On the Benefits of Selectivity in Pseudo-Labeling for Unsupervised Multi-Source-Free Domain Adaptation

Maohao Shen[1] **Yuheng Bu**[1] **Gregory Wornell**[1]

[1]Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, USA

Abstract

Due to privacy, storage, and other constraints, there is a growing need for unsupervised domain adaptation techniques in machine learning that do not require access to the data used to train a collection of source models. Existing methods for such multi-source-free domain adaptation typically train a target model using supervised techniques in conjunction with pseudo-labels for the target data, which are produced by the available source models. However, we show that assigning pseudo-labels to only a subset of the target data leads to improved performance. In particular, we develop an information-theoretic bound on the generalization error of the resulting target model that demonstrates an inherent bias-variance trade-off controlled by the subset choice. Guided by this analysis, we develop a method that partitions the target data into pseudo-labeled and unlabeled subsets to balance the trade-off. In addition to exploiting the pseudo-labeled subset, our algorithm further leverages the information in the unlabeled subset via a traditional unsupervised domain adaptation feature alignment procedure. Experiments on multiple benchmark datasets demonstrate the superior performance of the proposed method.

1 INTRODUCTION

Machine learning models trained in a standard supervised manner suffer from the problem of domain shift [Quiñonero-Candela et al., 2008], i.e., directly applying the model trained on the source domain to a distinct target domain usually results in poor generalization performance. Unsupervised Domain Adaptation (UDA) techniques have been proposed to mitigate this issue by transferring the knowledge learned from a labeled source domain to an unlabeled target domain. One prevailing UDA strategy to resolve the domain shift issue is domain alignment, i.e., learning domain-invariant features either by minimizing the discrepancy between the source and target data [Long et al., 2015, 2018, Peng et al., 2019] or through adversarial training [Ganin and Lempitsky, 2015, Tzeng et al., 2017]. However, the traditional UDA method requires the access of labeled source data and only applies to the single source domain adaptation, which cannot fulfill the emerging challenges in many real-world applications.

In practice, the source data might not be available due to various reasons, including privacy preservation, i.e., the data that contains sensitive information such as health and financial status is unsuitable to be made public; and storage limitation, i.e., large-scale dataset such as high-resolution videos require substantial storage space. Due to these practical concerns, the source-free domain adaptation (SFDA) problem has attracted increasing attentions [Yang et al., 2020, Kim et al., 2020, Liang et al., 2020, Li et al., 2020], which aims to address the data-free challenge by adapting the pretrained source model to the unlabeled target domain.

The other practical concern is that the source data is usually collected from multiple domains with different underlying distributions such as the street scene from different cities [Cordts et al., 2016] and the biomedical images with different modalities [Dou et al., 2018]. Taking such practical consideration into account, multi-source domain adaptation (MSDA) [Guo et al., 2018, Peng et al., 2019] aims to adapt to the target domain by properly aggregating the knowledge from multiple source domains.

A more challenging scenario is to combine both the data-free and multi-source settings, i.e., the source data collected from multiple domains with distinct distributions is not accessible due to some practical constraints. For example, federated learning [Truong et al., 2021] aggregates the information learned from a group of heterogeneous users. To preserve user privacy, the data of each user is stored locally, and only the trained models are transmitted to the central server.

We consider the Multi-Source-Free Domain Adaptation (MSFDA) problem to overcome these two challenges. Note that the MSFDA problem is still less explored, and few methods have been proposed. Ahmed et al. [2021] uses a self-supervised pseudo-labeling method extended from the SFDA method [Liang et al., 2020] and ensembles the source models with trainable domain weights. Similarly, Dong et al. [2021] proposes a confident-anchor-induced pseudo-labeling method. All these methods focus on improving the pseudo-labeling quality for all target domain data but ignore that pseudo-labels generated by pretrained models are inevitably noisy and will induce bias during training. In addition, they do not explicitly address the domain shift issue in source and target domains. To address these limitations, we tackle the MSFDA problem based on two crucial insights, i.e., (1) assigning reliable pseudo-labels to a subset of target data could reduce the bias, and (2) it is important to leverage traditional domain alignment strategy to address the domain shift issue.

Consider the simple binary classification example in Figure 1, where the source domain decision boundary can successfully separate the source domain data. However, directly applying the source model to the target domain will generate incorrect pseudo-labels for target data close to the decision boundary, which will induce bias during training. One strategy to resolve this issue is partitioning the target domain data into a labeled subset with reliable pseudo-labels and a remaining unlabeled subset based on a well-designed confidence measure. Then the bias caused by noisy pseudo-labels can be mitigated by only using the labeled subset in supervised training. In addition, to tackle the distribution shift between the labeled and unlabeled subsets, we can further enforce the feature alignment between these two subsets as in the standard UDA method. Motivated by these observations, we also provide theoretical justifications for the proposed selective pseudo-labeling algorithm.

Our main contributions are summarized as follows:

- We develop an information-theoretic bound on the generalization error for the MSFDA problem, which demonstrates an inherent bias-variance trade-off controlled by the subset selection.
- Inspired by our analysis, we propose a novel solution for the MSFDA problem to balance the trade-off based on the two crucial ideas, i.e., (1) partition the target data into pseudo-labeled and unlabeled subsets based on a well-designed confidence measure; (2) leverage the information in the unlabeled subset via a traditional feature alignment procedure.
- Experiments across multiple representative benchmark datasets demonstrate the superior performance of the proposed algorithm over existing methods.

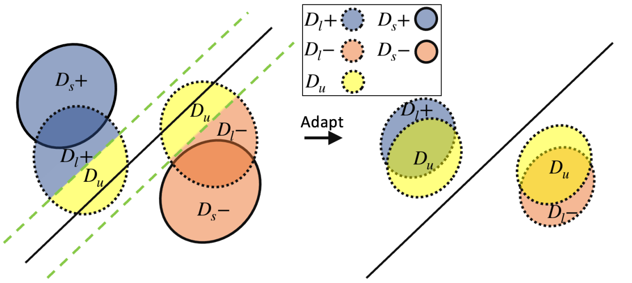

Figure 1: A binary classification example: Blue region denotes + class, and Orange region denotes - class. The regions with solid line and dashed line stand for source and target data, respectively. The source decision boundary (solid black line) can separate the source domain distributions ($\mathcal{D}_s+$ and $\mathcal{D}_s-$) perfectly, but will generate incorrect pseudo-labels for target domain data. Our idea is to use confidence thresholding (green dashed line) to partition the target data into pseudo-labeled subset $\mathcal{D}_l$ and unlabeled subset $\mathcal{D}_u$(Yellow region). After adaption, $\mathcal{D}_l$ and $\mathcal{D}_u$ should be aligned together and classified by the source decision boundary correctly.

2 RELATED WORK

In this section, we mainly discuss the literature on the following three variants of the traditional UDA that are closely relevant to the MSFDA problem.

Multi-source Domain Adaptation: Multi-source domain adaptation aims to transfer the knowledge from multiple distinct source domains to a target domain. Early theoretical works provide theoretical guarantees for the later empirical works by formally characterizing the connection between the source and target domains. Ben-David et al. [2010] introduces the $\mathcal{H}\Delta\mathcal{H}$ distance to measure the discrepancy between the target and source domains, and Mansour et al. [2009] assumes that the target distribution can be approximated by a linear combination of source distributions.

