Adaptive Weighted Co-Learning for Cross-Domain Few-Shot Learning

Abdullah Alchihabi[1]
abdullahalchihabi@cmail.carleton.ca
Marzi Heidari[1]
marziheidari@cmail.carleton.ca
Yuhong Guo[1,2]
yuhong.guo@carleton.ca

[1] School of Computer Science
Carleton University
Ottawa, Canada

[2] Canada CIFAR AI Chair
Amii, Canada

Abstract

Due to the availability of only a few labeled instances for the novel target prediction task and the significant domain shift between the well annotated source domain and the target domain, cross-domain few-shot learning (CDFSL) induces a very challenging adaptation problem. In this paper, we propose a simple Adaptive Weighted Co-Learning (AWCoL) method to address the CDFSL challenge by adapting two independently trained source prototypical classification models to the target task in a weighted co-learning manner. The proposed method deploys a weighted moving average prediction strategy to generate probabilistic predictions from each model, and then conducts adaptive co-learning by jointly fine-tuning the two models in an alternating manner based on the pseudo-labels and instance weights produced from the predictions. Moreover, a negative pseudo-labeling regularizer is further deployed to improve the fine-tuning process by penalizing false predictions. Comprehensive experiments are conducted on multiple benchmark datasets and the empirical results demonstrate that the proposed method produces state-of-the-art CDFSL performance.

1 Introduction

Deep learning has achieved great success on a wide set of computer vision tasks, ranging from image classification to image segmentation and object detection. Such success however has been contingent on the availability of a large set of annotated training instances, which induces significant data annotation cost. To alleviate this annotation burden, Few-Shot Learning (FSL) methods have been developed to exploit a base set of classes with a large number of labeled instances to help train deep prediction models for target tasks over novel classes, where only a few instances are labeled for each class. Standard FSL methods are designed for an in-domain setting, where data for the base classes and the novel task are from the same domain, and their performance degrades as the dissimilarity between the base dataset (source domain) and the novel target task (target domain) increases [9], which raises significant demands for cross-domain few-shot learning.

Cross-Domain Few-Shot Learning (CDFSL) expands the FSL study by leveraging labeled base datasets from a remote source domain to help FSL in an annotation-strenuous target domain. CDFSL is much more challenging than its in-domain counterpart due to the limited few-shot labeled instances for the novel target prediction task and the significant domain shift between the source and target domains, and has just started gaining attention from the research community. Some approaches have been developed to tackle CDFSL by employing data augmentation and data generation techniques to increase the number of available labeled instances [13, 36], while several others have deployed ensemble learning or hierarchical variational memory techniques to learn diverse features and facilitate model adaptation from the source domain to the target domain [1, 4]. However, these methods either have limited scalability for higher-shot learning problems [36], induce significant computational cost [1, 4], or require a large set of unlabeled target-domain samples to be available during source-domain training [13].

In this paper, we propose a simple but novel Adaptive Weighted Co-Learning (AWCoL) method for CDFSL. The proposed method jointly fine-tunes two simple prototypical classification models that were independently pre-trained on the source-domain dataset for the target few-shot prediction task in a weighted adaptive co-learning manner. Specifically, for each model we propose to use a weighted moving average (WMA) strategy to generate robust probabilistic predictions for the query instances of the target task, where the class prototypes are computed using the support instances. Pseudo-labels and the corresponding confidence weights for the query instances can be determined based on the average probabilistic predictions produced by the two models. The two prototypical models are then fine-tuned in an alternating fashion on the pseudo-labeled query instances by minimizing a weighted cross-entropy loss, aiming to effectively exploit the more diverse unlabeled queries for stable co-learning. Moreover, we also adopt a negative pseudo-label regularizer to further assist the fine-tuning process by pushing the predictions away from the negative pseudo-labels. To evaluate the proposed method, we conduct comprehensive experiments on eight benchmark datasets for CDFSL. The proposed AWCoL method demonstrates superior performance compared to the existing state-of-the-art CDFSL methods.

2 Related Works

2.1 Few-Shot Learning

Standard in-domain few-shot learning (FSL) methods can be broadly categorized into the following four main groups: generative and augmentation-based methods, transfer learning methods, meta-learning methods, and metric learning methods. The generative and augmentation-based methods generate new instances to increase the training set diversity [10, 19, 24, 26, 39]. The transfer learning methods focus on adapting models pre-trained on the base dataset to the novel target task with fine-tuning techniques [3, 7, 8, 14, 38]. Meta-learning approaches learn to perform few-shot learning by simulating few-shot tasks on the annotated base dataset [5, 16]. Metric learning methods aim to exploit the base dataset to induce good similarity/distance metrics [20, 25, 27, 28, 34]. Among these methods, ProtoNet [27] represents each class using a prototype and learns a metric space in which instances are classified based on their distances to the class prototypes. RelationNet [28], GNN [25] and Transductive Propagation Network (TPN) [20] utilize similarities between the support instances and query instances to perform few-shot learning.

