SECURE DOMAIN ADAPTATION WITH MULTIPLE SOURCES

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-source unsupervised domain adaptation (MUDA) is a recently explored learning framework within UDA, where the goal is to address the challenge of annotated data scarcity in a target domain via transferring knowledge from multiple source domains with annotated data. When the source domains are distributed, data privacy and security can become a significant concern, e.g., medical domains, yet existing MUDA methods overlook this concern. We develop an algorithm to address MUDA when source domains' data cannot be shared. Our method is based on aligning the distributions of the source and target domains indirectly via internally learned distributions in an intermediate embedding space. Our theoretical analysis supports our approach and extensive empirical results demonstrate our algorithm is effective and compares favorably against existing MUDA methods.

1 INTRODUCTION

Advances in deep learning have led to significant performance boost in a wide range of machine learning (ML) applications (Russakovsky et al., 2015). However, deep learning suffers from limited performance in domains with scarce labeled data. Even if a deep network can be trained initially, distributional drifts of data during testing, i.e., domain-shift (Torralba & Efros, 2011), lead to sub-optimal performance of deep networks. The naive solution to tackle this challenge is to retrain the models. However, this requires annotating large datasets persistently, which is a time-consuming and a laborious manual process. Unsupervised Domain Adaptation (UDA) (Long et al., 2016) is a learning setting to address the challenge of domain-shift in a *target domain* with *unannotated data* through transferring knowledge from a related *source domain* at which labeled data is accessible.

An effective approach to address UDA is to map data points from a source domain and a target domain into a latent embedding space at which distributions for both domains are aligned. Since domain-shift is mitigated in the latent space, a source-trained classifier receiving latent features as input would generalize well on the target domain. The latent embedding space is usually modeled as the output-space of a deep encoder network trained to match the source and target distributions. This process can implemented using adversarial learning (Hoffman et al., 2018; Dou et al., 2019; Tzeng et al., 2017; Bousmalis et al., 2017), where the distributions are matched indirectly through generator and discriminator networks. Alternatively, a distributional probability metric can be minimized to align the two distributions directly (Chen et al., 2019; Sun et al., 2017; Lee et al., 2019).

Recently, single-source unsupervised domain adaptation (SUDA) has been extended to multi-source unsupervised domain adaptation (MUDA) to benefit from several sources of knowledge (Xu et al., 2018; Guo et al., 2018; Peng et al., 2019a; Redko et al., 2019; Zhao et al., 2020; Wen et al., 2020b; Lin et al., 2020; Guo et al., 2020; Tasar et al., 2020; Venkat et al., 2020a). The goal in MUDA is to benefit from the collective knowledge that is encoded in several distinct annotated source domains to improve model generalization on an unannotated target domain. Since several source domains exist, domain-shift and category-shift between pairs of source-domains are new challenges that need to be addressed. Most existing MUDA algorithms consider the annotated source datasets are centrally accessible. However, it is natural to assume that in practical settings the source datasets are distributed among independent entities. The naive solution is to transfer all the datasets to a central server to address MUDA. However, sharing data in some applications, e.g, medical domains, may not be feasible due to data privacy as well as communication bandwidth limitations. Additionally,

sharing the source datasets may be infeasible due to security concerns. We develop an algorithm to relax the need for centralized processing of data for MUDA. Our contributions include:

- We address the challenge of data privacy for multi-source UDA by maintaining full privacy between any pair of domains. In our approach, the training datasets for source domains are not shared and only high-level learned knowledge from sources is shared with the target.
- We propose an efficient distributed optimization process for MUDA to process each dataset locally and to encode high-level learned knowledge in a latent embedding space.
- We provide theoretical justification for our method by proving our algorithm minimizes an upper bound on the target error. We also provide extensive experimental results on four standard benchmark datasets to demonstrate that our method is effective.

2 Related work

Single-Source Unsupervised Domain Adaptation: The problem of single-source UDA has been studied extensively. The primary approach for UDA in recent works involves training a deep neural network jointly on a labeled source domain and an unlabeled target domain to align the distribution for both domains in a latent space. This has been achieved with generative adversarial networks (Goodfellow et al., 2014) to encourage domain alignment by optimizing a domain discriminator tasked with discerning source features from generated target features (Hoffman et al., 2018; Dhouib et al., 2020; Luc et al., 2016; Tzeng et al., 2017; Sankaranarayanan et al., 2018). Another primary approach is to directly minimize the distance between the two distributions (Long et al., 2015; 2017b; Morerio et al., 2018). However, SUDA algorithms do not leverage inter-domain relations in the presence of several source domains and hence, do not generalize well for MUDA.

