MARRS: Modern Backbones Assisted Co-training for Rapid and Robust Semi-Supervised Domain Adaptation

Saurabh Kumar Jain Sukhendu Das
Visualization and Perception Lab, Department of Computer Science
Engineering, Indian Institute of Technology, Madras, India
cs21s043@cse.iitm.ac.in, sdas@iitm.ac.in

Abstract

*Semi-Supervised Domain Adaptation (SSDA) aims to develop domain invariant models from scarcely labeled target domain in addition to the fully labeled source domain. Current SSDA works are often applied in conjunction with ResNet34 backbone, which makes them overlook the advantages of utilizing other backbones. Hence, in this paper, we investigate the impact of employing different modern backbones in SSDA and propose a novel solution named **M**odern Backbones **A**ssisted Co-training for **R**apid and **R**obust **S**emi-Supervised Domain Adaptation (MARRS), that uses discriminative features of two modern backbones for training linear classifiers using the well established co-training framework. To induce diversity among classifiers for effective co-training, we propose a novel module that produces diversity at three levels, namely image, backbone, and feature distribution levels. Experiments reveal that MARRS not only achieves state-of-the-art performance across all popular SSDA datasets, but also drastically cuts the computation time compared to other SSDA approaches, making MARRS a rapid and robust solution for SSDA. We also provide extensive ablation experiments to verify our framework's vitality and primary design choices.*

1. Introduction

Semi-Supervised Domain Adaptation (SSDA) is the special case of unsupervised domain adaptation (UDA) [9, 23, 44] where only a few labeled samples from the target domain are available. SSDA is the more practical problem because it requires minimal labeling effort and still offers a promising boost in performance in comparison with unsupervised domain adaptation (UDA). Current SSDA techniques [16–18, 33, 38, 46, 47] proposed various methodologies and loss functions to reduce inter-domain and intra-domain gap, by using conventional ResNet34 [12] backbone. But, ResNet34 as a backbone has primarily two drawbacks. First, from the ***computation aspect***, due to the less discriminative ResNet34 features, current SSDA works have to use complex learning techniques along with intricate loss functions during training, which help them to boost accuracy to a certain level but makes them computationally expensive (cf. Fig. 6). Second, from the ***accuracy aspect***, we discovered that a simple combination of modern backbone ConvNeXt-XL [22] with a basic semi-supervised learning technique ENT [11] achieves a mean accuracy of 75.8% on DomainNet [30], 1-shot task, better than best mean accuracy reported using ResNet34 backbone. These drawbacks demonstrate that the tradition of using ResNet34 backbone in SSDA is outdated as well as inefficient, and motivates us to explore the use of modern backbones in SSDA. However, our study (cf. Table 4) reveals that directly utilizing modern backbones or an ensemble of them in SSDA will not fully exploit their strengths and results into relatively smaller gains. This raises the

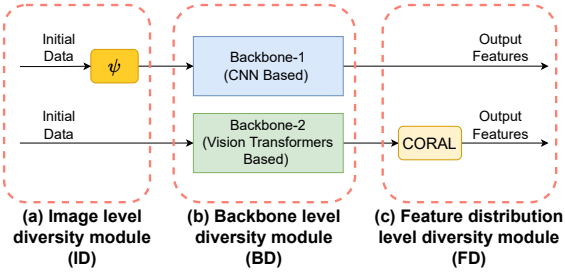


Figure 1. **Outline of the proposed diversity module. (a) ID:** By applying weak augmentation (ψ) on only one of the backbones initial data. **(b) BD:** By employing different family backbones, *i.e.* one from CNN family and another from Vision Transformers family. **(c) FD:** By using CORAL on output features of only one of the backbones. Our work applies ψ at backbone-1 and CORAL at backbone-2. However, other three designs based on the proposed diversity module can be obtained by applying (i) ψ at backbone-2 and CORAL at backbone-1; (ii) both ψ and CORAL at backbone-1; (iii) both ψ and CORAL at backbone-2. Notably, these designs are also effective for inducing diversity (cf. Table 3).

question: *is it possible to attain more significant gains in SSDA without using any complex adaptation techniques, by carefully leveraging high-quality generalizable and discriminative features from modern backbones?*. The answer is *yes*, our framework **MARRS** not only achieves state-of-the-art performance, but also in a relatively shorter amount of time which exhibits the practical advantage of our work in situations where both accuracy and computation cost are important concerns (*e.g*. edge computing [36]). MARRS consists of the following stages:

(i) Feature extraction: At this stage, we propose a novel diversity module which is shown in Fig. 1 for obtaining diverse views of data. It introduces diversity at image, backbone and feature distribution levels which makes resultant classifiers diverse in three different dimensions. Importantly, at feature distribution level, we propose the novel idea of employing CORAL [40] to induce diversity. Using the proposed diversity module, we extract and store the features of the initial data by taking ConvNeXt-XL [22] and Swin-L [21] as two modern backbones. These features extracted once and remains unaltered throughout the training, which makes our method computationally efficient.

(ii) Classifier training: After obtaining diverse features of the data from the first stage, we train two linear classifiers. To combine their strengths, we use co-training [2]. The main idea of co-training is to train classifiers using their labeled data and then use them to create pseudo-labels for each other on unlabeled data. To learn more compact representations, we also use consistency regularization [1,39,45] which is a powerful solution in semi-supervised learning.

By combining the above two stages, we obtain our novel SSDA framework, MARRS. Further, to make our framework suitable for deploying in resource constrained situations, we use knowledge distillation [14,25,26,49] to transfer the knowledge from our MARRS-trained classifiers to a smaller model like MobileNetV2 [35].

Following are the main contributions of our work: (1) We develop a novel SSDA solution, MARRS, which utilizes strong transferable feature representation of modern backbones by training two linear classifiers via co-training. (2) A novel diversity module is proposed to make the classifiers diverse at three levels during co-training. Extensive experiments backed with ablation studies show the vitality of our diversity module. (3) MARRS is the first to explore the effective use of modern backbones in SSDA and achieves state-of-the-art results across all popular SSDA datasets in relatively less time. Hence, it successfully addresses both drawbacks ($i.e.$ low accuracy and high computation time) of training with ResNet34 backbone. (4) we also train a smaller model MobileNetV2 containing only 3.4M parameters, which makes deployment of SSDA algorithms feasible even in mobile and AR/VR devices. Experiments reveal that the MobileNetV2 results are also superior to the results of all previous SSDA methods.

2. Related Work

Semi-Supervised Domain Adaptation (SSDA). Unlike UDA [9,10,24], SSDA reduces the discrepancy between the source and target distributions by using few labeled samples from the target domain. MME [33] proposed to solve the SSDA problem by using adversarial minimax loss. Further, APE [16] introduces the intra-domain discrepancy problem in SSDA and solves it using three techniques attraction, perturbation, and exploration. CLDA [38] uses class-wise and instance-wise contrastive losses to reduce inter-domain and intra-domain gap respectively. Recently, MCL [46] propose to use inter-domain and intra-domain consistency learning for solving SSDA. But, all these methods share a common practice of using conventional ResNet34 [12] backbone, which limits their applicability in complex domain adaptation settings. However, the utilization of modern backbones in SSDA is still unexplored. Notably, one parallel work PACE [20] is recently reported on arXiv. But, it uses an ensemble of 28 modern backbones, which makes it unsuitable for memory-constrained situations. In contrast, our work investigates the effective usage of backbones and leverages only two modern backbones to get superior results across all datasets in a significantly shorter period of time.