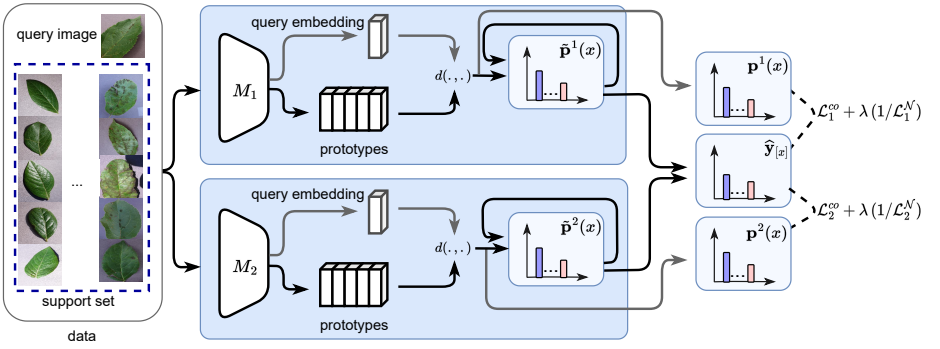

Figure 1: The framework of the proposed Adaptive Weighted Co-Learning (AWCoL) method with two prototypical models, M_1 and M_2, that are pre-trained independently in the source domain: At each fine-tuning iteration in the target domain, each model generates probabilistic predictions for the query instances using a weighted moving average strategy. The predictions from both models are then combined to determine the positive pseudo-labels, negative pseudo-labels, and adaptive weights for the query instances, which are used to fine-tune the two models in the target domain in an adaptive co-learning manner.

2.2 Cross-Domain Few-Shot Learning

By broadening the applicable domains, cross-domain few-shot learning (CDFSL) has recently started drawing more attention [9, 23]. Most works try to induce better generalizable representations for CDFSL. Li *et al*. proposed a ranking distance calibration with fine-tuning (RDC-FT) method that constructs a non-linear subspace to reduce task-irrelevant features [17]. The Hierarchical Variational neural Memory framework (HVM) learns hierarchical prototypes with variational inference [4]. Data augmentation and data generation techniques have also been used for CDFSL [13, 36]. Adversarial Task Augmentation (ATA) has been deployed to improve meta-learning models for cross-domain few-shot classification [36]. Fu *et al*. proposed a meta Style Adversarial training (StyleAdv) method that perturbs the original style of samples using a novel style attack method, enabling the model to generalize to challenging styles and inducing cross-domain robustness [6]. In addition, self-supervised learning based methods have also been developed for CDFSL [2]. Das *et al*. proposed a Contrastive learning and feature selection system (ConFeSS) to bridge the domain shift with self-supervised representation learning [2]. Zhou *et al*. proposed a two-branch Local-global Distillation Prototypical Network (LDP-net) that leverages cross-domain transferable semantic features by enforcing local-global semantic consistency between the two branches through knowledge distillation [41].

A few other works [18, 21, 29, 32] have employed alternative CDFSL settings such as Meta-Dataset [29] and Meta-Album [32] CDFSL settings where models are trained on several source-domain datasets and tested on multiple target-domain datasets. In this work, we focus on the CDFSL setting in [9] as it is the most widely studied CDFSL setting.

3 The Proposed Method

Problem Setup CDFSL aims to train a model in an annotation-abundant source domain, and then adapt/fine-tune the model for the novel few-shot prediction task in the target do-

main. The source domain and the target domain are assumed to have different distributions in the input feature space ($\mathcal{P}_s \neq \mathcal{P}_t$) and disjoint classes ($\mathcal{Y}_s \cap \mathcal{Y}_t = \emptyset$) in the output label space. In the target domain, the model has access to a labelled support set $S = \{(x_i, y_i)\}_{i=1}^{N_s}$ and is evaluated on a query set $Q = \{(x_i, y_i)\}_{i=1}^{N_q}$, where N_s and N_q are the sizes of the support and query sets respectively. The support set typically consists of K instances from each of the N classes, resulting a total of $N_s = N \times K$ labeled support instances. This setup is commonly dubbed as N-way K-shot learning.

In this section, we present the proposed Adaptive Weighted Co-Learning (AWCoL) method to address the CDFSL problem. The overall framework of AWCoL is illustrated in Figure 1. The method provides a simple adaptive co-learning mechanism for fine-tuning two models that were independently pre-trained in the source domain. In particular, we consider pre-training two simple prototypical FSL models in the source domain. The models are then fine-tuned for the target FSL task in the target domain using an adaptive weighted co-learning procedure. In each iteration of the fine-tuning, a weighted moving average (WMA) strategy is applied to each of the two models independently to generate probabilistic predictions for the unlabeled query instances. The predictions from the two models are then combined to produce pseudo-labels and adaptive weights for the query instances, which are subsequently used to fine-tune the models in an adaptive co-learning manner. In addition, negative pseudo-labels generated from the predictions can also be used to enhance the fine-tuning process by pushing the models away from the predicted negative class labels. Overall, the two models are fine-tuned in an alternating fashion to yield a stable co-learning process, which will be elaborated below.

3.1 Source-domain Pre-training

We use a simple prototypical FSL model [27] as our base classification model, due to its simplicity and efficiency. In the well-annotated source domain, we pre-train two classic prototypical few-shot learning models independently using two different sets of randomly sampled N-way K-shot FSL tasks, which we refer to as M_1 and M_2. Each of the two models has its own feature encoder f parameterized by θ that maps an input image x to a feature embedding vector. For each model, given each sampled training FSL task with a support set S and a query set Q, a prototype embedding vector $\mathbf{c}_n$ can be computed from the support instances for each class n using the feature extractor f, such that:

$$\mathbf{c}_n = \frac{1}{K} \sum_{(x,y) \in S_n} f_\theta(x), \tag{1}$$

where S_n denotes the set of K support instances from class n. With the class prototypes, the prototypical classifier predicts the class probability vector $\mathbf{p}(x) = [\mathbf{p}_1(x), \cdots, \mathbf{p}_N(x)]$ for each query instance x by calculating the negative distances between $f_\theta(x)$ and each class prototype vector and then normalizing them using the softmax function, such that:

$$\mathbf{p}_j(x) = \frac{\exp(-d(f_\theta(x), \mathbf{c}_j))}{\sum_{n=1}^{N} \exp(-d(f_\theta(x), \mathbf{c}_n))}, \tag{2}$$

where $d(\cdot,\cdot)$ is a distance function and $\mathbf{p}_j(x)$ is the predicted probability of instance x belonging to class j.