Many existing algorithms aim to mitigate the distribution shift issue between source and target domains. Discrepancy-based methods try to align the domain distribution by minimizing discrepancy loss, such as maximum mean discrepancy (MMD) [Guo et al., 2018], Rényi-divergence [Hoffman et al., 2018], moment distance [Peng et al., 2019], and a combination of several different discrepancy metrics [Guo et al., 2020]. Adversarial methods align the features between source and target domains by training a feature extractor that fools the discriminator under different loss functions, including $\mathcal{H}$-divergence [Zhao et al., 2018], traditional GAN loss [Xu et al., 2018], and Wasserstein distance [Li et al., 2018, Wang et al., 2019, Zhao et al., 2020].

Moreover, the domain weighting strategy is also widely used to quantify the contribution of each source domain, including uniform weights [Zhu et al., 2019], source model accuracy based weights [Peng et al., 2019], Wasserstein distance-based weights [Zhao et al., 2020]. More recently, Wang et al. [2020] proposes to aggregate the knowledge from multiple domains by learning a graph model.

Source-free Domain Adaptation However, all the methods mentioned above require source data and cannot be directly applied to the data-free setting. To this end, recent efforts have been made to tackle the source-free problem. Adversarial learning-based methods utilize generative models by either generating new samples from the source domain [Kurmi et al., 2021] or generating labeled samples from target distribution by conditional GAN [Li et al., 2020]. Another popular strategy is using the pseudo-labeling technique. Liang et al. [2020] uses self-supervised pseudo-labeling and maximizes the mutual information loss. Similarly, Kim et al. [2020] proposes a confidence-based filtering method to further improve the quality of pseudo-labels.

Multi-source-free Domain Adaptation To overcome both the data-free and multi-source challenges, Ahmed et al. [2021] uses a self-supervised pseudo-labeling method based on the SFDA algorithm in [Liang et al., 2020], and combines the source models using trainable domain weights. Similarly, [Dong et al., 2021] also focuses on improving the pseudo-labeling algorithm with a confident-anchor-induced pseudo-label generator. However, both these methods ignore the fact that pseudo-labels are inevitably noisy, and incorrect pseudo-labels will induce bias in training.

3 PROBLEM FORMULATION

Let $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ denotes the input space and $\mathcal{Y} = \{1, \cdots, K\}$ denotes the label space. A domain is a joint distribution P_{XY} on the instance space $\mathcal{Z}$. In this work, we aim to jointly adapt multiple pretrained models corresponding to m different source domains $\{P^{s_j}_{XY}\}_{j=1}^{m}$ to a new target domain P^t_{XY}.

Let $\boldsymbol{h}^{s_j} : \mathcal{X} \to \mathbb{R}^K$ denote the pretrained model for source domain j, which is a vector-valued function predicting the conditional distribution $P^{s_j}_{Y|X}$, i.e., $\boldsymbol{h}^{s_j} = [h^{s_j}_1, \cdots, h^{s_j}_K]^\top$, $\sum_{k=1}^{K} h^{s_j}_k = 1$ and $h^{s_j}_k \geq 0$. Each pretrained model can be decomposed into a feature extractor $\boldsymbol{f}^{s_j} : \mathcal{X} \to \mathbb{R}^{l_j}$, followed by a classifier $\boldsymbol{g}^{s_j} : \mathbb{R}^{l_j} \to \mathbb{R}^K$, where l_j is the dimension of feature representation. Thus, we can denote the prediction of input $\boldsymbol{x}$ using model $\boldsymbol{h}^{s_j}$ as $\boldsymbol{h}^{s_j}(\boldsymbol{x}) = (\boldsymbol{g}^{s_j} \circ \boldsymbol{f}^{s_j})(\boldsymbol{x})$.

Denote $\mathcal{D}^t \triangleq \{\boldsymbol{x}^t_i\}_{i=1}^{n}$ as the unlabeled target domain dataset, where $\boldsymbol{x}^t_i$ are i.i.d. generated from the target marginal distribution P^t_X. We denote the loss function as $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$, and our goal is to find a target model $\boldsymbol{h}$ that can minimize the population risk of target domain, i.e., $\mathcal{L}_P\left(\boldsymbol{h}, P^t_{XY}\right) \triangleq \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim P^t_{XY}}[\ell\left(\boldsymbol{h}\left(\boldsymbol{x}\right), \boldsymbol{y}\right)]$.

4 THEORETICAL ANALYSIS

In this section, we provide theoretical analysis for the multi-source-free domain adaptation problem. We show that there exists a bias and variance trade-off in using pseudo-labels generated from the pretrained source models, and we need to balance this trade-off in our algorithm design.

Suppose that for some unlabeled target samples $\boldsymbol{x}^t_i \in \mathcal{D}^t$, we can obtain its pseudo-label $\tilde{\boldsymbol{y}}^t_i$ by leveraging the pretrained models, and denote the subset of pseudo-labeled data as $\mathcal{D}_l \triangleq \{(\boldsymbol{x}^t_i, \tilde{\boldsymbol{y}}^t_i)\}_{i=1}^{n_l}$. To ensure that model learned by minimizing the empirical risk over $\mathcal{D}_l$, i.e.,

$$\mathcal{L}_E(\boldsymbol{h}, \mathcal{D}_l) \triangleq \frac{1}{n_l} \sum_{i=1}^{n_l} \ell(\boldsymbol{h}(\boldsymbol{x}^t_i), \tilde{\boldsymbol{y}}^t_i) \tag{1}$$

generalizes well to the target domain, we have the following upper bound on generalization gap, i.e., the difference between the population risk and the empirical risk.

Theorem 4.1. *Suppose that the samples of $\mathcal{D}_l$ are i.i.d. generated from the distribution $P^{\mathcal{D}_l}_{XY}$, the hypothesis space $\mathcal{H}$ has finite Natarajan dimension $d_N(\mathcal{H})$, then for zero-one loss function and any $\boldsymbol{h} \in \mathcal{H}$, there exists a constant C such that with probability $1 - \delta$,*

$$\mathcal{L}_P(\boldsymbol{h}, P^t_{XY}) - \mathcal{L}_E(\boldsymbol{h}, \mathcal{D}_l) \leq$$
$$C\sqrt{\frac{d_N(\mathcal{H}) \log K + \log \frac{1}{\delta}}{n_l}} + \sqrt{\frac{1}{2} D(P^{\mathcal{D}_l}_{XY} \| P^t_{XY})}. \tag{2}$$

Theorem 4.1 states that the generalization gap can be controlled by two terms that corresponds to variance and bias. The first term in equation 2 can be interpreted as the variance, since it only depends on the hypothesis space $\mathcal{H}$ and has the order of $\mathcal{O}(1/\sqrt{n_l})$. And the second term can be viewed as bias, where KL divergence $D(P^{\mathcal{D}_l}_{XY} \| P^t_{XY})$ is used to measure the discrepancy between the two domains.

In the following, we contrast two methods for single-source domain adaptation to show that it is important to only use a subset of the pseudo-labels generated from the source models to balance between the bias and variance in Theorem 4.1.

Unselective-pseudo-labeling: Generate pseudo-labels for all samples in $\mathcal{D}^t$ using source model $\boldsymbol{h}^s$. Then, the distribution of $\mathcal{D}_l$ is given by $P^{\mathcal{D}_l}_{XY} = P^t_X \otimes P^s_{Y|X}$. Although generating pseudo-labels for the entire $\mathcal{D}^t$ implies $n_l = n$, which gives a small variance term in Theorem 4.1, the bias term is given by

$$D(P^{\mathcal{D}_l}_{XY} \| P^t_{XY}) = D(P^s_{Y|X} \| P^t_{Y|X} | P^t_X). \tag{3}$$

If $P^t_{Y|X}(Y = \boldsymbol{y}|X = \boldsymbol{x}^t_i) = 0$ and $P^s_{Y|X}(Y = \boldsymbol{y}|X = \boldsymbol{x}^t_i) \neq 0$ for some $\boldsymbol{x}^t_i \in \mathcal{D}^t$ due to model mismatch, it leads to a large bias $D(P^s_{Y|X} \| P^t_{Y|X} | P^t_X) = \infty$.

