Unsupervised Multi-source Domain Adaptation Without Access to Source Data

Sk Miraj Ahmed[1,*], Dripta S. Raychaudhuri[1,*], Sujoy Paul[2,*,†], Samet Oymak[1], Amit K. Roy-Chowdhury[1]
[1] University of California, Riverside, [2] Google Research
{sahme047@, drayc001@, spaul003@, oymak@ece., amitrc@ece.}ucr.edu

Abstract

Unsupervised Domain Adaptation (UDA) aims to learn a predictor model for an unlabeled domain by transferring knowledge from a separate labeled source domain. However, most of these conventional UDA approaches make the strong assumption of having access to the source data during training, which may not be very practical due to privacy, security and storage concerns. A recent line of work addressed this problem and proposed an algorithm that transfers knowledge to the unlabeled target domain from a single source model without requiring access to the source data. However, for adaptation purposes, if there are multiple trained source models available to choose from, this method has to go through adapting each and every model individually, to check for the best source. Thus, we ask the question: can we find the optimal combination of source models, with no source data and without target labels, whose performance is no worse than the single best source? To answer this, we propose a novel and efficient algorithm which automatically combines the source models with suitable weights in such a way that it performs at least as good as the best source model. We provide intuitive theoretical insights to justify our claim. Furthermore, extensive experiments are conducted on several benchmark datasets to show the effectiveness of our algorithm, where in most cases, our method not only reaches best source accuracy but also outperforms it.

1. Introduction

Deep neural networks have achieved proficiency in a multiple array of vision tasks [11, 25, 18, 35], however, these models have consistently fallen short in adapting to visual distributional shifts [27]. Human recognition, on the other hand, is robust to such shifts, such as reading text in a new font or recognizing objects in unseen environments. Imparting such robustness towards distributional shifts to deep models is fundamental in applying these models to practical scenarios.

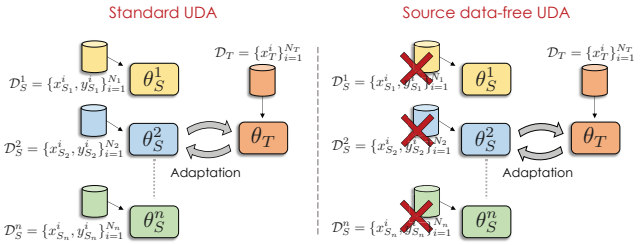

Figure 1. **Problem setup.** Standard unsupervised multi-source domain adaptation (UDA) utilizes the source data, along with the models trained on the source, to perform adaptation on a target domain. In contrast, we introduce a setting which adapts multiple models without requiring access to the source data.

Unsupervised domain adaptation (UDA) [2, 37] seeks to bridge this performance gap due to domain shift via adaptation of the model on small amounts of unsupervised data from the target domain. The majority of current approaches [7, 14] optimize a two-fold objective: (i) minimize the empirical risk on the source data, (ii) make the target and source features indistinguishable from each other. Minimizing distribution divergence between domains by matching the distribution statistical moments at different orders have also been explored extensively [42, 33].

A shortcoming of all the above approaches is the transductive scenario in which they operate, i.e., the source data is required for adaptation purposes. In a real-world setting, source data may not be available for a variety of reasons. Privacy and security are the primary concern, with the data possibly containing sensitive information. Another crucial reason is storage issues, i.e., source datasets may contain videos or high-resolution images and it might not be practical to transfer or store on different platforms. Consequently, it is imperative to develop unsupervised adaptation approaches which can adapt the source models to the target domain *without access to the source data*.

Recent works [21, 23] attempt this by adapting a single source model to a target domain without accessing the source data. However, an underlying assumption of these methods is that the most correlated source model is provided by an oracle for adaptation purposes. A more challenging and

*Equal Contribution
†Work done while SP was a PhD student at UC Riverside.

practical scenario entails adaptation from a *bag of source models* - each of these source domains are correlated to the target by different amounts and adaptation involves not only incorporating the combined prior knowledge from multiple models, but simultaneously preventing the possibility of negative transfer. In this paper, we introduce the problem of unsupervised *multi-source adaptation without access to source data*. We develop an algorithm based on the principles of pseudo-labeling and information maximization and provide intuitive theoretical insights to show that our framework guarantees performance better than the best available source and minimize the effect of negative transfer.

To solve this problem of multiple source model adaptation without accessing the source data, we deploy *Information Maximization (IM)* loss [23] on the weighted combination of target soft labels from all the source models. We also use the pseudo-label strategy inspired from deep cluster method [4], along with the IM loss to minimize noisy cluster assignment of the features. The overall optimization jointly adapts the feature encoders from sources as well as the corresponding source weights, combining which the target model is obtained.

Main Contributions. We address the problem of multiple source UDA, with no access to the source data. Towards solving the problem, we make the following contributions:

• We propose a novel UDA algorithm which operates without requiring access to the source data. We term it as Data frEe multi-sourCe unsupervISed domain adaptatiON (DECISION). Our algorithm automatically identifies the optimal blend of source models to generate the target model by optimizing a carefully designed unsupervised loss.

• Under intuitive assumptions, we establish theoretical guarantees on the performance of the target model which shows that it is consistently at least as good as deploying the single best source model, thus, minimizing negative transfer.

• We validate our claim by extensive numerical experiments, demonstrating the practical benefits of our approach.

2. Related works

In this section we present a brief overview of the literature in the area of unsupervised domain adaptation in both the single and multiple sources scenario, as well as the closely related setting of hypothesis transfer learning.

Unsupervised domain adaptation. UDA methods have been used for a variety of tasks, including image classification [44], semantic segmentation [32] and object detection [15]. Besides the feature space adaptation methods based on the paradigms of moment matching [42, 33] and adversarial learning [7, 44], recent works have explored pixel space adaptation via image translation [14]. All existing UDA methods require access to labeled source data, which may not be available in many applications.

Hypothesis transfer learning. Similar to our objective, hypothesis transfer learning (HTL) [40, 34, 1] aims to transfer learnt source hypotheses to a target domain without access to source data. However, data is assumed to be labeled in the target domain in contrast to our scenario, limiting its applicability to real-world settings. Recently, [21, 23] extend the standard HTL setting to unsupervised target data (U-HTL) by adapting single source hypotheses via pseudo-labeling. Our paper takes this one step further by introducing multiple source models, which may or may not be positively correlated with the target domain.

METHOD	MULTIPLE DOMAINS	NO SOURCE DATA	SOURCE MODEL	UNLABELED TARGET DATA
UDA [14]	✗	✗	✓	✓
MSDA [33]	✓	✗	✓	✓
HTL [40]	✗	✓	✓	✗
U-HTL [23]	✗	✓	✓	✓
DECISION(Ours)	✓	✓	✓	✓

Table 1. **Comparison to different adaptation settings by attributes demonstrated in the paper.** Our proposed setting satisfies all the criteria desired in a holistic adaptation framework.