Each model can be trained consecutively on the set of randomly sampled FSL tasks in the source domain by minimizing the following cross-entropy loss over the query instances:

$$\mathcal{L}_{CE}(Q) = \sum_{(x,y) \in Q} \ell_{CE}(\mathbf{1}_y, \mathbf{p}(x)), \tag{3}$$

where ℓ_{CE} is the cross-entropy loss function, $\mathbf{p}(x)$ and $\mathbf{1}_y$ are the predicted class probability vector and the ground-truth one-hot label indicator vector (with value 1 at the y-th entry) respectively for the query instance x. The output of the source-domain pre-training step is two independently trained prototypical classification models, M_1 and M_2, with separate feature encoders f_1 and f_2 parameterized by θ_1 and θ_2 respectively.

3.2 Weighted Moving Average based Probabilistic Prediction Generation

Given the significant domain shift between the source domain and target domain in the CDFSL problem, it is important to adapt the models pre-trained in the source domain to the novel FSL task in the target domain through fine-tuning. Due to the limited number of labeled support instances, S, available for the FSL target task, we propose to exploit the unlabeled query instances, Q, with pseudo-labels for transductive model fine-tuning, aiming to increase training data diversity and avoid over-adaptation.

The pseudo-labels for query instances will be produced based on the predictions of the two models, M_1 and M_2. In order to generate robust predictions across fine-tuning iterations, we adopt a weighted moving average (WMA) strategy to produce probabilistic predictions for the query instances using each of the two models separately. Specifically, let M_m denote either one of the two models, such that $m \in \{1,2\}$. Then at each fine-tuning iteration, we calculate the prototype vector for each class from the support instances with the feature extractor f_m in the given model M_m using Eq. (1). Next for each query instance x, we calculate its class prediction probability vector $\mathbf{p}^m(x)$ using Eq. (2) under the current model M_m. Instead of using this predicted probability vector directly, we maintain a WMA class prediction probability vector $\tilde{\mathbf{p}}^m(x)$ for each query instance x across the fine-tuning iterations, which can be updated in each iteration using the new predictions $\mathbf{p}^m(x)$ as follows:

$$\tilde{\mathbf{p}}^m(x) = (1-\alpha_m)\,\tilde{\mathbf{p}}^m(x) + \alpha_m\,\mathbf{p}^m(x) \tag{4}$$

where $\alpha_m \in (0,1)$ is a trade-off hyper-parameter that controls the combination weights between the class prediction probability vector $\mathbf{p}^m$ of the current iteration and the WMA class prediction probability vector $\tilde{\mathbf{p}}^m$ from the previous iteration, and hence determines the updating degree. With such a WMA update, we expect to maintain stable improvements for pseudo-label predictions over the query instances and avoid local oscillations.

To further improve the stability of the WMA prediction generation, we employ a rectified annealing schedule to update the hyper-parameter α_m in each iteration as follows:

$$\alpha_m = \max(\alpha_{\min}, \gamma\,\alpha_m), \tag{5}$$

where $\alpha_{\min}$ provides a lower bound for α_m, and $\gamma \in (0,1)$ is a reduction ratio hyper-parameter that gradually reduces the α_m value across iterations. This annealing schedule allows the method to perform larger updates to the WMA class prediction probability vectors with larger α_m values in the early fine-tuning iterations, while decreasing the updating degree with smaller α_m in later iterations of fine-tuning, aiming to help the fine-tuning process reach convergence.

3.3 Weighted Adaptive Co-Learning

With the WMA class prediction probability vectors for the query instances produced by M_1 and M_2, we propose to fine-tune the two models together using a weighted adaptive co-

learning procedure by effectively integrating their predictions. Specifically, we simply take a softmax rescaled average of the WMA predictions from the two models to produce the co-prediction probability vector $\tilde{\mathbf{p}}^{\text{co}}(x)$ for each query instance x, such that:

$$\tilde{\mathbf{p}}^{\text{co}}(x) = \text{softmax}(\text{avg}(\tilde{\mathbf{p}}^1(x), \tilde{\mathbf{p}}^2(x))). \tag{6}$$

Here we apply the softmax function to rescale the average probability vector towards less skewed predictions, aiming to avoid early stage poor local optima for the co-learning procedure and help stabilize the fine-tuning process.

The pseudo-labels for the query instances can then be determined based on the integrated co-prediction probability vectors. Specifically, for a query instance x, its pseudo-label $\widehat{y}_{[x]}$ will be a scalar index indicating the class with the highest prediction probability:

$$\widehat{y}_{[x]} = \text{argmax}_{n\in\{1,\cdots,N\}}\ \tilde{\mathbf{p}}^{\text{co}}_n(x) \tag{7}$$

where N is the total number of classes for the target FSL task. For convenience, we further transform the pseudo-label $\widehat{y}_{[x]}$ into a one-hot pseudo-label indicator vector $\widehat{\mathbf{y}}_{[x]} \in \{0,1\}^N$ that has a single 1 value at its $\widehat{y}_{[x]}-$th entry.

By producing pseudo-labels from the integrated predictions of the two models, we expect to yield more accurate pseudo-labels than using each individual model alone. Nevertheless, noisy labels, i.e., mistakes, are still unavoidable among the predicted pseudo-labels. To alleviate the negative impact of the potential pseudo-label noise, we further propose to calculate a dynamic and adaptive weight for each unlabeled query instance x based on the current co-predictions:

$$w_{[x]} = \max_{n\in\{1,\cdots,N\}}\ \tilde{\mathbf{p}}^{\text{co}}_n(x) \tag{8}$$

This adaptive weighting mechanism allows us to assign different weights to different query instances based on the co-prediction confidence of the generated pseudo-labels, such that the query instances with more confident pseudo-labels—less likely to be noise—get larger weights.