Selective-pseudo-labeling: The aforementioned issue can be fixed by only generating pseudo-labels for a specific subset of $\mathcal{D}^t$. Suppose that $\mathcal{X}_{\mathcal{D}_l} \subset \mathcal{X}$ satisfies the condition

$$P^s_{Y|X}(\cdot|X = \boldsymbol{x}) \approx P^t_{Y|X}(\cdot|X = \boldsymbol{x}), \quad \forall \boldsymbol{x} \in \mathcal{X}_{\mathcal{D}_l}. \quad (4)$$

If we only generate pseudo-labels for $\boldsymbol{x}^t \in \mathcal{X}_{\mathcal{D}_l}$, then $P^{\mathcal{D}_l}_{XY} = P^{\mathcal{D}_l}_X \otimes P^s_{Y|X}$, where

$$P^{\mathcal{D}_l}_X(\boldsymbol{x}) = \frac{P^t_X(\boldsymbol{x})\mathbb{1}\{\boldsymbol{x} \in \mathcal{X}_{\mathcal{D}_l}\}}{P^t_X(\mathcal{X}_{\mathcal{D}_l})}, \quad (5)$$

and we have

$$D(P^{\mathcal{D}_l}_{XY} \| P^t_{XY}) \approx D(P^{\mathcal{D}_l}_X \| P^t_X) = -\log P^t_X(\mathcal{X}_{\mathcal{D}_l}). \quad (6)$$

Thus, as long as $P^t_X(\mathcal{X}_{\mathcal{D}_l})$ is large enough, i.e., the ratio of pseudo-labelled target samples n_l/n is large, the bias term can be controlled by a constant in this case. Thus, by balancing between the bias-variance trade-off, we can obtain better performance by carefully selecting $\mathcal{X}_{\mathcal{D}_l}$.

In practice, it is impossible to check the condition $P^s_{Y|X}(\cdot|X = \boldsymbol{x}) \approx P^t_{Y|X}(\cdot|X = \boldsymbol{x})$, since the target distribution $P^t_{Y|X}(\cdot|X = \boldsymbol{x})$ is unknown. Thus, we make the following assumption on different source domains, which allows us to identify the set $\mathcal{X}_{\mathcal{D}_l}$ by performing majority vote using *multiple* source models.

Assumption 1. *For each sample $\boldsymbol{x}^t_i \in \mathcal{D}^t$, denote $\tilde{y}^{s_j}_i \triangleq \arg\max_k h^{s_j}_k(\boldsymbol{x}^t_i)$ as the pseudo-label generated by each source model. If more than half of the pseudo-labels generated by the source models are the same, i.e., $\tilde{y}^t_i = \tilde{y}^{s_j}_i$ for $j \in \mathcal{S}$, where the index set $|\mathcal{S}| > m/2$, then the true target label $y^t_i = \tilde{y}^t_i$.*

Assumption 1 implies that if the predictions made by the majority of the source models are the same, then it gives the true label in the target domain, which formally states how the source domains are related to the target domain.

5 METHOD

In this section, inspired by our theoretical results, we present the proposed method for the MSFDA problem. Specifically, our algorithm contains three major components, including (1) generating reliable pseudo-labels for $\mathcal{D}^t$ and partitioning it into labeled set $\mathcal{D}_l$ and unlabeled set $\mathcal{D}_u$; (2) supervised training using labeled subset $\mathcal{D}_l$; and (3) unsupervised training by aligning the features between $\mathcal{D}_l$ and $\mathcal{D}_u$.

5.1 $\mathcal{D}_l$ AND $\mathcal{D}_u$ PARTITION

Motivated by Theorem 4.1 and the discussion of selective pseudo-labeling strategy, the core idea of the proposed algorithm is to balance the bias and variance trade-off in the MSFDA problem by partitioning the unlabelled target domain data $\mathcal{D}^t$ into two subsets, i.e., $\mathcal{D}_l \triangleq \{(\boldsymbol{x}^t_i, \tilde{y}^t_i)\}^{n_l}_{i=1}$ with reliable pseudo-labels and the remaining unlabeled subset $\mathcal{D}_u \triangleq \{\boldsymbol{x}^t_i\}^{n_u}_{i=1}$. Ideally, the labeled subset $\mathcal{D}_l$ should contain as much correctly pseudo-labeled data as possible. Then, our goal is to design a confidence score that can identify accurate pseudo-labels and filter out noisy labels. Specifically, we adopt a pseudo-label denoising trick to improve the quality of pseudo-labels and a confidence thresholding method to partition $\mathcal{D}_l$ and $\mathcal{D}_u$.

Pseudo-Label Denoising Pseudo-labeling techniques are widely used in semi-supervised learning [Lee et al., 2013, Shi et al., 2018, Rizve et al., 2021]. However, in the multi-source data-free domain adaptation problem, the predictions made by the pretrained source models on target data can be very noisy, so it is crucial to combine it with other labeling criterion to improve the quality of the pseudo-labels. Inspired by Zhang et al. [2021], we propose a prototype-based pseudo-label denoising method.

For each target sample $\boldsymbol{x}^i_t \in \mathcal{D}_t$, its pseudo-label is generated based on two different criterion: (1) the label distribution directly obtained from source models, i.e., $\boldsymbol{h}^{s_j}(\boldsymbol{x}^t_i) = (\boldsymbol{g}^{s_j} \circ \boldsymbol{f}^{s_j})(\boldsymbol{x}^t_i)$, and (2) the clustering structure contained in the feature space of each source model. Specifically, we define the label distribution $[q^{s_j}_1, \cdots, q^{s_j}_K]^\top$ using the distance to prototypes in feature space by ignoring the classifier $\boldsymbol{g}^{s_j}$, where

$$q^{s_j}_k(\boldsymbol{x}^t_i) = \frac{\exp\left(-\left\|\boldsymbol{f}^{s_j}(\boldsymbol{x}^t_i) - \boldsymbol{\eta}^{s_j}_k\right\|/\tau\right)}{\sum_{k'=1}^K \exp\left(-\left\|\boldsymbol{f}^{s_j}(\boldsymbol{x}^t_i) - \boldsymbol{\eta}^{s_j}_{k'}\right\|/\tau\right)}. \quad (7)$$

Here, τ denotes the softmax temperature, and $\boldsymbol{\eta}^{s_j}_k$ is the prototype of k-th class given by source model $\boldsymbol{h}^{s_j}$, which is computed only using samples from $\mathcal{D}_l$, i.e.,

$$\boldsymbol{\eta}^{s_j}_k = \frac{\sum_{\boldsymbol{x}^t_i \in \mathcal{D}_l} \boldsymbol{f}^{s_j}(\boldsymbol{x}^t_i) \cdot \mathbb{1}\{\tilde{y}^t_i = k\}}{\sum_{\boldsymbol{x}^t_i \in \mathcal{D}_l} \mathbb{1}\{\tilde{y}^t_i = k\}}. \quad (8)$$

If we further assume these two labeling criterion are independent with each other, then the probability that they agree on the same pseudo-label k is the product of these two distribution, i.e.,

$$p^{s_j}_k(\boldsymbol{x}^t_i) = h^{s_j}_k(\boldsymbol{x}^t_i) \cdot q^{s_j}_k(\boldsymbol{x}^t_i), \quad (9)$$

which can be interpreted as a pseudo-label confidence score on target sample $\boldsymbol{x}^t_i$ using model $\boldsymbol{h}^{s_j}$.