Multi-source domain adaptation. Multi-source domain adaptation (MSDA) extends the standard UDA setting by incorporating knowledge from multiple source models. Latent space transformation methods [50] aim to align the features of different domains by optimizing a discrepancy measure or an adversarial loss. Discrepancy based methods seek to align the domains by minimizing measures such as maximum mean discrepancy [10, 50] and Rényi-divergence [13]. Adversarial methods aim to make features from multiple domains indistinguishable to a domain discriminator by optimizing GAN loss [47], $\mathcal{H}-$divergence [49] and Wasserstein distance [46, 22]. Domain generative methods [36, 24] use some form of domain translation, such as the CycleGAN [51], to perform adaptation at the pixel level. All these methods assume access to the source data during adaptation.

3. Methodology

Problem setting. We address the problem of jointly adapting multiple models, trained on a variety of domains, to a new target domain with access to only samples without annotations from the target. In this work, we will be considering the adaptation of classification models with K categories and the input space being $\mathcal{X}$. Formally, let us consider we have a set of source models $\{\theta_S^j\}_{j=1}^n$, where the j^{th} model $\theta_S^j : \mathcal{X} \to \mathbb{R}^K$, is a classification model learned using the source dataset $\mathcal{D}_S^j = \{x_{S_j}^i, y_{S_j}^i\}_{i=1}^{N_j}$, with N_j data points, where $x_{S_j}^i$ and $y_{S_j}^i$ denote the i-th source image and the corresponding label respectively. Now, given a target unlabeled dataset $\mathcal{D}_T = \{x_T^i\}_{i=1}^{N_T}$, the problem is to learn a classification model $\theta_T : \mathcal{X} \to \mathbb{R}^K$, using only the learned source

models, without any access to the source datasets. Note that this is different from multi-source domain adaptation methods in literature, which also utilize the source data while learning the target model θ_T.

Overall Framework. We can decompose each of the source models into two modules: the feature extractor $\phi_S^i : \mathcal{X} \to \mathbb{R}^{d_i}$ and the classifier $\psi_S^i : \mathbb{R}^{d_i} \to \mathbb{R}^K$. Here, d_i refers to the feature dimension of the i-th model while K refers to the number of categories. We aim to estimate the target model θ_T by combining knowledge only from the given source models in a manner that automatically rejects poor source models, i.e., those which are irrelevant for the target domain.

At the core of our framework lies a model aggregation scheme [28, 13], wherein we learn a set of weights $\{\alpha_i\}_{i=1}^n$ corresponding to each of the source models, such that, $\alpha_k \geq 0$ and $\sum_{k=1}^n \alpha_k = 1$. These weights represent a probability mass function over the source domains, with a higher value implying higher transferability from that particular domain, and are used to combine the source hypotheses accordingly. However, unlike previous works, we jointly adapt each individual model and simultaneously learn these weights by utilizing solely the unlabeled target instances. In what follows, we describe our training strategy used to achieve this in detail.

3.1. Weighted Information Maximization

As we do not have access to the labeled source or target data, we propose to fix the source classifiers,$\{\psi_S^i\}_{i=1}^n$, since it contains the class distribution information of the source domain and adapt solely the feature maps $\{\phi_S^i\}_{i=1}^n$ via the principle of information maximization [3, 19, 30, 23]. Our motivation behind the adaptation process stems from the *cluster assumption* [5] in semi-supervised learning, which hypothesizes that the discriminative model's decision boundaries should be located in regions of the input space which are not densely populated. To achieve this, we minimize a conditional entropy term (i.e., for a given input example) [9] as follows:

$$\mathcal{L}_{\text{ent}} = -\mathbb{E}_{x_T \in \mathcal{D}_T} \Big[\sum_{j=1}^{K} \delta_j(\theta_T(x_T)) \log(\delta_j(\theta_T(x_T))) \Big] \quad (1)$$

where $\theta_T(x_T) = \sum_{j=1}^n \alpha_j \theta_S^j(x_T)$, and $\delta(\cdot)$ denotes the softmax operation with $\delta_j(v) = \frac{\exp(v_j)}{\sum_{i=1}^K \exp(v_i)}$ for $v \in \mathbb{R}^K$. Intuitively, if a source θ_S^j has good transferability on the target and consequently, has smaller value of the conditional entropy, optimizing the term (1) over $\Big\{\theta_S^j, \alpha_j\Big\}$, will result in higher value of α_j than rest of the weights.

While entropy minimization effectively captures the cluster assumption when training with partial labels, in an unsupervised setting, it may lead to degenerate solutions, such as, always predicting a single class in an attempt to minimize conditional entropy. To control such degenerate solutions, we incorporate the idea of class diversity: configurations in which class labels are assigned evenly across the dataset are preferred. A simple way to encode our preference towards class balance is to maximize the entropy of the empirical label distribution [3] as follows,

$$\mathcal{L}_{\text{div}} = \sum_{j=1}^{K} -\bar{p}_j \log \bar{p}_j \quad (2)$$

where $\bar{p} = \mathbb{E}_{x_T \in \mathcal{D}_T}[\delta(\theta_T(x_T))]$. Combining the terms (1) and (2), we arrive at,

$$\mathcal{L}_{\text{IM}} = \mathcal{L}_{\text{div}} - \mathcal{L}_{\text{ent}} \quad (3)$$

which is the empirical estimate of the mutual information between the target data and the labels under the aggregate model θ_T. Although maximizing this loss makes the predictions on the target data more confident and globally diverse, it may sometime still fail to restrict erroneous label assignment. Inspired by [23], we propose a pseudo-labeling strategy in an effort to contain this mislabeling.

3.2. Weighted Pseudo-labeling

As a result of domain shift, information maximization may result in some instances being clubbed with the wrong class cluster. These wrong predictions get reinforced over the course of training and lead to a phenomenon termed as *confirmation bias* [43]. Aiming to contain this effect we adopt a self-supervised clustering strategy [23] inspired from the DeepCluster technique [4].

First, we calculate the cluster centroids induced by each source model for the whole target dataset as follows,

$$\mu_{k_j}^{(0)} = \frac{\sum_{x_T \in \mathcal{D}_T} \delta_k(\hat{\theta}_S^j(x_T)) \hat{\phi}_S^j(x_T)}{\sum_{x_T \in \mathcal{D}_T} \delta_k(\hat{\theta}_S^j(x_T))} \quad (4)$$

where the cluster centroid of class k obtained from source j at iteration i is denoted as $\mu_{k_j}^{(i)}$, and $\hat{\theta}_S^j = (\psi_S^j \circ \hat{\phi}_S^j)$ denotes the source from the previous iteration. These source-specific centroids are combined in accordance to the current aggregation weights on each source model as follows,

$$\mu_k^{(0)} = \sum_{j=1}^{n} \alpha_j \mu_{k_j}^{(0)} \quad (5)$$

Next, we compute the pseudo-label of each sample by assigning it to its nearest cluster centroid in the feature space,

$$\hat{y}_T^{(0)} = \arg\min_k \|\hat{\theta}_T(x_T) - \mu_k^{(0)}\|_2^2 \quad (6)$$

We reiterate this process to get the updated centroids and pseudo-labels as follows,

$$\mu_{k_j}^{(1)} = \frac{\sum_{x_T \in \mathcal{D}_T} \mathbb{1}\{\hat{y}_T^{(0)} = k\} \hat{\phi}_S^j(x_T)}{\sum_{x_T \in \mathcal{D}_T} \mathbb{1}(\hat{y^t}_0 = k)} \quad (7)$$

