Understanding Cross-Domain Few-Shot Learning Based on Domain Similarity and Few-Shot Difficulty

Jaehoon Oh[*]
KAIST DS
Daejeon, South Korea
jhoon.oh@kaist.ac.kr

Sungnyun Kim[*]
KAIST AI
Seoul, South Korea
ksn4397@kaist.ac.kr

Namgyu Ho[*]
KAIST AI
Seoul, South Korea
itsnamgyu@kaist.ac.kr

Jin-Hwa Kim
NAVER AI Lab, SNU AIIS
Seongnam, South Korea
j1nhwa.kim@navercorp.com

Hwanjun Song[†]
NAVER AI Lab
Seongnam, South Korea
hwanjun.song@navercorp.com

Se-Young Yun[†]
KAIST AI
Seoul, South Korea
yunseyoung@kaist.ac.kr

Abstract

Cross-domain few-shot learning (CD-FSL) has drawn increasing attention for handling large differences between the source and target domains–an important concern in real-world scenarios. To overcome these large differences, recent works have considered exploiting small-scale unlabeled data from the target domain during the pre-training stage. This data enables self-supervised pre-training on the target domain, in addition to supervised pre-training on the source domain. In this paper, we empirically investigate which pre-training is preferred based on *domain similarity* and *few-shot difficulty* of the target domain. We discover that the performance gain of self-supervised pre-training over supervised pre-training becomes large when the target domain is dissimilar to the source domain, or the target domain itself has low few-shot difficulty. We further design two pre-training schemes, mixed-supervised and two-stage learning, that improve performance. In this light, we present six findings for CD-FSL, which are supported by extensive experiments and analyses on three source and eight target benchmark datasets with varying levels of domain similarity and few-shot difficulty. Our code is available at `https://github.com/sungnyun/understanding-cdfsl`.

1 Introduction

Few-shot learning (FSL) is a machine learning paradigm to learn novel classes from *few* examples with supervised information [66, 69]. Unlike standard supervised learning, a model is pre-trained on the source dataset consisting of *base* classes and then transferred into the target dataset consisting of *novel* classes with few examples, where base and novel classes are disjoint but share similar data domains. However, this underlying assumption is not applicable to real-world scenarios because source (base classes) and target (novel classes) domains are different in general. This leads to poor generalization performance because of the change in feature and label distributions, posing a new challenge in FSL [24, 64].

In this regard, *cross-domain few-shot learning* (CD-FSL) is gaining immense attention with the BSCD-FSL (Broader Study of CD-FSL) benchmark [24], which enables us to evaluate real-world few-shot learning tasks. The BSCD-FSL benchmark is a collection of four different datasets with varying

[*] Equal contribution.
[†] Corresponding authors.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

levels of domain similarity to large-scale natural image collections, such as ImageNet [11]. Although there are two possible directions for FSL, meta-learning [17, 35, 64] and transfer learning [4, 12, 62], transfer learning has been reported to have higher performance than meta-learning approaches in cross-domain scenarios. Therefore, following the transfer learning pipeline, recent studies for CD-FSL [47, 30] have mainly focused on improving the pre-training phase before fine-tuning on the target labeled data with novel classes.

To address the challenge of different domains, there have been recent efforts to leverage *unlabeled* examples from the target domain as auxiliary data for pre-training, in addition to labeled examples from the source domain. For example, along with the supervised cross-entropy loss, STARTUP [47] and Dynamic Distillation [30] incorporate distillation loss and FixMatch-like loss for self-supervision, respectively. In other words, they develop sophisticated pre-training approaches that can leverage source and target data together. However, the basic pre-training schemes, supervised learning (SL) on the source domain and self-supervised learning (SSL) on the target domain, have not been thoroughly studied with respect to their pros and cons in CD-FSL.

In this paper, we establish an *empirical understanding* of the effectiveness of SL and SSL for a better pre-training process in CD-FSL. To this end, we begin by scrutinizing an opposing finding of the previous works [47, 30]. We discover that readily available SSL methods, *e.g.*, SimCLR [3], can outperform the standard SL method for pre-training, even when the amount of unlabeled target data for SSL is much smaller than that of labeled source data for SL (see Section 4).

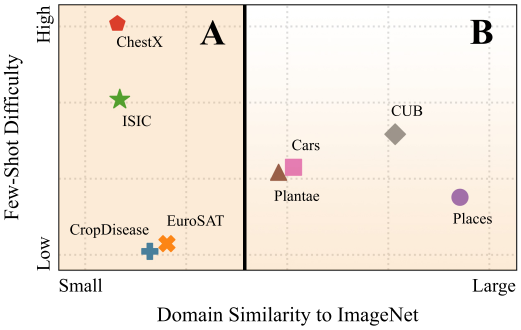


Figure 1: Our insights on the pre-training approaches. (A) SSL is preferred for all datasets with small domain similarity. (B) SL is preferred for high-difficulty datasets with large domain similarity. The formal definitions of similarity and difficulty are explained in Section 3.2.

Next, we investigate why the CD-FSL performance depends on different pre-training schemes using the two properties: *domain similarity* and *few-shot difficulty*. **Domain Similarity** is the similarity between the source and target domains, which is known to affect the transferability of the source domain features into the target domain [10, 36]. However, we find it insufficient to identify the effectiveness of SL and SSL based on domain similarity alone. To solve this conundrum, we propose **Few-Shot Difficulty** as a measure of the inherent hardness of a dataset, based on the upper bound of empirical FSL performance. By grounding our analysis on these two metrics, we discover coherent insights on CD-FSL pre-training schemes, depicted in Figure 1. Our analyses point to two conclusions: (A) When domain similarity is small, SSL is preferred due to the limited transferability of source information. On the other hand, (B) SL is preferred when domain similarity is large and few-shot difficulty is high, because supervision from the source dataset achieves stronger performance compared to self-supervision on difficult target data (see Section 5).

Finally, to investigate whether SL and SSL can synergize, we design a joint learning scheme using both SL and SSL, coined as *mixed-supervised learning* (MSL). It is observed that SL and SSL can synergize when they have similar performances. Furthermore, we extend our analysis to a *two-stage* pre-training scheme, motivated by recent works on CD-FSL [47, 30]. We observe that this generally improves performance because the SL pre-trained model provides a good initialization for the second phase of pre-training (see Section 6).

2 Related Work

2.1 Few-Shot Learning (FSL)

FSL has been mainly studied in the literature based on two approaches, meta-learning and transfer learning. In the meta-learning approach, a model is trained on the meta-train set (*i.e.*, source data) in an episodic manner, mimicking the evaluation procedure, such that fast adaptation is possible on the meta-test set (*i.e.*, few-shot target data). This family of approaches include learning a good initialization [38, 17, 18, 48, 43], learning a metric space [66, 54, 58], and learning an update rule [50, 1, 19]. By contrast, in the transfer-based approach [4, 12, 62], a model is pre-trained on the source dataset following the general supervised learning procedure in a mini-batch manner, and subsequently fine-tuned on the target dataset for evaluation.

2.2 Cross-Domain Few-Shot Learning (CD-FSL)

CD-FSL has addressed a more challenging and realistic scenario where the source and target domains are dissimilar [24, 64]. Such a cross-domain setting makes it difficult to transfer source information into the target domain owing to large domain differences [44, 36, 40, 73]. In general, the most recent methods have been developed on top of the fine-tuning paradigm because this paradigm outperforms the traditional meta-learning approach such as FWT [24]. STARTUP [47] and Dynamic Distillation [30] are the two representative algorithms, and they suggested using small-scale unlabeled data from the target domain in pre-training such that a pre-trained model can be well-adaptable for the target domain. Specifically, both algorithms first train a teacher network with cross-entropy loss on labeled source data. Then, STARTUP trains the student network with cross-entropy loss on the source data together with two unsupervised losses on the target data: distillation loss [28] and self-supervised loss (*i.e.*, SimCLR [3]). Dynamic Distillation trains the student network with cross-entropy loss on labeled source data and KL loss based on FixMatch [55] on unlabeled target data.

2.3 Self-Supervised Learning (SSL)

SSL has attracted attention as a method of learning useful representations from unlabeled data [14, 13, 72, 46, 42]. When this field first emerged, hand-crafted pretext tasks, such as solving jigsaw puzzles [41] and predicting rotations [20], were designed and utilized for training. In recent times, there has been an effort to use contrastive loss, which enhances representation learning based on augmentation and negative samples [3, 61, 26, 2]. This contrastive loss encourages the alignment of positive pairs and uniformity of data distribution on the hypersphere [67]. This improves the transferability of a model by encouraging it to contain lower-level semantics compared to supervised approaches [31]. However, this advantage is conditional on the availability of numerous negative samples. To alleviate such constraint, non-contrastive approaches that do not use negative samples have been proposed [23, 5, 71, 63]. In our empirical study, we use two contrastive approaches, SimCLR [3] and MoCo [26], and two non-contrastive approaches, BYOL [23] and SimSiam [5]. The details of each algorithm are described in Appendix A.

For the completeness of our survey, we include prior works that address SSL for cross-domain and/or few-shot learning. Kim et al. [32] addressed self-supervised pre-training under label-shared cross-domain, while our setting does not share the label space between domains. Ericsson et al. [16] observed that SSL on the source data improves performance on the BSCD-FSL dataset. However, domain-specific SSL (*i.e.*, SSL on target data) was not addressed. Cole et al. [8] showed that adding data from different domains can lead to performance degradation when data is numerous. Phoo and Hariharan [47] and Islam et al. [30] argued that plain SSL methods struggle to outperform SL for CD-FSL. We investigate domain-specific SSL and demonstrate its superiority, which opposes the finding from previous studies.

3 Overview

We clarify the scope of our empirical study, propose formal definitions of domain similarity and few-shot difficulty, and describe experimental configurations. Table 1 summarizes the notations used in this paper.

Table 1: Summary of the notations.

Notation	Description
$\mathcal{D}_B, \mathcal{D}_N$	Source and target datasets, $\mathcal{D}_B \cap \mathcal{D}_N = \emptyset$
$\mathcal{C}_B, \mathcal{C}_N$	Base classes for $\mathcal{D}_B$ and novel classes for $\mathcal{D}_N$
$\mathcal{D}_U$ $(\subset \mathcal{D}_N)$	Unlabeled target data for SSL
$\mathcal{D}_L$ $(\subset \mathcal{D}_N)$	Labeled target data for evaluation, $\mathcal{D}_U \cap \mathcal{D}_L = \emptyset$
n, k	# classes and examples for n-way k-shot
$\mathcal{D}_S$ $(\subset \mathcal{D}_L)$	A support set with size nk for fine-tuning
$\mathcal{D}_Q$ $(\subset \mathcal{D}_L)$	A query set for evaluation, $\mathcal{D}_S \cap \mathcal{D}_Q = \emptyset$
f	A feature extractor (backbone network)
h_{sl}	A classification head for SL during pre-training
h_{ssl}	A projection head for SSL during pre-training
g	A classification head during fine-tuning

3.1 Scope of the Empirical Study

Our objective is to learn a feature extractor f on base classes $\mathcal{C}_B$ in source data $\mathcal{D}_B$, which can extract informative representations for novel classes

$\mathcal{C}_N$ in target data $\mathcal{D}_N$. Typically, a classifier g is fine-tuned and the model $g \circ f$ is evaluated using labeled target examples $\mathcal{D}_L$ ($\subset \mathcal{D}_N$) after pre-training f on the source data $\mathcal{D}_B$ under the condition that the base classes are largely different from the novel classes.

Following the recent literature [47, 30], we further assume that additional unlabeled data $\mathcal{D}_U$ ($\subset \mathcal{D}_N$) is available in the pre-training phase. We follow the split strategy used in Phoo and Hariharan [47], where 20% of the target data $\mathcal{D}_N$ is used as the unlabeled data $\mathcal{D}_U$ for pre-training. Note that the size of the unlabeled portion is very small (*e.g.*, only a few thousand examples) compared to large-scale datasets typically considered for self-supervised learning. In this problem setup, the pre-training phase of CD-FSL can be carried out based on *three* learning strategies:

- **Supervised Learning**: Let f and h_{sl} be the feature extractor and linear classifier for the base classes $\mathcal{C}_B$, respectively. Then, a model $h_{\mathsf{sl}} \circ f$ is pre-trained only for the labeled source data $\mathcal{D}_B$ by minimizing the standard cross-entropy loss ℓ_{ce} in a mini-batch manner,[3]

$$\mathcal{L}_{\mathsf{sl}}(f, h_{\mathsf{sl}}; \mathcal{D}_B) = \frac{1}{|\mathcal{D}_B|} \sum_{(x,y) \in \mathcal{D}_B} \ell_{\mathsf{ce}}(h_{\mathsf{sl}} \circ f(x), y). \tag{1}$$

- **Self-Supervised Learning**: Let h_{ssl} be the projection head. Then, a model $h_{\mathsf{ssl}} \circ f$ is pre-trained only for the unlabeled target data $\mathcal{D}_U$, which is much smaller than the labeled source data, by minimizing (non-)contrastive self-supervised loss ℓ_{self} (*e.g.*, NT-Xent),[1]

$$\mathcal{L}_{\mathsf{ssl}}(f, h_{\mathsf{ssl}}; \mathcal{D}_U) = \frac{1}{2|\mathcal{D}_U|} \sum_{x \in \mathcal{D}_U} \Big[\ell_{\mathsf{self}}\big(z_1; z_2; \{z^-\}\big) + \ell_{\mathsf{self}}\big(z_2; z_1; \{z^-\}\big) \Big] \tag{2}$$

$$\text{where} \;\; z_i = h_{\mathsf{ssl}} \circ f(A_i(x)),$$

  and $A_i(x)$ is the i-th augmentation of the same input x. This training loss forces z_1 to be similar to z_2 and dissimilar to the set of negative features $\{z^-\}$. In addition, there are non-contrastive SSL methods that do not rely on negative examples, *i.e.*, $\{z^-\} = \emptyset$. We provide a more detailed explanation of SSL losses, including multiple (non-)constrastive approaches in Appendix A.

- **Mixed-Supervised Learning**: MSL exploits labeled as well as unlabeled data from different domains simultaneously. MSL can be intuitively formulated by minimizing the interpolation of their losses in Eqs. (1) and (2),

$$\mathcal{L}_{\mathsf{msl}}(f, h_{\mathsf{sl}}, h_{\mathsf{ssl}}; \mathcal{D}_B, \mathcal{D}_U) = (1 - \gamma) \cdot \mathcal{L}_{\mathsf{sl}}(f, h_{\mathsf{sl}}; \mathcal{D}_B) + \gamma \cdot \mathcal{L}_{\mathsf{ssl}}(f, h_{\mathsf{ssl}}; \mathcal{D}_U), \tag{3}$$

  where $0 < \gamma < 1$ and the feature extractor f is hard-shared and trained through SL and SSL losses with a balancing hyperparameter γ. This can be a generalization of STARTUP and Dynamic Distillation, which use Eq. (3) in the second pre-training phase with a moderate modification after the typical pre-training phase using SL.

Our analysis focuses on pre-training and fine-tuning schemes due to the superiority of transfer-based methods over typical FSL algorithms such as MAML [17], which is shown in [24]. Based on the three learning strategies above, we conduct an empirical study to gain an in-depth understanding of their effectiveness in the pre-training phase, providing deep insight into the following questions:

1. Which is more effective for pre-training, using only SL or SSL? ▷ Section 4
2. How to apply domain similarity and few-shot difficulty to identify the more effective pre-training scheme between SL and SSL, for CD-FSL? ▷ Section 5
3. Can MSL, a combination of SL and SSL, as well as a two-stage scheme improve performance? ▷ Section 6

3.2 Domain Similarity and Few-Shot Difficulty

We present a procedure for estimating the two metrics on datasets, which are used to analyze the pre-training schemes. First, we use *domain similarity* introduced in [10], which is based on Earth Mover's Distance (EMD [52]) because the distance between the two domains can be considered as

[3] The batch loss on the entire data is used for ease of exposition.

the cost of *moving* images from one domain to the other in the transfer learning context [10, 36]. Further details on this metric, *e.g.*, advantages of EMD, are explained in Appendix D.

We can easily compute EMD using the retrieved sample representations.[4] We create the prototype vector $\mathbf{p}_i$, which is an averaged representation for all examples belonging to class i. Next, let $i \in \mathcal{C}_B$ and $j \in \mathcal{C}_N$ be a class in base (source) and novel (target) classes, respectively. Then, the domain similarity between the source and target data is formulated as

$$\mathrm{Sim}(\mathcal{D}_B, \mathcal{D}_N) = \exp\bigl(-\alpha \, \mathrm{EMD}(\mathcal{D}_B, \mathcal{D}_N)\bigr) \quad \text{where } \mathrm{EMD}(\mathcal{D}_B, \mathcal{D}_N) = \frac{\sum_{i \in \mathcal{C}_B, \, j \in \mathcal{C}_N} f_{i,j} \, d_{i,j}}{\sum_{i \in \mathcal{C}_B, \, j \in \mathcal{C}_N} f_{i,j}}$$

$$\text{subject to } \; f_{i,j} \geq 0, \quad \sum_{i \in \mathcal{C}_B, \, j \in \mathcal{C}_N} f_{i,j} = 1, \; \sum_{j \in \mathcal{C}_N} f_{i,j} \leq \frac{|\mathcal{D}_B[i]|}{|\mathcal{D}_B|}, \; \sum_{i \in \mathcal{C}_B} f_{i,j} \leq \frac{|\mathcal{D}_N[j]|}{|\mathcal{D}_N|}, \tag{4}$$

where $d_{i,j} = ||\mathbf{p}_i - \mathbf{p}_j||_2$; $f_{i,j}$ is the optimal flow between $\mathbf{p}_i$ and $\mathbf{p}_j$ subject to the constraints for EMD; $\mathcal{D}[i]$ returns all examples of the specified class i in $\mathcal{D}$; and α is typically set to 0.01 [10]. Namely, EMD can be interpreted as the weighted distance of all combinations between the base and novel classes. The larger similarity indicates that source and target data share similar domains.

Next, we propose *few-shot difficulty*, which quantifies the difficulty of a dataset based on the empirical upper bound of few-shot performance in our problem setup, regardless of its relationship to the source dataset. To capture the upper bound of FSL performance, we use 20% of the target dataset as labeled data to pre-train the model in a supervised manner. Then, the pre-trained model is evaluated on the remaining unseen target data for the 5-way k-shot classification task.[5] As the generalization capability indicates the hardness [56], the classification accuracy for unseen data is used and converted into the few-shot difficulty using an exponential function with a hyperparameter β (the default value is 0.01),

$$\mathrm{Diff}(\mathcal{D}, k) = \exp(-\beta \, \mathrm{Acc}(\mathcal{D}, k)), \tag{5}$$

where $\mathrm{Acc}(\mathcal{D}, k)$ returns the average of 5-way k-shot classification accuracy over 600 episodes for the given data $\mathcal{D}$. Note that in our paper, k is set to 5, but the order of difficulty is the same regardless of k. High few-shot difficulty implies that the achievable accuracy is low even when there is no domain difference between pre-training and evaluation.

3.3 Experimental Configurations

Cross-Domain Datasets. We use ImageNet, tieredImageNet, and miniImageNet as source datasets for generality. Regarding the target domain, we prepare eight datasets with varying domain similarity and few-shot difficulty; domain similarity is computed based on both the source and target datasets, while few-shot difficulty is computed based on the target dataset. To summarize their order in Figure 1, ***domain similarity*** *to ImageNet: Places > CUB > Cars > Plantae > EuroSAT > CropDisease > ISIC > ChestX*, and ***few-shot difficulty****: ChestX > ISIC > CUB > Cars > Plantae > Places > EuroSAT > CropDisease*. For instance, Places data has the largest domain similarity to ImageNet, while ChestX has the highest few-shot difficulty. Appendix B provides the details of each dataset. The detailed values for domain similarity and few-shot difficulty are reported in Appendix D and E, respectively. These are visualized in Appendix F. We also provide the results on the case when source and target domains are the same, *i.e.*, the standard FSL setting, in Appendix O.