Domain Weights As we have multiple source models in the MSFDA problem, it is critical to aggregate the information from multiple source models to generate more accurate pseudo-labels. One naive approach is to average the predictions from multiple models uniformly [Zhu et al., 2019]. However, each source domain may have different transferability to the target domain, and the contribution of each source domain to the target domain may not be the same.

Without accessing the target data labels, it is hard to know the exact performance of pretrained models on the target domain. One possible way is to use the prediction uncertainty to measure the transfer-ability of pretrained models. Intuitively, a source model with more prediction uncertainty is more likely to perform worse in the target domain. Thus, we can define the domain weights as the softmax over the entropy of the predictions given by each source model, i.e.,

$$\boldsymbol{w}_j = \frac{\exp\left(-\frac{1}{n}\sum_{\boldsymbol{x}_i^t \in \mathcal{D}^t} \mathrm{H}\left(\boldsymbol{h}^{s_j}(\boldsymbol{x}_i^t)\right)\right)}{\sum_{j'=1}^{m} \exp\left(-\frac{1}{n}\sum_{\boldsymbol{x}_i^t \in \mathcal{D}^t} \mathrm{H}\left(\boldsymbol{h}^{s_{j'}}(\boldsymbol{x}_i^t)\right)\right)}, \tag{10}$$

where $\mathrm{H}(\boldsymbol{h}) \triangleq -\sum_{k=1}^{K} h_k \log h_k$ denotes the entropy of the model output, which measures the uncertainty of each pretrained model in the target domain.

Finally, combining the pseudo-label denoising trick with the domain weights discussed above, we can generate the pseudo-label by computing the weighted average of the confidence score given by each source model, i.e.,

$$\tilde{y}_i^t = \arg\max_k \sum_{j=1}^{m} \boldsymbol{w}_j \cdot p_k^{s_j}(\boldsymbol{x}_i^t). \tag{11}$$

Confidence Thresholding After generating the pseudo-labels, we use confidence-thresholding to partition the unlabeled target data $\mathcal{D}^t$ into $\mathcal{D}_l$ and $\mathcal{D}_u$. The quantity $\sum_{j=1}^{m} \boldsymbol{w}_j \cdot p_k^{s_j}(\boldsymbol{x}_i^t)$ can be interpreted as the weighted confidence score of class k. If this score is high, the two labeling criteria for multiple source models agree on the pseudo-labeling with high certainty, which implies that the generated pseudo-label is more likely to be correct. Thus, by setting a confidence threshold α, we partition $\mathcal{D}_l$ and $\mathcal{D}_u$ based on the following rule

$$\begin{cases} (\boldsymbol{x}_i^t, \tilde{y}_i^t) \in \mathcal{D}_l, & \text{if } \max_k \sum_{j=1}^{m} \boldsymbol{w}_j \cdot p_k^{s_j}(\boldsymbol{x}_i^t) > \alpha, \\ \boldsymbol{x}_i^t \in \mathcal{D}_u, & \text{Otherwise.} \end{cases} \tag{12}$$

Notice that the optimal threshold is unknown in practice, so we empirically set the value of α, and more details can be found in the Appendix B.2.

5.2 SUPERVISED TRAINING USING $\mathcal{D}_l$

The subset $\mathcal{D}_l$ with reliable pseudo-labels enables us to train the target model in a supervised manner. Since we cannot access the source training data, we fix the source model classifier $\boldsymbol{g}^{s_j}$, and update each feature extractor $\boldsymbol{f}^{s_j}$ to adapt the target domain, which implicitly incorporate the information from the source domain. Therefore, the cross-entropy loss for each feature extractor $\boldsymbol{f}^{s_j}$ is given by

$$\mathcal{L}_{\text{ce}}\left(\boldsymbol{f}^{s_j}, \mathcal{D}_l\right) = -\frac{1}{n_l} \sum_{\boldsymbol{x}_i^t, \tilde{y}_i^t \in \mathcal{D}_l} \sum_{k=1}^{K} \mathbb{1}\{\tilde{y}_i^t = k\} \log h_k^{s_j}\left(\boldsymbol{x}_i^t\right).$$

Moreover, the features learned from $\mathcal{D}_l$ should also be discriminative and diversely distributed, which both benefits the prototype estimation and the feature alignment. $\mathcal{L}_{\text{IM}}$ is the information maximization loss [Gomes et al., 2010, Hu et al., 2017] that can encourage the source models to make individual certain but global diverse predictions.

$$\begin{aligned} \mathcal{L}_{\text{IM}}\left(\boldsymbol{f}^{s_j}, \mathcal{D}_l\right) = & -\frac{1}{n_l} \sum_{\boldsymbol{x}_i^t \in \mathcal{D}_l} \sum_{k=1}^{K} h_k^{s_j}\left(\boldsymbol{x}_i^t\right) \log\left(h_k^{s_j}\left(\boldsymbol{x}_i^t\right)\right) \\ & + \sum_{k=1}^{K} \overline{p}_k^{s_j} \log \overline{p}_k^{s_j}, \end{aligned} \tag{13}$$

where $\overline{p}_k^{s_j} \triangleq \frac{1}{n_l} \sum_{\boldsymbol{x}_i^t \in \mathcal{D}_l} h_k^{s_j}\left(\boldsymbol{x}_i^t\right)$. Different from previous works [Liang et al., 2020, Ahmed et al., 2021], we only apply information maximization loss on $\mathcal{D}_l$ to avoid incorrect pseudo-labels in $\mathcal{D}_u$. Thus, the training objectives on $\mathcal{D}_l$ is given as $\mathcal{L}_{\mathcal{D}_l} = \mathcal{L}_{\text{ce}} + \mathcal{L}_{\text{IM}}$.

5.3 FEATURE ALIGNMENT BETWEEN $\mathcal{D}_l$ AND $\mathcal{D}_u$

Different from the labeled set $\mathcal{D}_l$, the samples in $\mathcal{D}_u$ are unlabeled since the pseudo-labels generated via equation 10 are noisy. We will show that we can still leverage the information contained in $\mathcal{D}_u$ by using the traditional unsupervised domain adaptation technique, which aligns the feature distributions of both $\mathcal{D}_l$ and $\mathcal{D}_u$.

For sample $\boldsymbol{x}_i^t \in \mathcal{D}_l$, the pseudo-label is more reliable since $P_{Y|X}^{s_j}(\cdot|X = \boldsymbol{x}_i^t) \approx P_{Y|X}^{t}(\cdot|X = \boldsymbol{x}_i^t)$, for the majority of the source models. However, $P_{Y|X}^{s_j}(\cdot|X = \boldsymbol{x}_i^t) \neq P_{Y|X}^{t}(\cdot|X = \boldsymbol{x}_i^t)$ for data $\boldsymbol{x}_i^t \in \mathcal{D}_u$, which implies that the data generating distributions for $\mathcal{D}_l$ and $\mathcal{D}_u$ are different, i.e., $P_{XY}^{\mathcal{D}_l} \neq P_{XY}^{\mathcal{D}_u}$. Similar to traditional UDA where the source and target domain have different distributions, there exists a distribution shift between $\mathcal{D}_l$ and $\mathcal{D}_u$. Therefore, we can use traditional UDA strategy, i.e., treat $\mathcal{D}_l$ as the labeled "source" data and $\mathcal{D}_u$ as the unlabeled "target" data, and enforce the feature alignment between $\mathcal{D}_l$ and $\mathcal{D}_u$.