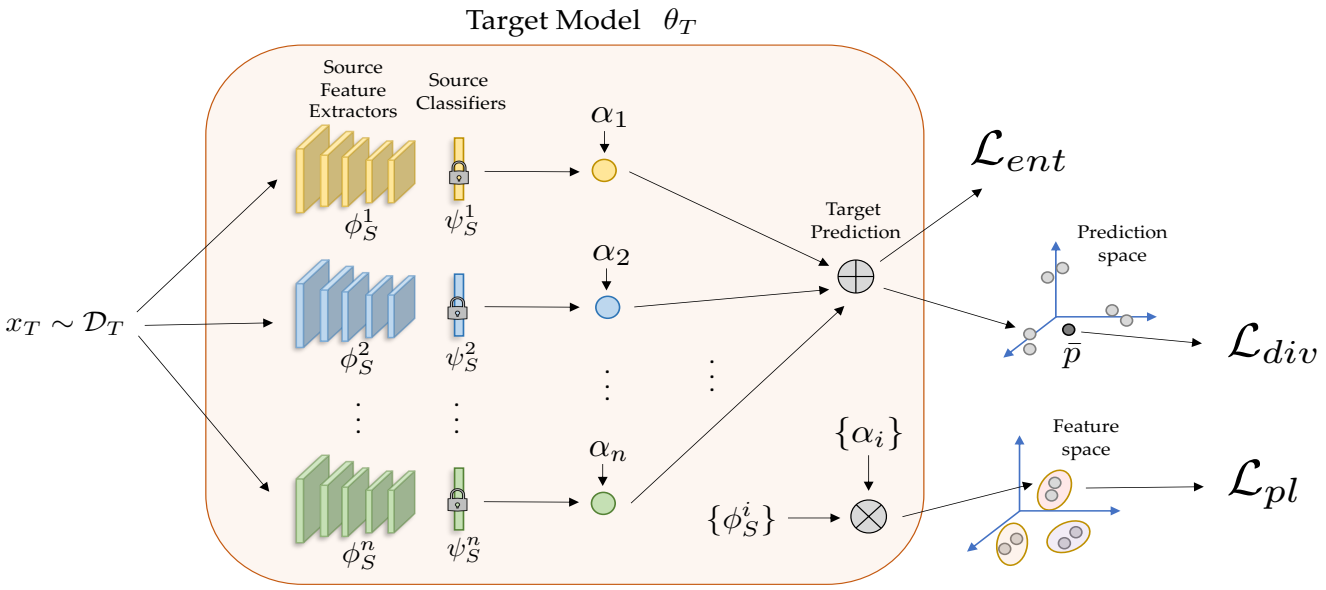

Figure 2. **Overall framework of our approach:** We freeze the final classification layers of all the sources and jointly optimize for the source feature encoders along with it's corresponding weights to get the target predictor by combining those.

$$\mu_k^{(1)} = \sum_{j=1}^{n} \alpha_j \mu_{k_j}^{(1)} \tag{8}$$

$$\hat{y}_T^{(1)} = \arg\min_k \|\hat{\theta}_T(x_T) - \mu_k^{(1)}\|_2^2 \tag{9}$$

where $\mathbb{1}(\cdot)$ is an indicator function which gives a value of 1 when the argument is true. While this alternating process of computing cluster centroids and pseudo-labeling can be repeated multiple times to get stationary pseudo-labels, one round is sufficient for all practical purposes. We then obtain the cross-entropy loss w.r.t. these pseudo-labels as follows:

$$\mathcal{L}_{\text{pl}}(Q_T, \theta_T) = -\mathbb{E}_{x_T \in \mathcal{D}_T} \sum_{k=1}^{K} \mathbb{1}\{\hat{y}_T = k\} \log \delta_k(\theta_T(x_T)). \tag{10}$$

Note that the pseudo-labels are updated regularly after a certain number of iterations as discussed in Section 5.

3.3. Optimization

In summary, given n source hypothesis $\{\theta_S^j\}_{j=1}^n = \{\psi_S^j \circ \phi_S^j\}_{j=1}^n$ and target data $\mathcal{D}_T = \{x_T^i\}_{i=1}^{n_T}$, we fix the classifier from each of the sources and optimize over the parameters of $\{\phi_S^j\}_{j=1}^n$ and the aggregation weights $\{\alpha_j\}_{j=1}^n$. The final objective is given by,

$$\mathcal{L}_{tot} = \mathcal{L}_{\text{ent}} - \mathcal{L}_{\text{div}} + \lambda \mathcal{L}_{\text{pl}} \tag{11}$$

The above objective is used to solve the following optimization problem,

$$\begin{aligned} \underset{\{\phi_S^j\}_{j=1}^n, \{\alpha_j\}_{j=1}^n}{\text{minimize}} \quad & \mathcal{L}_{tot} \\ \text{subject to} \quad & \alpha_j \geq 0, \forall j \in \{1, 2, \ldots, n\}, \\ & \sum_{j=1}^{n} \alpha_j = 1 \end{aligned} \tag{12}$$

Once we obtain the optimal set of ϕ_S^{j*} and α_j^*, the optimal target hypothesis is computed as $\theta_T = \sum_{j=1}^n \alpha_j^*(\psi_S^j \circ \phi_S^{j*})$. To solve the optimization (12) we follow the steps of Algorithm (1) stated below.

4. Theoretical Insights

Theoretical motivation behind our approach. Our algorithm aims to find the optimal weights $\{\alpha_j\}_{j=1}^n$ for each source and takes a convex combination of the source predictors to obtain the target predictor. Here, we shall show that under intuitive assumptions on the source and target distributions, there exists a simple choice of target predictor, which can perform better than or equal to the best source model being applied directly on the target data.

Formally, let L be a loss function which maps the pair of model-predicted label and the ground-truth label to a scalar. Denote the expected loss over k-th source distribution Q_S^k using the source predictor θ via $\mathcal{L}(Q_S^k, \theta) = \mathbb{E}_x[L(\theta(x), y)] = \int_x L(\theta(x), y) Q_S^k(x) dx$. Now let θ_S^k be the optimal source predictor given by $\theta_S^k = \arg\min_\theta \mathcal{L}(Q_S^k, \theta)\ \forall\ 1 \leq k \leq n$. Let us also assume

Algorithm 1: Algorithm to Solve Eq. 12

Input: Trained source models $\{\theta_S^j\}_{j=1}^n = \{\psi_S^j \circ \phi_S^j\}_{j=1}^n$, unlabeled target data $\{x_T^i\}_{i=1}^{N_T}$,weight parameters $\{\alpha_j\}_{j=1}^n$,max number of epochs E, regularization parameter λ,number of batches B

Output: Optimal feature enocoders $\{\phi_S^{j*}\}_{j=1}^n$, optimal source weights $\{\alpha_j^*\}_{j=1}^n$

