv:2106.01085*, 2021. 3

[79] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021. 3

[80] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34:18408–18419, 2021. 20

[81] Zhun Zhong, Linchao Zhu, Zhiming Luo, Shaozi Li, Yi Yang, and Nicu 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. 3

[82] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 6, 15