Specifically, we adopt the adversarial training strategy proposed in [Ganin and Lempitsky, 2015, Xu et al., 2018] to perform feature alignment. For each source model $\boldsymbol{h}^{s_j}$, we train a separate neural network $\boldsymbol{d}^{s_j} : \mathbb{R}^{l_j} \to [0, 1]$ as discriminator to distinguish the features computed using $\mathcal{D}_l$ and $\mathcal{D}_u$, and update the feature extractor $\boldsymbol{f}^{s_j}$ to fool the discriminator. The feature alignment loss is given by the following adversarial loss,

$$\begin{aligned} \mathcal{L}_{\text{adv}}\left(\boldsymbol{f}^{s_j}, \boldsymbol{d}^{s_j}, \mathcal{D}_l, \mathcal{D}_u\right) = & \, \mathbb{E}_{\boldsymbol{x}_i^t \in \mathcal{D}_l} \ln \boldsymbol{d}^{s_j}\left(\boldsymbol{f}^{s_j}\left(\boldsymbol{x}_i^t\right)\right) \\ & + \mathbb{E}_{\boldsymbol{x}_i^t \in \mathcal{D}_u} \ln(1 - \boldsymbol{d}^{s_j}\left(\boldsymbol{f}^{s_j}\left(\boldsymbol{x}_i^t\right)\right)). \end{aligned} \tag{14}$$

In summary, we fix the classifier $\boldsymbol{g}^{s_j}$ for each source model, and alternating between the training of the discriminator $\boldsymbol{d}^{s_j}$

Algorithm 1 Unsupervised Multi-source-free Domain Adaptation

```
Input: pretrained source models $\{\boldsymbol{h}^{s_j} = \boldsymbol{g}^{s_j} \circ \boldsymbol{f}^{s_j}\}_{j=1}^{m}$,
       target domain unlabeled data $\mathcal{D}^t = \{\boldsymbol{x}_i^t\}_{i=1}^{n}$, and
       maximum iterations $T$
Initialize the domain weights $\boldsymbol{w}_j$ by equation 10
Initialize the pseudo-label $\tilde{y}_i^t$ for each target data $\boldsymbol{x}_i^t$ by equation 11 using pretrained models $\{\boldsymbol{h}^{s_j}\}_{j=1}^{m}$ and domain weights $\boldsymbol{w}_j$
for $\tau = 1, 2, \ldots, T$ do
    Update $\mathcal{D}_l$ and $\mathcal{D}_u$ by equation 12
    for $j = 1, 2, \ldots, m$ do
        // Update each source model
        Update discriminators $\boldsymbol{d}^{s_j^{(\tau)}}$ by maximizing equation 14.
        Update feature extractor $\boldsymbol{f}^{s_j^{(\tau)}}$ by minimizing equation 15.
    end
    Update the pseudo-label $\tilde{y}_i^t$ using domain weights $\boldsymbol{w}_j$ and updated models $\{\boldsymbol{h}^{s_j^{(\tau)}}\}_{j=1}^{m}$ by equation 11.
end
Output: Updated models $\{\boldsymbol{h}^{s_j^{(T)}}\}_{j=1}^{m}$, domain weights $\boldsymbol{w}_j$
```

to maximize $\mathcal{L}_{\text{adv}}$ and the training of the feature extractor $\boldsymbol{f}^{s_j}$ to minimize the joint loss, i.e.,

$$\begin{aligned} &\max_{\boldsymbol{d}^{s_j}} \mathcal{L}_{\text{adv}}\left(\boldsymbol{f}^{s_j}, \boldsymbol{d}^{s_j}, \mathcal{D}_l, \mathcal{D}_u\right) \\ &\min_{\boldsymbol{f}^{s_j}} \mathcal{L}_{\text{ce}}\left(\boldsymbol{f}^{s_j}, \mathcal{D}_l\right) + \lambda_{\text{IM}} \mathcal{L}_{\text{IM}}\left(\boldsymbol{f}^{s_j}, \mathcal{D}_l\right) \\ &\qquad + \lambda_{\text{adv}} \mathcal{L}_{\text{adv}}\left(\boldsymbol{f}^{s_j}, \boldsymbol{d}^{s_j}, \mathcal{D}_l, \mathcal{D}_u,\right) \end{aligned} \tag{15}$$

where λ_{IM} and λ_{adv} are hyper-parameters that balance different regularization terms.

5.4 ALGORITHM

The overall algorithm is shown in Algorithm 1. All the models are retrained in a disjoint manner using the updated pseudo-labels at each iteration. As the quality of each model increases, we expect to see more correctly labeled samples in $\mathcal{D}_l$ and fewer unlabeled samples in $\mathcal{D}_u$. When the algorithm converges, the final prediction of the target data $\boldsymbol{x}_i^t$ is obtained by the ensemble of multiple updated models.

6 EXPERIMENT RESULTS

In this section, we describe the details of experiment settings in Section 6.1, present the main results in 6.2, and discuss our takeaways in section 6.3.

6.1 SETTINGS

Datasets We conduct extensive evaluations our methods using the following five benchmark datasets: **Digits-Five** [Peng et al., 2019] contains five different domains, including MNIST (MN), SVHN (SV), USPS (US), MNIST-M (MM), and Synthetic Digits (SY). **Office-31** [Saenko et al., 2010] contains 31 categories collected from three different office environments, including Amazon(A), Webcam (W), and DSLR (D). **Office-Caltech** [Gong et al., 2012] contains four domains with 10 categories, which is extended from the Office-31 dataset by adding the additional domain Caltech-256 (C). **Office-Home** [Venkateswara et al., 2017] is a more challenging dataset with 65 categories collected from four different office environments, including Art (A), Clipart (C), Real-world (R), and Product (P). **DomainNet** [Peng et al., 2019] is so far the largest and most challenging domain adaptation benchmark, which contains about 0.6 million images with 345 categories collected from six different domains, including Clipart (C), Infograph (I), Painting (P), Quickdraw (Q), Real (R), and Sketch (S).

Baselines To demonstrate the solid empirical performance of our method, we mainly compare it with the recently proposed SOTA multi-source-free domain adaptation method DECISION [Ahmed et al., 2021]. We provide another baseline by evaluating ensemble prediction accuracy on the target domain using the pretrained source models, denote it as Source-ens. In addition, we also include several SOTA single-source-free domain adaptation methods, including BAIT [Yang et al., 2020], SFDA [Kim et al., 2020], SHOT [Liang et al., 2020], and MA [Li et al., 2020]. These single-source-free methods also do not require access to the source data, and we compare their multi-source ensemble results by taking the average of predictions from the multiple retrained source models after adaptation.

6.2 RESULTS

The results on Digits-Five, Office-31, Office-Caltech, Office-Home, DomainNet are shown in the corresponding Tables. Across these five datasets, our proposed method can outperform all baseline methods in terms of average accuracy. The SOTA method Ahmed et al. [2021] shows strong performance over single-source baselines, but our proposed method can still outperform it with significant improvements over several domain adaptation tasks.

Digits-Five Our method achieves a performance gain of 14.4% over the ensemble of pretrained source models on this dataset. In addition, for some challenging tasks where the ensemble of source models gives a poor performance on the target domain, e.g., M-MNIST, our proposed method outperforms the SOTA method by a large margin of 3.4%.

Office-31 It can be seen from Table 2 that most baseline

Table 1: **Results on Digit-Five (5 domains):** MN,SV,US,MM and SY stand for MNIST, SVHN, USPS, MNIST-M and Synthetic Digits datasets, respectively. Source-ens denotes the ensemble prediction of multiple pretrained source models.

