# Unsupervised Meta-learning via Few-shot Pseudo-supervised Contrastive Learning

**Huiwon Jang*** **Hankook Lee*** **Jinwoo Shin**
Korea Advanced Institute of Science and Technology (KAIST)
{huiwoen0516, hankook.lee, jinwoos}@kaist.ac.kr

## Abstract

*Unsupervised meta-learning* aims to learn generalizable knowledge across a distribution of tasks constructed from unlabeled data. Here, the main challenge is how to construct diverse tasks for meta-learning without label information; recent works have proposed to create, e.g., pseudo-labeling via pretrained representations or creating synthetic samples via generative models. However, such a task construction strategy is fundamentally limited due to heavy reliance on the immutable pseudo-labels during meta-learning and the quality of the representations or the generated samples. To overcome the limitations, we propose a simple yet effective unsupervised meta-learning framework, coined ***P**seudo-**s**upervised **Co**ntrast (**PsCo**)*, for few-shot classification. We are inspired by the recent self-supervised learning literature; PsCo utilizes a *momentum network* and a *queue* of previous batches to improve pseudo-labeling and construct diverse tasks in a progressive manner. Our extensive experiments demonstrate that PsCo outperforms existing unsupervised meta-learning methods under various in-domain and cross-domain few-shot classification benchmarks. We also validate that PsCo is easily scalable to a large-scale benchmark, while recent prior-art meta-schemes are not.

## 1 Introduction

*Meta-learning* [1] aims to learn general knowledge about how to solve unseen, yet relevant tasks from prior experiences solving diverse tasks. *Few-shot classification* [2, 3] is the most popular application of meta-learning, whose goal is to classify test samples of unseen classes after (meta-)training with few labeled samples. The common approach is to construct a distribution of tasks (i.e., $N$-way $K$-shot) and optimize a model to generalize across tasks. This approach has shown good performance but suffers from limited scalability as constructing tasks requires a lot of human-annotated labels. To mitigate the issue, *unsupervised meta-learning (UML)* [4, 5, 6, 7, 8] attempts to apply meta-learning to unlabeled data. In particular, they have suggested various ways to construct synthetic tasks: assigning pseudo-labels [4, 5]; utilizing generative models [6, 7, 8]. They have achieved moderate performance in few-shot learning benchmarks, but are fundamentally limited as: (a) the pseudo-labels are fixed during meta-learning and impossible to correct mislabeled samples; (b) the generative models heavily rely on the quality of generated samples and are cumbersome to scale into large-scale setups.

To overcome the limitations of the existing UML approaches, in this paper, we ask whether one can (a) progressively improve a pseudo-labeling strategy during meta-learning, and (b) construct more diverse tasks without generative models. We draw inspiration from recent advances in *self-supervised learning* literature [9, 10], which has shown remarkable success in representation learning without labeled data. In particular, we utilize (a) a *momentum network* to improve pseudo-labeling progressively via temporal ensemble; and (b) a *momentum queue* to construct diverse tasks using previous mini-batches in an online manner.

*Equal contribution

6th Workshop on Meta-Learning at NeurIPS 2022, New Orleans.

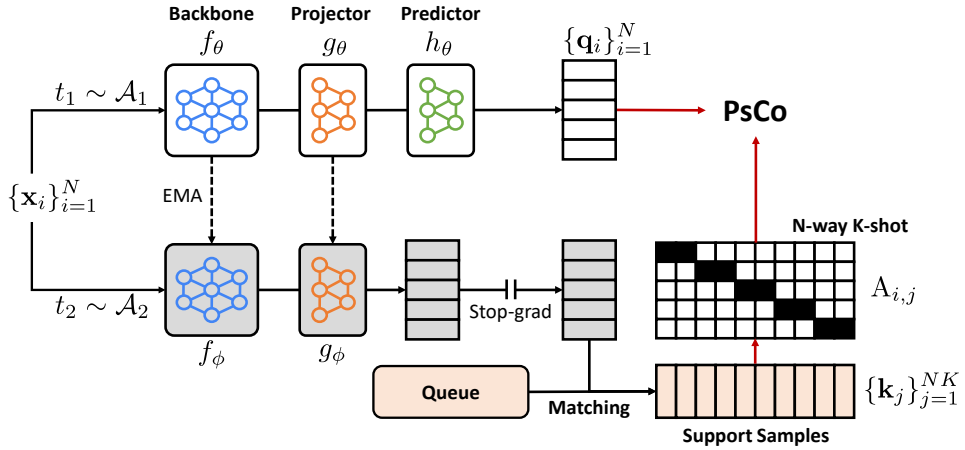



Figure 1: An overview of the proposed **P**seudo-**s**upervised **Co**ntrast (PsCo). PsCo constructs an $N$-way $K$-shot few-shot classification task using the current mini-batch $\{\mathbf{x}_i\}$ and the queue of previous mini-batches; and then, it learns the task via contrastive learning.

Formally, we propose **P**seudo-**s**upervised **Co**ntrast (PsCo), a novel and effective unsupervised meta-learning framework, for few-shot classification. Our key idea is to construct few-shot classification tasks using the current and previous mini-batches based on the momentum network and the momentum queue. Specifically, given a random mini-batch of $N$ unlabeled samples, we treat them as $N$ queries (i.e., test samples) of different $N$ labels, and then select $K$ shots (i.e., training samples) from the queue of previous mini-batches with the momentum network. To further improve the selection procedure, we utilize top-$K$ sampling after applying a matching algorithm, Sinkhorn-Knopp [11]. Finally, we optimize our model via supervised contrastive learning [10] for solving the $N$-way $K$-shot task. Remark that our task construction strategy (a) is progressively improved during meta-learning with the momentum network, and (b) constructs diverse tasks since the shots can be selected from the approximately entire dataset with the queue. Our framework is illustrated in Figure 1.

Throughout extensive experiments, we demonstrate the effectiveness of PsCo, under various few-shot classification benchmarks. First, PsCo achieves state-of-the-art performance under both Omniglot [12] and miniImageNet [13] few-shot benchmarks (e.g., $58.03\% \rightarrow 63.26\%$ for 5-way 5-shot tasks of miniImageNet); its performance is even competitive with supervised meta-learning methods (see Table 1a). Next, PsCo also shows superiority under cross-domain few-shot learning scenarios (see Table 1b). We also demonstrate the scalability of PsCo to a large-scale benchmark in Table 1c.

## 2 Method

In this section, we introduce ***P**seudo-**s**upervised **Co**ntrast* (*PsCo*), a novel and effective framework for unsupervised few-shot learning. The problem statement and contrastive learning are in Section 2.1. The details of our frameworks are in Section 2.2. We provide pseudocode of PsCo in Appendix A.