Evaluation Pipeline. We follow the standard evaluation pipeline of CD-FSL [24]. The evaluation process is performed in an episodic manner, where each episode represents a distinct few-shot task. Each episode is comprised of a support set $\mathcal{D}_S$ and a query set $\mathcal{D}_Q$, which are sampled from the entire labeled target data $\mathcal{D}_L$. The support set $\mathcal{D}_S$ and query set $\mathcal{D}_Q$ consist of n classes that are randomly selected among the entire set of novel classes $\mathcal{C}_N$. For the n-way k-shot setting, k examples are randomly drawn from each class for the support set $\mathcal{D}_S$, while k_q (typically 15) examples for the query set $\mathcal{D}_Q$. Thus, the support and query set are defined as,

$$\mathcal{D}_S = \{(x_i^s, y_i^s)\}_{i=1}^{n \times k} \text{ and } \mathcal{D}_Q = \{(x_i^q, y_i^q)\}_{i=1}^{n \times k_q}. \tag{6}$$

[4] To extract the representation of images, we follow Li et al. [36] by using a large model trained on a large-scale dataset, ResNet101 pre-trained on ImageNet. Note that Cui et al. [10] used JFT dataset [57], which is not released for public use. Furthermore, we measure domain similarity using different feature extractors, described in Table 6 of Appendix D. Our analysis is consistent regardless of the feature extractor used.

[5] We use a few-shot learning task instead of classification on the entire data, preventing the performance from being distorted by other factors, such as data imbalance and the number of classes.

Table 2: 5-way k-shot CD-FSL performance (%) of the models pre-trained by SL and SSL. We report the average accuracy and its 95% confidence interval over 600 few-shot episodes. B and S indicate base and strong augmentation, respectively. The best accuracy is marked in bold for each backbone.

Source Data	Pre-train Scheme	Method	Aug.	EuroSAT		CropDisease		ISIC		ChestX	
				k=1	k=5	k=1	k=5	k=1	k=5	k=1	k=5
ImageNet	SL	Default	B	66.14±.83	84.73±.51	74.18±.82	92.81±.45	31.11±.55	44.10±.58	22.48±.39	25.51±.44
tiered ImageNet	SL	Default	B	61.81±.88	79.87±.67	66.82±.90	87.19±.59	30.35±.60	41.67±.55	22.34±.38	25.08±.45
			S	60.07±.88	79.95±.66	65.70±.94	86.34±.60	29.75±.56	40.60±.58	22.11±.42	25.20±.41
-	SSL	SimCLR	B	70.37±.86	87.80±.46	90.94±.69	97.44±.29	34.13±.69	44.37±.66	21.41±.41	25.05±.42
			S	**84.30**±.73	**94.12**±.32	**91.00**±.76	**97.46**±.34	**36.39**±.66	47.85±.65	21.55±.41	25.26±.44
		MoCo	B	51.21±.93	68.19±.74	70.22±.95	87.11±.60	27.79±.53	36.60±.59	21.44±.43	24.28±.43
			S	69.11±.98	81.01±.73	80.08±.97	92.48±.52	29.54±.59	39.28±.58	21.74±.42	24.58±.44
		BYOL	B	60.98±.91	84.88±.56	81.58±.78	96.82±.27	35.31±.64	**49.26**±.64	22.65±.42	**28.80**±.49
			S	66.16±.86	87.83±.48	85.77±.73	96.93±.30	34.53±.62	47.59±.63	**22.75**±.41	28.36±.46
		SimSiam	B	44.06±.86	61.03±.72	75.36±.82	92.31±.44	26.99±.52	35.68±.52	22.02±.41	26.06±.46
			S	70.80±.88	85.10±.57	84.72±.80	96.05±.36	30.17±.56	39.51±.55	22.17±.40	26.56±.46

(a) ResNet18 is used as a backbone.

Source Data	Pre-train Scheme	Method	Aug.	EuroSAT		CropDisease		ISIC		ChestX	
				k=1	k=5	k=1	k=5	k=1	k=5	k=1	k=5
mini ImageNet	SL	Default	B	64.03±.91	82.72±.59	73.38±.87	91.53±.49	30.68±.58	41.77±.59	22.64±.40	26.26±.45
			S	65.03±.88	84.00±.56	72.82±.87	91.32±.49	29.91±.54	40.84±.56	22.88±.42	27.01±.44
-	SSL	SimCLR	B	66.77±.84	86.39±.48	89.33±.66	96.82±.32	33.32±.63	44.50±.64	22.26±.42	24.34±.42
			S	**79.50**±.78	**92.36**±.37	**89.49**±.74	**97.24**±.33	**34.90**±.64	**46.76**±.61	21.97±.41	25.62±.43
		MoCo	B	48.70±.92	66.85±.72	68.77±.92	87.67±.57	27.76±.54	38.03±.57	21.55±.42	24.48±.44
			S	76.20±.89	89.54±.46	80.19±.99	93.41±.53	30.20±.55	41.14±.57	21.64±.40	24.49±.43
		BYOL	B	61.18±.82	83.11±.57	80.50±.75	94.85±.35	33.02±.62	46.72±.65	22.90±.41	27.40±.47
			S	66.45±.80	86.55±.50	80.10±.76	94.53±.41	33.50±.59	45.99±.63	23.11±.42	27.71±.44
		SimSiam	B	44.57±.82	63.67±.67	82.83±.73	95.37±.34	30.74±.60	41.28±.62	22.76±.42	27.50±.47
			S	71.66±.88	85.21±.59	81.25±.77	95.13±.37	31.80±.59	41.44±.59	**23.22**±.41	**27.83**±.46

(b) ResNet10 is used as a backbone.

For evaluation, a classifier g is fine-tuned on the support set $\mathcal{D}_S$, using features extracted from the fixed pre-trained backbone f. Note that g is for the evaluation purpose different from h_{sl} and h_{ssl} for pre-training. The fine-tuned model $g \circ f$ is then tested on the query set $\mathcal{D}_Q$. We set $n = 5$ and $k = \{1, 5\}$, and the accuracy is averaged over 600 episodes following convention [24, 47].

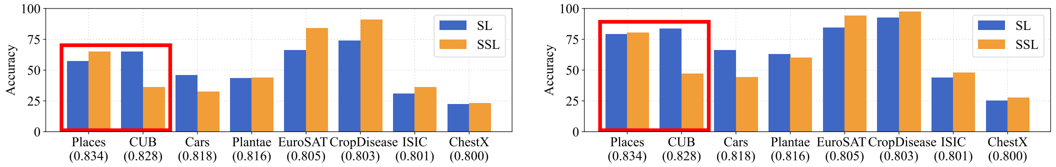
Implementation. We use different backbone networks depending on the source data. For ImageNet and tieredImageNet, ResNet18 is used as the backbone, while ResNet10 is used for miniImageNet. For ResNet18 pre-trained on ImageNet with SL, we use the model provided by PyTorch [45] repository. This setup is exactly the same for all pre-training schemes. Additional details on the training setup are provided in Appendix C.

4 Supervised Learning on Source vs. Self-Supervised Learning on Target

We begin by investigating the superiority of SSL on the target dataset over SL on the source dataset for pre-training. We compare the CD-FSL performance of pre-trained models using four representative (widely cited) SSL methods (SimCLR [3], MoCo [26], BYOL [23], and SimSiam [5]) with that of an SL method (Default) in Table 2. Four different domain datasets from the BSCD-FSL benchmarks (EuroSAT, CropDisease, ISIC, and ChestX) are used as target data. Table 2 provides empirical evidence of the findings in this section. Recent literature has reported that SSL pre-training does *not* work better than SL for the CD-FSL task because of insufficient unlabeled examples in the target domain [47, 30]. However, our observation contradicts this previous finding.

OBSERVATION 4.1. *SSL on the target domain can achieve remarkably higher performance over SL on the labeled source domain, even with small-scale (i.e., a few thousand) unlabeled target data.*

EVIDENCE. SSL methods are observed to outperform SL in most cases, even though SSL does not leverage source data for pre-training. In particular, SSL methods show much higher performance compared to the model pre-trained on the entire ImageNet dataset, which has more than 1.2M training examples. This leads to the conclusion that SSL on the target domain can be better than SL on the source domain for CD-FSL pre-training. In other words, unlabeled target data available at the pre-training phase is worth more than labeled source data, even if the unlabeled target data is much smaller (*e.g.*, 8k examples for CropDisease) than the labeled source data. In Appendix G, we show that SSL can outperform SL using even smaller portions of unlabeled target data.



(a) 5-way 1-shot

(b) 5-way 5-shot

Figure 2: 5-way k-shot CD-FSL performance (%) of SL and SSL according to domain similarity (values in x-axis), with ImageNet source data. The red box shows that SL outperforms SSL in the second largest domain similarity, while SSL outperforms SL in the largest domain similarity.

OBSERVATION 4.2. *SSL achieves significant performance gains with strong data augmentation.*

EVIDENCE. In addition, the results in Table 2 provide the performance sensitivity to data augmentation. For this study, two types of augmentation are used: (1) base augmentation from [3], which consists of random resized crop, color jitter, horizontal flip, and normalization, and (2) strong augmentation from [30], which adds Gaussian blur and random gray scale to the base (see the detail of the augmentations in Appendix C). With strong augmentation, SSL methods exhibit significant performance gains of up to 27.50%p compared to base augmentation, *i.e.*, MoCo on EuroSAT in Table 2(b). However, SL does not benefit from strong augmentation as SSL does. This has also been observed in the literature [3]. Therefore, the performance of SSL can be further improved for CD-FSL if more suitable augmentation is applied. Based on this observation, we use strong augmentation for SSL as the default setup in the rest of our paper.

Meanwhile, the superiority among SSL algorithms varies with target dataset. In Table 2, we observe that SimCLR performs best in EuroSAT and CropDisease, while in ISIC, SimCLR and BYOL both perform well. For ChestX, BYOL and SimSiam show good performance. The SSL methods can be categorized into two groups: contrastive (SimCLR and MoCo) and non-contrastive (BYOL and SimSiam). For the rest of our paper, we focus our analysis on SimCLR and BYOL, which are representative methods from each group with robust performance. The results for other target datasets are presented in Appendix H.

5 Closer Look at Domain Similarity and Few-Shot Difficulty

We investigate why the CD-FSL performance depends on different pre-training schemes, *i.e.,* SL or SSL, based on the two metrics: domain similarity and few-shot difficulty in Eqs. (4) and (5). We analyze the relationship between few-shot performance and the two metrics on various target datasets and provide insights for developing a more effective pre-training approach.

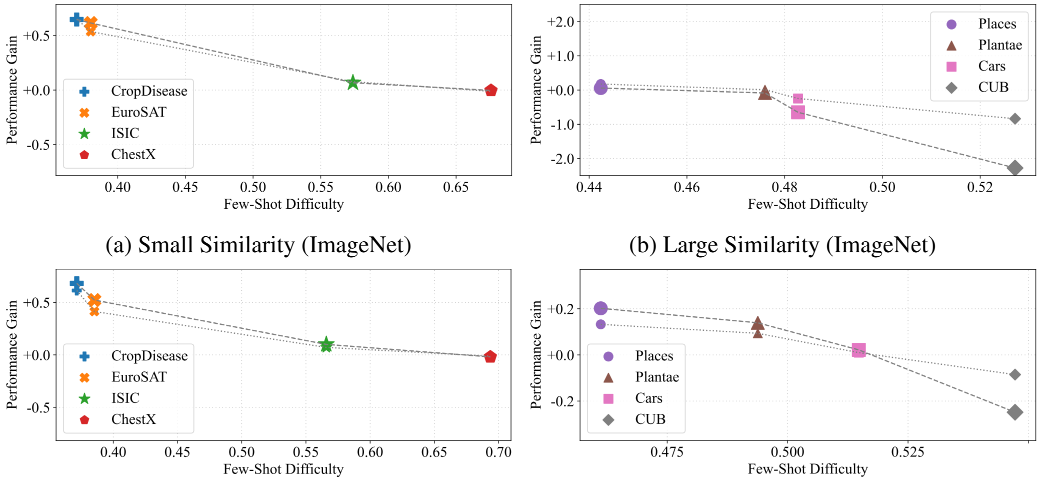
Including BSCD-FSL, we consider four additional datasets from different domains: Places, CUB, Cars, and Plantae. Note that these additional datasets are known to be more similar to ImageNet than the BSCD-FSL datasets are [16], and our estimated similarity shows the same trend. We mainly use ImageNet as the source dataset to make our analysis more reliable. We analyze their domain similarity and few-shot difficulty and display them in Figure 1, where ImageNet is used as source data for domain similarity. In this section, to select the SSL method for each dataset, we use SimCLR for all datasets except ChestX, where BYOL is used, based on the performance observed in Section 4.

5.1 Domain Similarity

Figure 2 shows the CD-FSL performance of the pre-trained models using SL and SSL for eight target datasets with varying domain similarity, where all the datasets are sorted by domain similarity. A common belief about domain similarity is that, as domain similarity increases, it is more beneficial for pre-training to use a large amount of labeled source data [10, 36, 16]. Our analysis shows that this belief is partially true.

OBSERVATION 5.1. *SL does not consistently benefit from large domain similarity.*

EVIDENCE. For the aforementioned belief to be true, the performance gain of SL over SSL should be greater as domain similarity increases. However, although SL outperforms SSL in the CUB dataset



(a) Small Similarity (ImageNet)

(b) Large Similarity (ImageNet)

(c) Small Similarity (miniImageNet)

(d) Large Similarity (miniImageNet)

Figure 3: 5-way k-shot performance gain of SSL over SL for the two dataset groups according to the few-shot difficulty (small markers: k=1, large markers: k=5). Results are shown for two source datasets: ImageNet and miniImageNet, each with their corresponding backbones.

with the second largest domain similarity, in the Places dataset with the largest domain similarity, SSL rather exhibits higher CD-FSL accuracy than SL (see the red box in Figure 2). Furthermore, in the ChestX dataset with the smallest domain similarity, SL and SSL have similar performances. These results demonstrate that unlike prior belief, large domain similarity does not always guarantee the superiority of SL. In other words, there is an inconsistency that cannot be explained solely by domain similarity, and we explore why this inconsistency occurs by taking few-shot difficulty into account.

5.2 Few-Shot Difficulty

In this sense, we study the impact of few-shot difficulty by categorizing the eight datasets into two groups: one with small domain similarity (*i.e.*, BSCD-FSL) and another with large domain similarity (*i.e.*, other datasets). Figure 3 shows the performance gain of SSL over SL for datasets with varying few-shot difficulty for each group. The performance gain of SSL over SL is defined as $(\text{Error}_{\mathsf{sl}} - \text{Error}_{\mathsf{ssl}})/\text{Error}_{\mathsf{sl}}$, which indicates the relative improvement of the classification error.

<u>OBSERVATION</u> 5.2. *Performance gain of SSL over SL becomes greater at smaller domain similarity or lower few-shot difficulty.*

EVIDENCE. For both groups, the performance gain of SSL over SL becomes greater as few-shot difficulty decreases. In particular, the performance gain is the greatest on the CropDisease and Places datasets with the lowest few-shot difficulty in each group, while the performance gain is the least on the ChestX and CUB datasets with the highest few-shot difficulty in each group. For the target data with higher few-shot difficulty, *it may not be easy to learn discriminative representations by solely using SSL without label supervision*.

Meanwhile, comparing the two groups (BSCD-FSL vs. other datasets), it is observed that the performance gain of SSL over SL is significantly worse for the group with large domain similarity. Namely, the performance gain is near or less than zero when domain similarity is large because features learned from SL with label supervision can be better transferred. Note that the negative value of performance gain means that SL outperforms SSL. Furthermore, the performance gain is closely related to the source dataset size for the datasets with large similarity (see Figures 3(b) and 3(d)). For instance, on the CUB dataset, the performance gain (k=5) is -2.276 and -0.249 for ImageNet and miniImageNet, respectively. However, when domain similarity is small (see Figures 3(a) and 3(c)), the source dataset size does not significantly affect the performance gain of SSL over SL.

In summary, we first conclude that SSL is advantageous to SL when the target domain is extremely dissimilar to the source domain (*i.e.*, the performance gain is greater than 0), which is in line with

Table 3: 5-way 5-shot CD-FSL performance (%) of the models pre-trained by SL, SSL, and MSL including their two-stage versions. ResNet18 is used as the backbone model, and ImageNet is used as the source data for SL and MSL. The balancing coefficient γ in Eq. (3) of MSL is set to be 0.875. Datasets are grouped by domain similarity and sorted by few-shot difficulty in ascending order in each group (CropDisease < ChestX | Places < CUB). The best results are marked in bold.

	Pre-train Scheme	Method	Small Similarity				Large Similarity			
			CropDisease	EuroSAT	ISIC	ChestX	Places	Plantae	Cars	CUB
Single-Stage	SL	Default	92.81±.45	84.73±.51	44.10±.58	25.51±.44	79.22±.64	63.21±.82	**66.38**±.80	**83.93**±.66
	SSL	SimCLR	**97.46**±.34	**94.12**±.32	47.85±.65	25.26±.44	80.43±.61	60.07±.84	44.55±.74	47.36±.79
		BYOL	96.93±.30	87.83±.48	47.59±.63	28.36±.46	72.47±.63	61.02±.82	48.56±.76	51.31±.78
	MSL	SimCLR	96.50±.35	90.11±.40	45.38±.63	26.05±.44	**82.56**±.58	64.76±.83	51.84±.79	64.53±.80
		BYOL	96.74±.31	90.82±.40	**49.14**±.70	**29.58**±.47	81.27±.59	**67.39**±.81	46.76±.73	69.67±.82

(a) Performance comparison for single-stage schemes.

	Pre-train Scheme	Method	CropDisease	EuroSAT	ISIC	ChestX	Places	Plantae	Cars	CUB
Two-Stage	SL→SSL	SimCLR	**97.88**±.30	**95.28**±.27	48.38±.60	25.25±.44	84.40±.53	66.35±.82	51.31±.84	57.11±.88
		BYOL	97.58±.26	91.82±.39	49.32±.63	28.27±.48	78.87±.60	67.83±.82	54.70±.84	60.60±.82
	SL→MSL	SimCLR	97.49±.30	91.70±.35	47.43±.62	26.24±.44	**85.76**±.52	69.24±.81	58.97±.82	81.51±.72
		BYOL	97.09±.31	90.89±.40	**50.72**±.67	**30.20**±.48	83.29±.55	**74.16**±.77	68.87±.80	84.34±.67
	SL→MSL+	STARTUP	96.06±.33	89.70±.41	46.02±.59	27.24±.46	85.00±.52	69.40±.84	68.43±.82	**89.60**±.55
		DynDistill	97.60±.35	92.28±.46	50.06±.86	29.65±.67	82.22±.81	71.49±1.06	**69.45**±1.12	86.54±1.88

(b) Performance comparison for two-stage schemes.

Observation 4.1. This implies supervision with a huge amount of source data cannot overcome domain differences. However, when domain similarity is large, the few-shot difficulty must be considered to determine a better strategy between SSL and SL. Namely, SL becomes more preferable as few-shot difficulty increases due to the benefits from supervision on the source dataset. The same trend is observed when tieredImageNet is used as the source dataset (Appendix I).

6 Advanced Scheme: MSL and Two-Stage

In this section, we further study SL and SSL in a more advanced scheme from the domain similarity and few-shot difficulty perspective, in line with previous observations. We first investigate whether SL and SSL can synergize by studying MSL. Next, we analyze the two-stage pre-training scheme used in recent works [47, 30].

6.1 Can SL and SSL Synergize?

To identify whether SL and SSL can complement each other, we first consider a mixed-loss pre-training scheme, MSL, described in Eq. (3). We define that synergy between SL and SSL occurs when MSL is superior to both SL and SSL. Table 3(a) summarizes the performance of the models under each pre-training scheme on eight target datasets, grouped by their domain similarity (BSCD-FSL vs. other datasets) and then sorted by the few-shot difficulty in ascending order. In MSL, the hyperparameter γ is set to be 0.875 found by a grid search, detailed in Appendix J.

OBSERVATION 6.1. *SL and SSL can synergize when SL and SSL have similar performances.*

EVIDENCE. In Table 3(a), it is observed that SL and SSL can synergize (*i.e.*, MSL > SL, SSL) on four datasets: ISIC, ChestX, Places, and Plantae. SL and SSL have similar performances on these datasets, as shown by the large markers (k=5) in Figures 3(a) and 3(b). MSL can learn diverse features, owing to differences in training domains (i.e, source vs. target) and learning frameworks (i.e., supervised vs. unsupervised), which allows for synergy [16, 37, 22, 21]. However, when either SL or SSL significantly outperforms the other, MSL does not perform best. In addition, MSL performance can be improved further in the large similarity group by emphasizing the SL component through a larger batch size (Appendix K).

6.2 Extension to Two-Stage Approach

We extend the single-stage to two-stage approaches, extracting more sophisticated target representations. In two-stage pre-training, a model is pre-trained *in prior* with labeled source data in the first phase and further trained through SSL or MSL in the second phase, *i.e.*, SL → SSL or SL → MSL. This pipeline has been adopted by recent algorithms, such as STARTUP [47] and DynDistill [30], but they additionally maintain an extra network or incorporate the knowledge distillation in the second phase, *i.e.*, SL → MSL^+. Table 3(b) summarizes the CD-FSL performance of two-stage schemes.