Setting	**Method**	**MN**	**SV**	**US**	**MM**	**SY**	**Avg**
Single-source	BAIT [Yang et al., 2020]	96.2	60.6	96.7	87.6	90.5	86.3
	SFDA [Kim et al., 2020]	95.4	57.4	95.8	86.2	84.8	83.9
	SHOT [Liang et al., 2020]	98.9	58.3	97.7	90.4	83.9	85.8
	MA [Li et al., 2020]	98.4	59.1	98.0	90.8	84.5	86.2
Multi-source	Source-ens	96.7	76.8	93.8	66.7	77.6	82.3
	DECISION [Ahmed et al., 2021]	**99.2**	89.1	97.6	93.5	96.6	95.2
	Ours	99.1	**90.7**	**98.2**	**96.9**	**98.4**	**96.7**

Table 2: **Results on Office-31 (3 domains) and Office-Caltech (4 domains):** A,C,D and W stand for Amazon, Caltech, DSLR and Webcam datasets, respectively.

Setting	**Method**	**D,W**	**A,D**	**A,W**	**Avg**	**C,D,W**	**A,D,W**	**A,C,W**	**A,C,D**	**Avg**
		→ A	**→ W**	**→ D**		**→ A**	**→ C**	**→ D**	**→ W**	
Single-source	BAIT [Yang et al., 2020]	71.1	98.5	98.8	89.5	97.5	95.7	97.5	98.0	97.2
	SFDA [Kim et al., 2020]	73.2	93.8	96.7	87.9	97.3	94.6	97.1	97.6	96.7
	SHOT [Liang et al., 2020]	75.0	94.9	97.8	89.3	95.7	95.8	96.8	99.6	97.0
	MA [Li et al., 2020]	75.2	96.1	97.3	89.5	95.7	95.6	97.2	99.8	97.1
Multi-source	Source-ens	62.5	95.8	98.7	86.0	94.9	91.7	98.7	98.6	96.0
	DECISION [Ahmed et al., 2021]	74.7	97.6	98.8	90.4	95.3	95.1	98.7	99.3	97.1
	Ours	**75.7**	**98.1**	**100.0**	**91.3**	**96.2**	**95.8**	**99.4**	**100.0**	**97.9**

Table 3: **Results on Office-home (4 domains):** A,C,R and P stand for Art, Clipart, Real-world and Product datasets, respectively.

Setting	**Method**	**C,R,P→ A**	**A,R,P→ C**	**A,C,P→ R**	**A,C,R→ P**	**Avg**
Single-source	BAIT [Yang et al., 2020]	71.1	59.6	77.2	79.4	71.8
	SFDA [Kim et al., 2020]	69.3	57.5	76.8	79.1	70.7
	SHOT [Liang et al., 2020]	72.2	59.3	82.9	82.8	74.3
	MA [Li et al., 2020]	72.5	57.4	81.7	82.3	73.5
Multi-source	Source-ens	67.0	52.1	78.6	74.8	68.1
	DECISION [Ahmed et al., 2021]	74.2	59.1	**84.7**	84.5	75.6
	Ours	**75.5**	**64.1**	84.6	**85.1**	**77.3**

Table 4: **Results on DomainNet (6 domains):** C,I,P,Q,R and S stand for Clipart, Infograph, Painting, Quickdraw, Real, and Sketch, respectively.

Setting	**Method**	**C**	**I**	**P**	**Q**	**R**	**S**	**Avg**
Single-source	BAIT [Yang et al., 2020]	57.5	22.8	54.1	14.7	64.6	49.2	43.8
	SFDA [Kim et al., 2020]	57.2	23.6	55.1	16.4	65.5	47.3	44.2
	SHOT [Liang et al., 2020]	58.6	25.2	55.3	15.3	70.5	52.4	46.2
	MA [Li et al., 2020]	56.8	24.3	53.5	15.7	66.3	48.1	44.1
Multi-source	Source-ens	49.4	20.8	48.3	10.6	63.8	46.4	39.9
	DECISION [Ahmed et al., 2021]	63.2	22.3	54.6	**18.2**	67.9	51.4	46.3
	Ours	**65.7**	**25.5**	**56.5**	16.1	**69.1**	**53.1**	**47.7**

methods perform very well on the tasks A,W → D and A,D → W, which implies that Domain D and W are similar. Moreover, our proposed method can still exhibit a further improvement and achieve 100% on the A,W → D task.

Office-Caltech Extended from Office-31 dataset with additional domain C, the average performance of our method increases to 97.9% and outperforms all other baseline methods while also achieving 100% accuracy on the A,C,D →

Table 5: **Performance Upper-bound Results on Office-Home:** Upper-bound denotes the performance upper-bound.

Setting	Method	C,R,P → A	A,R,P → C	A,C,P → R	A,C,R → P	Avg
Data Free	Ours	75.5	64.1	84.6	85.1	77.3
	upper-bound	**82.3**	**81.4**	**90.4**	**93.7**	**87.0**

W domain adaptation task.

Office-Home This large dataset is more challenging than other Office datasets. For the most difficult task of this dataset: A, R, P → C, our approach significantly outperforms the SOTA method [Ahmed et al., 2021] by 5.0%.

DomainNet This is the most challenging domain adaptation benchmark so far. Our method still shows superior performance over all baseline methods, except on Quickdraw task. Besides, our method shows significant improvement on Clipart task.

6.3 DISCUSSIONS

In this subsection, we discuss the following three aspects of the proposed method to better understand the data-free domain adaptation problem.

Performance Limit: We have shown that the proposed MSFDA algorithm outperforms the SOTA methods, and it is interesting to see if there is more room for improvement in this data-free domain adaptation problem. To this end, we need to construct an upper bound for the MSFDA problem to help us to understand the performance limit. Without any supervision, the domain adaptation performance highly depends on the quality of pretrained models, i.e., the pretrained source model with high transfer-ability will easily adapt to the target domain, and vice versa. According to our theoretical analysis, the best performance we can achieve is to utilize all correct pseudo-labeled data for training and ignore the remaining noisy pseudo-labeled data, which leads to an upper bound for the performance of the MSFDA problem.

However, it is impossible to guarantee that all the data in $\mathcal{D}_l$ have accurate pseudo-labels in practice. To construct an upper bound for the performance of the data-free domain adaptation problem, we assume that an “oracle” can identify the samples with correct pseudo-labels. Suppose we use this “oracle” to partition the subsets $\mathcal{D}_l$ and $\mathcal{D}_u$ in our proposed algorithm instead of the confidence thresholding in equation 12. In this case, the resulting prediction accuracy can be viewed as a limit on the performance. The results are shown in Table 5. Without source data, this performance upper bound is the best we can achieve. However, notice that we cannot access such “oracle” in practice. The upper bound is unreachable, and the goal of the data-free domain adaptation algorithm is to narrow the performance gap.

Table 6: **Ablation Study Results on Office-Home:** w/o alignment and w/o denoise denote the performance without the domain alignment loss and without pseudo-label denoise method, respectively.

Setting	Method	C,R,P → A	A,R,P → C	A,C,P → R	A,C,R → P	Avg
Data Free	w/o alignment	75.0	63.7	83.5	84.4	76.7
	w/o denoise	74.9	62.6	83.6	83.8	76.2
	Ours	**75.5**	**64.1**	**84.6**	**85.1**	**77.3**

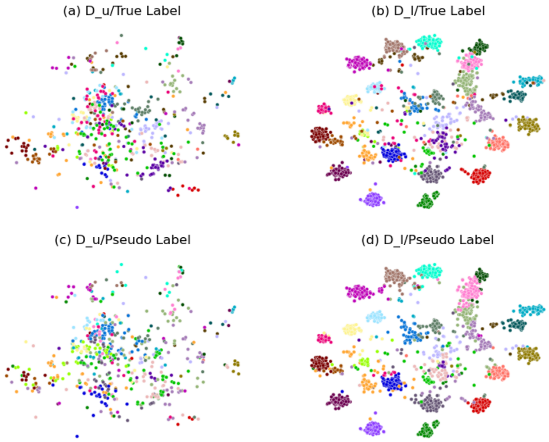

Figure 2: t-SNE visualization on D,W → A domain adaptation task of Office-31 dataset: (a) plot of $\mathcal{D}_u$ (705 data) labeled with ground-truth label. (b) plot of $\mathcal{D}_l$ (2112 data) labeled with ground-truth label. (c) plot of $\mathcal{D}_u$ (705 data) labeled with pseudo-label. (d) plot of $\mathcal{D}_l$ (2112 data) labeled with pseudo-label.