Co-training. Disagreement-based learning [50] is a very important technique in semi-supervised learning, and co-training [2, 4, 5] is one of the representative of it. In co-training, each model is trained on confident predictions of the other model. Diversity among members is a vital requirement in any disagreement-based learning technique. Hence, some prior works [29,31] use adversarial techniques to generate diverse views, but it makes their learning complex and unstable, which often results in the generation of absurd views. Some works [34, 48] propose to use classifiers with different initialization but using them with the same backbone results in less assurance of learning different and complementary information. For image classification, [8] shows the advantages of using image level and feature level augmentations together. Yet, to the best of our knowledge, no work on disagreement based learning has taken advantage of this crucial finding. We address all these mentioned shortcomings by proposing a stable and learning-free module that efficiently introduces diversity among classifiers at image, backbone and feature distribution levels.

3. Methodology

In SSDA, we have access to label rich source dataset $\mathcal{D}_s = \left\{\left(x_s^i, y_s^i\right)\right\}_{i=1}^{n_s}$ sampled from a distribution $P_S(X, Y)$ and labeled target dataset with less number of annotated samples $\mathcal{D}_{tl} = \left\{\left(x_{tl}^i, y_{tl}^i\right)\right\}_{i=1}^{n_{tl}}$, along with a relatively large number of unlabeled samples $\mathcal{D}_{tu} = \left\{x_{tu}^i\right\}_{i=1}^{n_{tu}}$ from

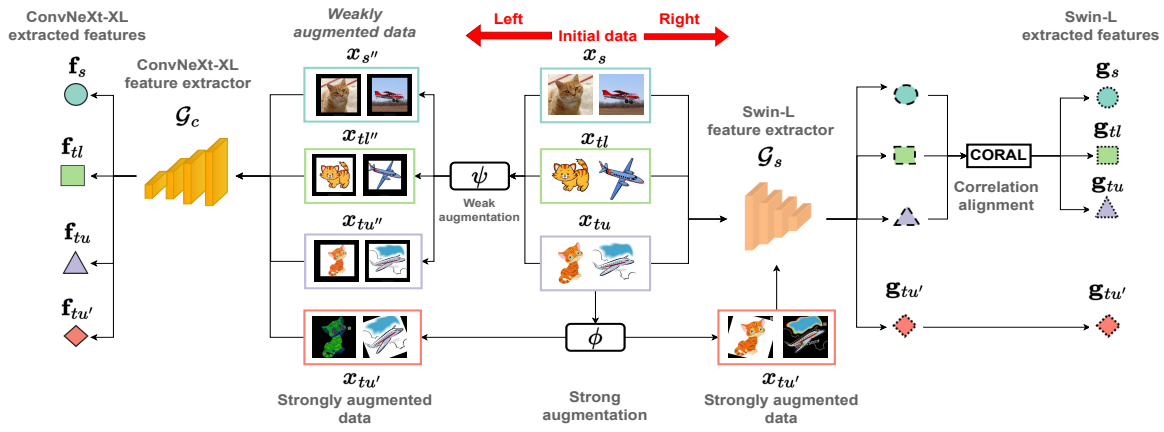


Figure 2. **Overview of Feature extraction.** (1) For Swin-L (**Right**): entire initial data (*i.e.* source data, target labeled data and target unlabeled data) along with strong augmented version of target unlabeled data are passed to get output features. Output features of source, target labeled, and target unlabeled data are then passed through CORAL module to get the final features. (2) For ConvNeXt-XL (**Left**): weak augmented version of initial data along with strong augmented version of target unlabeled data are passed to get output features.

a distribution $P_T(X,Y)$, such that $P_S(X,Y) \neq P_T(X,Y)$. n_s, n_{tl} and n_{tu} represent number of instances in $\mathcal{D}_s, \mathcal{D}_{tl}$ and $\mathcal{D}_{tu}$ respectively. Each data point x_s^i (x_{tl}^i) from $\mathcal{D}_s(\mathcal{D}_{tl})$ is associated with a label y_s^i (y_{tl}^i) belonging to one of the K different classes of the dataset. Our goal is to train an SSDA model using $\mathcal{D}_s, \mathcal{D}_{tl}$ and $\mathcal{D}_{tu}$ and evaluate it on $\mathcal{D}_{tu}$ using its labels which are available at test time.

3.1. Stage I: Feature extraction

Our work, MARRS uses co-training during the second stage (cf. Sec. 3.2). Since co-training benefits from diverse data views, we propose a novel module that consists of 3 different modules to introduce diversity at three levels:

(i) Backbone level diversity module (BD): Vision Transformers (ViT) based networks offer several advantages, such as multi-head self-attention and larger scalability. Similarly, CNN-based designs also have their own set of benefits, including various in-built inductive biases, higher throughput, and ease of implementation. So, to combine the diverse yet complementary advantages of CNN and ViT for stronger consensus, we propose to use ConvNeXt-XL [22] denoted as $\mathcal{G}_c$ from CNN family and Swin-L [21] denoted as $\mathcal{G}_s$ from ViT family as two fixed modern backbones in our framework. However, analysis of MARRS with other modern backbones can also be found in the supplementary.

(ii) Image level diversity module (ID): Since weak augmentation (ψ) [37] produces realistic yet diverse variations of images, we use ψ to bring diversity at the image level. But, applying ψ to both $\mathcal{G}_c$ and $\mathcal{G}_s$ data will not result in diversity. As a result, we propose to use ψ for only one of the backbones data, $\mathcal{G}_c$ in this case. We choose perspective preserving padding as ψ in our framework.

(iii) Feature distribution level diversity module (FD): We introduce a new idea of using CORAL [40] to induce diversity among models at the feature distribution level. CORAL is a simple linear algebraic-based technique traditionally used for aligning source and target data feature distributions. During alignment, it changes features of the data points. Hence, we can also interpret it as a feature level data augmentation technique and then use it to induce diversity at the feature distribution level by applying it to the features of only one of the backbones, $\mathcal{G}_s$ in this case.