OBSERVATION 6.2. *Two-stage pre-training schemes are better than their single-stage counterparts.*

EVIDENCE. Two-stage pre-training approaches generally achieve much higher performance than their single-stage counterparts, *i.e.*, SL → SSL outperforms SSL, and SL → MSL outperforms MSL. When SL is used separately in the first phase, it appears to provide a good initialization for the second phase because a converged extractor on the source data is better than a random extractor [40]. Also, the benefit of the two-stage pre-training is significant when domain similarity is large. This observation is promising for practitioners because pre-trained models on ImageNet or bigger datasets are readily accessible. In addition, our simple two-stage methods, without any additional techniques, are shown to achieve comparable performance to the meticulously designed two-stage approaches such as STARTUP, even though our main goal is analysis of basic pre-training methods. Appendix L summarizes the full results including meta-learning based algorithms.

7 Conclusion

We established a thorough empirical understanding of CD-FSL. Our work is a pioneering study that unveils hidden findings in the empirical use of CD-FSL. We believe it can inspire subsequent studies like theoretical analysis, which our paper did not cover. In particular, we focused on the effectiveness of SL, SSL, and MSL, which can be realized with single- and two-stage pre-training schemes. We (1) observed that their performances are closely related to domain similarity between the source and target datasets and few-shot difficulty of the target dataset, and (2) proposed how they can be effectively combined for pre-training. Through our empirical study, we presented six findings that have been either misunderstood or unexplored. To justify all the findings, extensive experiments were conducted on benchmarks with varying degrees of domain similarity and few-shot difficulty.

Acknowledgements

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] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas. Learning to learn by gradient descent by gradient descent. In *NeurIPS*, 2016.

[2] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

[3] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[4] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang. A closer look at few-shot classification. In *ICLR*, 2019.

[5] X. Chen and K. He. Exploring simple siamese representation learning. In *CVPR*, 2021.

[6] X. Chen, H. Fan, R. Girshick, and K. He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[7] N. Codella, V. Rotemberg, P. Tschandl, M. E. Celebi, S. Dusza, D. Gutman, B. Helba, A. Kalloo, K. Liopyris, M. Marchetti, et al. Skin lesion analysis toward melanoma detection 2018:

A challenge hosted by the international skin imaging collaboration (isic). *arXiv preprint arXiv:1902.03368*, 2019.

[8] E. Cole, X. Yang, K. Wilber, O. Mac Aodha, and S. Belongie. When does contrastive visual representation learning work? *arXiv preprint arXiv:2105.05837*, 2021.

[9] T. M. Cover. *Elements of information theory*. John Wiley & Sons, 1999.

[10] Y. Cui, Y. Song, C. Sun, A. Howard, and S. Belongie. Large scale fine-grained categorization and domain-specific transfer learning. In *CVPR*, 2018.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[12] G. S. Dhillon, P. Chaudhari, A. Ravichandran, and S. Soatto. A baseline for few-shot image classification. In *ICLR*, 2020.

[13] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.

[14] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. Riedmiller, and T. Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(9):1734–1747, 2015.

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

[16] L. Ericsson, H. Gouk, and T. M. Hospedales. How well do self-supervised models transfer? In *CVPR*, 2021.

[17] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

[18] C. Finn, K. Xu, and S. Levine. Probabilistic model-agnostic meta-learning. *arXiv preprint arXiv:1806.02817*, 2018.

[19] S. Flennerhag, A. A. Rusu, R. Pascanu, F. Visin, H. Yin, and R. Hadsell. Meta-learning with warped gradient descent. In *ICLR*, 2020.

[20] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018.

[21] R. Gontijo-Lopes, Y. Dauphin, and E. D. Cubuk. No one representation to rule them all: Overlapping features of training methods. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=BK-4qbGgIE3`.

[22] T. G. Grigg, D. Busbridge, J. Ramapuram, and R. Webb. Do self-supervised and supervised methods learn similar visual representations? *arXiv preprint arXiv:2110.00528*, 2021.

[23] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[24] Y. Guo, N. C. Codella, L. Karlinsky, J. V. Codella, J. R. Smith, K. Saenko, T. Rosing, and R. Feris. A broader study of cross-domain few-shot learning. In *ECCV*, 2020.

[25] 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.

[26] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

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

[28] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In *NeurIPSW*, 2015.

[29] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[30] A. Islam, C.-F. Chen, R. Panda, L. Karlinsky, R. Feris, and R. J. Radke. Dynamic distillation network for cross-domain few-shot recognition with unlabeled data. *arXiv preprint arXiv:2106.07807*, 2021.

[31] A. Islam, C.-F. Chen, R. Panda, L. Karlinsky, R. Radke, and R. Feris. A broad study on the transferability of visual representations with contrastive learning. *arXiv preprint arXiv:2103.13517*, 2021.

[32] D. Kim, K. Saito, T.-H. Oh, B. A. Plummer, S. Sclaroff, and K. Saenko. Cds: Cross-domain self-supervised pre-training. In *ICCV*, 2021.

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

[34] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In *ICCVW*, 2013.

[35] K. Lee, S. Maji, A. Ravichandran, and S. Soatto. Meta-learning with differentiable convex optimization. In *CVPR*, 2019.

[36] H. Li, P. Chaudhari, H. Yang, M. Lam, A. Ravichandran, R. Bhotika, and S. Soatto. Rethinking the hyperparameters for fine-tuning. *arXiv preprint arXiv:2002.11770*, 2020.

[37] H. Liu, J. Z. HaoChen, A. Gaidon, and T. Ma. Self-supervised learning is more robust to dataset imbalance. *arXiv preprint arXiv:2110.05025*, 2021.

[38] D. Maclaurin, D. Duvenaud, and R. Adams. Gradient-based hyperparameter optimization through reversible learning. In *ICML*, 2015.

[39] S. P. Mohanty, D. P. Hughes, and M. Salathé. Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7:1419, 2016.

[40] B. Neyshabur, H. Sedghi, and C. Zhang. What is being transferred in transfer learning? *arXiv preprint arXiv:2008.11687*, 2020.

[41] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, pages 69–84. Springer, 2016.

[42] M. Noroozi, H. Pirsiavash, and P. Favaro. Representation learning by learning to count. In *ICCV*, pages 5898–5906, 2017.

[43] J. Oh, H. Yoo, C. Kim, and S.-Y. Yun. BOIL: Towards representation change for few-shot learning. In *ICLR*, 2021.

[44] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2009.

[45] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *NeurIPS*. 2019.

[46] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *CVPR*, pages 2536–2544, 2016.

[47] C. P. Phoo and B. Hariharan. Self-training for few-shot transfer across extreme task differences. In *ICLR*, 2021.

[48] A. Raghu, M. Raghu, S. Bengio, and O. Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. *arXiv preprint arXiv:1909.09157*, 2019.

[49] A. Ramdas, S. J. Reddi, B. Póczos, A. Singh, and L. Wasserman. On the decreasing power of kernel and distance based nonparametric hypothesis tests in high dimensions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[50] S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2016.

[51] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, and R. S. Zemel. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*, 2018.

[52] Y. Rubner, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to image databases. In *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*, pages 59–66. IEEE, 1998.

[53] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[54] J. Snell, K. Swersky, and R. S. Zemel. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*, 2017.

[55] 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. In *NeurIPS*, 2020.

[56] H. Song, M. Kim, S. Kim, and J.-G. Lee. Carpe diem, seize the samples uncertain" at the moment" for adaptive batch selection. In *CIKM*, 2020.

[57] C. Sun, A. Shrivastava, S. Singh, and A. Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017.

[58] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018.

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

[60] M. Tan and Q. Le. Efficientnetv2: Smaller models and faster training. In *International Conference on Machine Learning*, pages 10096–10106. PMLR, 2021.

[61] Y. Tian, D. Krishnan, and P. Isola. Contrastive multiview coding. In *CECCV*, 2020.

[62] Y. Tian, Y. Wang, D. Krishnan, J. B. Tenenbaum, and P. Isola. Rethinking few-shot image classification: a good embedding is all you need? In *ECCV*, 2020.

[63] Y. Tian, X. Chen, and S. Ganguli. Understanding self-supervised learning dynamics without contrastive pairs. In *International Conference on Machine Learning*, pages 10268–10278. PMLR, 2021.

[64] H.-Y. Tseng, H.-Y. Lee, J.-B. Huang, and M.-H. Yang. Cross-domain few-shot classification via learned feature-wise transformation. In *ICLR*, 2020.

[65] 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 *CVPR*, 2018.

[66] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. Matching networks for one shot learning. In *NeurIPS*, 2016.

[67] T. Wang and P. Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, 2020.

[68] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers. ChestX-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *CVPR*, 2017.

[69] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)*, 53(3):1–34, 2020.

[70] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[71] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021.

[72] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *ECCV*, 2016.

[73] W. Zhang, L. Deng, L. Zhang, and D. Wu. Overcoming negative transfer: A survey. *arXiv preprint arXiv:2009.00909*, 2020.

[74] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] In the abstract and introduction, we established the contributions and scope of our paper as the six findings for CD-FSL, which are also described in Figure 1 in a compact manner.
   (b) Did you describe the limitations of your work? [Yes] Refer to Conclusion. As our work is a pioneering study, we exhaustively analyzed CD-FSL and provided several insightful observations; however, there was a lack of theoretical analysis.
   (c) Did you discuss any potential negative societal impacts of your work? [No] We have checked the ethics guidelines and think no corresponding aspects were found.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] We read and ensure it.
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Refer to Abstract. We provided the code URL, where we also described how to set up the data.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Refer to Appendix B and C for dataset and implementation details.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We added the 95% confidence interval of 600 episodes, following other few-shot learning works.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Refer to Appendix C.1 for the training details, including the amount of resources used.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We provided the full details on the existing algorithms and datasets in Appendix A and B.
   (b) Did you mention the license of the assets? [Yes] We included the license information of datasets and our own code assets in the code URL attached in Abstract.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Refer to Abstract. We attached our modified code asset as a form of URL, and this will be open via GitHub.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Self-Supervised Learning Methods

A.1 SimCLR

SimCLR [3] is one of the simplest yet high-performance contrastive learning methods. Its key idea is mapping the semantically similar examples to be close in the representation space while dissimilar examples to be distant. The similar examples are often called positive samples, and the dissimilar ones are called negative samples. Formally, all examples in the current batch $\{x_k\}_{k=1:B}$ with size B are augmented to generated an augmented batch $\{\tilde{x}_{2k-1}, \tilde{x}_{2k}\}_{k=1:B}$, where $\tilde{x}_{2k-1}$ and $\tilde{x}_{2k}$ are the examples differently augmented from the same input x_k. Then, the representations $\{z_{2k-1}, z_{2k}\}_{k=1:B}$ are extracted from a feature extractor with projection layers. Based on the representations, SimCLR performs contrastive learning such that it minimizes the contrastive loss:

$$\mathcal{L}_{\text{SimCLR}} = \frac{1}{2B} \sum_{k=1}^{B} \Big[\ell(2k-1, 2k) + \ell(2k, 2k-1) \Big]$$

$$\text{where } \ell(i,j) = -\log \frac{\exp(\mathsf{sim}(z_i, z_j)/\tau)}{\sum_{n=1}^{2B} \mathbf{1}_{[n \neq i]} \exp(\mathsf{sim}(z_i, z_n)/\tau)}$$

where $\mathbf{1}$ is an indicator function, τ is a temperature hyperparameter, and $\mathsf{sim}(\mathbf{u}, \mathbf{v}) = \mathbf{u}^\top \mathbf{v} / \|\mathbf{u}\| \|\mathbf{v}\|$ measures cosine similarity between two vectors $\mathbf{u}$ and $\mathbf{v}$.

A.2 MoCo

MoCo (Momentum Contrast [26]) is a variant of SimCLR method, which leverages the memory bank and the momentum update of an encoder. Similar to SimCLR, MoCo also minimizes the contrastive loss with positive and negative samples; the positive sample is the other augmentation (view) from the same instance, but the negative samples are not those from the current batch. Instead, MoCo fetches the negative samples from the memory bank, which has been enqueued from the previous batches. To emphasize the usage of the memory bank, the anchor sample, which is contrasted by positive and negative samples, is called query q, while the others are called keys $\{k_0, k_1, \dots k_K\}$. A positive key k_0 is the augmentation from the same sample as q. Then, MoCo minimizes the following loss:

$$\mathcal{L}_{\text{MoCo}} = -\frac{1}{B} \sum_{i=1}^{B} \log \frac{\exp(\mathsf{sim}(q(i), k_0(i))/\tau)}{\sum_{j=0}^{K} \exp(\mathsf{sim}(q(i), k_j(i))/\tau)}$$

where the query representation and the key representations are extracted from different models. That is, $q = f_q(x_q)$ where f_q is a main encoder, while $k = f_k(x_k)$ where f_k is a momentum encoder that is updated by the moving average of its previous state and that of f_q. MoCo overcomes the dependency of the negative sample size on batch size, efficiently achieving the objective of SimCLR using the small memory bank and the additional network.

There are two versions of MoCo: MoCo-v1 [26] and MoCo-v2 [6]. Since MoCo-v2 is a simple improvement of MoCo-v1, such as cosine annealing, MLP projector, and different hyperparameters, we only considered the MoCo-v1 version in this paper.

A.3 BYOL

While SimCLR and MoCo used positive and negative samples to construct a contrastive task, BYOL (Bootstrap Your Own Latent [23]) achieves higher performances than state-of-the-art contrastive learning models without using the negative samples. That is, BYOL is a non-contrastive SSL method, completely free from the need for negative samples. To this end, BYOL minimizes a similarity loss between the two augmented views using two networks.

There are two networks involved: online network f_θ and target network f_ξ. This is a similar setting to MoCo; the online network is a main encoder and the target network is an encoder that is updated by weighted moving average. Given an image x, it augments x into two views $\tilde{x}$ and $\tilde{x}'$. Each view is represented by the encoder with a projector g_θ and g_ξ: $z_\theta = g_\theta(f_\theta(\tilde{x}))$ and $z'_\xi = g_\xi(f_\xi(\tilde{x}'))$. Then, by prediction layers q_θ, a prediction $q_\theta(z_\theta)$ is output and it is compared with the target projection. BYOL uses a mean squared error between the normalized prediction and target projection:

$$\mathcal{L}_{\text{BYOL}} = 2 - 2 \cdot \frac{\langle q_\theta(z_\theta), z'_\xi \rangle}{\|q_\theta(z_\theta)\|_2 \cdot \|z'_\xi\|_2}$$

BYOL also uses a symmetric loss function that passes $\tilde{x}'$ through the online network and $\tilde{x}$ through the target network. The two losses are summed, and the same thing is done for every sample in a batch.

A.4 SimSiam

SimSiam (Simple Siamese [5]) basically shares a similar idea to the BYOL model. The loss form is exactly the same, but SimSiam does not use an extra target network that is updated by momentum. Instead, SimSiam uses the same online network f_θ to output the representation of the two views $\tilde{x}, \tilde{x}'$, but blocks the gradient flow for the target projection. While Grill et al. [23] insisted in BYOL on the importance of a momentum encoder since it can prevent collapsing, Chen and He [5] found that a stop-gradient operation is a key to avoiding collapsing. Thus, SimSiam loss is described as follows:

$$\mathcal{L}_{\text{SimSiam}} = 2 - 2 \cdot \frac{\langle q_\theta(z_\theta), \mathsf{sg}(q_\theta(z'_\theta))\rangle}{\|q_\theta(z_\theta)\|_2 \cdot \|\mathsf{sg}(q_\theta(z'_\theta))\|_2}$$

where sg indicates the stop-gradient operation.

B Datasets Details

B.1 Datasets

In this paper, we used two source domain datasets and eight target domain datasets. Table 4 summarizes the referenced papers, number of classes, and number of samples of each dataset. For source domain datasets, we used miniImageNet and tieredImageNet, which are two different subsets of the ImageNet-1k dataset [11]. The source dataset for miniImageNet (miniImageNet-train) includes 64 base classes, while the target dataset for miniImageNet (miniImageNet-test) include 20 classes that are disjoint from miniImageNet-train, following Appendix O. Similarly, tieredImageNet is partitioned into a train and test set for the source data and target data, respectively. In our FSL experiments, we also reported the performance of SL model pre-trained on ImageNet. However, we did not actually pre-train with the ImageNet dataset, but fine-tuned from the pre-trained model offered by an official PyTorch [45] library.

The target domain datasets can be separated into two groups: BSCD-FSL benchmark [24] and non-BSCD-FSL. First, the BSCD-FSL benchmark includes CropDisease, EuroSAT, ISIC, and ChestX. These datasets are *supposed* to be distant from the miniImageNet source, with CropDisease most similar and ChestX most dissimilar. The criteria are perspective distortion, semantic content, and color depth. We followed Phoo and Hariharan [47] for the splitting procedure of the target dataset into a pre-training unlabeled set and a few-shot evaluation set. A short description of each dataset is provided below.

- **CropDisease** is a set of diseased plant images.
- **EuroSAT** is a set of satellite images of the landscapes.
- **ISIC** is a set of dermoscopy images of human skin lesions.
- **ChestX** is a set of X-Ray images on the human chest.

In addition to the BSCD-FSL benchmark, we introduced four target datasets that are more commonly used in the (CD-)FSL literature. They are Places, Plantae, Cars, and CUB. However, there is no standard rule to separate the pre-training set and the evaluation set for these four datasets. Thus, we sampled the images from each dataset. A short description and the sampling strategy of a dataset are provided below. Also, for the reproducibility of our work, we provide the code for the sampling procedure and the list of images we used.

- **Places** contains the images designed for scene recognition, such as bedrooms and streets, etc. However, because Places is an enormous dataset to use in the FSL context, we sampled 16 classes out of 365 classes (in a total of train, val, and test). Also, to make the dataset size smaller, we sampled 1,715 images per class, which is a reduced amount from the original 4,941 images per class on average.
- **Plantae** contains the plant images. Similar to Places, we sampled some images to reduce the dataset size. However, unlike Places, Plantae is a highly class-imbalanced set. Therefore, we sampled the top 69 classes that have many samples out of 2,917 classes.

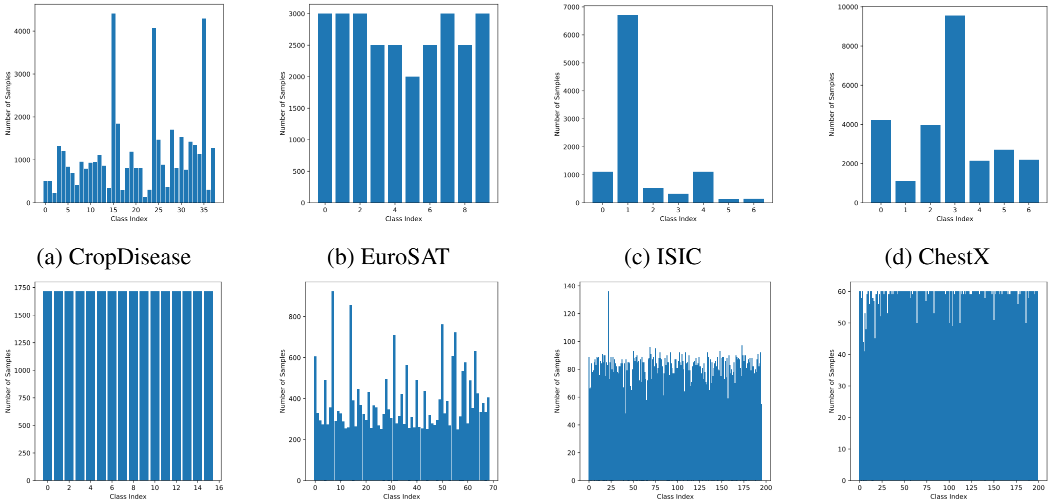
- **Cars** contains the images of 196 car models. We used the entire images that the Cars dataset has (train and test).
- **CUB** contains the images of 200 species of birds. We used the entire images that the CUB dataset has (train, val, and test).

Table 4: Summary of datasets we used in this paper. Note that we used a subset of images for Places and Plantae dataset.