Multi-Source Unsupervised Domain Adaptation: MUDA methods concomitantly leverage multiple streams of data for improved generalization on the target domain. Xu et al. (2018) minimize discrepancy between source and target domains by optimizing an adversarial loss. Peng et al. (2019a) adapt on multiple domains by aligning inter-domain statistics of the source domains in an embedding space. Guo et al. (2018) learn to combine domain specific predictions via meta-training. Venkat et al. (2020a) use pseudo-labels to improve domain alignment. Negative transfer across the source domains is an additional challenge for MUDA. Li et al. (2018) exploit domain similarity to avoid negative transfer by reasoning about domains in a shared embedding space. Zhu et al. (2019) achieve domain alignment by adapting deep networks at various levels of abstraction. Zhao et al. (2020) align target features against source trained features via optimal transport, then combine source domains proportionally with respect to their Wasserstein distance. Wen et al. (2020a) use a discriminator to exclude data samples with negative impact on generalization performance.

Privacy in Domain Adaptation: The importance of inter-domain privacy has been recognized and explored for single-source UDA, as in many important practical settings, privacy regulations limit possibility of sharing data (Peng et al., 2019b; Li et al., 2020; Liang et al., 2020; 2021b;a). Privacy preserving for MUDA is a relatively unexplored problem. Only recently, Ahmed et al. (2021) explored privacy-preserving MUDA via information maximization and pseudo-labeling. Unlike our approach, Ahmed et al. (2021) require simultaneous access to all the source models during the adaptation process and hence only relax sharing source datasets with the target domain. We address a more constrained setting, where privacy should be preserved both between source domains and source domains and target domain which is a more practical assumption.

Our approach builds on the idea of probability metric minimization, explored in UDA (Morerio et al., 2018; Bhushan Damodaran et al., 2018; Chen et al., 2019; Sun et al., 2017; Lee et al., 2019; Redko et al., 2019). To this end, a suitable probability metric is selected and minimized at the output-space of a deep encoder to enforce domain alignment. In this work, we used the Sliced Wasserstein Distance (SWD) (Rabin et al., 2011; Bonneel et al., 2015) for this purpose. SWD is a metric for approximating the optimal transport metric (Redko et al., 2019). It is a suitable choice for UDA because: i) It possesses non-vanishing gradients for two high-dimensional distributions with non-overlapping supports through exploiting the geometry of the embedding space. As a result, it is a suitable objective function for deep learning optimization which usually is solved using gradient-based techniques, e.g., stochastic gradient descent. ii) It can be computed efficiently based on a closed-form solution using only empirical samples, drawn from the two distributions.

3 PROBLEM FORMULATION

Let $\mathcal{S}_1, \mathcal{S}_2 \dots \mathcal{S}_n$ denote input distributions that represent n source domains and similarly $\mathcal{T}$ be the data distribution corresponding to an unlabeled target domain. We assume all domains have a common label-space $\mathcal{Y}$, but not necessarily sharing the same label distribution. For each source domain k, we observe the labeled samples $\{(\boldsymbol{x}^s_{k,1}, \boldsymbol{y}_{k,1}), \dots, (\boldsymbol{x}^s_{k,n^s_k}, \boldsymbol{y}_{k,n^s_k})\}$, where $\boldsymbol{x}^s_k \sim \mathcal{S}_k$. Additionally, we assume only unlabeled samples $\{\boldsymbol{x}^t_1, \dots, \boldsymbol{x}^t_{n^t}\}$ are accessible on the target domain $\mathcal{T}$. The goal is to train a model $f_\theta : \mathcal{S}_1 \cup \mathcal{S}_2 \cup \dots \mathcal{S}_n \cup \mathcal{T} \to \mathcal{Y}$ with learnable parameter θ, e.g., a deep network with weights θ, that minimizes true risk for the predicted labels on the target domain.

In the absence of labeled data in the target domain, we first train models on each source domain via empirical risk minimization (ERM), i.e., via minimizing cross-entropy loss on the sources' labeled datasets: $\theta_k = \arg\min_\theta \frac{1}{n^s_k} \sum_{i=1}^{n^s_k} \mathcal{L}_{ce}(f_\theta(\boldsymbol{x}^s_{k,i}), \boldsymbol{y}_{k,i})$. Since the target domain shares the same label space with the source domains, these models can be directly used on the target domain as a naive solution. However, given the distributional discrepancy between the source and target domains, generalization performance will be poor. The goal in MUDA is to benefit from the unannotated target dataset and the source-trained models in order to improve upon source model performance.