With the co-predicted pseudo-labels and adaptive weights for the query instances, we fine-tune the two models using an alternating co-learning procedure; i.e., alternatively updating each model across co-learning iterations. In a given iteration, for the chosen model M_m ($m \in \{1,2\}$), we update its model parameters θ_m by minimizing the following adaptive weighted cross-entropy loss on the pseudo-labeled query instances:

$$\mathcal{L}^{\text{co}}_m = \frac{1}{\sum_{x\in Q} w_{[x]}} \sum_{x\in Q} w_{[x]}\, \ell_{CE}\left(\widehat{\mathbf{y}}_{[x]}, \mathbf{p}^m(x)\right) \tag{9}$$

where $\mathbf{p}^m(x)$ is the prediction probability vector for instance x using the prototypical model M_m via Eq. (2).

3.4 Negative Pseudo-Label Regularization

In addition to exploiting the predicted positive pseudo-labels, we further propose to employ the negative pseudo-labels—i.e., the other class labels except the predicted ones—to regularize the co-learning process. Specifically, for a query instance x, in addition to determining the predicted positive pseudo-label $\widehat{y}_{[x]}$ using Eq. (7) from the integrated co-prediction probability vector $\tilde{\mathbf{p}}^{co}(x)$, we treat all the other classes $\{1,\cdots,N\}\backslash\{\widehat{y}_{[x]}\}$ as candidate negative

labels for x and randomly choose one to use as a negative pseudo-label:

$$\widehat{y}^{\mathcal{N}}_{[x]} = \text{Rand}(\{1,..,N\} \backslash \{\widehat{y}_{[x]}\}) \tag{10}$$

Same as before, we can transform this negative pseudo-label index into a one-hot negative pseudo-label indicator vector $\widehat{\mathbf{y}}^{\mathcal{N}}_{[x]} \in \{0,1\}^N$ with a single 1 at its $\widehat{y}^{\mathcal{N}}_{[x]}-$th entry.

For a multi-class classification problem, only one class label is the true label for each given query instance. Our hypothesis is that even when the predicted positive pseudo-label is a false positive label, a randomly selected negative pseudo-label can still possibly be a true negative label. Hence by minimizing the likelihood of the chosen negative pseudo-labels on query instances, one can push the model away from making true negative predictions and help alleviate the negative impact of the prediction noise in the positive pseudo-labels. To encode this hypothesis, we incorporate the selected negative pseudo-labels into the co-learning process by *maximizing* the following adaptive weighted cross-entropy loss on the query instances for each model M_m ($m \in \{1,2\}$):

$$\mathcal{L}^{\mathcal{N}}_m = \frac{1}{\sum_{x \in Q} w_{[x]}} \sum_{x \in Q} w_{[x]}\, \ell_{CE}\left(\widehat{\mathbf{y}}^{\mathcal{N}}_{[x]}, \mathbf{p}^m(x)\right) \tag{11}$$

where the adaptive weight $w_{[x]}$ and the label prediction probability vector $\mathbf{p}^m(x)$ are same as the ones used for positive pseudo-labels in Eq. (9).

3.5 Alternating Co-Learning Procedure

By integrating both the co-learning minimization loss in Eq. (9) and the negative pseudo-label based maximization loss in Eq. (11) together, we deploy the following adaptive co-learning loss to fine-tune each model M_m ($m \in \{1,2\}$):

$$\mathcal{L}_m = \mathcal{L}^{co}_m + \lambda\,(1/\mathcal{L}^{\mathcal{N}}_m) \tag{12}$$

where λ is a trade-off hyper-parameter that controls the relative contribution of the two loss terms. Minimizing this total loss will simultaneously minimize the positive pseudo-label based weighted adaptive co-learning loss $\mathcal{L}^{co}_m$ and maximize the negative pseudo-label based weighted cross-entropy loss $\mathcal{L}^{\mathcal{N}}_m$. In order to maintain a stable fine-tuning process and prevent oscillating co-prediction updates between the two models, we utilize an alternating updating mechanism to fine-tune the two models with the integrated total loss. In particular, we consecutively update each model for β iterations while keeping the other model fixed, and then switch to update the other model similarly. As a result, when calculating the co-prediction probability vector $\tilde{\mathbf{p}}^{\text{co}}(x)$ in each iteration, only one model's WMA predictions will be updated while anchoring on the fixed WMA predictions of the other model.

4 Experiments

4.1 Experimental Setup

Datasets We conducted comprehensive experiments to evaluate the performance of AW-CoL on eight CDFSL benchmark datasets. Following the standard CDFSL setting, we used Mini-ImageNet [34] as the source domain dataset in all experiments and employed the following eight datasets separately as the target domain datasets: ChestX [37], CropDiseases[22], ISIC [30], EuroSAT [12], Places [40], Planatae [33], Cars [15] and CUB [35]. We used the same train/validation/test splits as in [9] to ensure fair evaluation and comparison.

Table 1: Mean classification accuracy (95% confidence interval in brackets) on 8 target domain datasets for cross-domain 5-way 5-shot classification. "*" and "†" indicate the results reported in [9] and [36] respectively. The best results are indicated in bold font and the second best results are underlined.