Datasets	miniImageNet-train	miniImageNet-test	tieredImageNet-train	tieredImageNet-test
Reference	Vinyals et al. [66]	Vinyals et al. [66]	Ren et al. [51]	Ren et al. [51]
# of classes	64	20	351	160
# of samples	38,400	12,000	448,695	206,209
Datasets	CropDisease	EuroSAT	ISIC	ChestX
Reference	Mohanty et al. [39]	Helber et al. [27]	Codella et al. [7]	Wang et al. [68]
# of classes	38	10	7	7
# of samples	43,456	27,000	10,015	25,848
Datasets	Places	Plantae	Cars	CUB
Reference	Zhou et al. [74]	Van Horn et al. [65]	Krause et al. [34]	Welinder et al. [70]
# of classes	16	69	196	200
# of samples	27,440	26,650	16,185	11,788

Figure 4 shows the class distribution of each target dataset considered in our study. We observe major differences in the class distributions. For example, the EuroSAT, Places, and CUB datasets have overall balanced class distributions, while the ISIC dataset is extremely unbalanced, with the number of samples per class ranging from 115 to 9,547. We also see that the number of samples per class varies over the eight datasets. The average number of samples per class for the ChestX dataset is 3,693, while for the CUB dataset, this number goes down to only 59.

We posit that the class distribution contributes to the difficulty of each dataset, thus implicitly considered as part of our analysis of target datasets. However, we note that class imbalance is not the deciding factor in dataset difficulty. For example, the CropDisease dataset has a relatively imbalanced class distribution yet is shown to have very low difficulty in our study. Explicitly, the effects of class distribution on CD-FSL have not been studied in our paper.



(a) CropDisease

(b) EuroSAT

(c) ISIC

(d) ChestX

(e) Places

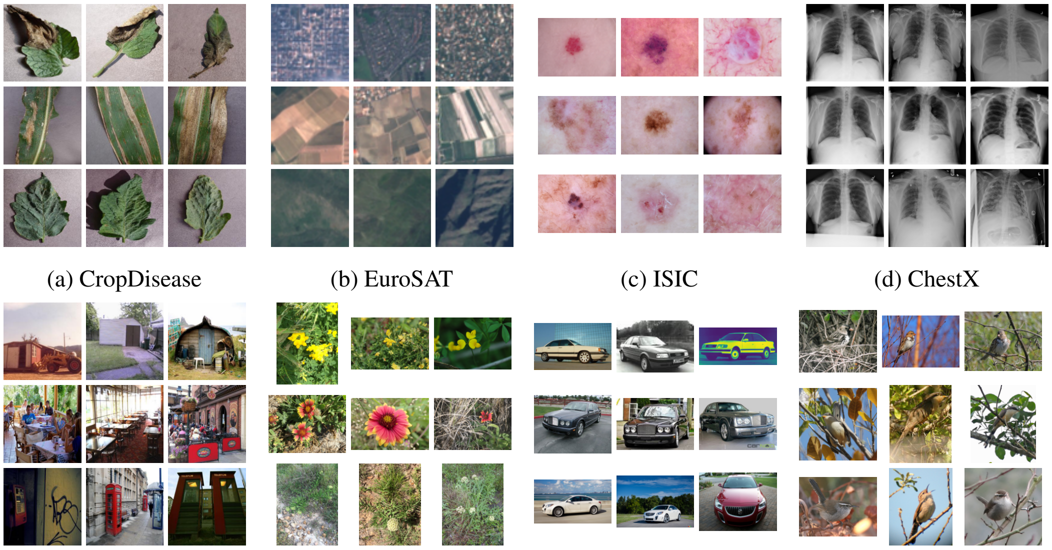
(f) Plantae

(g) Cars

(h) CUB

Figure 4: Class distributions of eight target datasets considered in our study.

B.2 Image Examples


(a) CropDisease (b) EuroSAT (c) ISIC (d) ChestX

(e) Places (f) Plantae (g) Cars (h) CUB

Figure 5: Image examples from eight target datasets considered in our study. Each row displays three samples from a distinct class randomly sampled from each target dataset.

We illustrate the qualitative characteristics of eight target domains for CD-FSL by showing nine randomly sampled examples from three distinct classes for each target dataset in Figure 5. The previous work [24] defined domain similarity to miniImageNet source with respect to perspective distortion, semantic content, and color depth. This can be seen in Figure 5(a)-(d); CropDisease consists of natural images regarding agriculture, EuroSAT contains satellite images taken from a fixed perspective, ISIC and ChestX contain images with fixed perspective and unique semantics, with ChestX being grayscale. On the other hand, non-BSCD-FSL datasets in Figure 5(e)-(h) depict familiar scenes or objects to human eyes.

Using the EMD (Earth Mover's Distance) analysis, we discovered that domain similarity is mainly determined by the semantic content and color depth of the images. For example, we find that ChestX and ISIC, which exhibit highly distinct semantic content, have high domain similarities with all three target datasets. CropDisease and EuroSAT have relatively higher domain similarity within the BSCD-FSL benchmark, and this can be attributed to the fact that the image subjects are from the natural image setting, albeit with fixed perspective and lack of either background or foreground. Places shows the highest domain similarity, which can be attributed to the existence of diverse subjects from the natural image domain, similar to the source datasets.

We can also observe that each target dataset has varying levels of difficulty. For example, for ChestX, it appears challenging to detect the small differences between the grayscale images and nearly impossible to distinguish any prominent features between classes to the untrained eye. On the other hand, classes from CropDisease are shown to have distinct features that are easily distinguishable.

C Implementation Details

C.1 Training Setup

We generally follow the training setup of previous works without validation dataset. Each model was pre-trained on a single RTX A5000, and each pre-training stage of 1000 epochs took 2.5–13.6 hours for ResNet18. SSL pre-training on CropDisease took 6.1, 8.1, 10.3, and 13.6 hours for SimCLR, MoCo, BYOL, and SimSiam, respectively. MSL pre-training took approximately $\times 1.5$ more time compared to SSL, and training time scaled linearly with the size of the target dataset for SSL and MSL.

We explain training details below:

SL Pre-Training We use an SGD optimizer with an initial learning rate of 0.1, the momentum of 0.9, and weight decay coefficient of 10^{-4} is used. When using miniImageNet as source data, we train the ResNet10 model for 1,000 epochs with batch size 64, learning rate decayed by 1/10 at epoch {400, 600, 800}. When using tieredImageNet as source data, we train the ResNet18 model for 90 epochs with batch size 256. For ImageNet, the pre-trained ResNet18 model offered by an official PyTorch [45] library is used.

SimCLR Pre-Training We follow the setting in Phoo and Hariharan [47] except batch size, learning rate, and augmentation method; SGD optimizer with momentum 0.9 and weight decay 10^{-4} is used. 1000 epochs are trained with batch size 32. Because SimCLR (including other SSL methods) uses a multi-viewed batch, it has an effective batch size of 64 by augmentations. The learning rate starts with 0.1 and is decayed by 1/10 times at epoch {400, 600, 800}. For the SimCLR loss, a two-layer projection head (i.e., Linear-ReLU-Linear) is added on top of the extractor. The projection head uses a hidden dimension of 512 and an output feature dimension of 128. The temperature value of NT-Xent loss (normalized temperature-scaled cross-entropy loss, Chen et al. [3]) is set to 1.0.

MoCo Pre-Training We use the same optimizer, epochs, and batch size as SimCLR pre-training. The projector of both query and key is one fully-connected layer with a feature of dimension 128. Also, we used a moving average coefficient of 0.999 for the momentum encoder and the memory bank size of 1,024. Note that the original MoCo [26] uses a considerable size of a memory bank (i.e., 65,536) because a large number of negative samples is required in self-supervised learning of ImageNet data. However, in the case of our self-supervised learning on small-size target data, a large memory bank is neither needed nor recommended. Moreover, a large number of negative samples can make the contrastive task too hard to optimize for extremely fine-grained images, as we observed in ChestX FSL performances. This is the main reason why MoCo rarely surpasses SimCLR in our experiments. Also, note that the hyperparameters are mainly suited to the SimCLR model, but we did not further search or tune the hyperparameters.

BYOL Pre-Training We use a different optimizer for BYOL; Adam [33] optimizer with the initial learning rate of 3×10^{-4}. The online and target projector are both composed of two-layer MLP (i.e., Linear-BatchNorm1D-ReLU-Linear) with a hidden dimension of 4,096 and a projection dimension of 256. The moving average coefficient for the target network is 0.99. A predictor after the online projector is also a two-layer MLP with a hidden dimension of 4,096 and an output prediction dimension of 256.

SimSiam Pre-Training SimSiam uses the same network structure as BYOL, except there is no auxiliary target encoder and target projector. Every other training setup is the same.

MSL Pre-Training When training the MSL model, the batch size of source data is 64 and that of target data is 32, because target data are augmented twice to make positive pairs. Although we pre-trained for 1,000 epochs, one epoch corresponds to an entire sweep over the target data. The source batch is randomly sampled at every iteration, independently from the epoch. A conflicting setting is that we used an SGD optimizer for SL pre-training and an Adam optimizer for BYOL pre-training. Therefore, in MSL (BYOL) pre-training, there were two choices of an optimizer. We confirmed with some experiments that the SGD optimizer better works for MSL (BYOL).

Two-Stage Pre-Training In Section 6, we extended the single-stage pre-training to the two-stage approaches. The initial model for the second stage is the SL model, which is exactly the same model as the above **SL Pre-Training**. The second stage of pre-training also follows the same procedure as the single-stage, both for SSL and MSL.

Fine-Tuning We follow the setting in Guo et al. [24]; SGD optimizer with learning rate 0.01, momentum 0.9, and weight decay 0.001 is used. Only the linear classifier is trained with a frozen pre-trained extractor, and 100 epochs are trained with batch size 4. Note that, for a fair comparison, we removed a projector or predictor that is additionally introduced in SSL pre-training.

C.2 Data Augmentations

We provide below the PyTorch-style code for the base and strong augmentation. A short description for each transform with our set parameter is as follows:

- RandomResizedCrop: Randomly crop a portion of an image and then resize it to 224x224.
- RandomColorJitter: Randomly change the brightness, contrast, and saturation, with a probability of 1.0.
- RandomHorizontalFlip: Randomly flip an image on a vertical axis, with a probability of 0.5.
- RandomGrayscale: Randomly convert image into grayscale, with a probability of 0.1.
- RandomGaussianBlur: Randomly blur an image with Gaussian blur of kernel size (5,5), with a probability of 0.3.

```
import torchvision.transforms as transforms

def parse_transform(transform, image_size=224):
    if transform == ‘RandomResizedCrop’:
        return transforms.RandomResizedCrop(image_size)

    elif transform == ‘RandomColorJitter’:
        return transforms.RandomApply(
               [transforms.ColorJitter(0.4,0.4,0.4,0.0)], p=1.0)

    elif transform == ‘RandomGrayscale’:
        return transforms.RandomGrayscale(p=0.1)

    elif transform == ‘RandomGaussianBlur’:
        return transforms.RandomApply(
               [transforms.GaussianBlur(kernel_size=(5,5))], p=0.3)

    elif transform == ‘Resize’:
        return transforms.Resize([image_size, image_size])

    elif transform == ‘Normalize’:
        return transforms.Normalize(mean=[0.485,0.456,0.406],
                                    std=[0.229,0.224,0.225])
    elif transform == ‘ToTensor’:
        return transforms.ToTensor()

def get_composed_transform(augmentation: str, image_size=224):
    if augmentation == ‘base’:
        transform_list = [‘RandomResizedCrop’, ‘RandomColorJitter’,
                          ‘RandomHorizontalFlip’, ‘ToTensor’, ‘Normalize’]

    elif augmentation == ‘strong’:
        transform_list = [‘RandomResizedCrop’, ‘RandomColorJitter’,
                          ‘RandomGrayscale’, ‘RandomGaussianBlur’,
                          ‘RandomHorizontalFlip’, ‘ToTensor’, ‘Normalize’]

    elif augmentation == ‘none’:
        transform_list = [‘Resize’, ‘ToTensor’, ‘Normalize’]


    transform_funcs = [parse_transform(x, image_size=image_size)
                       for x in transform_list]
    transform = transforms.Compose(transform_funcs)
    return transform
```

D Domain Similarity

To estimate the domain similarity, we follow Cui et al. [10] and Li et al. [36] by calculating EMD as the distance between two domains. EMD is informally defined as the minimum cost of moving one accumulation into another. EMD has advantages compared to other metric choices, such as Kullback-Leibler divergence (KLD), Jensen-Shannon divergence (JSD), or maximum mean discrepancy (MMD). We can compute EMD directly from the samples, whereas KLD and JSD require explicit expressions for the densities [9]. MMD can also be considered but is less powerful in high dimensions and highly dependent on the kernel and its hyperparameters [49].

The BSCD-FSL benchmark contains four datasets with varying levels of domain similarity: CropDisease, EuroSAT, ISIC, and ChestX. These datasets are known to be distant from the source dataset, miniImageNet. Guo et al. [24] provided the order of domain similarity for the BSCD-FSL benchmark based on three qualitative factors: perspective distortion, semantic contents, and color depth. However, our quantitative metric in Eq. (4) shows a somewhat different order of domain similarity between the four datasets in BSCD-FSL. The known similarity order for BSCD-FSL was *"CropDisease $>$ EuroSAT $>$ ISIC $>$ ChestX"* under the assumption that a dataset has domain similar to ImageNet if it has perspective distortion (*i.e.*, CropDisease), is natural (*i.e.*, CropDisease, EuroSAT), and has RGB color depth (*i.e.*, CropDisease, EuroSAT, ISIC).

In contrast, we observe a different order of *"EuroSAT $>$ CropDisease $\approx$ ISIC $>$ ChestX"* in Table 5 and Table 6 when using our quantitative metric. It turns out that semantic content and color depth are significant factors in deciding domain similarity; thus, ChestX is always the most dissimilar to the source domain. On the other hand, perspective distortion is less important than semantic content and color depth for determining domain similarity, considering that the order of CropDisease and EuroSAT is reversed. Therefore, we observe that EuroSAT is the closest dataset to the source domain. The change in the domain similarity rank of EuroSAT, when tieredImageNet is used as the source dataset, is discussed below.

Table 5: Domain Similarity. Earth Mover's Distance (EMD) and similarity (calculated by $\exp(-\alpha \times \text{EMD})$) are reported. The feature extractor used is ResNet101 provided by PyTorch [45]. Rank 1 dataset indicates that the source and target datasets are the most similar.

		Places	CUB	Cars	Plantae	EuroSAT	CropDisease	ISIC	ChestX
EMD	IN	18.14	18.91	20.13	20.31	21.67	21.88	22.25	22.28
	tieredIN	17.26	19.90	20.23	20.63	19.20	22.07	22.94	23.19
	miniIN	17.49	19.38	20.34	20.29	21.10	21.66	22.20	22.33
Sim	IN	0.834 (1)	0.828 (2)	0.818 (3)	0.816 (4)	0.805 (5)	0.803 (6)	0.801 (7)	0.800 (8)
	tieredIN	0.841 (1)	0.820 (3)	0.817 (4)	0.814 (5)	0.825 (2)	0.802 (6)	0.795 (7)	0.793 (8)
	miniIN	0.840 (1)	0.824 (2)	0.816 (4)	0.816 (3)	0.810 (5)	0.805 (6)	0.801 (7)	0.800 (8)

Domain similarity can differ according to the feature extractor used because it is based on representations. In the main paper, we use ResNet101 to extract representations because we use ResNet-like models for our few-shot classification tasks. However, other architectures (*e.g.*, DenseNet and ViT) can also be used. Table 6 shows the domain similarity to the ImageNet source dataset, measured using different feature extractors. To do this, we used the open-source library `timm`.[6] For the details of each architecture, please refer to the original papers: ResNet [25], MobileNetV2 [53], EfficientNet [59, 60], DenseNet [29], and ViT [15].

Although the exact ordering of domain similarity can change, it does not undermine the consistency of our analysis. In particular, we explain that:

- **Important point for Obs. 5.1.** From Obs. 5.1, we argue that larger domain similarity does not always guarantee the superiority of SL. To demonstrate this, we compare the performance of SL and SSL on Places and CUB. Namely, SSL is better than SL on Places, while SL is better than SSL on CUB, despite Places being closer to the source dataset. As we can see in Table 6, this observation does not change, even when using other feature extractors. In fact, when EfficientNet-b4 is used, the domain similarity ranking of Cars and EuroSAT are changed, exacerbating the inconsistency. Furthermore, when using ViT models, the similarity ranking of CUB moves down to fifth place,

[6] `http://github.com/rwightman/pytorch-image-models/`

despite it showing the largest margin between SL and SSL performance, in favor of SL on source data (refer to Figure 2).

- **Important point for Obs. 5.2.** From Obs. 5.2, we argue that for both groups, the performance gain of SSL over SL becomes greater as few-shot difficulty decreases. We first divide the eight target datasets into two groups according to the domain similarity. Within each similarity group, the performance gain of SSL over SL is highly related to the few-shot difficulty. As shown in Table 6, EuroSAT can be categorized into the large similarity group when using EfficientNet-b4, ViT-B/16, or ViT-L/16. However, because EuroSAT has lower few-shot difficulty than Places (refer to Figure 1), the superiority of SSL over SL on EuroSAT is consistently explained with few-shot difficulty, even when inside the large similarity group. In addition, the domain similarity ranking of CropDisease and ISIC based on ResNet101 is different from that based on other extractors. However, both datasets remain inside the small similarity group, hence does not affect Obs. 5.2.

Table 6: Domain Similarity to ImageNet measured across different architectures. Similarities (calculated by $\exp(-\alpha \times \text{EMD})$) are reported. The feature extractors used are ResNet101, provided by PyTorch [45], and others, by `timm` open-source library. Rank 1 dataset indicates that the source and target datasets are the most similar.

Extractor	Places	CUB	Cars	Plantae	EuroSAT	CropDisease	ISIC	ChestX
ResNet101 (main)	0.834 (1)	0.828 (2)	0.818 (3)	0.816 (4)	0.805 (5)	0.803 (6)	0.801 (7)	0.800 (8)
ResNet18	0.866 (1)	0.847 (2)	0.845 (3)	0.843 (4)	0.829 (5)	0.823 (7)	0.828 (6)	0.815 (8)
MobileNetV2	0.919 (1)	0.918 (2)	0.908 (4)	0.910 (3)	0.903 (5)	0.889 (7)	0.893 (6)	0.884 (8)
EfficientNet-b0	0.913 (1)	0.910 (2)	0.903 (3)	0.901 (4)	0.901 (5)	0.873 (7)	0.877 (6)	0.873 (8)
EfficientNet-b4	0.969 (1)	0.967 (2)	0.963 (5)	0.965 (4)	0.966 (3)	0.956 (7)	0.961 (6)	0.953 (8)
EfficientNetV2	0.930 (1)	0.927 (2)	0.924 (3)	0.924 (4)	0.924 (5)	0.906 (7)	0.914 (6)	0.896 (8)
DenseNet121	0.818 (1)	0.802 (2)	0.791 (3)	0.787 (4)	0.785 (5)	0.761 (7)	0.767 (6)	0.752 (8)
ViT-B/16	0.508 (1)	0.415 (5)	0.438 (4)	0.442 (3)	0.444 (2)	0.386 (7)	0.409 (6)	0.390 (8)
ViT-L/16	0.478 (1)	0.395 (5)	0.396 (4)	0.426 (2)	0.422 (3)	0.372 (7)	0.391 (6)	0.367 (8)

E Few-Shot Difficulty

We quantify few-shot difficulty using our empirical upper bound on each dataset following in Eq. (5). The few-shot difficulty depends on a backbone network and k. Table 7 describes few-shot difficulty according to the combination of our backbone (ResNet10 and ResNet18) and k. It is observed that the order of few-shot difficulty remains the same, except between ISIC and CUB when ResNet10 is used as a backbone and k=1. We point out that Obs. 5.2 still stands under this variation.

Table 7: Few-shot difficulty (ranking). 5-way k-shot performances are reported. To quantify the data difficulty, we designed the upper performance case, where we use SL pre-training with 20% of target data as labeled data. The k-shot difficulty is calculated by Diff@$k = \exp(-\beta \times \text{Perf@}k)$. Rank 1 dataset is the most difficult one.