Initialization: Freeze final classification layers $\{\psi_S^j\}_{j=1}^n$, set $\alpha_j = 1$ for all j

for $epoch = 1$ ***to*** E **do**
- Calculate pseudo-labels from equation (6)
- Calculate the mean embedding $\bar{p}$ from equation (2)
- **for** $iteration = 1$ ***to*** B **do**
  - Sample a mini batch from target and pass it through each of the source models
  - calculate all the losses from equation (1),(2) and (10)
  - calculate total loss from equation (3)
  - Update the parameters in $\{\phi_S^j\}_{j=1}^n$ and $\{\alpha_j\}_{j=1}^n$ from optimization(12)
  - Make α positive by setting $\alpha_j = 1/(1+e^{-\alpha_j})$
  - Normalize α by setting $\alpha_j = \alpha_j / \sum_{i=1}^n \alpha_i$
- **end**

end

that the target distribution is in the span of source distributions. We formalize this by expressing the target distribution as an affine combination of source distributions i.e., $Q_T(x) = \sum_{k=1}^n \lambda_k Q_S^k(x) : \lambda_k \geq 0, \sum_{k=1}^n \lambda_k = 1$. Under this assumption, if we express our target predictor as $\theta_T(x) = \sum_{k=1}^n \frac{\lambda_k Q_S^k(x)}{\sum_{j=1}^n \lambda_j Q_S^j(x)} \theta_S^k(x)$, then we establish our theoretical claim stated in Lemma 1.

Lemma 1. *Assume that the loss $L(\theta(x), y)$ is convex in its first argument and that there exists a $\lambda \in \mathbb{R}^n$ where $\lambda \geq 0$ and $\lambda^\top \mathbb{1} = 1$, such that the target distribution is exactly equal to the mixture of source distributions, i.e., $Q_T = \sum_{i=1}^n \lambda_i Q_S^i$. Set the target predictor as the following convex combination of the optimal source predictors*

$$\theta_T(x) = \sum_{k=1}^n \frac{\lambda_k Q_S^k(x)}{\sum_{j=1}^n \lambda_j Q_S^j(x))} \theta_S^k(x).$$

Recall the pseudo-labeling loss (10). Then, for this target predictor, over the target distribution, the unsupervised loss induced by the pseudo-labels and the supervised loss are both less than or equal to the loss induced by the best source predictor. In particular,

$$\mathcal{L}(Q_T, \theta_T) \leq \min_{1 \leq j \leq n} \mathcal{L}(Q_T, \theta_S^j).$$

Let $\alpha = \arg\min_{1 \leq j \leq n} \mathcal{L}(Q_T, \theta_S^j)$. Additionally, this inequality is strict if the entries of λ are strictly positive and there exists a source i for which the strict inequality $\mathcal{L}(Q_S^i, \theta_S^i) < \mathcal{L}(Q_S^i, \theta_S^\alpha)$ holds.

Proof. See proof in the supplementary. □

Observe that the expected loss $\mathcal{L}$ defined in Lemma 1 is the supervised loss where one does have the label information. Our proposed target predictor θ_T achieves a supervised loss at least as good as the best individual source model. Importantly, the inequality is strict under a natural mild condition: The best individual source model β (for the target Q_T) is strictly worse than some source model i on the source distribution Q_S^i. We also note the key differences between our algorithm and the predictor in Lemma 1. In our algorithm's combination rule, we fine-tune the feature extractors of each source model unlike Lemma 1. However each source has an individual weight which is agnostic to the source data, whereas Lemma 1 uses different weights per input instance. Below we provide an intuitive justification for choosing this input-agnostic weighting strategy.

Since we do not know the source distributions (due to the unavailability of source data), let us consider the least informative of all the distributions i.e. uniform distribution for sources by the *Principle of Maximum Entropy* [17]. This uniformity is assumed over the target support set $\mathcal{X}$. In what follows, we will consider the restrictions of the source distributions to the target support $\mathcal{X}$. Mathematically, our assumption is $Q_S^k(x) = c_k \mathcal{U}(x)$ when restricted to the support set $x \in \mathcal{X}$, where c_k is a scaling factor which captures the relative contribution of source k and $\mathcal{U}(x)$ has value 1. If we plug this value of the distribution in the combination rule in Lemma 1, we get $\theta_T(x) = \sum_{k=1}^n \frac{\lambda_k c_k}{\sum_{j=1}^n \lambda_j c_j} \theta_S^k(x)$ (see supplementary for more details). This term consisting of λ_k and c_k essentially becomes the weighting term α_k in our algorithm. We put this value of θ_T to solve the optimization (12) jointly with respect to this α_k and ϕ_S^k. Thus, our optimization will return us a favorable combination of source hypotheses, satisfying the bounds in Lemma 1, under the uniformity assumption of source distributions.

5. Experiments

Datasets. To test the effectiveness of our algorithm, we experiment on various visual benchmarks described as follows.
• *Office* [14]: In this benchmark DA dataset there are three domains under the office environment namely Amazon (A), DSLR (D) and Webcam (W) with a total of 31 object classes in each domain.

SOURCE	METHOD	MT,UP,SV,SY → MM	MM,UP,SV,SY → MT	MM,MT,SV,SY → UP	MM,MT,UP,SY → SV	MM,MT,UP,SV → SY	AVG.
Multiple(w)	DAN[26]	63.7	96.31	94.2	62.5	85.4	80.4
	DANN[6]	71.3	97.6	92.3	63.5	85.3	82.0
	MCD[38]	72.5	96.21	95.3	78.9	87.5	86.1
	CORAL[41]	62.5	97.2	93.4	64.4	82.7	80.1
	ADDA[44]	71.6	97.9	92.8	75.5	86.5	84.8
	M^3SDA-β[33]	72.8	98.4	96.1	81.3	89.6	87.6
Single(w/o)	Source-best	60.7	98.2	74.5	89.5	89.4	82.5
	Source-worst	21.3	64	29.3	7.4	25.7	29.5
	SHOT[23]-best	**94.0**	98.7	**97.9**	**83.5**	**97.5**	**94.3**
	SHOT[23]-worst	44.5	97.2	96.2	29.5	32.5	60.0
Multiple(w/o)	SHOT[23]-Ens	90.4	98.9	97.7	58.3	83.9	85.8
	DECISION(Ours)	93.0	**99.2**	97.8	82.6	**97.5**	94.0

Table 2. **Results on digit recognition.** MT, MM, UP, SV, SY are abbreviations of *MNIST, MNIST-M, USPS, SVHN* and *Synthetic Digits* respectively. Multiple and Single denotes the methods which uses multiple and single sources respectively for domain adaptation, while (w) and (w/o) are abbreviations of *with source data* and *without source data* respectively. *Source* is the accuracy with the unadapted models, whereas *-best* and *-worst* refer to the best and worst sources.

SOURCE	METHOD	A,D → W	A,W → D	D,W → A	AVG.
Single	Source-best	96.3	98.4	62.5	85.7
	Source-worst	75.6	80.9	62.0	72.8
	SHOT [23]-best	98.2	99.6	75.1	90.9
	SHOT [23]-worst	90.6	94.2	72.9	85.9
Multiple	SHOT [23]-Ens	94.9	97.8	75.0	89.3
	DECISION(Ours)	**98.4**	**99.6**	**75.4**	**91.1**

Table 3. **Results on Office**: A,D and W are abbreviations of *Amazon*, *DSLR* and *Webcam*. For single source methods, Source-best and Source-worst denote the best and worst unadapted source models, whereas SHOT-best, SHOT-worst are the best and worst accuracies of adapted source models.