In addition, we apply strong augmentation [6] to target unlabeled data in both backbones. A detailed feature extraction outline is shown in Fig. 2. It integrates all 3 proposed modules to extract final features. Let, $\mathbf{f}_s(\mathbf{g}_s), \mathbf{f}_{tl}(\mathbf{g}_{tl}), \mathbf{f}_{tu}(\mathbf{g}_{tu})$ and $\mathbf{f}_{tu'}(\mathbf{g}_{tu'})$ represent features of source data, target labeled data, target unlabeled data and strongly augmented version of target unlabeled data extracted from $\mathcal{G}_c$ ($\mathcal{G}_s$). After feature extraction, two groups of feature sets are produced, which are then used for learning two linear classifiers during the second stage. Feature sets obtained using $\mathcal{G}_c$ are $\mathcal{D}_s^c = (\mathbf{f}_s, y_s), \mathcal{D}_{tl}^c = (\mathbf{f}_{tl}, y_{tl}), \mathcal{D}_{tu}^c = \mathbf{f}_{tu}$ and $\mathcal{D}_{tu'}^c = \mathbf{f}_{tu'}$ and that using $\mathcal{G}_s$ are $\mathcal{D}_s^s = (\mathbf{g}_s, y_s)$, $\mathcal{D}_{tl}^s = (\mathbf{g}_{tl}, y_{tl}), \mathcal{D}_{tu}^s = \mathbf{g}_{tu}$ and $\mathcal{D}_{tu'}^s = \mathbf{g}_{tu'}$. In all cases, subscripts denote the type of feature and superscripts denote the feature extractor, *e.g.* $\mathcal{D}_s^c$ denotes source feature set obtained from $\mathcal{G}_c$. $\mathcal{D}_s^c, \mathcal{D}_{tl}^c, \mathcal{D}_s^s$ and $\mathcal{D}_{tl}^s$ denotes labeled feature sets containing features along with their available labels.

3.2. Stage II: Classifier training

Co-training [2] is a well-established semi-supervised learning (SSL) framework. Given diverse views of data, in co-training, each model is trained using the most confident predictions of its counterpart, which implicitly integrates the strengths of models and results into more accurate models. In our work, two models are: (i) linear classifier $\mathcal{F}_c$: single unbiased fully connected layer followed by the Soft-

max function, with weight $\boldsymbol{w}_c$ trained using $\mathcal{D}_s^c, \mathcal{D}_{tl}^c, \mathcal{D}_{tu}^c$ and $\mathcal{D}_{tu'}^c$; (ii) linear classifier $\mathcal{F}_s$: single unbiased fully connected layer followed by the Softmax function, with weight $\boldsymbol{w}_s$ trained using $\mathcal{D}_s^s, \mathcal{D}_{tl}^s, \mathcal{D}_{tu}^s$ and $\mathcal{D}_{tu'}^s$. Initially, classifiers $\mathcal{F}_c$ and $\mathcal{F}_s$ are trained on their individual labeled feature sets. After initial training, at each iteration, we pass target unlabeled feature sets $\mathcal{D}_{tu}^c$ and $\mathcal{D}_{tu}^s$ to classifiers $\mathcal{F}_c$ and $\mathcal{F}_s$ to obtain pseudo-label sets U^c and U^s respectively, which can be formulated as:

$$
\begin{aligned}
U^c = &\Big\{ \Big(y_{ps,c}^i = \arg\max_k p\left(k \mid \mathbf{f}_{tu}^i; \boldsymbol{w}_c\right) \Big); \\
&\quad \text{if } \max_k p\left(k \mid \mathbf{f}_{tu}^i; \boldsymbol{w}_c\right) > \tau \Big\} \quad i = 1, \ldots, n_{tu} \\
U^s = &\Big\{ \Big(y_{ps,s}^i = \arg\max_k p\left(k \mid \mathbf{g}_{tu}^i; \boldsymbol{w}_s\right) \Big); \\
&\quad \text{if } \max_k p\left(k \mid \mathbf{g}_{tu}^i; \boldsymbol{w}_s\right) > \tau \Big\} \quad i = 1, \ldots, n_{tu}
\end{aligned} \tag{1}
$$

where, $\mathbf{f}_{tu}^i$ and $\mathbf{g}_{tu}^i$ are i-th feature drawn from $\mathcal{D}_{tu}^c$ and $\mathcal{D}_{tu}^s$, $y_{ps,c}^i$ and $y_{ps,s}^i$ are pseudo-labels of i-th features obtained using $\mathcal{F}_c$ and $\mathcal{F}_s$ respectively, τ is the confidence threshold and p is the prediction of the classifier. By using the main principle of co-training to teach one model on confident predictions of another model, we constructed two labeled feature sets for co-training as: (i) $\mathcal{D}_{co}^c = \left\{\mathbf{f}_{tu}^i, y_{ps,s}^i\right\}_{i=1}^{n_{tu'}}$ denoting co-training feature set for $\mathcal{F}_c$ with labels from U^s and corresponding features from $\mathcal{D}_{tu}^c$; (ii) $\mathcal{D}_{co}^s = \left\{\mathbf{g}_{tu}^i, y_{ps,c}^i\right\}_{i=1}^{n_{tu''}}$ denoting co-training feature set for $\mathcal{F}_s$ which consists of labels from U^c and corresponding features from $\mathcal{D}_{tu}^s$. $n_{tu''}$ and $n_{tu'}$ denote the size of pseudo-label sets U^c and U^s respectively. We use standard cross-entropy loss for co-training, which can be written as:

$$
\mathcal{L}_{\text{ce}}(\mathcal{D}; \boldsymbol{w}) = -\frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} \left(y^i\right)_k \log\left(\mathcal{F}\left(x^i\right)\right)_k, \tag{2}
$$

where, $\boldsymbol{w}$ is the weight of classifier $\mathcal{F}$, $\mathcal{D}$ denotes the labeled dataset, (x^i, y^i) is the i-th sample of dataset $\mathcal{D}$, K is the number of classes and n is the number of samples in dataset $\mathcal{D}$. Inspired by the success of consistency regularization in previous works [1, 39, 45], we integrate it into our framework to learn a more robust model. In our work, we apply consistency regularization by setting predictions of target unlabeled data features as the pseudo-labels on predictions of strongly augmented target unlabeled data features. Strong data augmentation generates a broader range of highly perturbed data, and training on them makes the model learn only important characteristics about the data and therefore enhances the model's generalizability. Our work uses RandAugment [6] as a strong data augmentation technique. Only reliable target unlabeled data features ($i.e.$ data features with maximal probability score over a threshold τ_{tu}) are retained for loss estimation to limit the influence of inaccurate pseudo-labels. We have already computed features of target unlabeled data and features of strongly augmented version of target unlabeled data (cf. Sec. 3.1 for details). We apply cross-entropy loss using Eq. (2) for consistency regularization in which targets are pseudo-labels obtained from Eq. (1), and predictions are classifier outputs for corresponding strongly augmented target unlabeled data features. Hence, two labeled features sets constructed for consistency regularization are: (i) $\mathcal{D}_{cons}^c = \left\{\mathbf{f}_{tu'}^i, y_{ps,c}^i\right\}_{i=1}^{n_{tu''}}$ denoting consistency regularization feature set for $\mathcal{F}_c$ which contains labels from U^c and corresponding features from $\mathcal{D}_{tu'}^c$; (ii) $\mathcal{D}_{cons}^s = \left\{\mathbf{g}_{tu'}^i, y_{ps,s}^i\right\}_{i=1}^{n_{tu'}}$ denoting consistency regularization feature set for $\mathcal{F}_s$ with labels from U^s and corresponding features from $\mathcal{D}_{tu'}^s$. For labeled feature sets $\mathcal{D}_s^c, \mathcal{D}_{tl}^c, \mathcal{D}_s^s$ and $\mathcal{D}_{tl}^s$, we use cross-entropy loss with label-smoothing regularization [41]. It is a simple technique to reduce the over-fitting of a model on labeled data by regularizing the label with a small smoothing parameter ϵ, formulated as:

$$
\begin{aligned}
\mathcal{L}_{\text{lsr}}(\mathcal{D}; \boldsymbol{w}) &= -\frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} \left(y_{smooth}^i\right)_k \log\left(\mathcal{F}\left(x^i\right)\right)_k, \\
\left(y_{smooth}^i\right)_k &= \begin{cases} 1-\varepsilon; & \text{if } \left(y^i\right)_k = 1, \\ \varepsilon/K; & \text{otherwise.} \end{cases}
\end{aligned} \tag{3}
$$