Backbone	k	CropDisease	EuroSAT	ISIC	ChestX	Places	Plantae	Cars	CUB
RN18	1	96.92±.32	90.51±.55	42.83±.80	31.00±.60	63.97±.87	52.83±.89	48.71±.82	42.96±.76
	5	99.51±.10	96.74±.21	55.55±.67	39.19±.58	81.56±.57	74.24±.71	72.83±.67	64.03±.77
	20	99.69±.07	97.45±.17	61.32±.62	42.11±.56	86.10±.47	82.17±.64	82.08±.53	74.14±.66
Diff@1		0.379 (8)	0.405 (7)	0.652 (2)	0.733 (1)	0.527 (6)	0.590 (5)	0.614 (4)	0.651 (3)
Diff@5		0.370 (8)	0.380 (7)	0.574 (2)	0.676 (1)	0.442 (6)	0.476 (5)	0.483 (4)	0.527 (3)
Diff@20		0.369 (8)	0.377 (7)	0.542 (2)	0.656 (1)	0.423 (6)	0.440 (5)	0.440 (4)	0.476 (3)
RN10	1	92.44±.55	83.34±.68	42.89±.77	28.89±.55	57.25±.82	49.08±.83	43.32±.72	40.72±.73
	5	99.00±.15	95.37±.26	56.94±.65	36.59±.56	77.39±.63	70.56±.76	66.40±.68	60.29±.78
	20	99.54±.07	97.28±.18	63.93±.58	42.03±.55	84.62±.49	80.50±.65	78.56±.57	71.71±.67
Diff@1		0.397 (8)	0.435 (7)	0.651 (3)	0.749 (1)	0.564 (6)	0.612 (5)	0.648 (4)	0.666 (2)
Diff@5		0.372 (8)	0.385 (7)	0.566 (2)	0.694 (1)	0.461 (6)	0.494 (5)	0.515 (4)	0.547 (3)
Diff@20		0.370 (8)	0.378 (7)	0.528 (2)	0.657 (1)	0.429 (6)	0.447 (5)	0.456 (4)	0.488 (3)

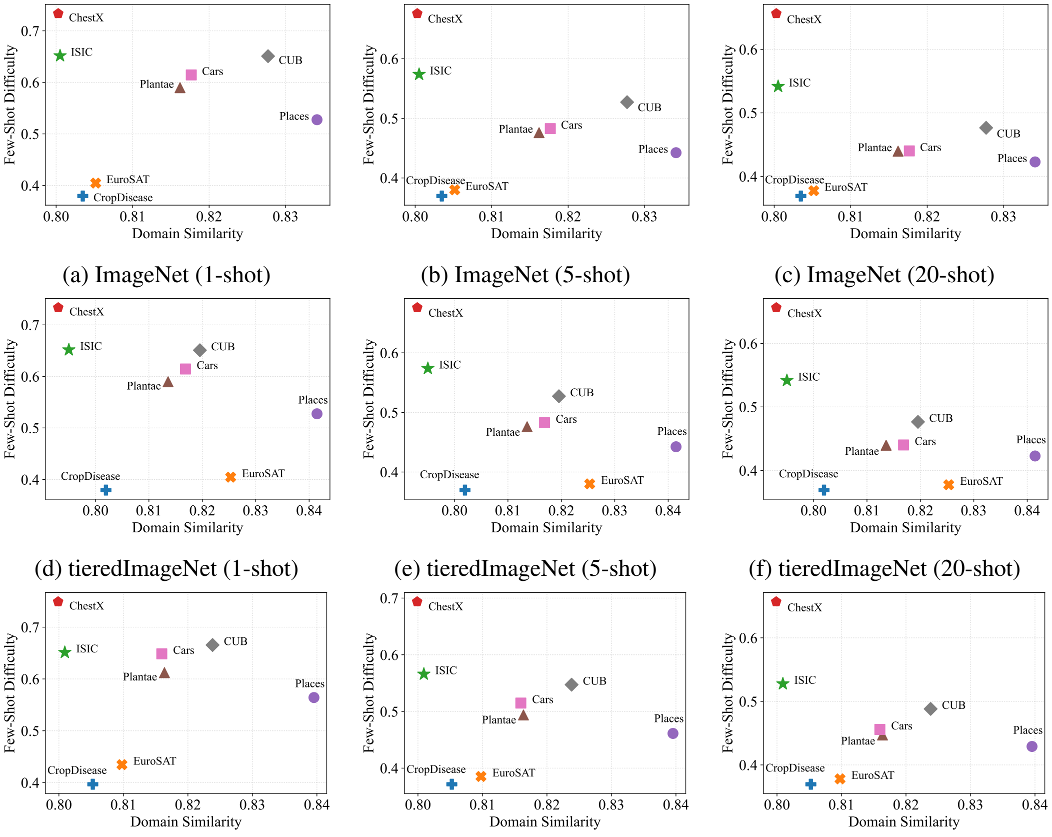
Few-shot Difficulty on Different Splits. We used the same 20% split of $\mathcal{D}_U$ as used in SSL pre-training for measuring the few-shot difficulty, but with label information. This is because the dataset partition for calculating few-shot difficulty (which is the rest 80%) should be matched with that for evaluating SL/SSL methods, for consistent analysis. However, few-shot difficulty can differ according to the dataset splits. To remedy this concern, we provide the few-shot performance using different splits when 20% of target dataset are used for pre-training with label information. Table 8 shows that the ranks of few-shot difficulty between datasets do not change even if dataset splits are changed.

Table 8: 5-way k-shot performances are reported. These performances are converted to few-shot difficulty. To quantify the data difficulty, we designed the upper performance case, where we use SL pre-training with 20% of target data as labeled data. To show the robustness of few-shot difficulty, accuracy is estimated three times using different splits for 20% of target data. ResNet18 is used as a backbone.

Split seed	k	CropDisease	EuroSAT	ISIC	ChestX	Places	Plantae	Cars	CUB
1	1	96.92±.32	90.51±.55	42.83±.80	31.00±.60	63.97±.87	52.83±.89	48.71±.82	42.96±.76
	5	99.51±.10	96.74±.21	55.55±.67	39.19±.58	81.56±.57	74.24±.71	72.83±.67	64.03±.77
	20	99.69±.07	97.45±.17	61.32±.62	42.11±.56	86.10±.47	82.17±.64	82.08±.53	74.14±.66
2	1	96.52±.36	90.84±.50	43.05±.74	30.03±.62	63.61±.90	54.94±.88	48.98±.85	43.84±.80
	5	99.51±.09	96.93±.20	55.92±.65	39.64±.55	81.65±.56	75.73±.72	72.95±.63	63.76±.78
	20	99.78±.05	97.67±.16	63.34±.56	45.96±.54	86.18±.45	82.74±.59	81.41±.49	75.41±.67
3	1	96.54±.34	89.23±.56	42.94±.78	29.36±.60	65.25±.84	54.20±.91	48.39±.79	43.95±.75
	5	99.54±.08	96.73±.19	56.78±.68	38.23±.57	82.44±.54	74.51±.79	71.71±.70	64.39±.75
	20	99.81±.05	97.60±.16	63.16±.62	44.60±.55	86.64±.45	82.94±.61	82.04±.50	75.79±.64

F Domain Similarity and Few-Shot Difficulty Visualizations

In this section, we provide visualizations of domain similarity and few-shot difficulty. Domain similarity is dependent on the source dataset, and few-shot difficulty is dependent on backbone network (*e.g.*, ResNet10 and ResNet18) and k. Figure 6 visualizes domain similarity and few-shot difficulty for eight datasets, as depicted in Appendix D and E. Figure 6(a,b,c) have the same domain similarity, Figure 6(d,e,f) have the same, and Figure 6(g,h,i) have the same, because domain similarity is based on the source dataset. For few-shot difficulty, Figure 6(a,d) have the same difficulty, 6(b,e) have the same, and (c,f) have the same, because few-shot difficulty is based on backbone network and k.



(a) ImageNet (1-shot)

(b) ImageNet (5-shot)

(c) ImageNet (20-shot)

(d) tieredImageNet (1-shot)

(e) tieredImageNet (5-shot)

(f) tieredImageNet (20-shot)

(g) miniImageNet (1-shot)

(h) miniImageNet (5-shot)

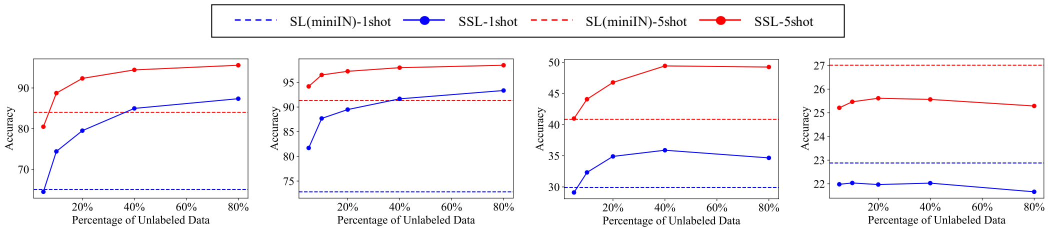
(i) miniImageNet (20-shot)

Figure 6: Domain similarity and few-shot difficulty for eight benchmark datasets.

G Performance of SSL according to the Ratio of Unlabeled Target Data

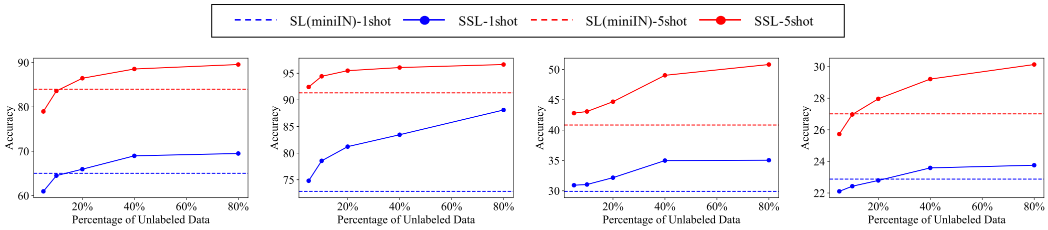
In this section, we evaluate few-shot performance of SSL (SimCLR and BYOL) according to the ratio of unlabeled target data when ResNet10 is used as a backbone network. Figure 7 and Figure 8 describe the few-shot performance according to the ratio of unlabeled target data when SimCLR and BYOL are used for SSL method, respectively. We control the ratio $\in \{5\%, 10\%, 20\%, 40\%, 80\%\}$. We further evaluate few-shot performance of SSL (SimCLR) when ResNet18 is used as a backbone network, depicted as Figure 9.

It is observed that except for ChestX, SimCLR with a small portion (even 5%) of target data as unlabeled data has better performance than SL that uses ImageNet, tieredImageNet, and miniImageNet. Note that ImageNet and tieredImageNet include around 1.3 million and 0.45 million samples with annotations, respectively. On the other hand, 5% of EuroSAT, CropDisease, and ISIC unlabeled data include around 1.4k, 2.2k, and 0.5k samples. It implies that the consistency between source and target domains is much more important than the number of data for pre-training.



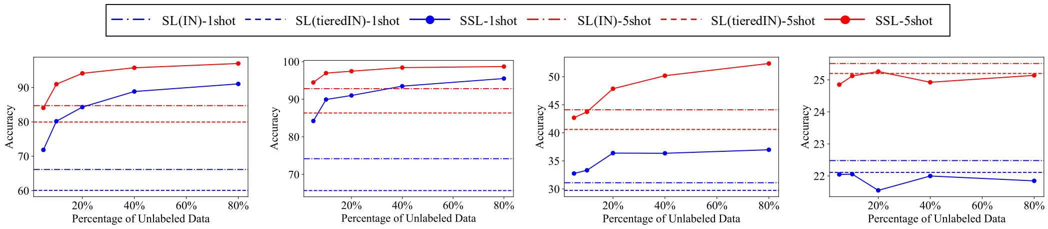
(a) EuroSAT (b) CropDisease (c) ISIC (d) ChestX

Figure 7: 5way-$k$shot performance of SSL (SimCLR) according to the ratio of unlabeled target data and SL (Section 4). ResNet10 is used as a backbone. Blue and red lines indicate 1-shot and 5-shot accuracy, respectively. Dotted and solid lines are accuracy of SL (miniIN) and SSL (SimCLR), respectively.



(a) EuroSAT (b) CropDisease (c) ISIC (d) ChestX

Figure 8: 5way-$k$shot performance of SSL (BYOL) according to the ratio of unlabeled target data and SL (Section 4). ResNet10 is used as a backbone. Blue and red lines indicate 1-shot and 5-shot accuracy, respectively. Dotted and solid lines are accuracy of SL (miniIN) and SSL (BYOL), respectively.



(a) EuroSAT (b) CropDisease (c) ISIC (d) ChestX

Figure 9: 5way-$k$shot performance of SSL according to the ratio of unlabeled target data and SL (Section 4). ResNet18 is used as a backbone. SimCLR is used for the SSL method.

H Performance of SL and SSL for the Other Datasets

Table 9 summarizes the few-shot performance of SL and SSL on non-BSCD-FSL datasets. Note that these datasets are known to be closer to the ImageNet than BSCD-FSL datasets [16] and our estimated similarity shows the same trend. We would like to highlight that unlike BSCD-FSL sets, these four target domains take a big advantage from the ImageNet dataset. SL pre-trained on ImageNet has comparable or even better (in Cars and CUB) performance than SSL.

Table 9: 5-way k-shot CD-FSL performance of the models pre-trained by SL and SSL, on four additional target datastes: Places, Plantae, Cars, and CUB. We report the average accuracy and its 95% confidence interval over 600 few-shot episodes. B and S indicate base and strong augmentations, respectively. The best results are marked in bold and the second best are underlined. We include the result when using the model pre-trained on the entire ImageNet data, which also uses the ResNet18 backbone as tieredImageNet experiments.

Source Data	Pre-train Scheme	Method	Aug.	Places		Plantae		Cars		CUB	
				k=1	k=5	k=1	k=5	k=1	k=5	k=1	k=5
ImageNet	SL	Default	B	57.47±.86	79.22±.64	43.66±.80	**63.21**±.82	**45.82**±.79	**66.38**±.80	**65.24**±.97	**83.93**±.66
tiered ImageNet	SL	Default	B	52.07±.86	72.12±.69	38.63±.74	54.76±.82	31.23±.65	42.59±.70	57.94±.93	76.86±.78
			S	52.82±.86	72.96±.67	34.99±.64	51.11±.76	31.05±.63	42.32±.69	54.18±.91	74.14±.80
Target Data	SSL	SimCLR	B	45.82±.85	62.07±.78	38.52±.74	53.89±.80	28.86±.68	37.05±.69	33.56±.67	43.99±.71
			S	**64.97**±.94	**80.43**±.61	**44.18**±.85	60.07±.84	32.46±.70	44.55±.74	36.15±.76	47.36±.79
		MoCo	B	39.64±.82	53.95±.77	35.17±.73	48.83±.76	27.40±.64	34.59±.67	29.67±.59	36.93±.61
			S	55.53±.74	71.50±.73	36.49±.73	49.15±.76	29.36±.67	38.44±.70	31.76±.66	40.81±.72
		BYOL	B	40.38±.72	60.06±.73	38.60±.72	57.81±.81	31.04±.66	41.79±.72	35.27±.67	49.61±.71
			S	51.76±.79	72.47±.63	42.16±.75	61.02±.82	34.54±.70	48.56±.76	36.50±.68	51.31±.78
		SimSiam	B	35.27±.68	48.12±.69	36.11±.76	48.63±.79	28.30±.64	35.24±.65	29.96±.62	37.61±.60
			S	52.56±.92	68.29±.74	36.19±.69	50.23±.76	31.21±.64	43.06±.67	33.73±.71	43.22±.74

(a) ResNet18 is used as a backbone.

Source Data	Pre-train Scheme	Method	Aug.	Places		Plantae		Cars		CUB	
mini ImageNet	SL	Default	B	51.84±.80	72.19±.70	37.28±.69	54.15±.74	30.79±.56	44.36±.69	**40.65**±.78	**58.54**±.81
			S	52.45±.78	72.92±.66	36.72±.67	53.26±.73	30.20±.54	44.39±.66	40.56±.78	58.10±.78
Target Data	SSL	SimCLR	B	44.06±.78	62.86±.78	38.43±.77	54.68±.80	28.59±.66	38.24±.73	33.88±.68	45.31±.73
			S	**58.75**±.93	**78.39**±.61	**42.65**±.80	**59.77**±.82	30.89±.66	**45.60**±.72	35.49±.73	47.69±.77
		MoCo	B	38.41±.74	54.65±.74	33.96±.69	47.51±.72	28.03±.66	36.19±.72	32.37±.65	40.55±.72
			S	52.05±.90	71.57±.70	36.36±.73	50.37±.78	28.25±.61	38.89±.69	33.53±.72	42.87±.74
		BYOL	B	40.60±.69	59.28±.71	39.27±.73	55.87±.79	30.11±.62	41.21±.69	34.74±.65	49.10±.74
			S	47.81±.75	68.14±.68	39.12±.71	55.31±.79	**31.53**±.65	43.92±.70	35.96±.70	49.34±.76
		SimSiam	B	39.27±.72	53.40±.74	37.12±.72	50.61±.81	28.49±.62	35.50±.67	30.37±.63	38.22±.62
			S	51.62±.81	69.77±.66	38.49±.73	53.10±.78	30.00±.59	40.92±.67	34.25±.71	44.85±.74

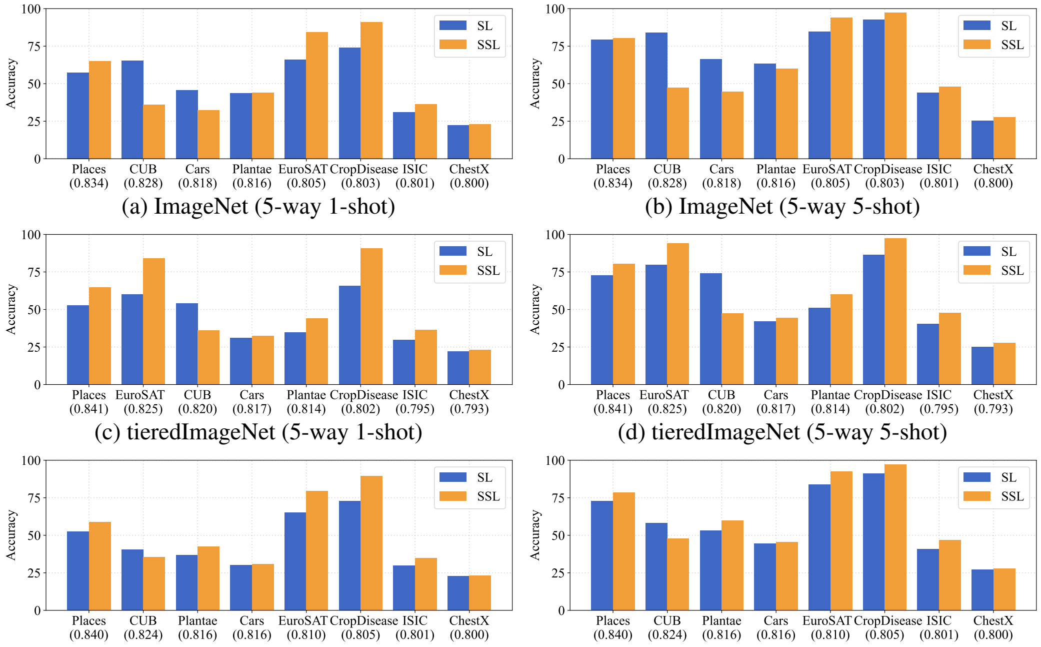
(b) ResNet10 is used as a backbone.

I Analyses on Other Source Datasets

In this section, we expand our analyses of SL and SSL on ImageNet source in Section 5, onto two additional source datasets: tieredImageNet and miniImageNet. Every observation that was previously identified on ImageNet is consistently made in the two additional source datasets.

I.1 Limitations of Domain Similarity (Observation 5.1)

Figure 10 shows the performance of SL and SSL for three source datasets and eight target datasets, according to domain similarity. Across all source datasets, we consistently find that domain similarity alone is not sufficient to explain the relative performance of SL, compared to SSL. As mentioned in Section 5, we observe that SSL can outperform SL even when domain similarity is large, as highlighted by the difference between Places and CUB shown in Figures 10(a,b) for ImageNet, which are identical to Figures 2(a,b) in the main paper. Similar observations are made between EuroSAT and CUB for tieredImageNet in Figures 10(c,d), and between Places and CUB for miniImageNet in Figures 10(e,f).