	ChestX	CropDisea.	ISIC	EuroSAT	Places	Planatae	Cars	CUB
MatchingNet*[34]	22.40$_{(0.70)}$	66.39$_{(0.78)}$	36.74$_{(0.53)}$	64.45$_{(0.63)}$	–	–	–	–
MAML*[6]	23.48$_{(0.96)}$	78.05$_{(0.68)}$	40.13$_{(0.58)}$	71.70$_{(0.72)}$	–	–	–	–
ProtoNet*[27]	24.05$_{(1.01)}$	79.72$_{(0.67)}$	39.57$_{(0.57)}$	73.29$_{(0.71)}$	58.54$_{(0.68)}$	46.80$_{(0.65)}$	41.74$_{(0.72)}$	55.51$_{(0.68)}$
MetaOpt*[16]	22.53$_{(0.91)}$	68.41$_{(0.73)}$	36.28$_{(0.50)}$	64.44$_{(0.73)}$	–	–	–	–
RelationNet†[28]	24.07$_{(0.20)}$	72.86$_{(0.40)}$	38.60$_{(0.30)}$	65.56$_{(0.40)}$	64.25$_{(0.40)}$	42.71$_{(0.30)}$	40.46$_{(0.40)}$	56.77$_{(0.40)}$
GNN†[25]	23.87$_{(0.20)}$	83.12$_{(0.40)}$	42.54$_{(0.40)}$	78.69$_{(0.40)}$	70.91$_{(0.50)}$	48.51$_{(0.40)}$	43.70$_{(0.40)}$	62.87$_{(0.50)}$
TPN †[20]	22.17$_{(0.20)}$	81.91$_{(0.50)}$	45.66$_{(0.30)}$	77.22$_{(0.40)}$	71.39$_{(0.40)}$	50.96$_{(0.40)}$	44.54$_{(0.40)}$	63.52$_{(0.40)}$
ATA†[36]	24.43$_{(0.20)}$	90.59$_{(0.30)}$	45.83$_{(0.30)}$	83.75$_{(0.40)}$	75.48$_{(0.40)}$	55.08$_{(0.40)}$	49.14$_{(0.40)}$	66.22$_{(0.50)}$
STARTUP [23]	26.94$_{(0.44)}$	93.02$_{(0.45)}$	47.22$_{(0.61)}$	82.29$_{(0.60)}$	–	–	–	–
HVM[4]	**27.15**$_{(0.45)}$	87.65$_{(0.35)}$	42.05$_{(0.34)}$	74.88$_{(0.45)}$	–	–	–	–
ConFeSS [2]	27.09	88.88	48.85	84.65	–	–	–	–
RDC-FT[17]	25.48$_{(0.20)}$	93.55$_{(0.30)}$	49.06$_{(0.30)}$	84.67$_{(0.30)}$	74.65$_{(0.40)}$	60.63$_{(0.40)}$	53.75$_{(0.50)}$	67.77$_{(0.40)}$
LDP-net[41]	26.88$_{(0.46)}$	91.89$_{(0.50)}$	48.44$_{(0.67)}$	84.05$_{(0.66)}$	75.47$_{(0.73)}$	59.64$_{(0.77)}$	53.06$_{(0.82)}$	73.34$_{(0.75)}$
StyleAdv[6]	26.24$_{(0.35)}$	96.51$_{(0.28)}$	53.05$_{(0.54)}$	91.64$_{(0.43)}$	79.35$_{(0.61)}$	64.10$_{(0.64)}$	56.44$_{(0.68)}$	70.90$_{(0.63)}$
AWCoL	24.50$_{(0.33)}$	**99.59**$_{(0.10)}$	**58.75**$_{(0.60)}$	**96.76**$_{(0.27)}$	**92.56**$_{(0.36)}$	**67.31**$_{(0.64)}$	**62.94**$_{(0.75)}$	**86.23**$_{(0.52)}$

Implementation Details We use ResNet10 [11] as the backbone network for the two prototypical classification models in AWCoL. The distance function employed in AWCoL is the squared L2 norm. We train each model in the source domain for 400 iterations with 100 randomly sampled FSL tasks and 15 query instances per class, using the Adam optimizer with a learning rate of 1e-3. We evaluate our proposed AWCoL method on 600 randomly selected few-shot tasks in each target domain. For each task, we randomly sample 5 classes and randomly select 15 images per class as the query set. We fine-tune the models for a total of 100 iterations for each novel target task using the Adam optimizer with a learning rate of 1e-3. Hyper-parameters λ, α_{min}, α_0 (the initial value for α_m), γ, and β take the values of 1e-2, 0.1, 0.5, 0.99 and 5 respectively.

4.2 Comparison Results

4.2.1 Learning with Few Shots

We evaluate the proposed AWCoL method on the standard cross-domain 5-way 5-shot tasks. We compare AWCoL with a set of representative FSL baselines (MatchingNet [34], MAML [6], ProtoNet [27], RelationNet [28], MetaOpt [16], GNN [25] and TPN [20]) as well as 7 state-of-the-art CDFSL methods (ATA [36], STARTUP [23], HVM [4], ConFeSS [2], RDC-FT [17], StyleAdv[6] and LDP-net[41]). The comparison results are reported in Table 1, where the top section of the table presents the results of the standard FSL methods and the bottom section presents the results of the CDFSL methods.

The table shows that most CDFSL methods naturally outperform the FSL methods on all the datasets given their ability to handle the significant domain shift between the source and target domains. The proposed method AWCoL outperforms all the FSL and CDFSL methods on all the datasets except ChestX. Only on ChestX, AWCoL is slightly outperformed by a few other CDFSL methods. Nevertheless, the performance gains produced by AWCoL over all the other methods are notable, exceeding 5%, 13% and 12% on EuroSAT, Places and CUB

Table 2: Mean classification accuracy (95% confidence interval in brackets) on 4 target domain datasets for cross-domain 5-way 20-shot and 50-shot classification. "*" indicates results reported in [9]. The best results are in bold font and the second best results are underlined.