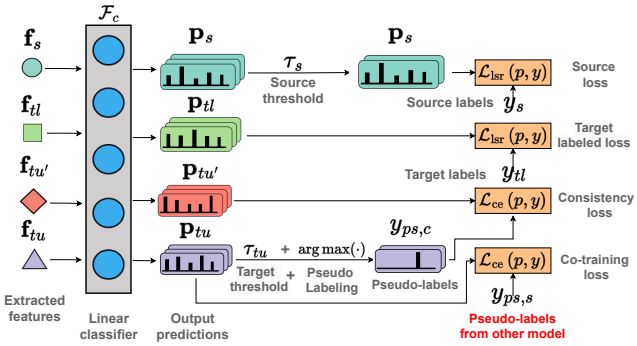


Figure 3. **Illustration of the proposed MARRS framework.** Extracted features are passed through a linear classifier for obtaining predictions. For reliable source data features ($i.e.$ data features with maximal probability score over a threshold τ_s) and target labeled data features, losses are calculated using their predictions and available ground truths. Confident predictions of target unlabeled data features are converted into pseudo-labels, which are then used for 2 purposes: (i) For calculating consistency loss on predictions of strongly augmented target unlabeled data features; (ii) For calculating co-training loss on predictions of target unlabeled data features of another classifier. For ease of visualization, training for only $\mathcal{F}_c$ is shown; training of $\mathcal{F}_s$ is obtained by replacing input features with features extracted from $\mathcal{G}_s$ and by using $y_{ps,c}$ as pseudo-labels for calculating co-training loss.

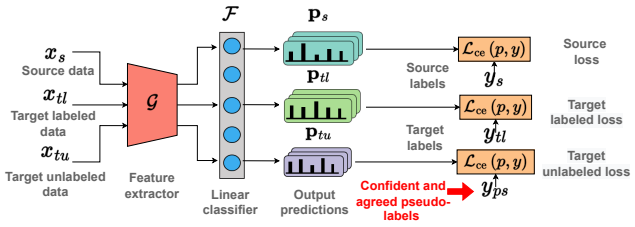


Figure 4. **Illustration of student model training.** Initial data (*i.e.* source data, target labeled data and target unlabeled data) are passed through the feature extractor $\mathcal{G}$ (MobileNetV2) followed by a classifier $\mathcal{F}$ which is a linear network. For source data and target labeled data, losses are calculated using their predictions and available labels. For target unlabeled data, loss is calculated using its predictions and agreed pseudo-labels with high confidence obtained from MARRS trained classifiers (See supplementary material for the detailed algorithm).

where, ϵ is the smoothing parameter, y^i_{smooth} denotes the i-th sample label after smoothing and other notations are same as Eq. (2). Importantly, in source features sets ($\mathcal{D}^c_s$ and $\mathcal{D}^s_s$), we considered only those source data features whose maximal probability score lies above a threshold (τ_s) because source data features with low confidence scores are unlikely to be informative. At test time, the average of predictions of $\mathcal{F}_c$ and $\mathcal{F}_s$ are used for classification. Fig. 3 and Algorithm 1 outlines the main steps of MARRS.

3.3. Knowledge Distillation

Knowledge Distillation (KD) [3, 14, 27, 32] is the process of capturing the knowledge of a large model or an ensemble of models into a more petite model without suffering a significant performance loss. The main reason for exploring knowledge distillation in our work is to eliminate the burden of keeping two large backbones at inference time for resource constrained situations (*e.g.* autonomous driving cars, video surveillance, robotics and augmented reality). Hence, we use KD to transfer the knowledge from the teacher model (MARRS-trained classifiers) to a student model. We use MobileNetV2 [35] as a student model because it is a compact yet efficient network designed primarily for mobile applications. MobileNetV2 also allows memory-efficient inference, which significantly reduces the memory footprint needed during inference time.

The most important component of a knowledge distillation algorithm is the knowledge itself. In our case, knowledge consists of the pseudo-labels (class with maximal probability) generated by the MARRS-trained classifiers on unlabeled data. To reduce the effect of uncertain pseudo-labels on learning, we use only those pseudo-labels for which both classifiers are confident (probability confidence score of both classifiers pseudo-labels are more than a threshold τ) and agreed (pseudo-labels of both classifiers

Algorithm 1: Proposed MARRS algorithm

Input : Linear classifiers $\mathcal{F}_c$ and $\mathcal{F}_s$, parameters $\boldsymbol{w}_c$ and $\boldsymbol{w}_s$, $\mathcal{F}_c$ feature sets $\mathcal{D}^c_s, \mathcal{D}^c_{tl}, \mathcal{D}^c_{tu}$ and $\mathcal{D}^c_{tu'}$, $\mathcal{F}_s$ feature sets $\mathcal{D}^s_s, \mathcal{D}^s_{tl}, \mathcal{D}^s_{tu}$ and $\mathcal{D}^s_{tu'}$, learning rate η, weight balancing parameters $\lambda_s, \lambda_{tl}, \lambda_{co}$ and λ_{cons}, outer iterations N_{outer}, inner iterations N_{inner}, confidence thresholds τ_s and τ_{tu}.