### 2.1 Preliminaries

**Problem statement.** Our goal is to learn generalizable knowledge from unlabeled data $\mathcal{D}_{\texttt{meta\_train}} := \{\mathbf{x}_i\}$ for quickly adapting to unseen but relevant $N$-way $K$-shot few-shot tasks $\{\mathcal{T}_i\} \sim \mathcal{D}_{\texttt{meta\_test}}$. Each $\mathcal{T}_i$ consists of query samples $\{\mathbf{x}_q\}_q$ and support samples $\mathcal{S} = \{(\mathbf{x}_s, \mathbf{y}_s)\}_{s=1}^{NK}$ where there are $K$ support samples for each label $\mathbf{y}_s \in \{1, \dots, N\}$.

**Contrastive learning** [14, 15, 9, 10] aims to learn meaningful representations by maximizing the similarity between similar (i.e., positive) samples, and minimizing the similarity between dissimilar (i.e., negative) samples on the representation space; the general form of the objectives is as follows:

$$\mathcal{L}_{\texttt{Contrast}}(\{\mathbf{q}_i\}_{i=1}^N, \{\mathbf{k}_j\}_{j=1}^M, \mathbf{A}; \tau) := -\frac{1}{N}\sum_{i=1}^{N}\frac{1}{\sum_{j=1}^{M}\mathrm{A}_{i,j}}\sum_{j=1}^{M}\mathrm{A}_{i,j}\log\frac{\exp(\mathbf{q}_i^\top\mathbf{k}_j/\tau)}{\sum_{k=1}^{M}\exp(\mathbf{q}_i^\top\mathbf{k}_k/\tau)}, \tag{1}$$

where $\mathbf{q}_i$ and $\mathbf{k}_i$ are query and key representations, respectively, and $\mathbf{A} \in \{0,1\}^{NM}$ such that $\mathrm{A}_{i,j} = 1$ if and only if $\mathbf{q}_i$ and $\mathbf{k}_j$ are positive. Following the literature [15, 16, 17, 18], in this paper, we use the following form: $\mathbf{q}_i = \mathrm{Normalize}(h_\theta \circ g_\theta \circ f_\theta(t_{i,1}(\mathbf{x}_i)))$ and $\mathbf{k}_i = \mathrm{Normalize}(g_\phi \circ f_\phi(t_{i,2}(\mathbf{x}_i)))$ where $t$ is a random data augmentation, $f$ is a backbone feature extractor, $g$ and $h$ are projection and prediction MLPs, respectively, $\phi$ is an exponential moving average (EMA) of the model parameter $\theta$.

### 2.2 PsCo: pseudo-supervised contrastive meta-learning

**Online pseudo-task construction.** Let $\mathcal{B} := \{\mathbf{x}_i\}_{i=1}^N \subset \mathcal{D}_{\texttt{meta\_train}}$ be a (random) current mini-batch and $\mathcal{Q} := \{\tilde{\mathbf{x}}_j\}_{j=1}^M$ be a queue of previous mini-batch samples. Our idea is to treat $\mathcal{B}$ as queries of $N$ different pseudo-labels and a subset of $\mathcal{Q}$ as $K$ support samples for each pseudo-label. To utilize all the samples efficiently and consistently, we follow MoCo [9]: we compute the momentum query representations $\mathbf{z}_i := \mathrm{Normalize}(g_\phi \circ f_\phi(t_{i,2}(\mathbf{x}_i)))$ and store them into the queue $\mathcal{Q}_\mathbf{z} := \{\tilde{\mathbf{z}}_j\}_{j=1}^M$.

To construct semantically meaningful few-shot tasks, we need two requirements: (i) shots and queries of the same label should be semantically similar, and (ii) all shots should be different. Based on these requirements, we formulate our assignment problem as follows:

$$\max_{\mathbf{A} \in \{0,1\}^{N \times M}} \sum_{i=1}^{N} \sum_{j=1}^{M} \mathrm{A}_{ij} \cdot \mathbf{z}_i^\top \tilde{\mathbf{z}}_j \quad \text{such that} \quad \sum_j \mathrm{A}_{ij} = K, \quad \sum_i \mathrm{A}_{ij} \leq 1. \tag{2}$$

Due to the expensive cost of solving the above assignment problem [19], we first find a (soft) equal-sized clustering assignment matrix $\tilde{\mathbf{A}} \in [0,1]^{N \times M}$ by Sinkhorn-Knopp [11] following [20, 21]. We then select top-$K$ elements for each row of $\tilde{\mathbf{A}}$ and finally construct an $N$-way $K$-shot pseudo-task consisting of (a) query samples $\mathcal{B} = \{\mathbf{x}_i\}_{i=1}^N$, (b) the support representations $\mathcal{S}_\mathbf{z} := \{\tilde{\mathbf{z}}_s\}_{s=1}^{NK}$, and (c) the pseudo-label assignment matrix $\mathbf{A} \in \{0,1\}^{N \times NK}$. We empirically observe that our task construction strategy satisfies the above requirements (i) and (ii) (see Appendix C).

**Meta-training.** Our objective $\mathcal{L}_{\texttt{PsCo}}$ for learning our pseudo-tasks is defined as follows:

$$\mathcal{L}_{\texttt{PsCo}} := \mathcal{L}_{\texttt{Contrast}}(\{\mathbf{q}_i\}_{i=1}^N, \mathcal{S}_\mathbf{z}, \mathbf{A}; \tau_{\texttt{PsCo}}), \tag{3}$$

where $\mathbf{q}_i := \mathrm{Normalize}(h_\theta \circ g_\theta \circ f_\theta(t_{i,1}(\mathbf{x}_i)))$ is query representations obtained by $\theta$. $\mathcal{S}_\mathbf{z} := \{\tilde{\mathbf{z}}_s\}_{s=1}^{NK}$ and $\mathbf{A} \in \{0,1\}^{N \times NK}$ are constructed by our task construction strategy. Since PsCo and MoCo [9] use the same architectural components, the MoCo objective $\mathcal{L}_{\texttt{MoCo}} := \mathcal{L}_{\texttt{Contrast}}(\{\mathbf{q}_i\}_{i=1}^N, \{\mathbf{z}_i\}_{i=1}^N \cup \mathcal{Q}_\mathbf{z}, \mathbf{A}_{\texttt{MoCo}}; \tau_{\texttt{MoCo}})$ can be incorporated into PsCo without additional costs. Note that $(\mathbf{A}_{\texttt{MoCo}})_{i,j} = 1$ if and only if $i = j$, and $\mathbf{z}_i := \mathrm{Normalize}(g_\phi \circ f_\phi(t_{i,2}(\mathbf{x}_i)))$ following [9]. We optimize our model $\theta$ via $\mathcal{L}_{\texttt{total}} := \mathcal{L}_{\texttt{PsCo}} + \mathcal{L}_{\texttt{MoCo}}$, and we update $\phi$ by EMA, i.e., $\phi \leftarrow m\phi + (1-m)\theta$.