• *Office-Caltech* [8]: This is an extension of the Office dataset, with Caltech-256 (C) added on top of the 3 existing domains by extracting 10 classes common to all domains.
• *Office-Home* [45]: Office-Home consists of four domains, namely, Art(Ar), Clipart(Cl), Product(Pr) and Real-world(Re). Each of these domains contain 65 object classes.
• *Digits*: The Digits dataset is a benchmark for DA in digit recognition. Following [33], we utilize five subsets, namely MNIST (MT), USPS (UP), SVHN (SV), MNIST-M (MM) and Synthetic Digits (SY) for our experiments.

In all of our experiments, we take turns and fix one of the domains as the target and the rest as the source domains. The source data is discarded after training the source models.

Baseline Methods. We compare our method against a wide array of baselines. Similar to our setting, SHOT [23] attempts unsupervised adaptation without source data. However, it adapts a single source at a time. We compare against a multi-source extension of SHOT via ensembling - we pass the target data through each of the adapted source model and take an average of the soft prediction to obtain the test label. In our comparisons, we name this method SHOT-ens. We also compare against single source baselines, namely SHOT-best and SHOT-worst, which refer to the best adapted source model and the worst one respectively, learned using SHOT. Additionally, we run comparisons against traditional multi-source adaptation methods M^3SDA-β[33], DAN [26], DANN [6], MCD [38], CORAL [41], ADDA [44], DCTN[47]. All these methods, except for SHOT, have access to source data during adaptation.

5.1. Implementation details

Network architecture. For the object recognition tasks, we use a pre-trained ResNet-50 [11] as the feature extractor backbone, similar to [33, 48]. Following [23, 6], we replace the penultimate fully-connected layer with a bottleneck layer and a task specific classifier layer. Batch normalization [16] is utilized after the bottleneck layer, along with weight normalization [39] in the final layer. For the digit recognition task, we use a variant of the LeNet [20] similar to [23].
Source model training. Following [23], we train the source models using smooth labels, instead of the usual one-hot encoded labels. This increases the robustness of the model and helps in the adaptation process by encouraging features to lie in tight, evenly separated clusters [29]. The maximum number of epochs for Digits, Office, Office-Home and Office-Caltech is set to 30, 100, 50 and 100, respectively. Additionally, for our experiments on digit recognition, we resize images from each domain to 32×32 and convert the

SOURCE	METHOD	AR,CL,PR → RW	AR,CL,RW → PR	AR,PR,RW → CL	CL,PR,RW → AR	AVG.
Single(w/o)	Source-best	74.1	78.3	46.2	65.8	66.1
	Source-worst	64.8	62.8	40.9	53.3	55.5
	SHOT[23]-best	81.3	83.4	57.2	72.1	73.5
	SHOT[23]-worst	80.8	77.9	53.8	66.6	69.8
Multiple(w/o)	SHOT[23]-Ens	82.9	82.8	59.3	72.2	74.3
	DECISION(Ours)	**83.6**	**84.4**	**59.4**	**74.5**	**75.5**

Table 4. **Results on Office-Home.**: AR,CL,RW and PR are abbreviations of *Art*, *Clipart*,*Real-world* and *Product*. We see that our method outperforms all the baselines including the best source accuracy as well as ensemble method. The abbreviations under the column SOURCE and METHOD are same as described in Table 2.

SOURCE	METHOD	A,C,D → W	A,C,W → D	C,D,W → A	A,D,W → C	AVG.
Multiple(w)	ResNet-101[11]	99.1	98.2	88.7	85.4	92.9
	DAN[26]	99.5	99.1	91.6	89.2	94.8
	DCTN[47]	99.4	99.0	92.7	90.2	95.3
	MCD[38]	99.5	99.1	92.1	91.5	95.6
	M^3SDA-β[33]	99.5	99.2	94.5	99.2	96.4
Single(w/o)	Source-best	98.9	99.3	94.8	86.5	94.9
	Source-worst	86.7	89.8	89.6	83.2	87.4
	SHOT-best	99.6	100	95.8	95.5	97.7
	SHOT-worst	97.3	96.2	95.7	93.9	95.8
Multiple(w/o)	SHOT-Ens	99.6	96.8	95.7	95.8	97.0
	DECISION(Ours)	**99.6**	**100**	**95.9**	**95.9**	**98.0**

Table 5. **Results on Office-Caltech Dataset**:A,D,C and W are abbreviations of *Amazon*, *DSLR*, *Caltech-256* and *Webcam*. Our method consistently outperform all the baselines across all the domains as target.The abbreviations under the column SOURCE and METHOD are same as described in Table 2.

gray-scale images to RGB.

Hyper-parameters. The entire framework is trained in an end-to-end fashion via back-propagation. Specifically, we utilize stochastic gradient descent with momentum value 0.9 and weight decay equalling 10^{-3}. The learning rate is set at 10^{-2} for the bottleneck and classifier layers, while the backbone is trained at a rate of 10^{-3}. In addition, we use the learning rate scheduling strategy from [6], where the initial rate is exponentially decayed as learning progresses. The batch size is set to 32. We use $\lambda = 0.3$ for all the object recognition tasks and $\lambda = 0.1$ for the digits benchmark. For adaptation, maximum number of epochs is set to 15, with the pseudo-labels updated at the start of every epoch. We use PyTorch [31] for all our experiments.

5.2. Digit recognition

The results on digit recognition are shown in Table 2. The digit benchmark is characterised by the presence of very poor sources in some scenarios, notably when treating MNIST-M, SVHN or Synthetic Digits as the target domain. For example, on SVHN as the target, the best and worst source models adapted using SHOT [23] exhibit a performance gap of more than 50%. Combining these models via uniform ensembling results in a predictor which greatly underperforms the best adapted source. In contrast, our method restricts this severe negative transfer via a joint adaptation over the models and the ensembling weights, and outperforms the baseline by $\mathbf{24.3}\%$. The corresponding increase in performance when using Synthetic Digits and MNISTM as the target are $\mathbf{13.5}\%$ and $\mathbf{2.6}\%$ respectively. Overall, we obtain an average increase of $\mathbf{8.2}\%$ across all the digit adaptation tasks over SHOT-Ens. In spite of such disparities among the sources, our framework also achieves performance at par with the best adapted source and actually outperforms the latter on the MNIST transfer task. We also outperform the traditional multi-source adaptation methods, which use source data, on all the tasks by an average of $\mathbf{6.4}\%$.

5.3. Object recognition

Office. The results for the 3 adaptation tasks on the Office dataset are shown in Table 3. We achieve performance at par with the best adapted source models on all the tasks

and obtain an average increase of $\mathbf{5.2}\%$ over SHOT-Ens. In the task of adapting to the Webcam (W) domain, negative transfer from the Amazon (A) model brings the ensemble performance down - our model is able to prevent this, and not only outperforms the ensemble by $\mathbf{3.5}\%$ but also achieves higher performance than the best adapted source.