(a) ImageNet (5-way 1-shot)

(b) ImageNet (5-way 5-shot)

(c) tieredImageNet (5-way 1-shot)

(d) tieredImageNet (5-way 5-shot)

(e) miniImageNet (5-way 1-shot)

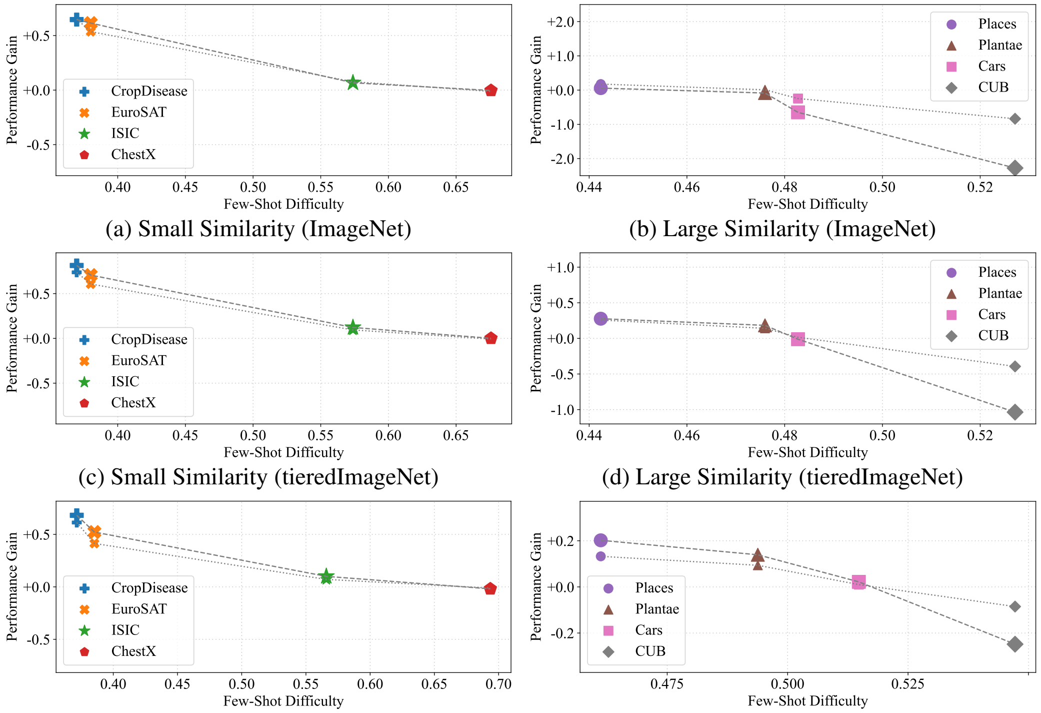
(f) miniImageNet (5-way 5-shot)

Figure 10: 5-way k-shot CD-FSL performance (%) of SL and SSL according to domain similarity. Target datasets are shown in order of domain similarity (values in x-axis) to ImageNet, tieredImageNet and miniImageNet, respectively. For SSL, SimCLR is used for all datasets except ChestX, for which BYOL is used.

I.2 When Does Performance Gain of SSL over SL Become Greater? (Observation 5.2)

Figure 11 shows the performance gain of SSL over SL for three source datasets and eight target datasets, according to few-shot difficulty, for two groups with different levels of domain similarity. Again, the identical observation is made for all three source datasets. When comparing the two groups (BSCD-FSL vs. others), larger performance gain is observed for the small domain similarity group (BSCD-FSL), compared to the latter (others). Within each group, the performance gain of SSL over SL increases with lower few-shot difficulty.

In addition, comparing between different source datasets, for target datasets with large similarity (Figure 11(b,d,f)), the performance gain of SSL over SL decreases by larger source dataset size. For example, on the CUB dataset, the performance gain (for $k = 5$) is -0.249, -1.035, and -2.276 for miniImageNet, tieredImageNet, and ImageNet, respectively. However, for target datasets with small similarity (Figure 11(a,c,e)), the performance gain of SSL over SL does not have a consistent trend according to the source dataset size.



(a) Small Similarity (ImageNet)

(b) Large Similarity (ImageNet)

(c) Small Similarity (tieredImageNet)

(d) Large Similarity (tieredImageNet)

(e) Small Similarity (miniImageNet)

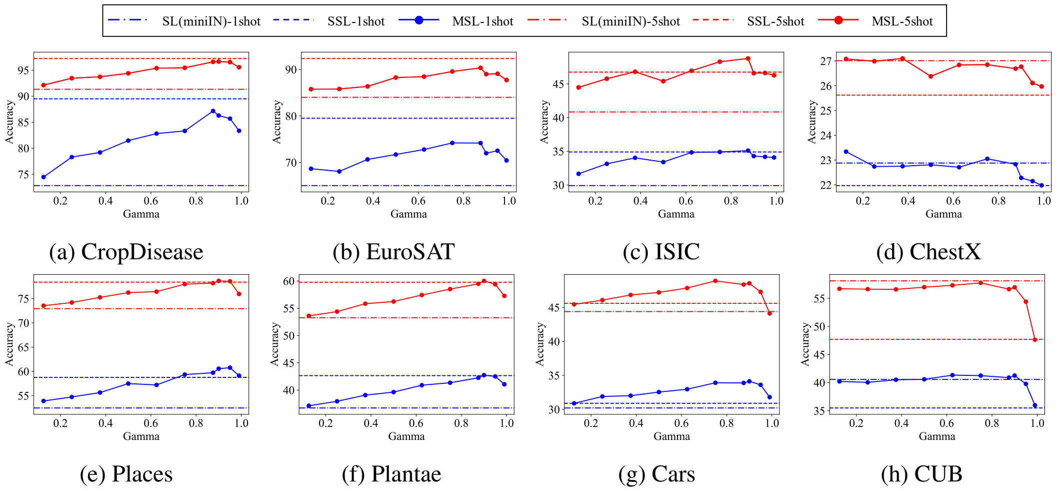
(f) Large Similarity (miniImageNet)

Figure 11: 5-way k-shot performance gains of SSL over SL for the two dataset groups according to the few-shot difficulty (small: k=1, large: k=5). Results are shown for three source datasets: ImageNet, tieredImageNet, and miniImageNet, each with their corresponding backbones. SimCLR is used for SSL in all target datasets except ChestX, for which BYOL is used.

J Hyperparameter γ in MSL Pre-Training

J.1 Choice of Hyperparameter γ

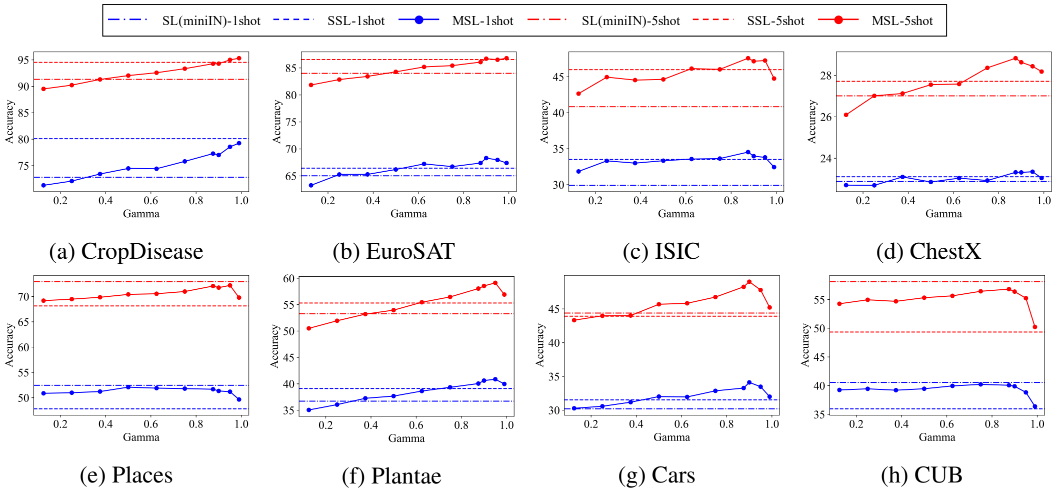
One important hyperparameter in MSL is a balancing weight γ (refer to Eq. (3)). We investigated how we should choose γ value. Figure 12 and Figure 13 describe the few-shot performance of MSL according to the balancing weight γ between SL and SSL when SimCLR or BYOL are used for SSL, respectively. In Figure 12, MSL performance (circle-marked solid lines) generally improves as γ increases from 0.125 to 0.875, i.e., the weight for SSL is getting larger, except for ChestX. In Section 4, we found that non-contrastive SSL method nicely worked on ChestX. Figure 13 shows that MSL with BYOL loss guarantees good performance on ChestX in $\gamma = 0.875$. We further increased γ to $\{0.9, 0.95, 0.99\}$, but there was an overall decreasing trend of accuracy, so we fixed γ to 0.875 in every MSL experiment in the paper.



(a) CropDisease (b) EuroSAT (c) ISIC (d) ChestX

(e) Places (f) Plantae (g) Cars (h) CUB

Figure 12: 5-way k-shot performance of MSL according to the balancing weight (i.e., γ) between SL and SSL (Section 6). ResNet10 is used as a backbone. SimCLR is used for the MSL and SSL method.



(a) CropDisease (b) EuroSAT (c) ISIC (d) ChestX

(e) Places (f) Plantae (g) Cars (h) CUB

Figure 13: 5-way k-shot performance of MSL according to the balancing weight (i.e., γ) between SL and SSL (Section 6). ResNet10 is used as a backbone. BYOL is used for the MSL and SSL method.

J.2 Dynamic Hyperparameter γ

Inspired by the two-stage pre-training schemes in Section 6, we investigate the effects of dynamically increasing γ during the single-stage pre-training. Specifically, we investigate a simple pre-training scheme in which γ linearly increases from 0 to 1 over the course of 1000 epochs (*i.e.,* single-stage MSL with $\gamma = 0 \nearrow 1$).

Table 10 and Table 11 describe the few-shot performance of the devised method (in the lowermost row), with performances of other methods displayed for ease of comparison. We observe that the performance of the devised method lies between that of standalone SL and SSL except for ChestX. The devised method typically underperforms two-stage pre-training as well as standalone MSL, indicating that it is not an effective method to exploit both SL and SSL dynamically.

Table 10: 5-way 1-shot CD-FSL performance (%) of the models pre-trained with varying configurations of γ in Eq. (3) of MSL. ResNet18 is used as the backbone model, and ImageNet is used as the source data for SL. The best results are marked in bold.

Pre-train Scheme	γ	Method	Small Similarity				Large Similarity			
			CropDisease	EuroSAT	ISIC	ChestX	Places	Plantae	Cars	CUB
SL	0	Default	74.18±.82	66.14±.83	31.11±.55	22.48±.39	57.47±.86	43.66±.80	45.82±.79	**65.24**±.97
SSL	1	SimCLR	91.00±.76	84.30±.73	36.39±.66	21.55±.41	64.97±.94	44.18±.85	32.46±.70	36.15±.76
		BYOL	85.77±.73	66.16±.86	34.53±.62	22.75±.41	51.76±.79	42.16±.75	34.54±.70	36.50±.68
MSL	0.875	SimCLR	88.38±.70	73.97±.79	34.02±.62	22.04±.40	65.13±.88	47.47±.86	36.96±.77	47.35±.87
		BYOL	86.47±.74	73.18±.83	**37.10**±.67	23.97±.44	61.40±.87	48.31±.86	33.31±.66	50.71±.87
SL → SSL	0→1	SimCLR	**92.24**±.70	**86.51**±.67	36.11±.67	21.75±.41	**71.05**±.92	49.02±.91	37.43±.79	42.40±.85
		BYOL	87.64±.70	74.05±.84	35.62±.65	23.01±.43	58.12±.87	48.28±.88	38.23±.75	42.48±.82
SL → MSL	0→0.875	SimCLR	91.46±.66	77.62±.76	34.46±.64	22.50±.41	69.50±.87	51.27±.91	40.39±.82	62.12±.93
		BYOL	88.37±.73	71.54±.78	36.08±.63	**24.42**±.45	63.40±.86	**53.65**±.88	**46.62**±.85	64.33±.93
MSL	$0 \nearrow 1$	SimCLR	81.32±.79	70.68±.82	32.70±.60	22.77±.41	61.36±.84	44.50±.83	36.27±.69	50.40±.86
		BYOL	77.37±.83	67.84±.82	34.70±.64	23.38±.41	59.18±.82	45.37±.83	36.18±.71	51.00±.85

Table 11: 5-way 5-shot CD-FSL performance (%) of the models pre-trained with varying configurations of γ in Eq. (3) of MSL. ResNet18 is used as the backbone model, and ImageNet is used as the source data for SL. The best results are marked in bold.

Pre-train Scheme	γ	Method	Small Similarity				Large Similarity			
			CropDisease	EuroSAT	ISIC	ChestX	Places	Plantae	Cars	CUB
SL	0	Default	92.81±.45	84.73±.51	44.10±.58	25.51±.44	79.22±.64	63.21±.82	66.38±.80	83.93±.66
SSL	1	SimCLR	97.46±.34	94.12±.32	47.85±.65	25.26±.44	80.43±.61	60.07±.84	44.55±.74	47.36±.79
		BYOL	96.93±.30	87.83±.48	47.59±.63	28.36±.46	72.47±.63	61.02±.82	48.56±.76	51.31±.78
MSL	0.875	SimCLR	96.50±.35	90.11±.40	45.38±.63	26.05±.44	82.56±.58	64.76±.83	51.84±.79	64.53±.80
		BYOL	96.74±.31	90.82±.40	49.14±.70	29.58±.47	81.27±.59	67.39±.81	46.76±.73	69.67±.82
SL → SSL	0→1	SimCLR	**97.88**±.30	**95.28**±.27	48.38±.60	25.25±.44	84.40±.53	66.35±.82	51.31±.84	57.11±.88
		BYOL	97.58±.26	91.82±.39	49.32±.63	28.27±.48	78.87±.60	67.83±.82	54.70±.84	60.60±.82
SL → MSL	0→0.875	SimCLR	97.49±.30	91.70±.35	47.43±.62	26.24±.44	**85.76**±.52	69.24±.81	58.97±.82	81.51±.72
		BYOL	97.09±.31	90.89±.40	**50.72**±.67	**30.20**±.48	83.29±.55	**74.16**±.77	**68.87**±.80	**84.34**±.67
MSL	$0 \nearrow 1$	SimCLR	94.83±.42	87.69±.50	44.48±.61	26.76±.45	80.62±.58	62.02±.83	52.97±.76	69.37±.79
		BYOL	93.98±.41	86.66±.50	47.61±.66	28.55±.47	79.71±.60	63.77±.83	54.13±.75	70.49±.79

K Increasing Batch Size on the Source Dataset for MSL

Naturally, the size of labeled source data and unlabeled target data differ greatly. For example, while ImageNet contains 1.3M training examples, Cars and CUB each contains 3,400 and 2,350 unlabeled examples when 20% of the data is used. Thus, during 1,000 epochs of MSL pre-training (where each epoch corresponds to one pass through the unlabeled target data), only 2-3 passes are completed through ImageNet (refer to **MSL Pre-Training** setup in Appendix C.1). Considering the effectiveness of SL when domain similarity is large, we posit that MSL under large domain similarity can benefit from higher batch size for on the source data, *i.e.*, allowing more passes through the source dataset. In particular, we fix the batch size for the target data to 64, and increase the batch size for the source data.

Table 12 describes CD-FSL performance according to the source batch size on Cars and CUB. It is shown that larger batch size for the source dataset can improve the MSL performance. We suppose that the MSL model with ImageNet obtains large generalization ability from large-scale data, gaining much larger benefit than miniImageNet or tieredImageNet source. This improvement is significant in Cars and CUB datasets because they are similar to ImageNet. In fact, ImageNet data already includes car types ($\sim$10 classes) and bird species ($\sim$59 classes).

Table 12: 5-way k-shot performance of MSL according to the source batch size when ImageNet is used as source.

Target Dataset	Method	Batch Size for Source Dataset	k=1	k=5
Cars	SimCLR	64 (default)	36.96±.77	51.84±.79
		128	38.54±.81	53.80±.84
		256	38.24±.78	54.18±.81
		512	38.98±.81	55.25±.81
	BYOL	64 (default)	33.31±.66	46.76±.73
		128	39.85±.81	58.01±.80
		256	41.45±.82	59.48±.80
		512	40.98±.80	59.48±.81
CUB	SimCLR	64 (default)	47.35±.87	64.53±.80
		128	49.91±.87	68.01±.81
		256	51.06±.85	69.51±.79
		512	51.48±.88	70.13±.79
	BYOL	64 (default)	50.71±.87	69.67±.82
		128	52.75±.87	72.26±.79
		256	54.17±.86	73.50±.79
		512	53.70±.86	73.31±.80

L Results Summary

L.1 Source Dataset: ImageNet

Table 13 and Table 14 describe 5-way 1-shot and 5-way 5-shot CD-FSL performance when ImageNet is used as the source dataset, respectively. Note that Table 14 is added for convenience and this is the same with Table 3 in the main paper. The results of STARTUP on BSCD-FSL (*i.e.*, CropDisease, EuroSAT, ISIC, and ChestX) target datasets are from Phoo and Hariharan [47]. The results on the other four target datasets are our reimplementation with their official code.[7] Also, Islam et al. [30] did not provide the results of DynDistill on ImageNet source dataset, so we reimplemented it with their official code.[8]

Table 13: 5-way 1-shot CD-FSL performance (%) of the models pre-trained by SL, SSL, and MSL including their two-stage versions. ResNet18 is used as the backbone model, and ImageNet is used as the source data for SL. The balancing coefficient γ in Eq. (3) of MSL is set to be 0.875. The best results are marked in bold and the second best are underlined.

	Pre-train Scheme	Method	Small Similarity				Large Similarity			
			CropDisease	EuroSAT	ISIC	ChestX	Places	Plantae	Cars	CUB
Single-Stage	SL	Default	74.18±.82	66.14±.83	31.11±.55	22.48±.39	57.47±.86	43.66±.80	**45.82**±.79	**65.24**±.97
	SSL	SimCLR	**91.00**±.76	**84.30**±.73	36.39±.66	21.55±.41	64.97±.94	44.18±.85	32.46±.70	36.15±.76
		BYOL	85.77±.73	66.16±.86	34.53±.62	22.75±.41	51.76±.79	42.16±.75	34.54±.70	36.50±.68
	MSL	SimCLR	88.38±.70	73.97±.79	34.02±.62	22.04±.40	**65.13**±.88	47.47±.86	36.96±.77	47.35±.87
		BYOL	86.47±.74	73.18±.83	**37.10**±.67	**23.97**±.44	61.40±.87	**48.31**±.86	33.31±.66	50.71±.87

(a) Performance comparison for single-stage schemes.

	Pre-train Scheme	Method	CropDisease	EuroSAT	ISIC	ChestX	Places	Plantae	Cars	CUB
Two-Stage	SL→SSL	SimCLR	**92.24**±.70	**86.51**±.67	**36.11**±.67	21.75±.41	**71.05**±.92	49.02±.91	37.43±.79	42.40±.85
		BYOL	87.64±.70	74.05±.84	35.62±.65	23.01±.43	58.12±.87	48.28±.88	38.23±.75	42.48±.82
	SL→MSL	SimCLR	91.46±.66	77.62±.76	34.46±.64	22.50±.41	69.50±.87	51.27±.91	40.39±.82	62.12±.93
		BYOL	88.37±.73	71.54±.78	36.08±.63	**24.42**±.45	63.40±.86	**53.65**±.88	**46.62**±.85	64.33±.93
	SL→MSL+	STARTUP	85.10±.74	73.83±.77	31.69±.59	23.03±.42	66.02±.87	49.78±.93	45.75±.84	**72.58**±.93
		DynDistill	87.53±1.01	77.24±1.06	34.55±1.82	24.02±1.59	60.84±1.08	49.90±1.22	46.55±1.21	63.80±1.32

(b) Performance comparison for two-stage schemes.

Table 14: 5-way 5-shot CD-FSL performance (%) of the models pre-trained by SL, SSL, and MSL including their two-stage versions. ResNet18 is used as the backbone model, and ImageNet is used as the source data for SL. The balancing coefficient γ in Eq. (3) of MSL is set to be 0.875. The best results are marked in bold and the second best are underlined.

	Pre-train Scheme	Method	Small Similarity				Large Similarity			
			CropDisease	EuroSAT	ISIC	ChestX	Places	Plantae	Cars	CUB
Single-Stage	SL	Default	92.81±.45	84.73±.51	44.10±.58	25.51±.44	79.22±.64	63.21±.82	**66.38**±.80	**83.93**±.66
	SSL	SimCLR	**97.46**±.34	**94.12**±.32	47.85±.65	25.26±.44	80.43±.61	60.07±.84	44.55±.74	47.36±.79
		BYOL	96.93±.30	87.83±.48	47.59±.63	28.36±.46	72.47±.63	61.02±.82	48.56±.76	51.31±.78
	MSL	SimCLR	96.50±.35	90.11±.40	45.38±.63	26.05±.44	**82.56**±.58	64.76±.83	51.84±.79	64.53±.80
		BYOL	96.74±.31	90.82±.40	**49.14**±.70	**29.58**±.47	81.27±.59	**67.39**±.81	46.76±.73	69.67±.82

(a) Performance comparison for single-stage schemes.