To successfully find the pseudo-label assignment matrix $\mathbf{A}$, we apply *weak augmentations for the momentum representations* as Zheng et al. [22] did. This reduces the noise in the representations and consequently enhances the performance of PsCo as $\mathbf{A}$ becomes more accurate (see Appendix C).

**Meta-test.** At the meta-test stage, we discard the momentum network $\phi$ and use only the online network $\theta$. Given $N$-way $K$-shot task $\mathcal{T}$ consisting of query samples $\{\mathbf{x}_q\}$ and support samples $\mathcal{S} = \{(\mathbf{x}_s, \mathbf{y}_s)\}_{s=1}^{NK}$, we first compute the query representation $\mathbf{q}_q := \mathrm{Normalize}(h_\theta \circ g_\theta \circ f_\theta(\mathbf{x}_q))$ and the support representations $\mathbf{z}_s := \mathrm{Normalize}\left(g_\theta \circ f_\theta(\mathbf{x}_s)\right)$. Then we predict a label by the following rule: $\hat{y} := \arg\max_y \mathbf{q}_q^\top \mathbf{c}_y$ where $\mathbf{c}_y := \mathrm{Normalize}(\sum_s \mathbf{1}_{y_s = y} \cdot \mathbf{z}_s)$ is the prototype vector. This is inspired by our $\mathcal{L}_{\text{PsCo}}$, which can be interpreted as minimizing distance from the mean (i.e., prototype) of the shot representations.[2]

Under cross-domain few-shot classification scenarios, the model $\theta$ should further adapt to the meta-test domain due to the dissimilarity from meta-training. We here suggest an efficient *adaptation scheme* using only a few labeled samples. Our idea is to consider the support samples as queries: compute the query representation $\mathbf{q}_s := \mathrm{Normalize}(h_\theta \circ g_\theta \circ f_\theta(\mathbf{x}_s))$ for each support sample $\mathbf{x}_s$, and construct the label assignment matrix $\mathbf{A}'$ as $\mathrm{A}'_{s,s'} = 1$ if and only if $y_s = y_{s'}$. Then we simply optimize only $g_\theta$ and $h_\theta$ via $\mathcal{L}_{\texttt{Contrast}}(\{\mathbf{q}_s\}, \{\mathbf{z}_s\}, \mathbf{A}'; \tau_{\texttt{PsCo}})$, for few iterations. We empirically observe that this adaptation scheme is effective under cross-domain settings (see Appendix C).

[2] $\mathcal{L}_{\text{PsCo}} = -\frac{1}{N} \sum_i \frac{1}{\tau_{\text{PsCo}}} \mathbf{q}_i^\top \left( \frac{1}{K} \sum_j \mathrm{A}_{i,j} \mathbf{z}_j \right) +$ term not depending on $\mathbf{A}$.

Table 1: Few-shot classification accuracy (%) on standard in-domain and cross-domain benchmarks. **Bold** entries indicate the best for each task configuration, among unsupervised meta-learning methods. MAML [3] and ProtoNet [2] are *supervised* meta-learning baselines using true labels.

(a) Standard few-shot benchmarks: Omniglot [12] and miniImageNet [13].

| | Omniglot (way, shot) | | | | miniImageNet (way, shot) | | | |
|---|---|---|---|---|---|---|---|---|
| **Method** | **(5,1)** | **(5,5)** | **(20,1)** | **(20,5)** | **(5,1)** | **(5,5)** | **(5,20)** | **(5,50)** |
| *Training from Scratch* | 52.50 | 74.78 | 24.91 | 47.62 | 27.59 | 38.48 | 51.53 | 59.63 |
| CACTUs [4] | 68.84 | 87.78 | 48.09 | 73.36 | 39.90 | 53.97 | 63.84 | 69.64 |
| UMTRA [5] | 83.80 | 95.43 | 74.25 | 92.12 | 39.93 | 50.73 | 61.11 | 67.15 |
| LASIUM [6] | 83.26 | 95.29 | - | - | 40.19 | 54.56 | 65.17 | 69.13 |
| Meta-GMVAE [7] | 94.92 | 97.09 | 82.21 | 90.61 | 42.82 | 55.73 | 63.14 | 68.26 |
| Meta-SVEBM [8] | 91.85 | 97.21 | 79.66 | 92.21 | 43.38 | 58.03 | 67.07 | 72.28 |
| **PsCo (Ours)** | **96.37** | **99.13** | **89.64** | **97.07** | **46.70** | **63.26** | **72.22** | **73.50** |
| MAML | 94.46 | 98.83 | 84.60 | 96.29 | 46.81 | 62.13 | 71.03 | 75.54 |
| ProtoNets | 98.35 | 99.58 | 95.31 | 98.81 | 46.56 | 62.29 | 70.05 | 72.04 |

(b) Cross-domain few-shot (5-way 5-shot) benchmarks [23] using miniImageNet-trained Conv5 models.

| **Method** | **CUB** | **Cars** | **Places** | **Plantae** | **CropDiseases** | **EuroSAT** | **ISIC** | **ChestX** |
|---|---|---|---|---|---|---|---|---|
| Meta-GMVAE | 47.48 | 31.39 | 57.70 | 38.27 | 73.56 | 73.83 | 33.48 | 23.23 |
| Meta-SVEBM | 45.50 | 34.27 | 51.27 | 38.12 | 71.82 | 70.83 | 38.85 | **26.26** |
| SimCLR | 52.11 | 37.40 | 60.10 | 43.42 | 79.90 | 79.14 | 42.83 | 25.14 |
| MoCo v2 | 53.23 | 38.65 | 59.09 | 43.97 | 80.96 | 79.94 | 43.43 | 25.24 |
| SwAV | 51.58 | 36.85 | 59.57 | 42.68 | 80.15 | 79.31 | 43.21 | 24.99 |
| **PsCo (Ours)** | **57.37** | **44.01** | **63.60** | **52.72** | **88.24** | **81.08** | **44.00** | 24.78 |
| MAML | 56.57 | 41.17 | 60.05 | 47.33 | 77.76 | 71.48 | 47.34 | 22.61 |
| ProtoNets | 56.74 | 38.98 | 59.39 | 45.89 | 76.01 | 64.91 | 40.62 | 23.15 |

(c) Cross-domain few-shot (5-way 5-shot) benchmarks [23] using ImageNet-trained ResNet-50 models.

| **Method** | **CUB** | **Cars** | **Places** | **Plantae** | **CropDiseases** | **EuroSAT** | **ISIC** | **ChestX** |
|---|---|---|---|---|---|---|---|---|
| MoCo v2 | 64.16 | 47.67 | 81.39 | 61.36 | 82.89 | 76.96 | 38.26 | **24.28** |
| **PsCo (Ours)** | **76.63** | **53.45** | **83.87** | **69.17** | **89.85** | **83.99** | **41.64** | 23.60 |

## 3 Experiments

**Setup.** Following Lee et al. [7], we primarily focus on standard few-shot benchmarks, Omniglot [12] and miniImageNet [13] with Conv4 and Conv5 architectures, respectively. All the implementation, standard and cross-domain benchmark details are described in Appendix D, E, and F, respectively.