Office-Home. On the Office-Home dataset, we outperform all baselines as shown in Table 4. Across all tasks, our method achieves a mean increase in accuracy of $\mathbf{2}\%$ over the respective best adapted source models. This can be attributed to the relatively small performance gap between the best and worst adapted sources in comparison to other datasets. This suggests that, as the performance gap between the best and worst performing sources gets smaller, or outlier sources are removed, our method can generalize even better to the target.

Office-Caltech. The results follow a similar trend on the Office-Caltech dataset, as shown in Table 5. With a mean accuracy of $\mathbf{98}\%$ across all tasks, we outperform all baselines.

5.4. Ablation study

Contribution of each loss. Our framework is trained using a combination of three distinct losses: $\mathcal{L}_{\text{div}}$, $\mathcal{L}_{\text{ent}}$ and $\mathcal{L}_{\text{pl}}$. We study the contribution of each component of our framework to the adaptation task in Table 6. First, we remove both the diversity loss and the pseudo-labeling, and train using only $\mathcal{L}_{\text{ent}}$. Next, we add in $\mathcal{L}_{\text{div}}$ and perform weighted information maximization. Finally, we also compare the results of solely using $\mathcal{L}_{\text{pl}}$.

METHOD	A,D → W	A,W → D	D,W → A	AVG.
$\mathcal{L}_{pl}$	97.6	98.5	75.3	90.5
$-\mathcal{L}_{ent}$	96.6	99.0	68.5	88.0
$-\mathcal{L}_{ent} + \mathcal{L}_{div}$	95.9	99.0	71.6	88.9
$-\mathcal{L}_{ent} + \mathcal{L}_{div} + \lambda\mathcal{L}_{pl}$	**98.4**	**99.6**	**75.4**	**91.1**

Table 6. **Loss-wise ablation.** Contribution of each component in adaptation on the Office dataset.

Analysis on the learned weights. Our framework jointly adapts the the source models and learns the weights on each such source. To understand the impact of the weights, we propose to freeze the feature extractors and optimize solely over the weights $\{\alpha_j\}_{i=1}^{n}$. Naturally, this setup yields better performance compared to trivially assigning equal weights to all source models, as shown in Table 7. More interestingly, the learned weights correctly indicate which source model performs better on the target and could serve as a proxy indicator in a model selection framework. See Figure 3.

Distillation into a single model. Since we are dealing with multiple source models, inference time is of the order $\mathcal{O}(m)$,

METHOD	AR,CL,PR → RW	AR,CL,RW → PR	AR,PR,RW → CL	CL,PR,RW → AR	AVG.
Source-Ens	67.6	51.4	77.7	80.1	69.2
DECISION-weights	**68.8**	**52.3**	**79.2**	**80.4**	**70.2**

Table 7. **Performance on freezing backbone network on Office-Home.** DECISION-weight is optimized solely over the source weights and consistently performs better than uniform weighting.

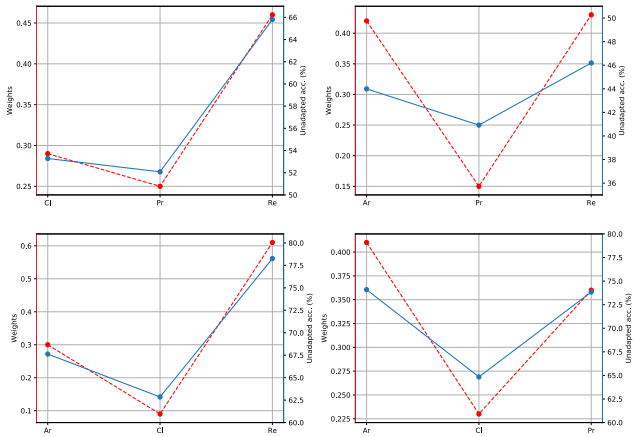

Figure 3. **Weights as model selection proxy.** The weights learnt by our framework on Office-Home correlates positively with the unadapted source model performance. (Left axis corresponds to the red plot and right to the blue plot, best viewed in color.)

where m is the number of source models. If m is large, this can lead to inference being quite time consuming. To ameliorate this overhead, we follow a knowledge distillation [12] strategy to obtain a single target model. Teacher supervision is obtained by linearly combining the adapted models via the learned weights. These annotations are subsequently used to train the single student model via vanilla cross-entropy loss. Results obtained using this strategy are presented in the supplementary.

6. Conclusions and Future Work

We developed a new UDA algorithm that can learn from and optimally combine multiple source models without requiring source data. We provide theoretical intuitions for our algorithm and verify its effectiveness in a variety of domain adaptation benchmarks. There are multiple exciting directions to pursue including: First, we suspect that our algorithm's performance can be further boosted by incorporating data augmentation techniques during training. Second, when there are too many source models to utilize, it would be interesting to study whether we can automatically select an optimal subset of the source models without requiring source data in an unsupervised fashion.

Acknowledgements. This work was partially supported by the ONR grant N00014-19-1-2264 and the NSF grants 2008020, 2046816.

References

[1] Sk Miraj Ahmed, Aske R Lejbolle, Rameswar Panda, and Amit K Roy-Chowdhury. Camera on-boarding for person re-identification using hypothesis transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12144–12153, 2020. 2

[2] 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(1-2):151–175, 2010. 1

[3] John S Bridle, Anthony JR Heading, and David JC MacKay. Unsupervised classifiers, mutual information and'phantom targets. In *Advances in neural information processing systems*, pages 1096–1101, 1992. 3

[4] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018. 2, 3

[5] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009. 3

[6] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015. 6, 7

[7] 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. 1, 2

[8] 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. 6

[9] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536, 2005. 3

[10] Jiang Guo, Darsh J Shah, and Regina Barzilay. Multi-source domain adaptation with mixture of experts. *arXiv preprint arXiv:1809.02256*, 2018. 2

[11] 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 (CVPR)*, pages 770–778, 2016. 1, 6, 7

[12] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 8

[13] Judy Hoffman, Mehryar Mohri, and Ningshan Zhang. Algorithms and theory for multiple-source adaptation. In *Advances in Neural Information Processing Systems*, pages 8246–8256, 2018. 2, 3

[14] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pages 1989–1998. PMLR, 2018. 1, 2, 5

[15] Han-Kai Hsu, Chun-Han Yao, Yi-Hsuan Tsai, Wei-Chih Hung, Hung-Yu Tseng, Maneesh Singh, and Ming-Hsuan Yang. Progressive domain adaptation for object detection. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 749–757, 2020. 2

[16] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 6

[17] Edwin T Jaynes. Information theory and statistical mechanics. *Physical review*, 106(4):620, 1957. 5

[18] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9404–9413, 2019. 1

[19] Andreas Krause, Pietro Perona, and Ryan G Gomes. Discriminative clustering by regularized information maximization. In *Advances in neural information processing systems*, pages 775–783, 2010. 3

[20] 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. 6

[21] 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 Conference on Computer Vision and Pattern Recognition*, pages 9641–9650, 2020. 1, 2

[22] Yitong Li, David E Carlson, et al. Extracting relationships by multi-domain matching. In *Advances in Neural Information Processing Systems*, pages 6798–6809, 2018. 2

[23] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. *arXiv preprint arXiv:2002.08546*, 2020. 1, 2, 3, 6, 7