Output: updated parameters $\boldsymbol{w}_c$ and $\boldsymbol{w}_s$.

for $p \leftarrow 1$ **to** N_{outer} **do**
  $U^c, U^s \leftarrow$ Calculate pseudo-label sets using $\mathcal{F}_c$ and $\mathcal{F}_s$ by Eq. (1) with $\tau = \tau_{tu}$.
  $\mathcal{D}^c_{co}, \mathcal{D}^s_{co} \leftarrow$ Obtain co-training feature sets using $U^c, U^s, \mathcal{D}^c_{tu}$ and $\mathcal{D}^s_{tu}$ (See Sec. 3.2).
  $\mathcal{D}^c_{cons}, \mathcal{D}^s_{cons} \leftarrow$ Obtain consistency regularization feature sets using $U^c, U^s, \mathcal{D}^c_{tu'}$ and $\mathcal{D}^s_{tu'}$ (See Sec. 3.2).
  for $q \leftarrow 1$ **to** N_{inner} **do**
    // Calculating Losses for $\mathcal{F}_c$
    $\mathcal{L}^c_s = \mathcal{L}_{\text{lsr}}(\mathcal{D}^c_s; \boldsymbol{w_c})$ (for source features whose $p(y \mid \mathbf{f}_s; \boldsymbol{w_c}) > \tau_s$) using Eq. (3).
    $\mathcal{L}^c_{tl} = \mathcal{L}_{\text{lsr}}(\mathcal{D}^c_{tl}; \boldsymbol{w_c})$ using Eq. (3).
    $\mathcal{L}^c_{co} = \mathcal{L}_{\text{ce}}(\mathcal{D}^c_{co}; \boldsymbol{w_c})$ using Eq. (2).
    $\mathcal{L}^c_{cons} = \mathcal{L}_{\text{ce}}(\mathcal{D}^c_{cons}; \boldsymbol{w_c})$ using Eq. (2).
    $\mathcal{L}^c_{tu} = \lambda_{co} * \mathcal{L}^c_{co} + \lambda_{cons} * \mathcal{L}^c_{cons}$
    $\mathcal{L}^c_{total} = \lambda_s * \mathcal{L}^c_s + \lambda_{tl} * \mathcal{L}^c_{tl} + \mathcal{L}^c_{tu}$
    // Calculating Losses for $\mathcal{F}_s$
    $\mathcal{L}^s_s = \mathcal{L}_{\text{lsr}}(\mathcal{D}^s_s; \boldsymbol{w_s})$ (for source features whose $p(y \mid \mathbf{g}_s; \boldsymbol{w_s}) > \tau_s$) using Eq. (3).
    $\mathcal{L}^s_{tl} = \mathcal{L}_{\text{lsr}}(\mathcal{D}^s_{tl}; \boldsymbol{w_s})$ using Eq. (3).
    $\mathcal{L}^s_{co} = \mathcal{L}_{\text{ce}}(\mathcal{D}^s_{co}; \boldsymbol{w_s})$ using Eq. (2).
    $\mathcal{L}^s_{cons} = \mathcal{L}_{\text{ce}}(\mathcal{D}^s_{cons}; \boldsymbol{w_s})$ using Eq. (2)
    $\mathcal{L}^s_{tu} = \lambda_{co} * \mathcal{L}^s_{co} + \lambda_{cons} * \mathcal{L}^s_{cons}$
    $\mathcal{L}^s_{total} = \lambda_s * \mathcal{L}^s_s + \lambda_{tl} * \mathcal{L}^s_{tl} + \mathcal{L}^s_{tu}$
    // Updating parameters
    $\boldsymbol{w}_c = \boldsymbol{w}_c - \eta \cdot \nabla \mathcal{L}^c_{total}$
    $\boldsymbol{w}_s = \boldsymbol{w}_s - \eta \cdot \nabla \mathcal{L}^s_{total}$
  end
end

are identical). We train student model on original datasets $\mathcal{D}_s, \mathcal{D}_{tl}$ and $\mathcal{D}_{tu}$ using same learning settings as in [33]. An outline of student model training is given in Fig. 4.

4. Experiments

Datasets: We use **DomainNet** [30], which is a large-scale adaptation dataset consisting of 6 domains with 345 categories. Following [33] we use 4 domains Clipart (C), Real (R), Sketch (S) and Painting (P), with 126 categories and perform 7 different cross-domain evaluations. We also

Table 1. **Accuracy** (%) **on *DomainNet***. Methods that use modern backbones are given in the shaded region. MARRS* (student model) uses MobileNetV2, and all other baselines use ResNet34. Best results are in **bold** and the second-best results are underlined.

Method	R → C		R→P		P→C		C→S		S→P		R→S		P→R		Mean	
	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	**1-shot**	**3-shot**
S+T	55.6	60.0	60.6	62.2	56.8	59.4	50.8	55.0	56.0	59.5	46.3	50.1	71.8	73.9	56.9	60.0
ENT	65.2	71.0	65.9	69.2	65.4	71.1	54.6	60.0	59.7	62.1	52.1	61.1	75.0	78.6	62.6	67.6
MME	70.0	72.2	67.7	69.7	69.0	71.7	56.3	61.8	64.8	66.8	61.0	61.9	76.1	78.5	66.4	68.9
CLDA	76.1	77.7	75.1	75.7	71.0	76.4	63.7	69.7	70.2	73.7	67.1	71.1	80.1	82.9	71.9	75.3
CDAC	77.4	79.6	74.2	75.1	75.5	79.3	67.6	69.9	71.0	73.4	69.2	72.5	80.4	81.9	73.6	76.0
MCL	77.4	79.4	74.6	76.3	75.5	78.8	66.4	70.9	74.0	74.7	70.7	72.3	82.0	83.3	74.4	76.5
PACE	82.4	84.2	84.5	84.9	82.6	84.5	74.6	76.0	**84.8**	85.3	74.0	75.4	91.7	92.5	82.1	83.3
MARRS*	84.2	85.0	84.7	85.0	85.6	85.7	75.4	**77.6**	83.0	84.6	73.7	76.2	90.2	91.1	82.4	83.6
MARRS	**84.5**	**85.5**	**85.1**	**85.9**	**86.1**	**86.1**	**76.0**	**77.6**	**84.8**	**86.1**	**74.5**	**77.3**	**91.9**	**92.9**	**83.3**	**84.5**

Table 2. **Accuracy** (%) **on *Office-Home***. Methods that use modern backbones are given in the shaded region. MARRS* (student model) uses MobileNetV2, and all other baselines use ResNet34. Best results are in **bold** and the second-best results are underlined.