**Standard few-shot benchmarks.** Table 1a shows the results of the few-shot classification with various (way, shot) tasks of Omniglot and miniImageNet. The proposed framework, PsCo, achieves state-of-the-art performance on both Omniglot and miniImageNet benchmarks under the unsupervised setting. For example, we obtain 5% accuracy gain (67.07 $\rightarrow$ 72.22) compared to the prior art, Meta-SVEBM [8], on miniImageNet 5-way 20-shot tasks. Moreover, the performance is even competitive with supervised meta-learning methods, ProtoNets [2], and MAML [3] as well.

**Cross-domain few-shot benchmarks.** To further evaluate the generalization ability across more diverse tasks, we evaluate PsCo on cross-domain few-shot classification benchmarks following Oh et al. [23]. The benchmark details are described in Appendix F. We here use our adaptation scheme (Section 2.2) with 50 iterations. We first evaluate various Conv5 models meta-trained on miniImageNet. Table 1b shows that PsCo outperforms all the baselines across all the benchmarks, except ChestX, which is too different from the distribution of miniImageNet [23]. Somewhat interestingly, PsCo performs better than supervised learning under these benchmarks.

We further validate that our meta-learning framework is applicable to the large-scale benchmark, ImageNet [24], using ResNet-50 [25]. Table 1c shows that (i) PsCo consistently outperforms MoCo v2 [16] under this setup (e.g., 12% accuracy gain in CUB), and (ii) PsCo much adds benefits from the large-scale dataset as we obtain a huge amount of performance gain on the benchmarks. These results show that our PsCo is applicable to large-scale unlabeled datasets.

**Ablation study.** We provide ablation experiments to validate PsCo's components in Appendix C.

## References


[1] Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer, 1998.

[2] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 4080–4090, 2017.

[3] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[4] Kyle Hsu, Sergey Levine, and Chelsea Finn. Unsupervised learning via meta-learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[5] Siavash Khodadadeh, Ladislau Bölöni, and Mubarak Shah. Unsupervised meta-learning for few-shot image classification. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 10132–10142, 2019.

[6] Siavash Khodadadeh, Sharare Zehtabian, Saeed Vahidian, Weijia Wang, Bill Lin, and Ladislau Bölоöi. Unsupervised meta-learning through latent-space interpolation in generative models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[7] Dong-Bok Lee, Dongchan Min, Seanie Lee, and Sung Ju Hwang. Meta-gmvae: Mixture of gaussian vae for unsupervised meta-learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[8] Deqian Kong, Bo Pang, and Ying Nian Wu. Unsupervised meta-learning via latent space energy-based model of symbol vector coupling. In *Fifth Workshop on Meta-Learning at the Conference on Neural Information Processing Systems*, 2021.

[9] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9726–9735. IEEE, 2020.

[10] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[11] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2292–2300, 2013.

[12] Brenden M. Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua B. Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the 33th Annual Meeting of the Cognitive Science Society, CogSci 2011, Boston, Massachusetts, USA, July 20-23, 2011*. cognitivescience-society.org, 2011.

[13] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

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

[15] Ting Chen, Simon Kornblith, Mohammad Norouzi 0002, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 2020.

[16] Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *CoRR*, abs/2003.04297, 2020.

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

[18] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[19] Lyle Ramshaw and Robert E Tarjan. On minimum-cost assignments in unbalanced bipartite graphs. *HP Labs, Palo Alto, CA, USA, Tech. Rep. HPL-2012-40R1*, 20, 2012.

[20] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[21] Yuki M. Asano, Christian Rupprecht 0001, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[22] Mingkai Zheng, Shan You, Fei Wang 0032, Chen Qian 0006, Changshui Zhang, Xiaogang Wang 0001, and Chang Xu 0002. Ressl: Relational self-supervised learning with weak augmentation. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 2543–2555, 2021.

[23] Jaehoon Oh, Sungnyun Kim, Namgyu Ho, Jin-Hwa Kim, Hwanjun Song, and Se-Young Yun. Understanding cross-domain few-shot learning: An experimental study. *CoRR*, abs/2202.01339, 2022.

[24] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li 0002. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255. IEEE, 2009.

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

[26] Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. In *Advances in neural information processing systems*, 2019.

[27] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[28] Yee Whye Teh, Max Welling, Simon Osindero, and Geoffrey E. Hinton. Energy-based models for sparse overcomplete representations. *Journal of Machine Learning Research*, 4:1235–1260, 2003.

[29] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015.

[30] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. CURL: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, 2020.

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

[32] Hung-Yu Tseng, Hsin-Ying Lee, Jia-Bin Huang, and Ming-Hsuan Yang 0001. Cross-domain few-shot classification via learned feature-wise transformation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[33] Yunhui Guo, Noel Codella, Leonid Karlinsky, James V. Codella, John R. Smith, Kate Saenko, Tajana Rosing, and Rogério Feris. A broader study of cross-domain few-shot learning. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXVII*, volume 12372 of *Lecture Notes in Computer Science*, pages 124–141. Springer, 2020.

[34] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. Caltech-ucsd birds 200. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[35] Jonathan Krause, Michael Stark 0003, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *2013 IEEE International Conference on Computer Vision Workshops, ICCV Workshops 2013, Sydney, Australia, December 1-8, 2013*, pages 554–561. IEEE Computer Society, 2013.

[36] Bolei Zhou, Àgata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba 0001. Places: A 10 million image database for scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40 (6):1452–1464, 2018.

[37] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alexander Shepard, Hartwig Adam, Pietro Perona, and Serge J. Belongie. The inaturalist species classification and detection dataset. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8769–8778. IEEE Computer Society, 2018.

[38] Sharada P Mohanty, David P Hughes, and Marcel Salathé. Using deep learning for image-based plant disease detection. *Frontiers in plant science*, 7:1419, 2016.

[39] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE J Sel. Topics in Appl. Earth Observ. and Remote Sensing*, 12(7):2217–2226, 2019.

[40] Noel C. F. Codella, David Gutman, M. Emre Celebi, Brian Helba, Michael A. Marchetti, Stephen W. Dusza, Aadi Kalloo, Konstantinos Liopyris, Nabin K. Mishra, Harald Kittler, and Allan Halpern. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In *15th IEEE International Symposium on Biomedical Imaging, ISBI 2018, Washington, DC, USA, April 4-7, 2018*, pages 168–172. IEEE, 2018.

[41] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M. Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 3462–3471. IEEE Computer Society, 2017.