[24] Chuang Lin, Sicheng Zhao, Lei Meng, and Tat-Seng Chua. Multi-source domain adaptation for visual sentiment classification. In *AAAI*, pages 2661–2668, 2020. 2

[25] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015. 1

[26] 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. 6, 7

[27] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2507–2516, 2019. 1

[28] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation with multiple sources. In *Advances in Neural Information Processing Systems*, pages 1041–1048, 2009. 3

[29] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? In *Advances in Neural Information Processing Systems*, pages 4694–4703, 2019. 6

[30] Samet Oymak and Talha Cihad Gulcu. Statistical and algorithmic insights for semi-supervised learning with self-training. *arXiv preprint arXiv:2006.11006*, 2020. 3

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019. 7

[32] Sujoy Paul, Yi-Hsuan Tsai, Samuel Schulter, Amit K Roy-Chowdhury, and Manmohan Chandraker. Domain adaptive semantic segmentation using weak labels. *arXiv preprint arXiv:2007.15176*, 2020. 2

[33] 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 International Conference on Computer Vision*, pages 1406–1415, 2019. 1, 2, 6, 7

[34] Michaël Perrot and Amaury Habrard. A theoretical analysis of metric hypothesis transfer learning. In *International Conference on Machine Learning*, pages 1708–1717, 2015. 2

[35] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016. 1

[36] Paolo Russo, Tatiana Tommasi, and Barbara Caputo. Towards multi-source adaptive semantic segmentation. In *International Conference on Image Analysis and Processing*, pages 292–301. Springer, 2019. 2

[37] 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. 1

[38] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3723–3732, 2018. 6, 7

[39] Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in neural information processing systems*, pages 901–909, 2016. 6

[40] Shashank Singh, Ananya Uppal, Boyue Li, Chun-Liang Li, Manzil Zaheer, and Barnabás Póczos. Nonparametric density estimation under adversarial losses. In *Advances in Neural Information Processing Systems*, pages 10225–10236, 2018. 2

[41] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. *arXiv preprint arXiv:1511.05547*, 2015. 6

[42] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2058–2065, 2016. 1, 2

[43] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017. 3

[44] 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. 2, 6

[45] 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. 6

[46] 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. 2

[47] 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. 2, 6, 7

[48] Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1426–1435, 2019. 6

[49] Han Zhao, Shanghang Zhang, Guanhang Wu, José MF Moura, Joao P Costeira, and Geoffrey J Gordon. Adversarial multiple source domain adaptation. In *Advances in neural information processing systems*, pages 8559–8570, 2018. 2

[50] Sicheng Zhao, Bo Li, Pengfei Xu, and Kurt Keutzer. Multi-source domain adaptation in the deep learning era: A systematic survey. *arXiv preprint arXiv:2002.12169*, 2020. 2

[51] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 2

Unsupervised Multi-source Domain Adaptation Without Access to Source Data

(Supplementary Material)

7. Proof of Lemma 1

Lemma 2. *Assume that the loss $L(\theta(x), y)$ is convex in its first argument and that there exists a $\lambda \in \mathbb{R}^n$ where $\lambda \geq 0$ and $\lambda^\top \mathbb{1} = 1$, such that the target distribution is exactly equal to the mixture of source distributions, i.e $Q_T = \sum_{i=1}^n \lambda_i Q_S^i$. Set the target predictor as the following convex combination of the optimal source predictors*

$$\theta_T(x) = \sum_{k=1}^{n} \frac{\lambda_k Q_S^k(x)}{\sum_{j=1}^n \lambda_j Q_S^j(x))} \theta_S^k(x).$$

Recall the pseudo-labeling loss (10). Then, for this target predictor, over the target distribution, the unsupervised loss induced by the pseudo-labels and the supervised loss are both less than or equal to the loss induced by the best source predictor. In particular,

$$\mathcal{L}(Q_T, \theta_T) \leq \min_{1 \leq j \leq n} \mathcal{L}(Q_T, \theta_S^j).$$

Proof. We can see that the left hand-side of the inequality can be upper-bounded by some loss as follows,

$$\begin{aligned}
\mathcal{L}(Q_T, \theta_T) = \int_x Q_T(x) L(\theta_T(x), y) &= \int_x Q_T(x) L\left(\sum_{i=1}^{n} \frac{\lambda_i Q_S^i(x)}{\sum_{j=1}^n \lambda_j Q_S^j(x))} \theta_S^i(x), y \right) dx \\
&\leq \int_x Q_T(x) \sum_{i=1}^{n} \frac{\lambda_i Q_S^i(x)}{\sum_{j=1}^n \lambda_j Q_S^j(x))} L(\theta_S^i(x), y) dx \quad \text{(from Jensen's inequality)} \\
&= \int_x Q_T(x) \sum_{i=1}^{n} \frac{\lambda_i Q_S^i(x)}{Q_T(x)} L(\theta_S^i(x), y) dx \quad \text{(from distribution assumption)} \\
&= \sum_{i=1}^{n} \lambda_i \int_x Q_S^i(x) L(\theta_S^i(x), y) dx \quad \text{(changing the order of summation)} \\
&= \sum_i \lambda_i \mathcal{L}(Q_S^i(x), \theta_S^i)
\end{aligned} \tag{13}$$

Now for the R.H.S. we can write this loss as follows,

$$\begin{aligned}
\mathcal{L}(Q_T, \theta_S^j) &= \int_x Q_T(x) L(\theta_S^j(x), y) dx \\
&= \int_x \sum_{i=1}^{n} \lambda_i Q_S^i(x) L(\theta_S^j(x), y) dx \\
&= \sum_{i=1}^{n} \lambda_i \int_x Q_S^i L(\theta_S^j(x), y) dx \\
&= \sum_{i=1}^{n} \lambda_i \mathcal{L}(Q_S^i(x), \theta_S^j)
\end{aligned} \tag{14}$$

Now recall from main paper that,

$$\theta_S^k = \arg\min_\theta \mathcal{L}(Q_S^k, \theta) \quad \text{for} \quad 1 \leq k \leq n.$$

.

This means θ_S^i is the best predictor for the source i, which has distribution Q_S^i. Thus we find that $\mathcal{L}(Q_S^i, \theta_S^i) \leq \mathcal{L}(Q_S^i, \theta_S^j)\ \forall j$, which implies $\sum_i \lambda_i \mathcal{L}(Q_S^i, \theta_S^i) \leq \sum_i \lambda_i \mathcal{L}(Q_S^i, \theta_S^j)$. This further implies that $\mathcal{L}(Q_T, \theta_T) \leq \mathcal{L}(Q_T, \theta_S^j)\ \forall j$, which in turn concludes the proof $\mathcal{L}(Q_T, \theta_T) \leq \min_{1 \leq j \leq n} \mathcal{L}(Q_T, \theta_S^j)$. Finally, suppose the entries of λ are strictly positive and let $\beta = \arg\min_j \mathcal{L}(Q_T, \theta_S^j)$. Observe that, if there is a source i such that the strict inequality $\mathcal{L}(Q_S^i, \theta_S^i) < \mathcal{L}(Q_S^i, \theta_S^\beta)$ holds, then the main claim of the lemma also becomes strict as we find

$$\mathcal{L}(Q_T, \theta_T) \leq \sum_i \lambda_i \mathcal{L}(Q_S^i, \theta_S^i) < \sum_i \lambda_i \mathcal{L}(Q_S^i, \theta_S^\beta) \leq \min_j \mathcal{L}(Q_T, \theta_S^j).$$