	ChestX		CropDiseases		ISIC		EuroSAT	
	20-shot	50-shot	20-shot	50-shot	20-shot	50-shot	20-shot	50-shot
MatchingNet*[34]	$23.61_{(0.86)}$	$22.12_{(0.88)}$	$76.38_{(0.67)}$	$58.53_{(0.73)}$	$45.72_{(0.53)}$	$54.58_{(0.65)}$	$77.10_{(0.57)}$	$54.44_{(0.67)}$
MAML*[6]	$27.53_{(0.43)}$	–	$89.75_{(0.42)}$	–	$52.36_{(0.57)}$	–	$81.95_{(0.55)}$	–
ProtoNet*[27]	$28.21_{(1.15)}$	$29.32_{(1.12)}$	$88.15_{(0.51)}$	$90.81_{(0.43)}$	$49.50_{(0.55)}$	$51.99_{(0.52)}$	$82.27_{(0.57)}$	$80.48_{(0.57)}$
MetaOpt*[16]	$25.53_{(1.02)}$	$29.35_{(0.99)}$	$82.89_{(0.54)}$	$91.76_{(0.38)}$	$49.42_{(0.60)}$	$54.80_{(0.54)}$	$79.19_{(0.62)}$	$83.62_{(0.58)}$
RelationNet*[28]	$26.63_{(0.92)}$	$28.45_{(1.20)}$	$80.45_{(0.64)}$	$85.08_{(0.53)}$	$41.77_{(0.49)}$	$49.32_{(0.51)}$	$74.43_{(0.66)}$	$74.91_{(0.58)}$
MatchingNet+FWT*[31]	$23.23_{(0.37)}$	$23.01_{(0.34)}$	$74.90_{(0.71)}$	$75.68_{(0.78)}$	$32.01_{(0.48)}$	$33.17_{(0.43)}$	$63.38_{(0.69)}$	$62.75_{(0.76)}$
ProtoNet+FWT*[31]	$26.87_{(0.43)}$	$30.12_{(0.46)}$	$85.82_{(0.51)}$	$87.17_{(0.50)}$	$43.78_{(0.47)}$	$49.84_{(0.51)}$	$75.74_{(0.70)}$	$78.64_{(0.57)}$
RelationNet+FWT*[31]	$26.75_{(0.41)}$	$27.56_{(0.40)}$	$78.43_{(0.59)}$	$81.14_{(0.56)}$	$43.31_{(0.51)}$	$46.38_{(0.53)}$	$69.40_{(0.64)}$	$73.84_{(0.60)}$
STARTUP[23]	$33.19_{(0.46)}$	$\underline{36.91}_{(0.50)}$	$\underline{97.51}_{(0.21)}$	$\underline{98.45}_{(0.17)}$	$58.63_{(0.58)}$	$64.16_{(0.58)}$	$89.26_{(0.43)}$	$91.99_{(0.36)}$
HVM[4]	$30.54_{(0.47)}$	$32.76_{(0.46)}$	$95.13_{(0.35)}$	$97.83_{(0.33)}$	$54.97_{(0.35)}$	$61.71_{(0.32)}$	$84.81_{(0.34)}$	$87.16_{(0.35)}$
ConFeSS [2]	$\underline{33.57}$	**39.02**	95.34	97.56	$\underline{60.10}$	$\underline{65.34}$	$\underline{90.40}$	$\underline{92.66}$
AWCoL	**34.13**$_{(0.47)}$	$36.72_{(0.48)}$	**99.93**$_{(0.03)}$	**99.98**$_{(0.02)}$	**72.40**$_{(0.52)}$	**73.40**$_{(0.51)}$	**98.98**$_{(0.12)}$	**98.73**$_{(0.16)}$

respectively. This highlights the significant performance gains obtained by our proposed method over the existing state-of-the-art CDFSL methods.

4.2.2 Learning with Higher Shots

We also conducted experiments to evaluate the performance of the proposed AWCoL method on higher-shot tasks in the target domain against a set of representative FSL baselines (MatchingNet [34], MAML [6], ProtoNet [27], RelationNet [28] and MetaOpt [16]) as well as 4 state-of-the-art CDFSL methods (FWT [31], STARTUP [23], HVM [4] and ConFeSS [2]). FWT has been applied jointly with three standard FSL methods: MatchingNet, ProtoNet and RelationNet. In particular, we evaluate our proposed AWCoL on 5-way 20-shot and 5-way 50-shot learning tasks on four target-domain datasets: ChestX, CropDiseases, ISIC and EuroSAT. The results are presented in Table 2, where the top section of the table reports results of the standard FSL methods and the bottom section reports results of the CDFSL methods.

Similar to the comparisons on 5-shot tasks, most CDFSL methods outperform the in-domain FSL methods across all datasets for both 20-shot and 50-shot tasks. It is also clear from the table that AWCoL outperforms all the other FSL and CDFSL methods on the CropDiseases, ISIC and EuroSAT datasets for both 20-shot and 50-shot learning tasks. The performance gains are notable, exceeding 12% and 8% on the ISIC dataset for the 20-shot and 50-shot learning tasks respectively. AWCoL is slightly outperformed on ChestX by ConFess and STARTUP in the case of 50-shot tasks. Nevertheless our proposed method still obtains the best result and third best result on ChestX for the cases of 20-shot and 50-shot tasks respectively.