Method	R→C	R→P	R→A	P→R	P→C	P→A	A→P	A→C	A→R	C→R	C→A	C→P	Mean
						1-shot							
S+T	52.1	78.6	66.2	74.4	48.3	57.2	69.8	50.9	73.8	70.0	56.3	68.1	63.8
ENT	53.6	81.9	70.4	79.9	51.9	63.0	75.0	52.9	76.7	73.2	63.2	73.6	67.9
MME	61.9	82.8	71.2	79.2	57.4	64.7	75.5	59.6	77.8	74.8	65.7	74.5	70.4
CLDA	60.2	83.2	72.6	81.0	55.9	66.2	76.1	56.3	79.3	76.3	66.3	73.9	70.6
CDAC	-	-	-	-	-	-	-	-	-	-	-	-	-
MCL	67.0	85.5	73.8	81.3	61.1	68.0	79.5	64.4	81.2	78.4	68.5	79.3	74.0
PACE	86.2	95.2	90.4	**94.7**	82.9	90.4	**95.2**	85.4	**94.4**	94.3	91.1	**94.7**	91.2
MARRS*	87.0	95.2	90.5	94.3	85.4	90.2	94.7	86.6	93.5	94.3	91.5	93.6	91.4
MARRS	**87.3**	**95.3**	**90.8**	**94.7**	**85.5**	**90.9**	**95.2**	**86.8**	93.7	**94.6**	**92.0**	93.8	**91.7**
						3-shot							
S+T	55.7	80.8	67.8	73.1	53.8	63.5	73.1	54.0	74.2	68.3	57.6	72.3	66.2
ENT	62.6	85.7	70.2	79.9	60.5	63.9	79.5	61.3	79.1	76.4	64.7	79.1	71.9
MME	64.6	85.5	71.3	80.1	64.6	65.5	79.0	63.6	79.7	76.6	67.2	79.3	73.1
CDAC	67.8	85.6	72.2	81.9	67.0	67.5	80.3	65.9	80.6	80.2	67.4	81.4	74.2
CLDA	66.0	87.6	76.7	82.2	63.9	72.4	81.4	63.4	81.3	80.3	70.5	80.9	75.5
MCL	70.1	88.1	75.3	83.0	68.0	69.9	83.9	67.5	82.4	81.6	71.4	84.3	77.1
PACE	**87.0**	95.7	**90.8**	95.1	85.0	**90.7**	95.3	86.3	**94.9**	94.9	91.2	**95.3**	91.9
MARRS*	86.3	**96.0**	90.6	95.2	86.4	90.0	**95.7**	86.4	94.5	**95.2**	91.2	95.1	91.9
MARRS	**87.0**	**96.0**	90.6	**95.3**	**86.5**	**90.7**	**95.7**	**86.5**	94.7	**95.2**	**91.8**	95.1	**92.1**

evaluate on **Office-Home** [42], which is a middle-size adaptation dataset consisting of 65 classes from four domains, namely Clipart (C), Real (R), Product (P), and Art (A). Following [33], we report performance for all possible 12 adaptation scenarios.

Implementation details: All experiments are implemented on a single NVIDIA RTX 2080 GPU using Pytorch [28]. We choose ConvNeXt-XL [22] and Swin-L [21] pretrained on ImageNet [7] as our two fixed backbone networks. We use gradient descent on the entire dataset with momentum and learning rate of 0.9 and 30 respectively, with no weight decay. In knowledge distillation, we use MobileNetV2 [35] as student model and set the value of τ to 0.7. Other experimental settings like learning rate, batch size, optimizers are same as [33] except for number of iterations in which we use only 10K/2.5K iterations as opposed to 50K/10K iterations on DomainNet/OfficeHome datasets. Additional experimental details are given in supplementary.

Baselines: We compare our framework MARRS with recent state-of-the-art SSDA approaches, including S+T, ENT [11], MME [33], CLDA [38], CDAC [17], MCL [46], and PACE [20]. Among baselines, S+T is trained on the only source and labeled target samples.

4.1. Main Results

Results of MARRS: Results of MARRS in Tabs. 1 and 2 emphasizes on four major findings: **(1)** MARRS outperforms all ResNet34 based baselines by a large margin across all evaluation settings in both datasets, which supports our motivation that efficient use of features from modern backbones can obtain superior performance in relatively less time. **(2)** MARRS gives comparatively higher performance in data efficient scenario ($i.e.$ 1-shot), where it outperforms SOTA ResNet34 based method (MCL [46]) by significant margin of 8.9% and 17.7% on mean accuracy of DomainNet and OfficeHome datasets respectively. **(3)** In both datasets, MARRS also beats PACE [20], which uses an ensemble of 28 modern backbones, unlike our method, which utilizes only 2 modern backbones. This shows that using modern backbones is not the only reason for marvelous result of MARRS, the design choice of leveraging modern backbones using co-training equipped with novel diversity module and other components are also an important reason for phenomenal performance of MARRS. **(4)** Gains of MARRS are more prominent in complex settings like C→S in DomainNet and P→C in OfficeHome, which shows the vitality of proposed framework in hard domain adaptation settings.

Table 3. **Comparison of performance among variations of MARRS** on mean test accuracy (%) across all settings. Best results are in **bold** and the second-best results are underlined.

Method	DomainNet		OfficeHome	
	1-shot	3-shot	1-shot	3-shot
MARRS$_{none}$	81.0	83.7	90.7	91.8
MARRS$_{both,both}$	82.8	83.9	91.5	91.8
MARRS$_{both,conv}$	82.9	84.4	91.5	92.0
MARRS$_{both,swin}$	83.0	84.3	91.6	**92.1**
MARRS$_{rev.}$	83.0	84.3	91.6	92.0
MARRS	**83.3**	**84.5**	**91.7**	**92.1**
MARRS$_{swin}$	81.1	82.6	90.8	91.4
MARRS$_{conv}$	82.0	83.4	91.0	91.5

Table 4. **Analysis of each component's relative importance** on mean test accuracy (%) of *DomainNet*, 1-shot task.

Method	Accuracy
Swin-L	78.2
ConvNext-XL	80.0
Ensemble	80.4
Co-training (A)	81.3
A+ID A+FD	81.8 82.4
A+ID+FD (B)	82.8
B+LS+CR (**MARRS**)	**83.3**

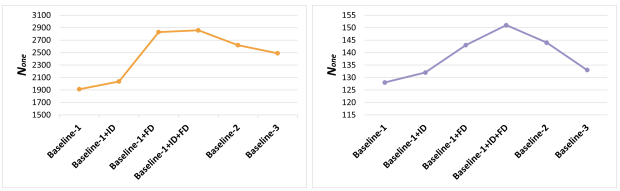
Table 5. **Analysis of MARRS with smaller backbones** using mean test accuracy (%) on *Office-Home* dataset.

Method	1-shot	3-shot
MME	70.4	73.1
CDAC	-	74.2
CLDA	70.6	75.5
MCL	74.0	77.1
MARRS$_{none}$	73.0	76.4
MARRS$_{both,both}$	73.9	76.0
MARRS	**74.6**	**77.3**

Results of MARRS*: We use MobileNetV2 as a student model (MARRS*) consisting of 3.4 M parameters that are nearly 6 times lesser than of ResNet34 (22 M) still, as can be seen from Tabs. 1 and 2, it outperforms all previous ResNet34 based methods by a large margin and modern backbone based method PACE with smaller margin on both datasets. These results show that it is also possible to deploy domain adaptation with high performance even on resource constrained devices ($i.e.$ mobile, AR/VR devices).

4.2. Ablation Studies

How effective is the proposed diversity module in inducing diversity?: For this, we conducted an experiment by excluding co-training from MARRS to correctly examine the effect of the proposed diversity module. Now, each classifier will use its own pseudo-label sets to construct two labeled feature sets, namely $\mathcal{D}^c_{co}$ and $\mathcal{D}^s_{co}$ (cf. Sec. 3.2). After training both classifiers $\mathcal{F}_c$ and $\mathcal{F}_s$, we calculate N_{one} representing the number of unlabeled examples on which *exactly one* of the classifiers have confidence. Since we know that if classifiers are diverse, they will be confident on different unlabeled examples. As a result, we may assert that diversity is directly proportional to N_{one}. Our proposed module consists of 3 different diversity modules: ID, FD, and BD (cf. Sec. 3.1 for details). Fig. 5 depicts that in the case of both datasets the value of N_{one} rises when either ID or FD is used, and it increases even more when both are used, which shows their complementary nature. However, the value of N_{one} falls when we use the same family backbones ($i.e.$ Baseline-2 and Baseline-3), which signifies the importance of BD in producing diversity. Yet, the values of N_{one} are larger than that of Baseline-1, which does not use ID and FD. It emphasizes that the ID and FD modules are capable of inducing diversity even with same backbones.