	Pre-train Scheme	Method	CropDisease	EuroSAT	ISIC	ChestX	Places	Plantae	Cars	CUB
Two-Stage	SL→SSL	SimCLR	**97.88**±.30	**95.28**±.27	48.38±.60	25.25±.44	84.40±.53	66.35±.82	51.31±.84	57.11±.88
		BYOL	97.58±.26	91.82±.39	49.32±.63	28.27±.48	78.87±.60	67.83±.82	54.70±.84	60.60±.82
	SL→MSL	SimCLR	97.49±.30	91.70±.35	47.43±.62	26.24±.44	**85.76**±.52	69.24±.81	58.97±.82	81.51±.72
		BYOL	97.09±.31	90.89±.40	**50.72**±.67	**30.20**±.48	83.29±.55	**74.16**±.77	68.87±.80	84.34±.67
	SL→MSL+	STARTUP	96.06±.33	89.70±.41	46.02±.59	27.24±.46	85.00±.52	69.40±.84	68.43±.82	**89.60**±.55
		DynDistill	97.60±.35	92.28±.46	50.06±.86	29.65±.67	82.22±.81	71.49±1.06	**69.45**±1.12	86.54±1.88

(b) Performance comparison for two-stage schemes.

[7] https://github.com/cpphoo/STARTUP

[8] https://github.com/asrafulashiq/dynamic-cdfsl

L.2 Source Dataset: tieredImageNet

Table 15 and Table 16 describe 5-way 1-shot and 5-way 5-shot CD-FSL performance when tieredImageNet is used as the source dataset, respectively. Phoo and Hariharan [47] did not provide the results of STARTUP on tieredImageNet source dataset, so we reimplemented it with their official code. The results of DynDistill on BSCD-FSL are from Islam et al. [30]; however, note that DynDistill used a larger ResNet-18 backbone model than our setting, which is provided by Tian et al. [62]. Also, the results on the other four target datasets are our reimplementation with their official code.

The difference of the result of tieredImageNet from the result of ImageNet as the source dataset is that one-stage MSL can outperform SL on Cars and CUB datasets. It is considered that bigger source dataset makes SL stronger, as we have addressed this issue in Appendix I.

Table 15: 5-way 1-shot CD-FSL performance (%) of the models pre-trained by SL, SSL, and MSL including their two-stage versions. ResNet18 is used as the backbone model, and tieredImageNet is used as the source data for SL. The balancing coefficient γ in Eq. (3) of MSL is set to be 0.875. The best results are marked in bold and the second best are underlined.

	Pre-train Scheme	Method	Small Similarity				Large Similarity			
			CropDisease	EuroSAT	ISIC	ChestX	Places	Plantae	Cars	CUB
Single-Stage	SL	Default	65.70±.94	60.07±.88	29.75±.56	22.11±.42	52.82±.86	34.99±.64	31.38±.61	54.18±.91
	SSL	SimCLR	**91.00**±.76	**84.30**±.73	36.39±.66	21.55±.41	**64.97**±.94	44.18±.85	32.46±.70	36.15±.76
		BYOL	85.77±.73	66.16±.86	34.53±.62	22.75±.41	51.76±.79	42.16±.75	34.54±.70	36.50±.68
	MSL	SimCLR	87.44±.72	77.42±.77	35.47±.64	21.95±.40	63.83±.93	46.47±.87	34.65±.74	50.41±.90
		BYOL	84.67±.78	68.45±.81	**37.30**±.66	**24.41**±.44	60.07±.87	**46.49**±.83	**37.88**±.75	**54.43**±.88

(a) Performance comparison for single-stage schemes.

	Pre-train Scheme	Method	CropDisease	EuroSAT	ISIC	ChestX	Places	Plantae	Cars	CUB
Two-Stage	SL→SSL	SimCLR	**92.41**±.70	**86.61**±.66	36.95±.67	21.75±.40	**68.51**±.94	47.92±.88	35.37±.77	44.74±.86
		BYOL	84.82±.76	66.92±.84	37.19±.66	24.23±.46	44.34±.79	44.32±.81	38.49±.78	44.40±.83
	SL→MSL	SimCLR	90.13±.69	80.20±.78	35.32±.64	22.18±.38	64.85±.92	48.00±.86	35.83±.75	60.87±.90
		BYOL	85.72±.76	53.92±.94	**39.41**±.68	**24.31**±.45	59.16±.86	**48.48**±.83	**41.02**±.78	61.98±.88
	SL→MSL^{+}	STARTUP	77.67±.83	69.60±.86	33.90±.63	23.13±.40	59.14±.87	41.80±.85	34.45±.66	**63.83**±.90
		DynDistill	84.41±.75	72.15±.75	33.87±.56	22.70±.42	52.21±1.15	43.06±1.12	38.51±1.03	58.67±1.30

(b) Performance comparison for two-stage schemes.

Table 16: 5-way 5-shot CD-FSL performance (%) of the models pre-trained by SL, SSL, and MSL including their two-stage versions. ResNet18 is used as the backbone model, and tieredImageNet is used as the source data for SL. The balancing coefficient γ in Eq. (3) of MSL is set to be 0.875. The best results are marked in bold and the second best are underlined.

	Pre-train Scheme	Method	Small Similarity				Large Similarity			
			CropDisease	EuroSAT	ISIC	ChestX	Places	Plantae	Cars	CUB
Single-Stage	SL	Default	86.34±.60	79.95±.66	40.60±.58	25.20±.41	72.96±.67	51.11±.76	45.18±.68	**74.14**±.80
	SSL	SimCLR	**97.46**±.34	**94.12**±.32	47.85±.65	25.26±.44	80.43±.61	60.07±.84	44.55±.74	47.36±.79
		BYOL	96.93±.30	87.83±.48	47.59±.63	28.36±.46	72.47±.63	61.02±.82	48.56±.76	51.31±.78
	MSL	SimCLR	96.68±.33	91.72±.37	47.55±.67	26.10±.45	**81.67**±.58	63.96±.82	48.81±.77	68.78±.82
		BYOL	96.41±.33	89.51±.42	**50.95**±.69	**30.04**±.47	80.16±.60	**67.09**±.80	**54.75**±.80	73.03±.82

(a) Performance comparison for single-stage schemes.

	Pre-train Scheme	Method	CropDisease	EuroSAT	ISIC	ChestX	Places	Plantae	Cars	CUB
Two-Stage	SL→SSL	SimCLR	**97.88**±.31	**95.39**±.26	50.28±.61	25.31±.44	**83.51**±.56	65.40±.82	48.91±.83	61.80±.84
		BYOL	96.25±.31	89.39±.45	53.00±.64	30.66±.48	71.57±.63	63.06±.79	55.04±.82	62.78±.80
	SL→MSL	SimCLR	97.43±.31	93.09±.33	49.66±.63	26.27±.44	83.03±.56	65.78±.85	52.22±.81	80.37±.76
		BYOL	96.64±.32	85.97±.50	**53.67**±.68	**30.84**±.51	80.76±.56	**69.77**±.79	**62.09**±.78	82.77±.69
	SL→MSL^{+}	STARTUP	92.87±.41	85.23±.59	48.20±.62	27.06±.42	78.00±.60	60.28±.82	51.14±.75	**83.36**±.66
		DynDistill	95.90±.34	89.44±.42	47.21±.56	27.67 ±.46	75.67±.86	64.32±1.08	59.14±1.15	79.26±.97

(b) Performance comparison for two-stage schemes.

L.3 Source Dataset: miniImageNet

Table 17 and Table 18 describe 5-way 1-shot and 5-way 5-shot CD-FSL performance when miniImageNet is used as the source dataset, respectively. The results of STARTUP and DynDistill on BSCD-FSL target datasets are from Phoo and Hariharan [47] and Islam et al. [30], respectively. The results on the other four target datasets are our reimplementation with their official codes. Similar to the results when tieredImageNet is used as the source dataset, one-stage MSL can outperform SL on Cars and CUB datasets. For a thorough comparison, we also report the results of meta-learning based approaches: MAML [17], MatchingNet [66], and RelationNet [58], where the numbers are from [64, 24, 30]. As previous studies on CD-FSL verified, meta-learning based algorithms are mostly outperformed by transfer learning based algorithms in the cross-domain setup.

Table 17: 5-way 1-shot CD-FSL performance (%) of the models pre-trained by SL, SSL, and MSL including their two-stage versions. ResNet10 is used as the backbone model, and miniImageNet is used as the source data for SL. The balancing coefficient γ in Eq. (3) of MSL is set to be 0.875. The best results are marked in bold and the second best are underlined.

	Pre-train Scheme	Method	Small Similarity				Large Similarity			
			CropDisease	EuroSAT	ISIC	ChestX	Places	Plantae	Cars	CUB
Single-Stage	-	MAML	-	-	-	-	-	-	-	-
		MatchingNet	46.86±.88	54.88±.90	27.37±.51	20.65±.29	49.86±.79	32.70±.60	30.77±.47	35.89±.51
		RelationNet	-	-	-	-	48.64±.85	33.17±.64	29.11±.60	42.44±.77
	SL	Default	72.82±.87	65.03±.88	29.91±.54	22.88±.42	52.45±.78	36.72±.67	30.20±.54	40.56±.78
	SSL	SimCLR	**89.49**±.74	**79.50**±.78	34.90±.64	21.97±.41	58.75±.93	42.65±.80	30.89±.66	35.49±.73
		BYOL	80.10±.76	66.45±.80	33.50±.59	23.11±.42	47.81±.75	39.12±.71	31.53±.65	35.96±.70
	MSL	SimCLR	87.15±.75	74.18±.80	35.10±.64	22.83±.41	**59.72**±.89	42.24±.80	33.89±.66	40.89±.79
		BYOL	74.16±.82	66.64±.81	**35.63**±.66	**24.07**±.47	53.60±.82	**43.94**±.79	**35.71**±.68	**42.73**±.78

(a) Performance comparison for single-stage schemes.

	Pre-train Scheme	Method	CropDisease	EuroSAT	ISIC	ChestX	Places	Plantae	Cars	CUB
Two-Stage	SL→SSL	SimCLR	**89.39**±.82	**82.64**±.73	35.09±.64	22.15±.40	**63.19**±.92	**46.30**±.85	34.85±.74	39.92±.79
		BYOL	82.61±.76	67.67±.77	**35.92**±.68	23.76±.45	53.72±.79	45.02±.79	37.40±.74	41.61±.75
	SL→MSL	SimCLR	86.18±.77	74.06±.85	33.91±.65	22.13±.40	61.56±.86	43.47±.79	35.78±.72	43.50±.82
		BYOL	75.77±.82	65.67±.83	35.23±.66	**24.47**±.44	54.86±.81	44.68±.78	**38.20**±.71	**45.82**±.79
	SL→MSL$^+$	STARTUP	75.93±.80	63.88±.84	32.66±.60	23.09±.43	48.87±.81	38.01±.73	31.79±.61	41.24±.75
		DynDistill	82.14±.78	73.14±.84	34.66±.58	23.38±.43	49.28±1.11	40.60±1.15	34.77±.98	42.51±1.11

(b) Performance comparison for two-stage schemes.

Table 18: 5-way 5-shot CD-FSL performance (%) of the models pre-trained by SL, SSL, and MSL including their two-stage versions. ResNet10 is used as the backbone model, and miniImageNet is used as the source data for SL. The balancing coefficient γ in Eq. (3) of MSL is set to be 0.875. The best results are marked in bold and the second best are underlined.

	Pre-train Scheme	Method	Small Similarity				Large Similarity			
			CropDisease	EuroSAT	ISIC	ChestX	Places	Plantae	Cars	CUB
Single-Stage	-	MAML	78.05±.68	71.70±.72	40.13±.58	23.48±.96	-	-	-	-
		MatchingNet	66.39±.78	64.45±.63	36.74±.53	22.40±.70	63.16±.77	46.53±.68	38.99±.64	51.37±.77
		RelationNet	68.99±.75	61.31±.72	39.41±.58	22.96±.88	63.32±.76	44.00±.60	37.33±.68	57.77±.69
	SL	Default	91.32±.49	84.00±.56	40.84±.56	27.01±.44	72.92±.66	53.26±.73	44.39±.66	58.10±.78
	SSL	SimCLR	**97.24**±.33	**92.36**±.37	46.76±.61	25.62±.43	**78.39**±.61	59.77±.82	45.60±.72	47.69±.77
		BYOL	94.53±.41	86.55±.50	45.99±.63	27.71±.44	68.14±.68	55.31±.71	43.92±.70	49.34±.76
	MSL	SimCLR	96.59±.35	90.34±.34	**48.78**±.62	26.69±.44	78.17±.61	59.48±.82	48.36±.75	56.63±.78
		BYOL	93.71±.41	87.21±.48	48.63±.66	**29.86**±.47	75.16±.64	**63.45**±.81	**53.33**±.76	**60.66**±.77

(a) Performance comparison for single-stage schemes.

	Pre-train Scheme	Method	CropDisease	EuroSAT	ISIC	ChestX	Places	Plantae	Cars	CUB
Two-Stage	SL→SSL	SimCLR	**96.84**±.40	**94.51**±.32	48.23±.59	24.59±.42	**81.52**±.56	64.37±.81	50.72±.80	55.06±.84
		BYOL	95.87±.35	90.01±.43	**50.33**±.67	29.94±.48	75.83±.62	64.93±.79	55.46±.79	59.78±.81
	SL→MSL	SimCLR	96.67±.33	90.18±.42	47.24±.63	26.47±.44	79.95±.61	61.34±.80	52.74±.79	61.33±.80
		BYOL	94.50±.39	87.96±.48	49.36±.66	**30.23**±.50	76.67±.63	**65.41**±.78	**58.62**±.78	**66.20**±.78
	SL→MSL$^+$	STARTUP	93.02±.45	82.29±.60	47.22±.61	26.94±.44	69.56±.66	55.40±.78	46.73±.73	60.00±.78
		DynDistill	95.54±.38	89.07±.47	49.36±.59	28.31±.46	70.98±.94	58.63±1.14	51.98±1.18	62.86±1.06

(b) Performance comparison for two-stage schemes.

L.4 Source Dataset: ImageNet (ResNet50)

Table 19 and Table 20 describe 5-way 1-shot and 5-way 5-shot CD-FSL performance when ResNet50 is used as a backbone and ImageNet is used as the source dataset, respectively. We find that the observations in our paper also hold for ResNet50.

Table 19: 5-way 1-shot CD-FSL performance (%) of the models pre-trained by SL, SSL, and MSL including their two-stage versions. ResNet50 is used as the backbone model, and ImageNet is used as the source data for SL. The balancing coefficient γ in Eq. (3) of MSL is set to be 0.875. The best results are marked in bold and the second best are underlined.

	Pre-train Scheme	Method	Small Similarity				Large Similarity			
			CropDisease	EuroSAT	ISIC	ChestX	Places	Plantae	Cars	CUB
Single-Stage	SL	Default	73.74±.90	67.38±.88	29.00±.51	22.03±.38	65.51±.91	46.52±.83	**51.25**±.90	**71.83**±.95
	SSL	SimCLR	**90.40**±.77	**81.82**±.77	35.26±.62	21.73±.41	63.98±.97	41.32±.80	32.91±.73	35.26±.75
		BYOL	87.17±.73	72.71±.83	34.33±.63	22.67±.42	53.33±.84	39.34±.76	31.58±.71	33.38±.68
	MSL	SimCLR	87.17±.73	72.71±.83	34.33±.63	22.67±.42	**68.24**±.90	45.85±.85	36.52±.77	47.53±.88
		BYOL	87.25±.82	72.47±.84	**36.68**±.67	**23.54**±.43	62.75±.87	**49.20**±.88	38.57±.77	48.72±.87

(a) Performance comparison for single-stage schemes.

	Pre-train Scheme	Method	CropDisease	EuroSAT	ISIC	ChestX	Places	Plantae	Cars	CUB
Two-Stage	SL→SSL	SimCLR	92.14±.72	**86.41**±.65	36.31±.69	21.72±.41	70.58±.93	49.36±.91	37.49±.79	43.20±.87
		BYOL	84.44±.97	69.11±.94	35.90±.69	21.62±.40	49.05±.89	37.40±.89	35.96±.75	36.95±.74
	SL→MSL	SimCLR	**92.62**±.62	76.30±.79	35.51±.67	22.48±.42	**73.05**±.88	54.08±.94	41.91±.85	61.51±.97
		BYOL	91.04±.68	73.78±.81	**37.27**±.67	**24.70**±.42	65.55±.86	**59.08**±.94	**49.65**±.89	**66.36**±.90

(b) Performance comparison for two-stage schemes.

Table 20: 5-way 5-shot CD-FSL performance (%) of the models pre-trained by SL, SSL, and MSL including their two-stage versions. ResNet50 is used as the backbone model, and ImageNet is used as the source data for SL. The balancing coefficient γ in Eq. (3) of MSL is set to be 0.875. The best results are marked in bold and the second best are underlined.

	Pre-train Scheme	Method	Small Similarity				Large Similarity			
			CropDisease	EuroSAT	ISIC	ChestX	Places	Plantae	Cars	CUB
Single-Stage	SL	Default	92.65±.47	85.95±.54	42.00±.60	25.04±.43	**86.46**±.50	66.82±.80	**73.49**±.80	**91.06**±.54
	SSL	SimCLR	97.00±.37	**93.34**±.34	46.16±.62	24.77±.43	79.02±.62	57.62±.85	43.97±.75	45.90±.77
		BYOL	96.69±.30	86.19±.50	43.44±.61	27.21±.45	73.73±.65	58.40±.81	41.87±.71	48.33±.74
	MSL	SimCLR	96.83±.34	89.02±.46	45.49±.61	26.15±.44	84.97±.54	63.69±.83	51.11±.81	65.23±.80
		BYOL	**97.27**±.31	90.46±.41	**49.06**±.69	**28.98**±.47	82.19±.57	**68.57**±.83	56.52±.81	66.99±.78

(a) Performance comparison for single-stage schemes.

	Pre-train Scheme	Method	CropDisease	EuroSAT	ISIC	ChestX	Places	Plantae	Cars	CUB
Two-Stage	SL→SSL	SimCLR	97.54±.34	**95.57**±.27	48.95±.63	24.75±.44	85.39±.51	66.46±.85	50.54±.85	58.48±.89
		BYOL	96.20±.38	91.94±.41	52.00±.66	26.42±.46	77.78±.61	66.95±.85	52.90±.81	54.37±.78
	SL→MSL	SimCLR	**98.18**±.26	91.48±.38	49.34±.64	26.62±.45	**87.87**±.49	72.39±.81	59.86±.85	81.46±.76
		BYOL	97.80±.27	92.14±.35	**53.04**±.67	**30.91**±.51	85.72±.52	**78.47**±.73	**72.98**±.80	**85.55**±.68

(b) Performance comparison for two-stage schemes.

L.5 Source Dataset: ImageNet (20- and 50-shots)

Table 21 and Table 22 describe 5-way 20-shot and 5-way 50-shot CD-FSL performance when ResNet18 is used as a backbone and ImageNet is used as the source dataset, respectively. We find that the results are consistent with our main analysis. For target datasets that have small similarity to the source dataset, it remains beneficial to perform SSL pre-training on the unlabeled target data to adapt to target domain features, compared to SL on source (Obs. 4.1). For target datasets with large similarity, the relative benefit of SSL on target data is larger when few-shot difficulty is low (Obs. 5.2). We note that SL performance significantly benefits from large k when similarity to the source domain is high. Furthermore, the observations about joint synergy via MSL (Obs. 6.1) and sequential synergy via two-stage pre-training (Obs. 6.2) consistently hold.

Table 21: 5-way 20-shot CD-FSL performance (%) of the models pre-trained by SL, SSL, and MSL including their two-stage versions. ResNet18 is used as the backbone model, and ImageNet is used as the source data for SL. The balancing coefficient γ in Eq. (3) of MSL is set to be 0.875. The best results are marked in bold and the second best are underlined.