Verbally, this strict inequality has a natural meaning that the model j is strictly worse than model i for the source data i. □

8. Detailed steps of combination rule under source distribution uniformity assumption

See the discussion after **Lemma 1** in the main paper for reference.

$$\begin{aligned}\theta_T(x) &= \sum_{k=1}^{n} \frac{\lambda_k Q_S^k(x)}{\sum_{j=1}^{n} \lambda_j Q_S^j(x)} \theta_S^k(x) \\ &= \sum_{k=1}^{n} \frac{\lambda_k c_k \mathcal{U}(x)}{\sum_{j=1}^{n} \lambda_j c_j \mathcal{U}(x)} \theta_S^k(x) \\ &= \sum_{k=1}^{n} \frac{\lambda_k c_k}{\sum_{j=1}^{n} \lambda_j c_j} \theta_S^k(x)\end{aligned} \tag{15}$$

9. Additional Experiments

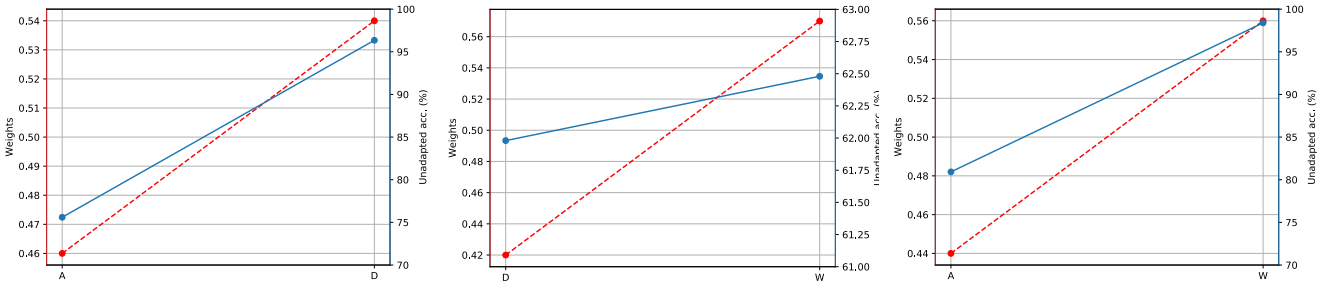

Figure 4. **Weights as model selection proxy.** The weights learnt by our framework on Office-31 correlates positively with the unadapted source model performance. (Left axis corresponds to the red plot and right to the blue plot, best viewed in color.)

From Figure 4, we can clearly see that for the model which gives higher accuracy for the unadapted scenario, it is automatically given higher weightage by our algorithm. As a result, we can easily infer about the quality of the source domain, in relation to the target, from the weights learnt by our framework.

Effect of weight on pseudo-labeling. We investigate the effect of the weight λ on $\mathcal{L}_{\text{pl}}$. We perform experiments on the Office dataset by varying the value of λ and plot the results in Figure 5. As shown in the plot, the proposed method performs best at $\lambda = 0.3$

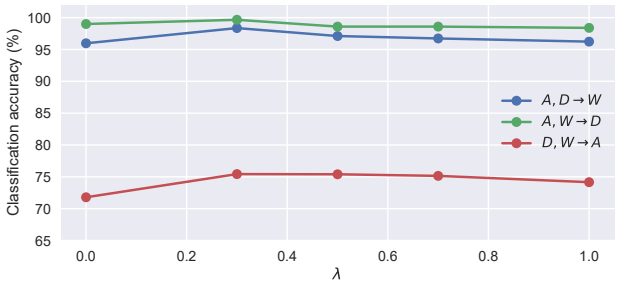

Figure 5. **Effect of λ.** The variations in classification as the weight on $\mathcal{L}_{\text{pl}}$ is varied. (Best viewed in color)

Effect of outlier source models. Our method is clearly robust to outlier source models. In Table 2 of the main paper, when *MNIST-M* is the target, transferring from only *USPS*, leads to an extremely poor performance of **21.3**% - here, USPS is a strong outlier. Despite the presence of such a poor source, our framework is mostly able to correctly negate the transfer from USPS, achieving a performance of **93**%, close to the best source performance of **94**%. On removing *USPS* as a source, DECISION outperforms the best source by achieving an accuracy of **94.5**%. In the future, we plan to actively use the weights to simultaneously remove poor sources while adaptation in order to boost the performance.

SOURCE	METHOD	C,P,I,S,R → Q	Q,P,I,S,R → C	Q,C,I,S,R → P	Q,C,P,S,R → I	Q,C,P,I,R → S	Q,C,P,I,S → R	AVG.
Multiple(w)	DAN[25]	16.2	39.1	33.3	11.4	29.7	42.1	28.6
	DCTN[46]	7.2	48.6	48.8	23.4	47.3	53.5	38.1
	MCD[37]	7.6	54.3	45.7	22.1	43.5	58.4	38.6
	M^3SDA-β[32]	6.3	58.6	52.3	26	49.5	62.7	42.5
Single(w/o)	Source-best	11.9	49.9	47.5	20	41.1	57.7	38
	Source-worst	2.3	12.2	2.2	1.1	8.7	4.8	5.2
	SHOT[22]-best	18.7	58.3	53	22.7	48.4	65.9	44.5
	SHOT[22]-worst	3.8	14.8	3.5	1	11.9	6.6	7
Multiple(w/o)	SHOT[22]-Ens	15.3	58.6	**55.3**	**25.2**	**52.4**	**70.5**	**46.2**
	DECISION(Ours)	**18.9**	**61.5**	54.6	21.6	51	67.5	45.9

Table 8. **Results on DomainNet**:Q,C,P,I,S and R are abbreviations of *quickdraw*, *clipart*, *painting*, *infograph*, *sketch* and *real*.

DomainNet [32]: This is a relatively new and large dataset where there are six domains under the common object categories, namely quickdraw (Q), clipart (C), painting (P), infograph (I), sketch (S) and real (R) with a total of 345 object classes in each domain. Experimental results on this dataset are shown in Table 8. Our method consistently outperforms the best adapted source baselines (SHOT-best) except for *infograph* as a target. However the average performance over all the domains as target is slightly less than the SHOT-Ens. Note that for *quickdraw* and *clipart* as target, our method outperforms all the state of the art methods including source free and with source data single and multi source state-of-the-art DA methods.

Distillation. Our results on using the distillation strategy outlined in Section 5.4 of the main paper are shown in Table 9. Despite the model compression, the performance remains consistent.

METHOD	OFFICE-HOME				OFFICE-CALTECH				OFFICE		
	RW	PR	CL	AR	A	C	D	W	A	D	W
DECISION (original)	83.6	84.4	59.4	74.5	95.9	95.9	100	99.6	75.4	99.6	98.4
DECISION (distillation)	83.7	84.4	59.1	74.4	96.0	95.7	99.4	99.6	75.4	99.6	98.1

Table 9. **Distillation results on object recognition tasks.** Performance remains consistent across all datasets despite distilling into a single target model.