[42] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

# A Pseudocode of PsCo

**Algorithm 1** **P**seudo-**s**upervised **Co**ntrast (PsCo): PyTorch-like Pseudocode

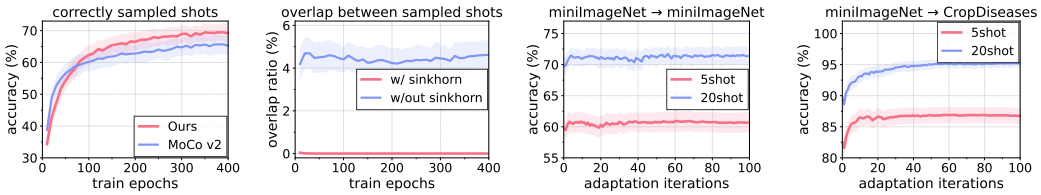
```
# f, g, h: backbone, projector, and predictor
# {f,g}_ema: momentum backbone, and projector
# queue: momentum queue (Mxd)
# mm: matrix multiplication, mul: element-wise multiplication

def PsCo(x):                              # x: a mini-batch of N samples
    x1, x2 = aug1(x), aug2(x)             # two augmented views of x
    q = h(g(f(x1)))                       # (Nxd) N query representations
    z = g_ema(f_ema(x2))                  # (Nxd) N query momentum representations
    sim = mm(z, queue.T)                  # (NxM) similarity matrix
    A_tilde = sinkhorn(sim)               # (NxM) soft pseudo-label assignment matrix
    s, A = select_topK(queue, A_tilde) # (NKxd) s: support momentum representations
                                          # (NxNK) A: pseudo-label assignment matrix
    logits = mm(q, s.T) / temperature
    loss = logits.logsumexp(dim=1) - mul(logits, A).sum(dim=1) / K
    return loss.mean()
```

# B Related works

**Unsupervised meta-learning.** Unsupervised meta-learning [4, 5, 7, 8, 6] links meta-learning and unsupervised learning by constructing synthetic tasks and extracting the meaningful information from unlabeled data. For example, CACTUs [4] cluster the data on the pretrained representations at the beginning of meta-learning to assign pseudo-labels. Instead of pseudo-labeling, UMTRA [5] and LASIUM [6] generate synthetic samples using data augmentations or pretrained generative networks like BigBiGAN [26]. Meta-GMVAE [7] and Meta-SVEBM [8] represent unknown labels via categorical latent variables using variational autoencoders [27] and energy-based models [28], respectively. In this paper, we suggest a novel online pseudo-labeling strategy to construct diverse tasks without help from any pretrained network or generative model. As a result, our method is easily applicable to large-scale datasets.

**Self-supervised learning.** Self-supervised learning (SSL) [29] has shown remarkable success for unsupervised representation learning across various domains, including vision [9, 15], speech [14], and reinforcement learning [30]. Among SSL objectives, contrastive learning [14, 15, 9] is arguably most popular for learning meaningful representations. In addition, recent advances have been made with the development of various architectural components: e.g., Siamese networks [29], momentum networks [9], and asymmetric architectures [18, 31]. In this paper, we utilize the SSL components to construct diverse few-shot tasks in an unsupervised manner.

## C Ablation study




(a) Pseudo-label quality (b) Shot overlap ratio (c) In-domain adaptation (d) Cross-domain adaptation

Figure 2: (a) Pseudo-label quality, measuring the agreement between pseudo-labels and true labels, (b) Shot overlap ratio, measuring whether the shots for each pseudo-label are disjoint, during meta-training. (c,d) Performance while adaptation on in-domain (miniImageNet) and cross-domain (CropDiseases) benchmarks, respectively. We obtain these results from 100 random batches.

Table 2: Component ablation studies on Omniglot.

| Momentum | Predictor | Sinkhorn | Top-K sampling | (5, 1) | (5, 5) | (20, 1) | (20, 5) |
|---|---|---|---|---|---|---|---|
| ✓ | ✓ | ✓ | ✓ | **96.37** | **99.13** | **89.64** | **97.07** |
| ✗ | ✓ | ✓ | ✓ | 90.32 | 96.78 | 76.17 | 90.41 |
| ✓ | ✗ | ✓ | ✓ | 90.21 | 96.86 | 76.15 | 90.53 |
| ✓ | ✓ | ✗ | ✓ | 95.81 | 98.94 | 88.25 | 96.57 |
| ✓ | ✓ | ✓ | ✗ | 94.95 | 98.81 | 86.32 | 96.05 |

**Component analysis.** In Table 2, we demonstrate the necessity of each component in PsCo by removing the components one by one: momentum encoder $\phi$, prediction head $h$, Sinkhorn-Knopp algorithm, and top-$K$ sampling for sampling support samples. We found that the momentum network $\phi$ and the prediction head $h$ are critical architectural components in our framework like recent self-supervised learning frameworks [18, 17]. To further validate that our task construction is progressively improved during meta-learning, we evaluate whether a query and a corresponding support sample have the same true label. Figure 2a shows that our task construction is progressively improved, i.e., the task requirement (i) described in Section 2.2 satisfies.

Table 2 also verifies the contribution of the Sinkhorn-Knopp algorithm and Top-$K$ sampling for the performance of PsCo. We further analyze the effect of the Sinkhorn-Knopp algorithm by measuring the overlap ratio of selected supports between different pseudo-labels. As shown in Figure 2b, there are almost zero overlaps when using the Sinkhorn-Knopp algorithm, which means the constructed task is a valid few-shot task, satisfying the task requirement (ii) described in Section 2.2.

**Adaptation effect on cross-domain.** To validate the effect of our adaptation scheme (Section 2.2), we evaluate the few-shot classification accuracy during the adaptation process on miniImageNet (i.e., in-domain) and CropDiseases (i.e., cross-domain) benchmarks. As shown in Figure 2d, we found that the adaptation scheme is more useful in cross-domain benchmarks than in-domain ones. Based on these results, we apply the scheme to only the cross-domain scenarios. We also found that our adaptation does not cause over-fitting since we only optimize the projection and prediction heads $g_\theta$ and $h_\theta$. The results for the adaptation effect on the whole benchmarks are represented in Table 3.