(a) Real→Sketch (b) Painting→Clipart

Figure 5. **Analysis of variation of N_{one}** on complex **(a)** Real→Sketch, setting of the *DomainNet* dataset; **(b)** Painting → Clipart, task of the *OfficeHome* dataset. **Baseline-1**: MARRS w/o co-training, ID and FD. **Baseline-2**: Baseline-1 with ID and FD, but with both backbones as Swin-L. **Baseline-3**: Baseline-1 with ID and FD, but with both backbones as ConvNeXt-XL.

Performance impact of the proposed diversity module: For extensive studies, we perform comparison with seven other variants of our framework MARRS. In **MARRS**, we applied weak augmentation (ψ) at $\mathcal{G}_c$ and CORAL at $\mathcal{G}_s$. **MARRS$_{\text{none}}$** is obtained by removing both ψ and CORAL from the framework. **MARRS$_{\text{both,both}}$** is obtained by applying both ψ and CORAL in both $\mathcal{G}_c$ and $\mathcal{G}_s$. **MARRS$_{\text{both,conv}}$** is obtained by using both ψ and CORAL in $\mathcal{G}_c$. Similarly, **MARRS$_{\text{both,swin}}$** is obtained by using both ψ and CORAL in $\mathcal{G}_s$. **MARRS$_{\text{rev.}}$** is the reverse of MARRS, obtained by using CORAL in $\mathcal{G}_c$ and ψ in $\mathcal{G}_s$. **MARRS$_{\text{conv}}$** and **MARRS$_{\text{swin}}$** are the MARRS with both backbones as ConvNeXt-XL and Swin-L respectively. Results in Table 3 shed light onto three important findings: **(1)** All four MARRS, MARRS$_{rev.}$, MARRS$_{both,conv}$ and MARRS$_{both,swin}$ are designs based on the proposed ID and FD modules and all are consistently outperforming MARRS$_{none}$ and MARRS$_{both,both}$ across all 4 adaptation settings, signifying the importance of our proposed image level and feature distribution level diversity modules which says that applying ψ or CORAL at only one of the backbones helps to introduce diversity, which ultimately leads to an improvement in performance. **(2)** Applying ψ at $\mathcal{G}_c$ and CORAL at $\mathcal{G}_s$ yields superior results when compared to the other three strategies, hence we use this order of applying ψ and CORAL in our framework MARRS. **(3)** Drop in performance of MARRS$_{swin}$ and MARRS$_{conv}$ highlight the importance of backbone level diversity module (BD), which emphasizes that different family backbones should be employed for larger diversity and higher performance gains.

Importance of individual components: A recent method PACE [20] uses an ensemble of 28 different modern backbones and still our method MARRS outperforms it by using just two modern backbones, which shows the impor-

Table 6. **Performance analysis of various feature level augmentation methods** on mean test accuracy (%) of *DomainNet* 3-shot task.

Method	Mean Accuracy
Gaussian noise (σ = 0.001)	83.7
Gaussian noise (σ = 0.01)	83.8
Gaussian noise (σ = 0.1)	83.4
Gaussian noise (σ = 1)	81.1
Interpolation	84.0
CORAL	**84.5**

Table 7. **Performance analysis of different weak augmentation** techniques on mean test accuracy (%) of *DomainNet*, 1-shot task.

Augmentation	Mean Accuracy
Color jitter	82.7
Grayscale	82.9
Horizontal flipping	83.1
Square padding	83.1
Perspective preserving Padding	**83.3**

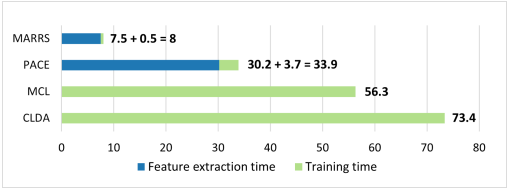


Figure 6. **Runtime (in hours)** on a single NVIDIA RTX 2080 GPU for *DomainNet*, 3-shot task. Runtime consists of feature extraction time (features are extracted and stored) and training time. MCL [46] and CLDA [38] are recent ResNet34 based methods.

tance of our design choice. Further, results in Table 4 show that simply using an ensemble gives only marginal improvements over standalone modern backbones. On using co-training, 0.9% increment in performance can be seen. Performance further improves on introducing either ID or FD. While utilizing both ID and FD, performance boost of 1.5% can be seen, which is a compelling improvement in complex and label scarce DomainNet 1-shot setting and shows the value of our novel diversity module. Finally, introducing Consistency regularization (CR) and Label smoothing (LS) further adds 0.5 % to mean accuracy.

Effectiveness of MARRS with smaller backbones: For fair comparison with prior works which use ResNet34 as a backbone, we also performed an experiment in which we replaced modern backbones with variations of ResNet34 backbone. For ensuring a *relaxed* version of our backbone level diversity module ($i.e.$ employ different backbones), we use two variants of ResNet34 namely ResNet34d [13] as $\mathcal{G}_c$ and skResNet34 [19] as $\mathcal{G}_s$, having 21.8M and 22.1M parameters respectively, which are comparable to ResNet34 with 22M parameters. Results in Table 5 highlight two key findings: **(1)** Superior performance than all ResNet34 based works without using any complex learning techniques, shows that our framework, MARRS is simple yet effective. However, performance gap now becomes thinner as we are training only linear classifiers using fixed features extracted from ResNet34 variants, which are not discriminative enough to give high performance without getting updated during training. **(2)** MARRS outperforms both MARRS$_{none}$ and MARRS$_{both,both}$. It demonstrates the versatility of proposed FD and ID modules, which can successfully produce diversity even with smaller backbones.

Comparison with other feature level augmentation methods: Table 6 shows the performance of MARRS with two other learning free feature level augmentation techniques (random Gaussian noise and Interpolation [8]), which are used for inducing feature distribution level diversity at the place of CORAL in MARRS. Random Gaussian noise performs well on smaller values of standard deviation (σ) and achieves maximum accuracy of 83.8% at σ=0.01. Interpolation produces relatively better results as it interpolates a given feature with its nearest neighbor feature, unlike adding just random Gaussian noise. On using either of the two, results are superior to all the baselines but inferior to the result gained with CORAL, supporting our novel idea that CORAL can be used as an effective learning free feature level augmentation in domain adaptation scenarios.

MARRS with different weak augmentation techniques: In Table 7, we analyse the effect of different weak augmentations (ψ) on the performance of MARRS. We discover that the optimum performance comes from utilizing perspective preserving padding as ψ. But the crucial finding is that performance of MARRS does not differ wildly and beats all prior works, even when using different ψ, which shows the stability of our framework MARRS.