	Pre-train Scheme	Method	Small Similarity				Large Similarity			
			CropDisease	EuroSAT	ISIC	ChestX	Places	Plantae	Cars	CUB
Single-Stage	SL	Default	97.21±.25	91.66±.33	53.78±.58	29.39±.42	**89.65**±.40	76.08±.72	**82.79**±.60	**94.48**±.38
	SSL	SimCLR	97.88±.29	**96.02**±.23	55.94±.60	28.36±.43	83.90±.50	64.91±.80	49.64±.78	55.41±.77
		BYOL	98.76±.15	94.60±.28	58.52±.55	35.26±.47	81.82±.52	72.32±.75	61.06±.79	64.38±.78
	MSL	SimCLR	98.78±.16	94.71±.25	56.39±.60	31.46±.43	88.64±.42	75.55±.73	66.05±.74	75.89±.68
		BYOL	**98.97**±.13	95.03±.24	**60.54**±.62	**37.35**±.49	88.06±.43	**78.40**±.69	59.74±.72	80.23±.64

(a) Performance comparison for single-stage schemes.

Two-Stage	SL→SSL	SimCLR	98.27±.26	**96.77**±.21	58.28±.57	28.57±.42	87.88±.42	71.52±.77	59.40±.82	67.79±.80
		BYOL	**99.20**±.12	96.60±.19	60.81±.60	36.00±.48	86.67±.43	78.91±.67	70.90±.76	73.95±.71
	SL→MSL	SimCLR	98.96±.15	95.58±.22	58.87±.59	31.78±.43	**90.69**±.38	79.85±.67	76.54±.70	90.25±.48
		BYOL	99.10±.14	95.82±.21	**62.43**±.59	**38.48**±.47	89.62±.40	**84.00**±.61	**84.39**±.60	**91.59**±.46

(b) Performance comparison for two-stage schemes.

Table 22: 5-way 50-shot CD-FSL performance (%) of the models pre-trained by SL, SSL, and MSL including their two-stage versions. ResNet18 is used as the backbone model, and ImageNet is used as the source data for SL. The balancing coefficient γ in Eq. (3) of MSL is set to be 0.875. The best results are marked in bold and the second best are underlined.

	Pre-train Scheme	Method	Small Similarity				Large Similarity			
			CropDisease	EuroSAT	ISIC	ChestX	Places	Plantae	Cars	CUB
Single-Stage	SL	Default	98.19±.17	93.70±.27	60.42±.55	33.27±.44	**91.49**±.34	80.96±.61	**89.79**±.44	**98.18**±.17
	SSL	SimCLR	97.99±.28	**96.35**±.22	58.40±.56	30.61±.43	84.44±.47	66.60±.74	53.17±.71	62.66±.73
		BYOL	99.04±.13	95.63±.24	62.74±.55	39.36±.49	84.26±.46	77.20±.68	70.37±.68	81.13±.57
	MSL	SimCLR	99.09±.13	95.93±.21	60.33±.57	35.52±.44	90.30±.35	80.41±.61	76.49±.66	89.97±.44
		BYOL	**99.27**±.10	96.00±.21	**64.64**±.58	**41.88**±.50	90.02±.36	**82.68**±.60	70.34±.68	92.40±.37

(a) Performance comparison for single-stage schemes.

Two-Stage	SL→SSL	SimCLR	98.29±.26	97.04±.19	61.40±.54	30.67±.43	88.44±.39	73.16±.72	63.73±.77	75.53±.68
		BYOL	**99.44**±.09	**97.35**±.16	66.71±.54	40.93±.50	88.74±.37	82.58±.60	81.17±.62	88.85±.45
	SL→MSL	SimCLR	99.34±.12	96.66±.18	64.44±.57	35.61±.45	**92.09**±.33	83.88±.56	85.50±.55	96.82±.24
		BYOL	99.37±.10	96.81±.18	**67.90**±.57	**43.59**±.49	91.36±.34	**87.59**±.49	**91.28**±.43	**97.15**±.23

(b) Performance comparison for two-stage schemes.

M Additional Pre-training Schemes

M.1 Alternative Two-Stage Schemes

In this section, we study alternative two-stage pre-training schemes. Namely, we consider MSL→SSL and SSL→MSL. By default, we use $\gamma = 0.875$ as the balancing hyperparameter for MSL within each scheme. However, for MSL→SSL, we also consider $\gamma = 0.125$ as a middle-ground between SL→SSL and MSL→SSL (with $\gamma = 0.875$).

Table 23 and Table 24 describe the few-show performances of the additional pre-training schemes (in the three lowermost rows), with other methods displayed for ease of comparison. We observe that MSL→SSL with either choice of γ can achieve the best performance for target datasets with small similarity to the source dataset, e.g., CropDisease, ISIC, ChestX. On the other hand, SSL→MSL generally underperforms other methods, unable to achieve best performance for any of the target datasets. We posit that this is because the latter stage of pre-training (MSL) is closer to the source dataset compared to the former (SSL), thus learning undesirable source information before few-shot adaptation. For the second stage of MSL→SSL, we used the SGD optimizer for BYOL as well as SimCLR, to match the optimizer used in the first stage, MSL.

Table 23: 5-way 1-shot CD-FSL performance (%) of the models according to different two-stage pre-training schemes. ResNet18 is used as the backbone model, and ImageNet is used as the source data. If not specified otherwise, the balancing coefficient γ in Eq. (3) of MSL is set to be 0.875. The best results are marked in bold.

Pre-train	Method	Small Similarity				Large Similarity			
Scheme		CropDisease	EuroSAT	ISIC	ChestX	Places	Plantae	Cars	CUB
SL → SSL	SimCLR	92.24±.70	**86.51**±.67	36.11±.67	21.75±.41	**71.05**±.92	49.02±.91	37.43±.79	42.40±.85
	BYOL	87.64±.70	74.05±.84	35.62±.65	23.01±.43	58.12±.87	48.28±.88	38.23±.75	42.48±.82
SL → MSL	SimCLR	91.46±.66	77.62±.76	34.46±.64	22.50±.41	69.50±.87	51.27±.91	40.39±.82	62.12±.93
	BYOL	88.37±.73	71.54±.78	36.08±.63	24.42±.45	63.40±.86	**53.65**±.88	**46.62**±.85	**64.33**±.93
MSL → SSL	SimCLR	92.18±.72	86.27±.68	36.20±.68	21.55±.41	67.12±.93	46.61±.88	34.86±.76	41.11±.82
($\gamma = 0.125$)	BYOL	84.33±.75	65.69±.83	36.13±.66	24.60±.44	48.03±.73	45.15±.78	40.62±.77	41.00±.76
MSL → SSL	SimCLR	**92.47**±.69	85.03±.69	36.22±.67	21.73±.41	67.74±.94	46.80±.86	35.08±.77	38.94±.80
($\gamma = 0.875$)	BYOL	92.09±.66	52.65±.90	**37.51**±.65	**24.73**±.44	46.13±.72	45.67±.81	38.80±.80	40.75±.76
SSL → MSL	SimCLR	84.42±.75	73.75±.82	33.80±.60	22.60±.42	62.63±.88	45.21±.82	38.51±.73	53.52±.89
	BYOL	78.26±.82	70.28±.78	35.37±.64	23.66±.41	60.88±.84	45.54±.84	40.27±.76	54.81±.86

Table 24: 5-way 5-shot CD-FSL performance (%) of the models according to different two-stage pre-training schemes. ResNet18 is used as the backbone model, and ImageNet is used as the source data. If not specified otherwise, the balancing coefficient γ in Eq. (3) of MSL is set to be 0.875. The best results are marked in bold.

Pre-train	Method	Small Similarity				Large Similarity			
Scheme		CropDisease	EuroSAT	ISIC	ChestX	Places	Plantae	Cars	CUB
SL → SSL	SimCLR	97.88±.30	**95.28**±.27	48.38±.60	25.25±.44	84.40±.53	66.35±.82	51.31±.84	57.11±.88
	BYOL	97.58±.26	91.82±.39	49.32±.63	28.27±.48	78.87±.60	67.83±.82	54.70±.84	60.60±.82
SL → MSL	SimCLR	97.49±.30	91.70±.35	47.43±.62	26.24±.44	**85.76**±.52	69.24±.81	58.97±.82	81.51±.72
	BYOL	97.09±.31	90.89±.40	50.72±.67	30.20±.48	83.29±.55	**74.16**±.77	**68.87**±.80	**84.34**±.67
MSL → SSL	SimCLR	97.59±.34	95.26±.27	50.22±.60	24.68±.44	83.91±.53	64.91±.82	47.90±.81	56.02±.87
($\gamma = 0.125$)	BYOL	96.89±.30	88.96±.43	52.26±.65	**31.19**±.51	72.16±.62	66.51±.77	58.57±.82	59.73±.80
MSL → SSL	SimCLR	97.76±.32	94.87±.28	48.98±.61	24.92±.43	83.84±.54	64.35±.82	48.07±.82	52.65±.84
($\gamma = 0.875$)	BYOL	**98.41**±.24	85.96±.49	**53.22**±.66	30.67±.49	72.13±.65	67.51±.81	55.17±.82	59.08±.80
SSL → MSL	SimCLR	95.93±.37	89.86±.41	46.38±.63	26.69±.45	82.00±.58	63.38±.82	56.54±.79	73.12±.76
	BYOL	94.76±.38	88.76±.43	48.16±.66	29.23±.48	81.08±.58	64.52±.80	59.02±.79	74.61±.76

M.2 Longer Training for SSL and MSL

Table 25 and Table 26 describe 5-way 1-shot and 5-way 5-shot CD-FSL performance of SSL and MSL when we increase the number of pre-training epochs from 1000 to 2000. We consider two schemes for this ablation: *Extended* and *Repeated*. In the *Extended* scheme, we simply adapt the milestones for the learning rate decay scheduler by doubling them from epochs 400, 600, 800 to 800, 1200, 1600. For the *Repeated* scheme, we apply the existing decay schedule in a cyclical manner; decaying the learning rate by a factor of 10 at epoch {400, 600, 800}, resetting the learning rate at epoch 1000, and again decaying at epoch {1400, 1600, 1800}. This scheme is used to isolate the effects of the learning rate reset that occurs during two-stage pre-training. Note that during two-stage pre-training, learning rate decay is applied to both stages independently, thus resulting in a jump in learning rate during the transition between stages. We follow the same implementation details as single-stage pre-training, unless explicitly stated.

We find that compared to standard single-stage pre-training, *Extended* pre-training for SSL achieves comparable 1-shot performance and minor overall improvement in 5-shot performance–up to 3.41% (Cars). On the other hand, MSL exhibits considerable performance increase overall under *Extended* pre-training. This is magnified under large domain similarity, where 5-shot performance improves by up to 18.02% (Cars). Considering this stark difference between SSL and MSL, we posit that longer training on MSL benefits from further extraction of source features, which are useful for similar target domains. However, we note that two-stage MSL still outperforms longer single-stage MSL, suggesting that SL pre-training is a more effective means of extracting source features. Comparing *Extended* and *Repeated* training, we find no major differences in performance, thus conclude that learning rate reset is not a major contributor in CD-FSL performance.

Table 25: 5-way 1-shot CD-FSL performance (%) of the models pre-trained by SSL and MSL under two schemes of longer pre-training: *Extended* and *Repeated*. ResNet18 is used as the backbone model, and ImageNet is used as the source data for SL. For the standard single-stage and two-stage pre-training, refer to Table 13 for comparison. The balancing coefficient γ in Eq. (3) of MSL is set to be 0.875.

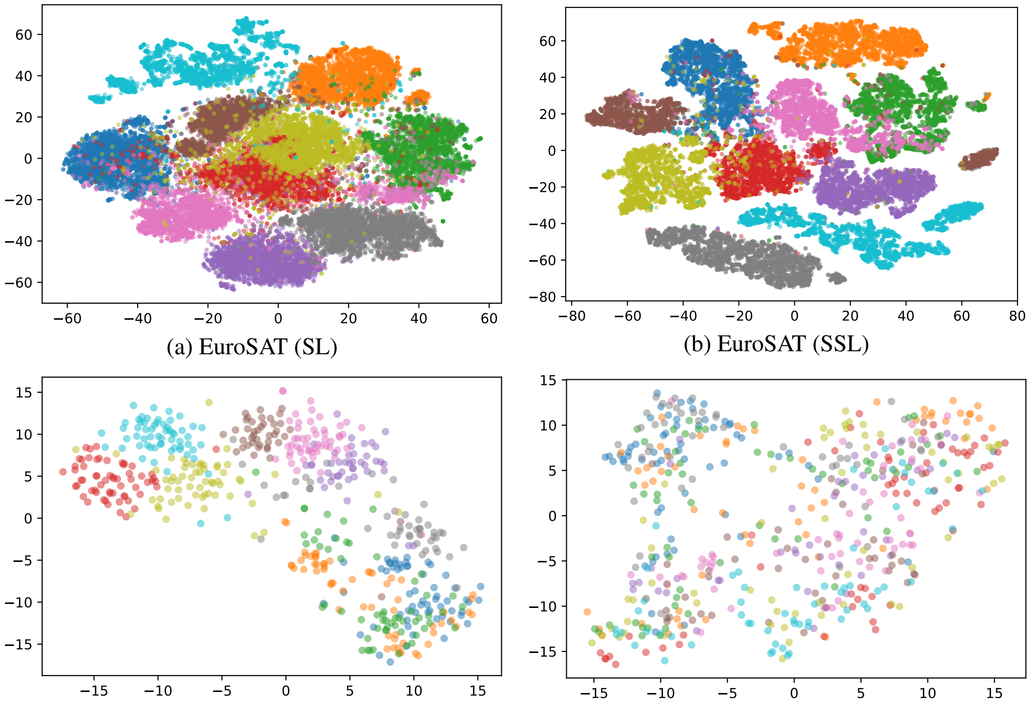
	Pre-train	Method	Small Similarity				Large Similarity			
	Scheme		CropDisease	EuroSAT	ISIC	ChestX	Places	Plantae	Cars	CUB
Extended	SSL	SimCLR	91.98±.76	86.34±.70	35.81±.67	21.44±.40	64.96±.95	43.81±.82	32.94±.72	35.92±.75
		BYOL	87.13±.74	64.33±.81	35.50±.62	23.11±.42	50.46±.82	41.92±.75	35.78±.70	36.67±.68
	MSL	SimCLR	89.30±.69	74.73±.80	34.50±.60	22.32±.42	65.27±.88	48.95±.87	39.70±.80	54.88±.91
		BYOL	90.90±.67	62.86±.86	36.22±.64	24.37±.42	50.94±.79	42.82±.76	44.35±.85	57.18±.91
Repeated	SSL	SimCLR	90.99±.77	85.96±.71	35.78±.64	21.70±.41	66.26±.96	44.53±.85	33.89±.74	36.86±.77
		BYOL	86.36±.78	64.95±.80	35.15±.63	23.48±.43	50.21±.78	41.58±.75	35.73±.72	38.74±.70
	MSL	SimCLR	88.75±.69	73.47±.80	33.58±.61	22.32±.42	65.84±.88	48.64±.85	39.87±.79	54.44±.91
		BYOL	89.70±.71	68.43±.82	36.76±.65	24.17±.44	53.47±.81	46.20±.80	43.80±.84	54.94±.84

Table 26: 5-way 5-shot CD-FSL performance (%) of the models pre-trained by SSL and MSL under two schemes of longer pre-training: *Extended* and *Repeated*. ResNet18 is used as the backbone model, and ImageNet is used as the source data for SL. For the standard single-stage and two-stage pre-training, refer to Table 14 for comparison. The balancing coefficient γ in Eq. (3) of MSL is set to be 0.875.

	Pre-train	Method	Small Similarity				Large Similarity			
	Scheme		CropDisease	EuroSAT	ISIC	ChestX	Places	Plantae	Cars	CUB
Extended	SSL	SimCLR	97.44±.36	95.14±.28	49.47±.59	24.36±.42	82.21±.57	60.77±.85	45.30±.75	47.57±.80
		BYOL	97.29±.28	89.54±.42	50.17±.64	29.16±.49	74.13±.65	62.91±.81	51.97±.78	53.35±.78
	MSL	SimCLR	97.39±.29	91.13±.37	47.28±.64	26.43±.44	84.32±.53	67.82±.82	56.97±.81	73.98±.79
		BYOL	97.87±.26	89.27±.43	50.18±.67	30.82±.49	77.55±.60	66.32±.78	64.78±.81	77.18±.75
Repeated	SSL	SimCLR	97.33±.36	94.99±.28	49.43±.59	24.74±.43	82.54±.56	61.30±.84	46.21±.76	48.43±.81
		BYOL	97.24±.27	89.92±.41	49.95±.63	29.64±.47	74.25±.63	62.86±.80	52.05±.79	55.73±.77
	MSL	SimCLR	97.12±.31	90.73±.38	46.23±.62	26.23±.43	83.94±.56	67.16±.81	56.88±.82	73.31±.81
		BYOL	97.71±.27	86.99±.52	49.60±.69	30.37±.50	78.41±.59	69.56±.80	64.23±.82	75.00±.78

N t-SNE Visualization of Pre-trained Models on the Target Domains

We provide t-SNE on the target datasets to visualize the difference of using the two types of pre-training models: supervised learning on the source domain and self-supervised learning on the target domain. Figure 14 describes t-SNE visualization of representations through SL/SSL models on the EuroSAT and CUB datasets. It is shown that on the EuroSAT dataset, representations through the SSL are clustered better than those through the SL; however, on the CUB dataset, representations through the SL are clustered better than those through the SSL. This implies that clustering ability of extractors trained through SL/SSL is related to the few-shot performance.



(a) EuroSAT (SL)

(b) EuroSAT (SSL)

(c) CUB (SL)

(d) CUB (SSL)

Figure 14: t-SNE visualization for each target dataset. ResNet18 is used as a backbone and ImageNet is used as the source dataset. Note that 10 classes are randomly sampled for the CUB dataset because CUB has 200 classes in total.

O Same Domain FSL Experiments

For the same domain FSL, although label space of the source and target datasets is still not shared, it is expected that SL is a better strategy and MSL improves the few-shot performance because MSL works like multi-task learning (MTL), improving the generalization ability. This is because the source and target datasets were collected in the same way.

To explain same-domain FSL experiments based on *domain similarity* and *few-shot difficulty*, we first provide them for the two datasets:

- Domain Similarity
  - miniImageNet ↔ miniImageNet-test: 0.832
  - tieredImageNet ↔ tieredImageNet-test: 0.869
- Few-shot Difficulty (k = 5)
  - miniImageNet-test: 0.467
  - tieredImageNet-test: 0.414

Interestingly, domain similarity of the miniImageNet-test dataset (0.832) is larger than that of every other benchmark, except for the Places (0.840). The Places dataset is found to be most similar to miniImageNet source. For the tieredImageNet-test dataset, domain similarity with tieredImageNet source (0.869) is larger than that of every other benchmark. Few-shot difficulty of miniImageNet-test is 0.467, which means slightly more difficult than Places and easier than Plantae. In addition, few-shot difficulty of tieredImageNet-test is 0.414, which means more difficult than EuroSAT and easier than Places.

Table 27 describes the few-shot performance under the same domain; miniImageNet → miniImageNet-test and tieredImageNet → tieredImageNet-test. As expected, they have large similarity and high difficulty. Therefore, (1) SL is more powerful than SSL, (2) MSL is a better strategy than both SL and SSL, and (3) two-stage pre-training boosts performance.

Table 27: 5-way k-shot FSL performance of the models pre-trained: miniImageNet → miniImageNet-test and tieredImageNet → tieredImageNet-test. We report the average accuracy and its 95% confidence interval over 600 few-shot episodes. B and S indicate base and strong augmentations, respectively. For MSL and SL→MSL, γ is set to 0.875.

Pre-train Scheme	Method	Aug.	miniImageNet		tieredImageNet	
			k=1	k=5	k=1	k=5
SL	Default	B	54.89±.80	77.92±.59	60.98±.92	78.88±.68
		S	57.30±.81	77.32±.65	60.77±.92	78.36±.71
SSL	SimCLR	B	42.69±.88	60.42±.81	51.63±.93	67.62±.84
		S	54.39±.92	71.62±.79	66.67±1.02	80.60±.75
	BYOL	B	39.32±.76	58.36±.80	48.11±.88	69.72±.80
		S	44.71±.80	63.66±.77	59.00±.97	78.59±.70
MSL	SimCLR	S	63.15±.85	80.03±.63	74.24±.94	86.90±.64
	BYOL	S	58.59±.84	78.17±.65	75.25±.92	88.37±.58
SL→SSL	SimCLR	S	65.48±.89	83.84±.59	68.27±1.04	81.22±.75
	BYOL	S	55.76±.83	78.43±.61	65.78±.99	80.93±.66
SL→MSL	SimCLR	S	**67.24**±.86	**85.02**±.52	74.14±.96	87.38±.62
	BYOL	S	61.10±.82	82.35±.58	**75.47**±.90	**88.72**±.58