Table 3: Before and after adaptation of PsCo in few-shot classification.

| Adaptation | miniImageNet | CUB | Cars | Places | Plantae | CropDiseases | EuroSAT | ISIC | ChestX |
|---|---|---|---|---|---|---|---|---|---|
| *5-way 5-shot* | | | | | | | | | |
| ✗ | 63.26 | 55.15 | 42.27 | 62.98 | 48.31 | 79.75 | 74.73 | 41.18 | 24.54 |
| ✓ | **63.30** | **57.38** | **44.01** | **63.60** | **52.72** | **88.24** | **81.08** | **44.00** | **24.78** |
| *5-way 20-shot* | | | | | | | | | |
| ✗ | 72.22 | 62.35 | 51.02 | 70.85 | 55.91 | 84.72 | 78.96 | 48.53 | 27.60 |
| ✓ | **73.00** | **68.58** | **57.50** | **73.95** | **64.53** | **94.95** | **87.65** | **54.59** | **27.69** |

Table 4: The ablation study with varying augmentation choices for $\mathcal{A}_1$ and $\mathcal{A}_2$ on miniImageNet.

| $\mathcal{A}_1$ | $\mathcal{A}_2$ | (5, 1) | (5, 5) | (5, 20) | (5, 50) |
|---|---|---|---|---|---|
| Strong | Strong | 44.54 | 60.04 | 68.61 | 71.20 |
| Strong | Weak | **46.70** | **63.26** | **72.22** | **73.50** |
| Weak | Strong | 43.56 | 58.77 | 67.21 | 69.46 |
| Weak | Weak | 40.68 | 55.05 | 63.32 | 65.82 |

Table 5: The ablation study with varying $K$ on miniImageNet.

| $K$ | (5, 1) | (5, 5) | (5, 20) | (5, 50) |
|---|---|---|---|---|
| 1 | 45.88 | 61.84 | 70.25 | 72.76 |
| 4 | **46.70** | **63.26** | **72.22** | **73.50** |
| 16 | 46.31 | 62.76 | 70.91 | 73.43 |
| 64 | 46.60 | 62.50 | 70.82 | 73.22 |

**Augmentations.** We here confirm that weak augmentation for the momentum network (i.e., $\mathcal{A}_2$) is more effective than strong augmentation unlike other self-supervised learning literature [15, 9]. We denote the standard augmentation consisting of both geometric and color transformations by *Strong*, and a weaker augmentation consisting of only geometric transformations as *Weak* (see details in Appendix D). As shown in Table 4, utilizing the weak augmentation for $\mathcal{A}_2$ is much beneficial since it is helpful for finding an accurate pseudo-label assignment matrix $\mathbf{A}$.

**Training $K$.** We also look at the effect of the training $K$, i.e. number of shots sampled online. We conduct the experiment with $K \in \{1, 4, 16, 64\}$. We observe that PsCo performs consistently well regardless of the choice of $K$ as shown in Table 5. The proper $K$ is suggested to obtain the best-performing models, e.g., $K = 4$ for miniImageNet and $K = 1$ for Omniglot are the best.

Table 6: Sensitivity of momentum $m$ on miniImageNet (way, shot).

| $m$ | **(5, 1)** | **(5, 5)** | **(5, 20)** | **(5, 50)** |
|---|---|---|---|---|
| 0.9 | 46.49 | 62.18 | 70.21 | 72.77 |
| 0.99 | **46.70** | **63.26** | **72.22** | **73.50** |
| 0.999 | 45.96 | 61.53 | 69.66 | 72.04 |

Table 7: Sensitivity of temperature $\tau_{\texttt{PsCo}}$ on miniImageNet (way, shot).

| $\tau_{\texttt{PsCo}}$ | **(5, 1)** | **(5, 5)** | **(5, 20)** | **(5, 50)** |
|---|---|---|---|---|
| 0.2 | 46.43 | 62.29 | 70.04 | 72.22 |
| 0.5 | 46.32 | 62.63 | 70.50 | 73.15 |
| 1.0 | **46.70** | **63.26** | **72.22** | **73.50** |

**Momentum and temperature hyperparameters.** For the small-scale experiments, we use a momentum of $m = 0.99$ and a temperature of $\tau_{\texttt{PsCo}} = 1$. We here provide more ablation experiments with varying the hyperparameters $m$ and $\tau_{\texttt{PsCo}}$. Table 6 and 7 show the sensitivity of hyperparameters on miniImageNet dataset. We observe that PsCo achieves good performance even for non-optimal hyperparameters.

## D Implementation details

We train our models via stochastic gradient descent (SGD) with a batch size of $N = 256$ for 400 epochs. Following Chen et al. [16], Chen and He [31], we use an initial learning rate of 0.03 with the cosine learning schedule, $\tau_{\texttt{MoCo}} = 0.2$, and a weight deacy of $5 \times 10^{-4}$. We use a queue size of $M = 16384$ since Omniglot [12] and miniImageNet [13] has roughly 100k meta-training samples. Following Lee et al. [7], we use Conv4 and Conv5 for Omniglot and miniImageNet, respectively, for the backbone feature extractor $f_\theta$. We describe the detailed architectures in Table 8. For projection and prediction MLPs, $g_\theta$ and $h_\theta$, we use 2-layer MLPs with a hidden size of 2048 and an output dimenension of 128. For the hyperparameters of PsCo, we use $\tau_{\texttt{PsCo}} = 1$ and a momentum parameter of $m = 0.99$ (see Appendix C for the hyperparameter sensitivity). For the number of shots during meta-learning, we use $K = 1$ for Omniglot and $K = 4$ for miniImageNet (see Table 5 for sensitivity of $K$). We use the last-epoch model for evaluation without any guidance from the meta-validation dataset.

Table 8: Pytorch-like architecture descriptions for standard few-shot benchmarks

| Backbone | Layer descriptions | Output shape |
|---|---|---|
| Conv4 | `[Conv2d(3×3, 64 filter), BatchNorm2d, ReLU, MaxPool2d(2×2)]` ×4 | $64 \times 2 \times 2$ |
| Conv5 | `[Conv2d(3×3, 64 filter), BatchNorm2d, ReLU, MaxPool2d(2×2)]` ×5 | $64 \times 2 \times 2$ |

**Augmentations.** We describe the augmentations for Omniglot and miniImagenet in Table 9. For Omniglot, because it is difficult to apply many augmentations to gray-scale images, we use the same rule for weak and strong augmentations. For miniImageNet, we use only geometric transformations for the weak augmentation following Zheng et al. [22].

Table 9: Pytorch-like augmentation descriptions for Omniglot and miniImageNet

| Dataset | Augmentation | Descriptions |
|---|---|---|
| Omniglot | Strong, Weak | `RandomResizeCrop(28, scale=(0.2, 1))`<br>`RandomHorizontalFlip()` |
| miniImagenet | Strong | `RandomResizedCrop(84, scale=(0.2, 1))`<br>`RandomApply([ColorJitter(0.4, 0.4, 0.4, 0.1)], p=0.1)`<br>`RandomGrayScale(p=0.2)`<br>`RandomHorizontalFlip()` |
| | Weak | `RandomResizedCrop(84, scale=(0.2, 1))`<br>`RandomHorizontalFlip()` |

**Training procedures.** To ensure performance of PsCo and self-supervised learning models, we use three independently-trained models with random seeds and report the average performance of them.

# E Setup for standard few-shot benchmarks

We here describe details of benchmarks and baselines in Section E.1 and E.2, respectively, for the standard few-shot classification experiments (Section 3).

## E.1 Datasets

**Omniglot** [12] is a $28 \times 28$ gray-scale dataset of 1623 characters with 20 samples each. We follow the setup of unsupervised meta-learning approaches [4]. We split the dataset into 120, 100, and 323 classes for meta-training, meta-validation, and meta-test respectively. In addition, the 0, 90, 180, and 270 degrees rotated views for each class become the different categories. Thus, we have a total of 6492, 400, and 1292 classes for meta-training, meta-validation, and meta-test respectively.