Ablation Study: It is essential to see the contributions of other components, e.g., the feature alignment loss $\mathcal{L}_{\text{adv}}$ and the pseudo-label denoising method used in our algorithm. We take the Office-Home dataset as an example to perform the ablation study. Specifically, we evaluate the performance of the following two variants of our proposed method: (1) w/o alignment, removing the feature alignment loss $\mathcal{L}_{\text{adv}}$, (2) w/o denoise, removing pseudo-label denoising method. It can be observed from Table 6, without feature alignment loss, the performance degrades marginally, which implies the supervised training on $\mathcal{D}_l$ can also implicitly encourage domain adaptation. Similarly, removing the pseudo-label denoising method also leads to slightly worse performance. These ablation study results imply the major performance improvement comes from the proposed selective pseudo-labeling strategy and further validate the importance of balancing the bias and variance trade-off for the MSFDA problem.

Visualization: To better understand why it is crucial to use the selective pseudo labeling method, we provide t-SNE plots in feature space to visualize the difference between $\mathcal{D}_l$ and $\mathcal{D}_u$, and each of them is labeled in two ways: using ground-truth labels and using pseudo labels directly generated from source models. We take task D,W → A of Office-31 dataset as our example and extract the target domain features from pretrained source models. The results are presented in Figure 2. From the figure, we can find that: (1) There are mismatches between pseudo labels of $\mathcal{D}_u$ and ground truth labels, i.e., the pseudo labels of $\mathcal{D}_u$ are much noisier than $\mathcal{D}_l$. Thus, we should only use data in D_l for training to avoid the bias in $\mathcal{D}_u$. (2) The feature of samples in $\mathcal{D}_l$ are more separately clustered compared to $\mathcal{D}_u$, which implies the existence of distribution shift between $\mathcal{D}_l$ and $\mathcal{D}_u$. To mitigate this discrepancy, we align the feature distributions for both $\mathcal{D}_l$ and $\mathcal{D}_u$ by minimizing the loss $\mathcal{L}_{\text{adv}}$ in the proposed method.

7 CONCLUDING REMARKS

This work develops a novel solution for the multi-source-free domain adaptation problem. We demonstrate the benefits of selective pseudo-labeling on target domain data from theoretical and empirical perspectives.

We provide a empirical performance upper bound of the MSFDA problem in our experiments, and it would be interesting to see that there is still some room for further improvements in this source-free setting. Therefore, identifying a better strategy to balance the trade-off and characterizing the fundamental limit of the MSFDA problem from a theoretical perspective are potential future works.

References

Sk Miraj Ahmed, Dripta S Raychaudhuri, Sujoy Paul, Samet Oymak, and Amit K Roy-Chowdhury. Unsupervised multi-source domain adaptation without access to source data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10103–10112, 2021.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

Amit Daniely, Sivan Sabato, Shai Ben-David, and Shai Shalev-Shwartz. Multiclass learnability and the ERM principle. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 207–232. JMLR Workshop and Conference Proceedings, 2011.

Jiahua Dong, Zhen Fang, Anjin Liu, Gan Sun, and Tongliang Liu. Confident anchor-induced multi-source free domain adaptation. *Advances in Neural Information Processing Systems*, 34, 2021.

Qi Dou, Cheng Ouyang, Cheng Chen, Hao Chen, and Pheng-Ann Heng. Unsupervised cross-modality domain adaptation of convnets for biomedical image segmentations with adversarial loss. *arXiv preprint arXiv:1804.10916*, 2018.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

Ryan Gomes, Andreas Krause, and Pietro Perona. Discriminative clustering by regularized information maximization. 2010.

Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2066–2073. IEEE, 2012.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. Multi-source domain adaptation for text classification via distancenet-bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7830–7838, 2020.

Jiang Guo, Darsh J Shah, and Regina Barzilay. Multi-source domain adaptation with mixture of experts. *arXiv preprint arXiv:1809.02256*, 2018.

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

Judy Hoffman, Mehryar Mohri, and Ningshan Zhang. Algorithms and theory for multiple-source adaptation. *arXiv preprint arXiv:1805.08727*, 2018.

Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama. Learning discrete representations via information maximizing self-augmented training. In *International conference on machine learning*, pages 1558–1567. PMLR, 2017.

Youngeun Kim, Donghyeon Cho, Kyeongtak Han, Priyadarshini Panda, and Sungeun Hong. Domain adaptation without source data. *arXiv preprint arXiv:2007.01524*, 2020.

Vinod K Kurmi, Venkatesh K Subramanian, and Vinay P Namboodiri. Domain impression: A source data free domain adaptation method. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 615–625, 2021.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, 2013.

Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9641–9650, 2020.

Yitong Li, Michael Murias, Samantha Major, Geraldine Dawson, and David E Carlson. Extracting relationships by multi-domain matching. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 6799–6810, 2018.

Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 6028–6039. PMLR, 2020.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

Mingsheng Long, Yue Cao, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Transferable representation learning with deep adaptation networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(12):3071–3085, 2018.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation with multiple sources. 2009.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1406–1415, 2019.

Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. Mit Press, 2008.

Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. *arXiv preprint arXiv:2101.06329*, 2021.

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European conference on computer vision*, pages 213–226. Springer, 2010.

Weiwei Shi, Yihong Gong, Chris Ding, Zhiheng MaXiaoyu Tao, and Nanning Zheng. Transductive semi-supervised deep learning using min-max features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 299–315, 2018.

Nguyen Truong, Kai Sun, Siyao Wang, Florian Guitton, and YiKe Guo. Privacy preservation in federated learning: An insightful survey from the GDPR perspective. *Computers & Security*, 110:102402, 2021.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5018–5027, 2017.

Hang Wang, Minghao Xu, Bingbing Ni, and Wenjun Zhang. Learning to combine: Knowledge aggregation for multi-source domain adaptation. In *European Conference on Computer Vision*, pages 727–744. Springer, 2020.

Haotian Wang, Wenjing Yang, Zhipeng Lin, and Yue Yu. Tmda: Task-specific multi-source domain adaptation via clustering embedded adversarial training. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 1372–1377. IEEE, 2019.

Ruijia Xu, Ziliang Chen, Wangmeng Zuo, Junjie Yan, and Liang Lin. Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3964–3973, 2018.

Shiqi Yang, Yaxing Wang, Joost van de Weijer, Luis Herranz, and Shangling Jui. Unsupervised domain adaptation without source data by casting a bait. *arXiv preprint arXiv:2010.12427*, 2020.

Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12414–12424, 2021.

Han Zhao, Shanghang Zhang, Guanhang Wu, José MF Moura, Joao P Costeira, and Geoffrey J Gordon. Adversarial multiple source domain adaptation. *Advances* *in neural information processing systems*, 31:8559–8570, 2018.

Sicheng Zhao, Guangzhi Wang, Shanghang Zhang, Yang Gu, Yaxian Li, Zhichao Song, Pengfei Xu, Runbo Hu, Hua Chai, and Kurt Keutzer. Multi-source distilling domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12975–12983, 2020.

Yongchun Zhu, Fuzhen Zhuang, and Deqing Wang. Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5989–5996, 2019.