4.3 Ablation Study

In order to investigate the contribution of each component of the proposed method, we conducted an ablation study to compare AWCoL with its seven variants: (1) "−w/o Co-Learn.", which drops the adaptive co-learning component and fine-tunes each model independently using its own WMA predictions. (2) "−w/o Alt. Update", which updates the two models

Table 3: Ablation study results in terms of mean classification accuracy (95% confidence interval within brackets) for cross-domain 5-way 5-shot classification tasks.

	ChestX	CropDisea.	ISIC	EuroSAT	Places	Planatae	Cars	CUB
AWCoL	$24.50_{(0.33)}$	$\mathbf{99.59}_{(0.10)}$	$\mathbf{58.75}_{(0.60)}$	$96.76_{(0.27)}$	$92.56_{(0.36)}$	$\mathbf{67.31}_{(0.64)}$	$\mathbf{62.94}_{(0.75)}$	$\mathbf{86.23}_{(0.52)}$
−w/o Co-Learn.	$24.08_{(0.42)}$	$81.65_{(0.62)}$	$42.93_{(0.60)}$	$72.72_{(0.72)}$	$69.15_{(0.73)}$	$46.16_{(0.65)}$	$42.60_{(0.72)}$	$56.35_{(0.73)}$
−w/o Alt. Update	$\mathbf{24.79}_{(0.40)}$	$84.31_{(0.63)}$	$44.42_{(0.61)}$	$75.34_{(0.69)}$	$68.91_{(0.70)}$	$49.40_{(0.67)}$	$44.53_{(0.77)}$	$58.63_{(0.72)}$
−w/o WMA	$24.47_{(0.44)}$	$84.19_{(0.62)}$	$43.96_{(0.63)}$	$75.31_{(0.70)}$	$68.85_{(0.71)}$	$48.19_{(0.67)}$	$40.05_{(0.68)}$	$58.05_{(0.70)}$
−w/o $\mathcal{L}_m^{co}$	$23.45_{(0.41)}$	$84.74_{(0.62)}$	$41.83_{(0.62)}$	$71.16_{(0.69)}$	$69.77_{(0.73)}$	$47.72_{(0.69)}$	$43.30_{(0.75)}$	$56.07_{(0.71)}$
−w/o $\mathcal{L}_m^{\mathcal{N}}$	$23.22_{(0.33)}$	$99.50_{(0.12)}$	$57.82_{(0.65)}$	$\mathbf{96.86}_{(0.24)}$	$\mathbf{93.13}_{(0.33)}$	$66.90_{(0.65)}$	$62.90_{(0.69)}$	$85.28_{(0.51)}$
−w/o Adapt. Weight	$23.62_{(0.34)}$	$98.52_{(0.09)}$	$57.45_{(0.62)}$	$96.65_{(0.27)}$	$\mathbf{93.06}_{(0.36)}$	$62.23_{(0.75)}$	$61.54_{(0.77)}$	$84.61_{(0.52)}$
−with $\mathcal{L}_m^{S}$	$22.28_{(0.26)}$	$99.22_{(0.12)}$	$54.98_{(0.63)}$	$93.97_{(0.36)}$	$86.94_{(0.49)}$	$46.38_{(0.64)}$	$37.28_{(0.70)}$	$77.77_{(0.57)}$

simultaneously instead of using the alternating update strategy. (3) "−w/o WMA", which drops the WMA prediction generating strategy by setting $\alpha_m = 1$. (4) "−w/o $\mathcal{L}_m^{co}$", which drops the adaptive co-learning loss term ($\mathcal{L}_m^{co}$) from the total loss function. (5) "−w/o $\mathcal{L}_m^{\mathcal{N}}$", which drops the negative pseudo-labeling regularization loss term ($\mathcal{L}_m^{\mathcal{N}}$) from the total loss function. (6) "−w/o Adapt. Weight", which drops the adaptive weighting component of AWCoL by setting all $w_{[x]} = 1$. (7) "−with $\mathcal{L}_m^{S}$", which adds a cross-entropy loss term on the labeled support instances to the total loss function for each model.

We compare AWCoL with all of its seven variants using the cross-domain 5-way 5-shot learning tasks on all the 8 datasets and report the results in Table 3. From the table, we can see that dropping any component from the proposed full model results in performance degradation in all the 48 cases except for only 4 cases: the "−w/o Alt. Update" variant on ChestX, the "−w/o Adapt. Weight" variant on Places, and the "−w/o $\mathcal{L}_m^{\mathcal{N}}$" variant on EuroSAT and Places. Even with the exception on ChestX, the "−w/o Alt. Update" variant demonstrates substantial performance drops from the full AWCoL method on all the other datasets, ranging from 14% (on ISIC) to 27% (on CUB). Such consistent performance drops across many datasets validate the essential contribution of each corresponding component of the proposed AWCoL. Meanwhile, with an additional loss term on the support instances, the "−with $\mathcal{L}_m^{S}$" variant produces results that are much inferior to the proposed AWCoL across all the 8 datasets. The possible reason is that the labeled support instances have already been used to produce the class prototypes for the cross-entropy loss on the query instances in AWCoL, hence an additional cross-entropy loss $\mathcal{L}_m^{S}$ may overfit the limited support instances.

5 Conclusion

In this paper, we proposed a weighted adaptive co-learning method to address the challenging cross-domain few-shot learning problem. The proposed method fine-tunes two prototypical classification models independently pre-trained in the source domain for the target FSL task. In each co-learning iteration, a weighted moving average strategy is deployed to generate probability predictions for the query instances using each model separately. The predictions of the two models are then combined to produce positive pseudo-labels, negative pseudo-labels, and adaptive weights for the query instances. We adopted an alternating fine-tuning mechanism to update each model separately by minimizing the weighted cross-entropy loss over the pseudo-labeled query instances while maximizing a similar cross-entropy loss with negative pseudo-labels to penalize false predictions. We conducted extensive experiments on eight CDFSL benchmark datasets. The results demonstrated the effectiveness of the proposed simple method compared to the state-of-the-art baselines in the literature.