**MiniImageNet** [13] is an $84 \times 84$ resized subset of ILSVRC-2012 [24] with 600 samples each. We split the dataset into 64, 16, and 20 classes for meta-training, meta-validation, and meta-test respectively as introduced in Ravi and Larochelle [13].

## E.2 Baselines

We compare our performance with unsupervised meta-learning, self-supervised learning, and supervised meta-learning methods. To be specific, (a) for the unsupervised meta-learning, we use CACTUs [4] of the best options (ACAI clustering for Omniglot and DeepCluster for miniImageNet), UMTRA [5], LASIUM [30] of the best options (LASIUM-RO-GAN for Omniglot and LASIUM-N-GAN for miniImageNet), Meta-GMVAE [7], Meta-SVEBM [8]; (b) for the self-supervised learning methods, we use SimCLR [15], MoCo v2 [16], and SwAV [20]; (c) for the supervised meta-learning, we use the results of MAML [3] and ProtoNets [2] reported in [4].

For training self-supervised learning methods in our experimental setups, we use the same architecture and hyperparameters. For the hyperparameter of temperature scaling, we use the value provided in each paper: $\tau_{\texttt{SimCLR}} = 0.5$ for SimCLR, $\tau_{\texttt{MoCo}} = 0.2$ for MoCo v2, and $\tau_{\texttt{SwAV}} = 0.1$ for SwAV. For evaluation, we use K-Nearest Neightobrs (K-NN) for self-supervised learning methods since their classification rules are not specified.

# F Setup for cross-domain few-shot benchmarks

We now describe the setup for cross-domain few-shot benchmarks, including detailed information on datasets, baseline experiments, implementational details, and the setup for large-scale experiments.

## F.1 Datasets

For the cross-domain few-shot benchmarks, we use eight different datasets. We describe the dataset information in Table 10. We use the dataset split described in Tseng et al. [32] for the benchmark of high-similarity and we use the dataset split described in Guo et al. [33] for the benchmark of low-similarity. Because we do not perform the meta-training procedure using the datasets of cross-domain benchmarks, we only utilize the meta-test splits on these datasets. We use the $84 \times 84$ resized samples for evaluation on small-scale experiments.

Table 10: Information of datasets for cross-domain few-shot benchmarks.

| ImageNet similarity | Datset | # of classes | # of samples |
|---|---|---|---|
| High | CUB [34] | 200 | 11,788 |
| | Cars [35] | 196 | 16,185 |
| | Places [36] | 365 | 1,800,000 |
| | Plantae [37] | 5089 | 675,170 |
| Low | CropDiseases [38] | 38 | 43,456 |
| | EuroSAT [39] | 10 | 27,000 |
| | ISIC [40] | 7 | 10,015 |
| | ChestX [41] | 7 | 25,848 |

## F.2 Baselines

We compare our performance with (a) previous in-domain state-of-the-art methods of unsupervised meta-learning, self-supervised learning models, and supervised meta-learning models.

**Unsupervised meta-learning models.** We use previous in-domain state-of-the-art methods of unsupervised meta-learning models, Meta-GMVAE[7] and Meta-SVEBM [8]. We use the miniImageNet pretrained parameters that the paper provided, and follow the meta-test procedure of each model to evaluate the performance.

**Self-supervised learning models.** We use SimCLR [15], MoCo v2 [16], and SwAV [20] of miniImageNet pretrained parameters as our baselines. Because self-supervised learning models are pretrained on miniImageNet, we additionally fine-tune the models with a linear classifier to let the models adapt to each domain. Following the setting provided in Guo et al. [33], Oh et al. [23], we detach the head of the models $g_\theta$ and attach the linear classifier $c_\psi$ to the model. We freeze the base network $f_\theta$ while fine-tuning and only $c_\psi$ is learned. We fine-tune the models via SGD with an initial learning rate of 0.01, a momentum of 0.9, weight decay of 0.001, and a batch size of $N = 4$ for 100 epochs.

**Supervised meta-learning models.** We use MAML [3] and ProtoNets [2] of Conv5 architectures of miniImageNet pretrained. Following the procedure of Snell et al. [2], we train the models via Adam [42] with a learning rate of 0.001 and cut the learning rate in half for every training of 2000 episodes. We train them for 60K episodes and use the model of the best validation accuracy. We train them through a 5-way 5-shot, and the rest of the hyperparameters are referenced in their respective papers. We observe that their performances are similar to the performance described in Table 1.

## F.3 Evaluation details

To evaluate our method, we apply our adaptation scheme. Following Section 2.2, we freeze the base network $f_\theta$. We train only projection head $g_\theta$ and prediction head $h_\theta$ via SGD with an initial learning rate of 0.01, a momentum of 0.9, and weight decay of 0.001 as self-supervised learning models are fine-tuned. We only apply 50 iterations of our adaptation scheme when reporting performance.

### F.4 Large-scale setup

Here, we describe the setup for large-scale experiments. For evaluating, we use the same protocol with the small-scale experiments, except the scale of images is $224 \times 224$.

**Augmentations.** For large-scale experiments, we use $224 \times 224$-scaled data. Thus, we use similar yet slightly different augmentation schemes with small-scale experiments. Following the strong augmentation used in Chen et al. [16, 15], we additionally apply `GaussianBlur` as a random augmentation. We use the same configuration for the weak augmentation. For evaluation, we resize the images into $256 \times 256$ and then apply the `CenterCrop` to make $224 \times 224$ images by following Guo et al. [33].

**ImageNet pretraining.** We pretrain MoCo v2 [16] and our PsCo of ResNet-18/50 [25] via SGD with a batch size of $N = 256$ for 200 epochs. Following [16, 31], we use an initial learning rate of 0.03 with the cosine learning schedule, $\tau_{\text{MoCo}} = 0.2$ and a weight decay of 0.0001. We use a queue size of $M = 65536$ and momentum of $m = 0.999$. For the parameters of PsCo, we use $\tau_{\text{PsCo}} = 0.2$ and $K = 16$ as the queue is 4 times bigger.

## G Experimental results with 95% confidence interval

We here provide the experimental results of Table 1, 1b, and 1c with 95% confidence intervals in Table 11, 12, and 13, respectively.