To this end, we decompose the model f_θ into a feature extractor encoder $g_{\boldsymbol{u}}(\cdot) : \mathbb{R}^{d_1 \times d_2 \times 3} \to \mathbb{R}^{d_h}$ and a classifier subnetwork $h_{\boldsymbol{v}}(\cdot) : \mathbb{R}^{d_h} \to \mathbb{R}^{|\mathcal{Y}|}$ with learnable parameters $\boldsymbol{u}$ and $\boldsymbol{v}$, such that $f(\cdot) = (h \circ g)(\cdot)$. Here, we assumed input data points are images of size $d_1 \times d_2 \times 3$ and the latent embedding shape is of size d_h. In a SUDA setting, we can improve generalization of each source-specific model on the target domain by aligning the distributions of the source and the target domain in the latent embedding space. Specifically, we can minimize a distributional discrepancy metric $D(\cdot, \cdot)$ across both domains, e.g., SWD loss, to update the learnable parameters: $\boldsymbol{u}^A_k = \arg\min_{\boldsymbol{u}} D(g_{\boldsymbol{u}}(\mathcal{S}_k), g_{\boldsymbol{u}}(\mathcal{T}))$. By aligning the two distributions, the source trained classifier h_k will generalize well on the target domain $\mathcal{T}$. In the MUDA setting, the goal is to improve upon SUDA by benefiting from the collective knowledge of the source domains to make predictions on the target domain. This can be done via a weighted average of predictions made by each of the domain-specific models, i.e., models with learnable parameters $\theta^A_k = (\boldsymbol{u}^A_k, \boldsymbol{v}_k)$. Thus, for a sample $\boldsymbol{x}^t_i$ in the target domain, the model prediction will be $\sum_{k=1}^n w_k f_{\theta^A_k}(X^t_i)$, where w_k denotes a set of learnable weights associated with the source domains. The weights are set according to model reliability.

We note the above general approach requires joint access to source and target data during adaptation. We consider a more challenging setting, where we lose access to the source domains once training is finished, as well as forbid interaction between source models during adaptation. This privacy focused assumption is realistic in applications with sensitive and private data, e.g., medical data. Hence, the source domain distribution in the embedding space, i.e., $g(\mathcal{S}_k)$ will become inaccessible. To circumvent this challenge, we rely on intermediate distributional estimates.

4 PROPOSED ALGORITHM

Our proposed approach for MUDA with private data is visualized in Figure 1. As it can be seen, our approach is based on two levels of hierarchies. We first adapt each source-trained model while preserving privacy (left and middle subfigures). We then combine predictions of the source-specific models on the target domain according to their reliability (right subfigure). To tackle the challenge of data privacy, we approximate the distributions of the source domains in the embedding space as a multi-modal distribution and use these distributional estimates for domain alignment (Figure 1, left). We can benefit from these estimates because once source training is completed, the input distribution should be mapped into a $|\mathcal{Y}|-$modal distribution to enable the classifier subnetwork to separate the classes. Note, each separated distributional mode encodes one of the classes (see Figure 1, left). To approximate these internal distributions we use Gaussian Mixture Models, with mean and covariance parameters μ_k, Σ_k. Since we have access to labeled data points on the source domains, we can learn μ_k and Σ_k in a supervised fashion. Let $\mathbb{1}_c(x)$ denotes the indicator function for $x = c$, then the maximum likelihood estimates for the GMM parameters would be:

$$\mu_{k,c} = \frac{\sum_{i=1}^{n^s_k} \mathbb{1}_c(\boldsymbol{y}_{k,i}) g_{\boldsymbol{u}_k}(\boldsymbol{x}^s_{k,i})}{\sum_{i=1}^{n^s_k} \mathbb{1}_c(\boldsymbol{y}_{k,i})}, \Sigma_{k,c} = \frac{\sum_{i=1}^{n^s_k} \mathbb{1}_c(\boldsymbol{y}_{k,i}) (g_{\boldsymbol{u}_k}(\boldsymbol{x}^s_{k,i}) - \mu_{k,c})(g_{\boldsymbol{u}_k}(\boldsymbol{x}^s_{k,i}) - \mu_{k,c})^T}{\sum_{i=1}^{n^s_k} \mathbb{1}_c(\boldsymbol{y}_{k,i})} \quad (1)$$

Learning μ_k and Σ_k for each domain k, enables us to sample class conditionally from the GMMs and approximate $g(\mathcal{S}_k)$ in the absence of the source dataset to implement domain alignment.