References

[1] Thomas Adler, Johannes Brandstetter, Michael Widrich, Andreas Mayr, David Kreil, Michael Kopp, Günter Klambauer, and Sepp Hochreiter. Cross-domain few-shot learning by representation fusion. In *arXiv preprint arXiv:2010.06498*, 2020.

[2] Debasmit Das, Sungrack Yun, and Fatih Porikli. ConfeSS: A framework for single source cross-domain few-shot learning. In *International Conference on Learning Representations (ICLR)*, 2022.

[3] Guneet Singh Dhillon, Pratik Chaudhari, Avinash Ravichandran, and Stefano Soatto. A baseline for few-shot image classification. In *International Conference on Learning Representations (ICLR)*, 2019.

[4] Yingjun Du, Xiantong Zhen, Ling Shao, and Cees G. M. Snoek. Hierarchical variational memory for few-shot learning across domains. In *International Conference on Learning Representations (ICLR)*, 2022.

[5] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning (ICML)*, 2017.

[6] Yuqian Fu, Yu Xie, Yanwei Fu, and Yu-Gang Jiang. Styleadv: Meta style adversarial training for cross-domain few-shot learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[7] Weifeng Ge and Yizhou Yu. Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[8] Yunhui Guo, Honghui Shi, Abhishek Kumar, Kristen Grauman, Tajana Rosing, and Rogerio Feris. Spottune: transfer learning through adaptive fine-tuning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[9] Yunhui Guo, Noel C Codella, Leonid Karlinsky, James V Codella, John R Smith, Kate Saenko, Tajana Rosing, and Rogerio Feris. A broader study of cross-domain few-shot learning. In *European Conference on Computer Vision (ECCV)*, 2020.

[10] Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *International Conference on Computer Vision (ICCV)*, 2017.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[12] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. In *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019.

[13] Ashraful Islam, Chun-Fu Richard Chen, Rameswar Panda, Leonid Karlinsky, Rogerio Feris, and Richard J Radke. Dynamic distillation network for cross-domain few-shot recognition with unlabeled data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[14] Taewon Jeong and Heeyoung Kim. Ood-maml: Meta-learning for few-shot out-of-distribution detection and classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[15] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *International Conference on Computer Vision workshops*, 2013.

[16] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[17] Pan Li, Shaogang Gong, Chengjie Wang, and Yanwei Fu. Ranking distance calibration for cross-domain few-shot learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[18] Wei-Hong Li, Xialei Liu, and Hakan Bilen. Cross-domain few-shot learning with task-specific adapters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7161–7170, 2022.

[19] Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[20] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sungju Hwang, and Yi Yang. Learning to propagate labels: Transductive propagation network for few-shot learning. In *International Conference on Learning Representations (ICLR)*, 2019.

[21] Yanbin Liu, Juho Lee, Linchao Zhu, Ling Chen, Humphrey Shi, and Yi Yang. A multi-mode modulator for multi-domain few-shot classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8453–8462, 2021.

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

[23] Cheng Perng Phoo and Bharath Hariharan. Self-training for few-shot transfer across extreme task differences. In *International Conference on Learning Representations (ICLR)*, 2020.

[24] Scott Reed, Yutian Chen, Thomas Paine, Aäron van den Oord, SM Ali Eslami, Danilo Rezende, Oriol Vinyals, and Nando de Freitas. Few-shot autoregressive density estimation: Towards learning to learn distributions. In *International Conference on Learning Representations (ICLR)*, 2018.

[25] Victor Garcia Satorras and Joan Bruna Estrach. Few-shot learning with graph neural networks. In *International Conference on Learning Representations (ICLR)*, 2018.

[26] Eli Schwartz, Leonid Karlinsky, Joseph Shtok, Sivan Harary, Mattias Marder, Abhishek Kumar, Rogerio Feris, Raja Giryes, and Alex Bronstein. Delta-encoder: an effective sample synthesis method for few-shot object recognition. *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[27] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[28] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[29] Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, et al. Meta-dataset: A dataset of datasets for learning to learn from few examples. *arXiv preprint arXiv:1903.03096*, 2019.

[30] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. In *Scientific data*, 2018.

[31] Hung-Yu Tseng, Hsin-Ying Lee, Jia-Bin Huang, and Ming-Hsuan Yang. Cross-domain few-shot classification via learned feature-wise transformation. In *International Conference on Learning Representations (ICLR)*, 2020.

[32] Ihsan Ullah, Dustin Carrión-Ojeda, Sergio Escalera, Isabelle Guyon, Mike Huisman, Felix Mohr, Jan N van Rijn, Haozhe Sun, Joaquin Vanschoren, and Phan Anh Vu. Meta-album: Multi-domain meta-dataset for few-shot image classification. *arXiv preprint arXiv:2302.08909*, 2023.

[33] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[34] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[35] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. California Institute of Technology, 2011.

[36] Haoqing Wang and Zhi-Hong Deng. Cross-domain few-shot classification via adversarial task augmentation. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

[37] 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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[38] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[39] Ruixiang Zhang, Tong Che, Zoubin Ghahramani, Yoshua Bengio, and Yangqiu Song. Metagan: An adversarial approach to few-shot learning. In *Advances in Neural information processing systems (NeurIPS)*, 2018.

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

[41] Fei Zhou, Peng Wang, Lei Zhang, Wei Wei, and Yanning Zhang. Revisiting prototypical network for cross domain few-shot learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.