A PROOF OF THEOREM 4.1

The gap between the empirical risk $\mathcal{L}_E(\boldsymbol{h}, \mathcal{D}_l)$ over $\mathcal{D}_l$ and $\mathcal{L}_P(\boldsymbol{h}, P^t_{XY})$ can be written as

$$\begin{aligned}
&\mathcal{L}_P(\boldsymbol{h}, P^t_{XY}) - \mathcal{L}_E(\boldsymbol{h}, \mathcal{D}_l) \\
&= \mathcal{L}_P(\boldsymbol{h}, P^t_{XY}) - \mathcal{L}_P(\boldsymbol{h}, P^{\mathcal{D}_l}_{XY}) + \mathcal{L}_P(\boldsymbol{h}, P^{\mathcal{D}_l}_{XY}) - \mathcal{L}_E(\boldsymbol{h}, \mathcal{D}_l) \\
&\leq |\mathcal{L}_P(\boldsymbol{h}, P^{\mathcal{D}_l}_{XY}) - \mathcal{L}_E(\boldsymbol{h}, \mathcal{D}_l)| + |\mathcal{L}_P(\boldsymbol{h}, P^t_{XY}) - \mathcal{L}_P(\boldsymbol{h}, P^{\mathcal{D}_l}_{XY})|.
\end{aligned} \tag{16}$$

We note that the first term is simply the generalization error of supervised learning using n_l i.i.d. samples generated from the distribution $P^{\mathcal{D}_l}_{XY}$. Since $\boldsymbol{h} \in \mathcal{H}$ has finite Natarajan dimension, by Natarajan dimension theory (see [Daniely et al., 2011] equation (6)), the following upper bound holds for some constant $C > 0$ with probability at least $1 - \delta$,

$$|\mathcal{L}_P(\boldsymbol{h}, P^{\mathcal{D}_l}_{XY}) - \mathcal{L}_E(\boldsymbol{h}, \mathcal{D}_l)| \leq C\sqrt{\frac{d_N(\mathcal{H}) \log K + \log \frac{1}{\delta}}{n_l}}, \tag{17}$$

where K is the number of different classes for the label.

As for the second term in (equation 16), it can be upper bounded via the Donsker-Varadhan variational representation of the relative entropy between two probability measures P and Q defined on $\mathcal{X}$:

$$D(P\|Q) = \sup_{g \in \mathcal{G}} \left\{ \mathbb{E}_P[g(X)] - \log \mathbb{E}_Q[e^{g(X)}] \right\}, \tag{18}$$

where the supremum is over all measurable functions $\mathcal{G} = \{g : \mathcal{X} \to \mathbb{R}, \text{ s.t. } \mathbb{E}_Q[e^{g(X)}] < \infty\}$. It then follows that for any $\lambda \in \mathbb{R}$,

$$D(P^{\mathcal{D}_l}_{XY} \| P^t_{XY}) \geq \mathbb{E}_{P^{\mathcal{D}_l}_{XY}}[\lambda \ell(\boldsymbol{h}(X), Y)] - \log \mathbb{E}_{P^t_{XY}}[e^{\lambda \ell(\boldsymbol{h}(X), Y)}]. \tag{19}$$

Since the loss function ℓ is bounded between $[0, 1]$, we can show that $\ell(\boldsymbol{h}(X), Y)$ is $\frac{1}{2}$-sub-Gaussian, i.e.,

$$\log \mathbb{E}_{P^t_{XY}}\left[e^{\lambda(\ell(\boldsymbol{h}(X), Y) - \mathbb{E}_{P^t_{XY}}[\ell(\boldsymbol{h}(X), Y)])}\right] \leq \frac{\lambda^2}{8}. \tag{20}$$

Thus, the following inequality holds for all $\lambda \in \mathbb{R}$,

$$\begin{aligned}
D(P^{\mathcal{D}_l}_{XY} \| P^t_{XY}) &\geq \mathbb{E}_{P^{\mathcal{D}_l}_{XY}}[\lambda \ell(\boldsymbol{h}(X), Y)] - \log \mathbb{E}_{P^t_{XY}}[e^{\lambda \ell(\boldsymbol{h}(X), Y)}] \\
&\geq \lambda \left(\mathbb{E}_{P^{\mathcal{D}_l}_{XY}}[\ell(\boldsymbol{h}(X), Y)] - \mathbb{E}_{P^t_{XY}}[\ell(\boldsymbol{h}(X), Y)] \right) - \frac{\lambda^2}{8} \\
&= \lambda \left(\mathcal{L}_P(\boldsymbol{h}, P^t_{XY}) - \mathcal{L}_P(\boldsymbol{h}, P^{\mathcal{D}_l}_{XY}) \right) - \frac{\lambda^2}{8},
\end{aligned} \tag{21}$$

which gives a non-negative parabola in λ, whose discriminant must be non-positive, which implies

$$\mathcal{L}_P(\boldsymbol{h}, P^t_{XY}) - \mathcal{L}_P(\boldsymbol{h}, P^{\mathcal{D}_l}_{XY}) \leq \sqrt{\frac{1}{2} D(P^{\mathcal{D}_l}_{XY} \| P^t_{XY})}. \tag{22}$$

Combining equation 22 with equation 18 completes the proof.

B ADDITIONAL EXPERIMENT SETTINGS

B.1 IMPLEMENTATION DETAILS

For a fair comparison, we follow the experiment settings in previous works [Ahmed et al., 2021, Liang et al., 2020]. For the Digits-Five benchmark, we use a variant of LeNet [LeCun et al., 1998] as our pretrained model. We resize the image samples from different digit domains to the same size (32 × 32) and convert the gray-scale images to RGB. For all office benchmarks and DomainNet, we use the pretrained ResNet-50 [He et al., 2016] as the backbone of the feature extractor, followed by a bottleneck layer with batch normalization and a classifier layer with weight normalization. We train the models on different source domain datasets and then retrain the pretrained models on the remaining single target domain dataset. The weights of the classifier are frozen during model training. The maximum number of training iterations is set to 20. For model optimization, we use SGD with momentum value 0.9 and weight decay 10^{-3}, the learning rate for backbone, bottleneck layer, and classifier layer is set to 10^{-2}, 10^{-2} and 10^{-3}, respectively. The hyper-parameter in the loss function equation 15 is set to be $\lambda_{\text{IM}} = 1$, $\lambda_{\text{adv}} = 1$. The confidence threshold α is set to be the mean of confidence score with a step decay strategy, more details can be found in appendix B.2. The batch size for Digit, Office and DomainNet datasets is set to 64, 32, and 32, respectively. All experiments are implemented in PyTorch using Titan RTX GPUs with 24 GB memory.

B.2 CONFIDENCE THRESHOLDING

The confidence thresholding aims to select correct-pseudo labeled data while filtering out the noisy data, but the optimal threshold value α is hard to find in practice due to lack of true labels of target domain data. Instead of manually tuning the value, we find a simple but effective way to set the value of α that can achieve good empirical performance. We set α based on the mean value of weighted confidence score defined in equation 11 with a step decay method, i.e., $\alpha^{(\tau)} = \beta \cdot \gamma^{\tau} \frac{1}{n_l} \sum_{\boldsymbol{x}_i^t \in \mathcal{D}_t} \sum_{j=1}^{m} \boldsymbol{w}_j \cdot p_k^{s_j}(\boldsymbol{x}_i^t)$, where τ is the iteration index. Default value of β are set to be 1.0, 0.5 and 0.9 for Digit-Five, Office, and DomainNet, respectively. Default value of γ are set to be 0.8, 0.8 and 1.0 for Digit-Five, Office, and DomainNet, respectively.