Table 11: Few-shot classification accuracy (%) on Omniglot and miniImageNet with a 95% confidence interval over 2000 few-shot tasks.

| | Omniglot (way, shot) | | | | miniImageNet (way, shot) | | | |
|---|---|---|---|---|---|---|---|---|
| **Method** | **(5, 1)** | **(5, 5)** | **(20, 1)** | **(20, 5)** | **(5, 1)** | **(5, 5)** | **(5, 20)** | **(5, 50)** |
| SimCLR | 92.13±0.30 | 97.06±0.13 | 80.95±0.21 | 91.60±0.12 | 43.35±0.42 | 52.50±0.39 | 61.83±0.35 | 64.85±0.32 |
| MoCo v2 | 92.66±0.28 | 97.38±0.12 | 82,13±0.21 | 92.34±0.11 | 41.92±0.41 | 50.94±0.38 | 60.23±0.35 | 63.45±0.33 |
| SwAV | 93.13±0.27 | 97.32±0.13 | 82.63±0.21 | 92.12±0.12 | 43.24±0.42 | 52.41±0.39 | 61.36±0.35 | 64.52±0.33 |
| PsCo (ours) | **96.37**±0.20 | **99.13**±0.07 | **89.60**±0.17 | **97.07**±0.07 | **46.70**±0.42 | **63.26**±0.37 | **72.22**±0.32 | **73.50**±0.29 |

Table 12: Few-shot classification accuracy (%) on cross-domain few-shot classification benchmarks of Conv5 pretrained on miniImageNet with a 95% confidence interval over 2000 few-shot tasks.

(a) Cross-domain few-shot benchmarks similar to miniImageNet.

| | CUB | | Cars | | Places | | Plantae | |
|---|---|---|---|---|---|---|---|---|
| **Method** | **(5, 5)** | **(5, 20)** | **(5, 5)** | **(5, 20)** | **(5, 5)** | **(5, 20)** | **(5, 5)** | **(5, 20)** |
| Meta-GMVAE | 47.48±0.47 | 54.08±0.45 | 31.39±0.34 | 38.36±0.35 | 57.70±0.47 | 65.08±0.38 | 38.27±0.40 | 45.02±0.37 |
| Meta-SVEBM | 45.50±0.83 | 54.61±0.91 | 34.27±0.79 | 46.23±0.87 | 51.27±0.82 | 61.09±0.85 | 38.12±0.86 | 46.22±0.85 |
| SimCLR | 52.11±0.45 | 61.89±0.45 | 37.40±0.35 | 50.05±0.39 | 60.10±0.40 | 69.93±0.35 | 43.42±0.37 | 54.92±0.36 |
| MoCo v2 | 53.23±0.45 | 62.81±0.45 | 38.65±0.35 | 51.77±0.39 | 59.09±0.40 | 69.08±0.36 | 43.97±0.37 | 55.45±0.36 |
| SwAV | 51.58±0.45 | 61.38±0.46 | 36.85±0.33 | 50.03±0.38 | 59.57±0.40 | 69.70±0.36 | 42.68±0.37 | 54.03±0.36 |
| PsCo (ours) | **57.38**±0.44 | **68.58**±0.41 | **44.01**±0.39 | **57.50**±0.40 | **63.60**±0.41 | **73.95**±0.36 | **52.72**±0.39 | **64.53**±0.36 |
| *MAML* | 56.57±0.43 | 64.17±0.40 | 41.17±0.40 | 48.82±0.40 | 60.05±0.42 | 67.54±0.37 | 47.33±0.41 | 54.86±0.38 |
| *ProtoNets* | 56.74±0.43 | 65.03±0.41 | 38.98±0.37 | 47.98±0.38 | 59.39±0.40 | 67.77±0.36 | 45.89±0.40 | 54.29±0.38 |

(b) Cross-domain few-shot benchmarks dissimilar to miniImageNet.

| | CropDiseases | | EuroSAT | | ISIC | | ChestX | |
|---|---|---|---|---|---|---|---|---|
| **Method** | **(5, 5)** | **(5, 20)** | **(5, 5)** | **(5, 20)** | **(5, 5)** | **(5, 20)** | **(5, 5)** | **(5, 20)** |
| Meta-GMVAE | 73.56±0.53 | 81.22±0.39 | 73.83±0.42 | 80.11±0.35 | 33.48±0.30 | 39.48±0.28 | 23.23±0.23 | 26.26±0.24 |
| Meta-SVEBM | 71.82±1.03 | 83.13±0.78 | 70.83±0.83 | 80.21±0.73 | 38.85±0.76 | 48.43±0.81 | **26.26**±0.65 | 28.91±0.69 |
| SimCLR | 79.90±0.39 | 88.73±0.28 | 79.14±0.38 | 85.05±0.32 | 42.83±0.29 | 51.35±0.27 | 25.14±0.23 | **29.21**±0.24 |
| MoCo v2 | 80.96±0.37 | 89.85±0.27 | 79.94±0.37 | 86.16±0.31 | 43.43±0.30 | 52.14±0.27 | 25.24±0.23 | 29.19±0.24 |
| SwAV | 80.15±0.39 | 89.24±0.28 | 79.31±0.39 | 85.62±0.31 | 43.21±0.30 | 51.99±0.27 | 24.99±0.23 | 28.57±0.24 |
| PsCo (ours) | **88.24**±0.31 | **94.95**±0.18 | **81.08**±0.35 | **87.65**±0.28 | **44.00**±0.30 | **54.59**±0.29 | 24.78±0.23 | 27.69±0.23 |
| *MAML* | 77.76±0.39 | 83.24±0.34 | 71.48±0.38 | 76.70±0.33 | 47.34±0.37 | 55.09±0.34 | 22.61±0.22 | 24.25±0.22 |
| *ProtoNets* | 76.01±0.40 | 83.64±0.33 | 64.91±0.38 | 70.88±0.33 | 40.62±0.31 | 48.38±0.29 | 23.15±0.22 | 25.72±0.23 |

Table 13: Few-shot classification accuracy (%) on cross-domain few-shot classification benchmarks of pretrained ResNet-18/50 on ImageNet with a 95% confidence interval (5-way 5-shot).

| **Methods** | **CUB** | **Cars** | **Places** | **Plantae** | **CropDiseases** | **EuroSAT** | **ISIC** | **ChestX** |
|---|---|---|---|---|---|---|---|---|
| *ResNet-18 pretrained* | | | | | | | | |
| MoCo v2 | 61.88±0.96 | 46.42±0.73 | 79.11±0.68 | 56.24±0.72 | 81.48±0.74 | 75.98±0.73 | 38.21±0.53 | 24.34±0.36 |
| **PsCo (Ours)** | **70.08**±0.87 | **50.73**±0.76 | **79.74**±0.64 | **61.55**±0.76 | **87.91**±0.57 | **79.92**±0.64 | **40.61**±0.52 | **25.03**±0.42 |
| *ResNet-50 pretrained* | | | | | | | | |
| MoCo v2 | 64.16±0.91 | 47.67±0.75 | 81.39±0.64 | 61.36±0.79 | 82.89±0.77 | 76.96±0.68 | 38.26±0.56 | **24.28**±0.39 |
| **PsCo (Ours)** | **76.63**±0.84 | **53.45**±0.76 | **83.87**±0.58 | **69.17**±0.70 | **89.85**±0.78 | **83.99**±0.52 | **41.64**±0.55 | 23.60±0.36 |