Runtime comparison: We compare runtime of our framework with a recent modern backbone based method PACE [20] and two ResNet34 based methods, namely CLDA [38] and MCL [46]. From Fig. 6, we can see that our method is **7 to 9** times faster than ResNet34 based methods and nearly **4.25** times faster than PACE. Despite being compute friendly, our method gives dominant performance across all evaluation settings, which encourages future SSDA works to incorporate modern backbones in their framework for robust and energy-efficient training [15, 43].

5. Conclusion

In this work, we propose a novel SSDA framework named MARRS. It integrates strong feature representation of modern backbones by training two linear classifiers using co-training. To induce diversity among classifiers, a novel three stage diversity module is proposed, including a simple yet promising new idea of using CORAL to induce diversity at feature level. For instances where resources are limited, a smaller student model is trained by utilizing the knowledge from MARRS-trained classifiers. Experiments on benchmark SSDA datasets show that both MARRS and MARRS-trained student model outperforms previous SSDA methods in significantly less time. In future work, we are keen to extend our method to more data efficient problems like source-free domain adaptation, few-shot domain adaptation and in other related computer vision tasks, namely video domain adaptation, adaptive object detection and depth estimation.

References

[1] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, 2019. 2, 4

[2] Avrim Blum and Tom. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, 1998. 2, 3

[3] Defang Chen, Jian-Ping Mei, Yuan Zhang, Can Wang, Zhe Wang, Yan Feng, and Chun Chen. Cross-layer distillation with semantic calibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021. 5

[4] Minmin Chen, Kilian Q Weinberger, and John Blitzer. Co-training for domain adaptation. In *Advances in Neural Information Processing Systems*, 2011. 2

[5] Minmin Chen, Kilian Q. Weinberger, and Yixin Chen. Automatic feature decomposition for single view co-training. In *International Conference on Machine Learning*, 2011. 2

[6] Ekin Dogus Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 3008–3017, 2020. 3, 4

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 6

[8] Terrance Devries and Graham W. Taylor. Dataset augmentation in feature space. In *International Conference on Learning Representations*, 2017. 2, 8

[9] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, 2015. 1, 2

[10] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016. 2

[11] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems*, 2004. 1, 6

[12] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1, 2

[13] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 558–567, 2018. 8

[14] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015. 2, 5

[15] Torsten Hoefler, Dan Alistarh, Tal Ben-Nun, Nikoli Dryden, and Alexandra Peste. Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. *The Journal of Machine Learning Research*, 22:241:1–241:124, 2021. 8

[16] Taekyung Kim and Changick Kim. Attract, perturb, and explore: Learning a feature alignment network for semi-supervised domain adaptation. In *European Conference on Computer Vision*, 2020. 1, 2

[17] Jichang Li, Guanbin Li, Yemin Shi, and Yizhou Yu. Cross-domain adaptive clustering for semi-supervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2505–2514, 2021. 1, 6

[18] Kai Li, Chang Liu, Handong Zhao, Yulun Zhang, and Yun Raymond Fu. Ecacl: A holistic framework for semi-supervised domain adaptation. In *IEEE International Conference on Computer Vision*, pages 8558–8567, 2021. 1

[19] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 510–519, 2019. 8

[20] Christopher Liao, Theodoros Tsiligkaridis, and Brian Kulis. Pick up the pace: Fast and simple domain adaptation via ensemble pseudo-labeling. *ArXiv*, abs/2205.13508, 2022. 2, 6, 7, 8

[21] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE International Conference on Computer Vision*, pages 9992–10002, 2021. 2, 3, 6

[22] Zhuang Liu, Hanzi Mao, Chaozheng Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 11966–11976, 2022. 1, 2, 3, 6

[23] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, 2018. 1

[24] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, 2018. 2

[25] Zhong Meng, Jinyu Li, Yong Zhao, and Yifan Gong. Conditional teacher-student learning. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6445–6449, 2019. 2

[26] Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. When does label smoothing help? In *Advances in Neural Information Processing Systems*, 2019. 2

[27] Peyman Passban, Yimeng Wu, Mehdi Rezagholizadeh, and Qun Liu. Alp-kd: Attention-based layer projection for knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021. 5

[28] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zach DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 6

[29] Jizong Peng, Guillermo Estrada, Marco Pedersoli, and Christian Desrosiers. Deep co-training for semi-supervised image segmentation. *Pattern Recognition*, abs/1903.11233, 2020. 2

[30] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *IEEE International Conference on Computer Vision*, pages 1406–1415, 2019. 1, 5

[31] Siyuan Qiao, Wei Shen, Zhishuai Zhang, Bo Wang, and Alan Loddon Yuille. Deep co-training for semi-supervised image recognition. In *European Conference on Computer Vision*, 2018. 2

[32] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *CoRR*, abs/1412.6550, 2015. 5

[33] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *IEEE International Conference on Computer Vision*, pages 8049–8057, 2019. 1, 2, 5, 6

[34] Kuniaki Saito, Y. Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. In *International Conference on Machine Learning*, 2017. 2

[35] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018. 2, 5, 6

[36] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3:637–646, 2016. 2

[37] Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6:1–48, 2019. 3

[38] Ankit Singh. Clda: Contrastive learning for semi-supervised domain adaptation. In *Advances in Neural Information Processing Systems*, 2021. 1, 2, 6, 8

[39] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*, 2020. 2, 4

[40] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016. 2, 3

[41] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016. 4

[42] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5385–5394, 2017. 6

[43] Yue Wang, Ziyu Jiang, Xiaohan Chen, Pengfei Xu, Yang Zhao, Yingyan Lin, and Zhangyang Wang. E2-train: Training state-of-the-art cnns with over 80% energy savings. In *Advances in Neural Information Processing Systems*, 2019. 8

[44] Garrett Wilson and Diane J Cook. A survey of unsupervised deep domain adaptation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(5):1–46, 2020. 1

[45] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems*, 2020. 2, 4

[46] Zizheng Yan, Yushuang Wu, Guanbin Li, Yipeng Qin, Xiaoguang Han, and Shuguang Cui. Multi-level consistency learning for semi-supervised domain adaptation. In *International Joint Conference on Artificial Intelligence*, 2022. 1, 2, 6, 8

[47] Luyu Yang, Yan Wang, Mingfei Gao, Abhinav Shrivastava, Kilian Q Weinberger, Wei-Lun Chao, and Ser-Nam Lim. Deep co-training with task decomposition for semi-supervised domain adaptation. In *IEEE International Conference on Computer Vision*, pages 8906–8916, 2021. 1

[48] Qing Yu, Atsushi Hashimoto, and Y. Ushiku. Divergence optimization for noisy universal domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2515–2524, 2021. 2

[49] Feng Zhang, Xiatian Zhu, and Mao Ye. Fast human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3512–3521, 2019. 2

[50] Zhi-Hua Zhou and Ming Li. Semi-supervised learning by disagreement. *Knowl. Inf. Syst.*, 24(3):415–439, sep 2010. 2