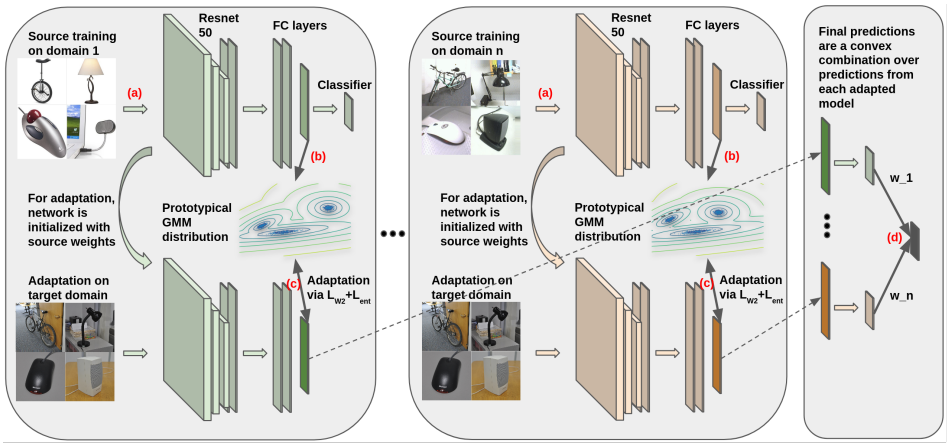

Figure 1: Block-diagram of the proposed approach: (a) source-specific model training is done independently for each source domain, potentially using different data storage (b) each latent source domain distribution is estimated via a GMM, (c) the source-trained network is adapted on the target domain by performing pairwise domain alignment between the GMM distribution and the unlabeled target data, and by minimizing conditional-entropy for the model target predictions (d) the final target domain predictions are obtained via a convex combinations of logits for each adapted model

We adapt the source-trained model by aligning the target distribution with the GMM distribution in the embedding space. To preserve privacy, for each source domain k we generate intermediate pseudo-domains A_k with pseudo-samples $\{\boldsymbol{z}^a_{k,1}, \ldots, \boldsymbol{z}^a_{k,n^a_k}\}$ by drawing random samples from the estimated GMM distribution. The pseudo-domain is used as an approximation of the corresponding source domain. To align the two distribution, we need to select a suitable distance metric $D(\cdot, \cdot)$. We rely on the SWD for this purpose due to its mentioned appealing properties. Since the prior probabilities on classes are not known in the target domain, optimizing the SWD may lead to clustering samples from different classes together, depending on the discrepancy between the label distributions. To compensate for this challenge, we take advantage of the conditional entropy loss (Grandvalet & Bengio, 2004) as a regularization term based on information maximization. The conditional entropy $\mathcal{L}_{ent}(f_\theta(\mathcal{T})) = \mathcal{L}_{CE}(f_\theta(\mathcal{T}), f_\theta(\mathcal{T}))$ acts as a soft clustering objective to enhance domain alignment. To ensure the feature extractor benefits from this added loss, the classifier is frozen during model adaptation. Our final loss used for source-specific adaptation is:

$$D(g(\mathcal{T}), A) + \gamma \mathcal{L}_{ent}(f_\theta(\mathcal{T})). \tag{2}$$

Once the source-specific adaptation is completed across all domains, the final model predictions on the target domain are obtained by combining probabilistic predictions returned by each of the n domain-specific models. The mixing weights are chosen as a convex vector $\boldsymbol{w} = (w_1 \ldots w_n)$, i.e., $w_i > 0$ and $\sum_i w_i = 1$. We use weighted averages to account for different levels of similarities between the target domain and each source domain. The choice of w is critical, as assigning large weights to a model which does not generalize well will harm MUDA performance on the target domain. We use the source-specific model prediction confidence on the target domain as a proxy for determining the weight according to pairwise domain similarities. We have provided empirical evidence for this choice in Section 6. We thus set a confidence threshold λ and assign w_k:

$$\tilde{w}_k \sim \sum_{i=1}^{n^t} \mathbb{1}(f_{\theta_k}(\boldsymbol{x}^t_i) > \lambda), \quad w_k = \tilde{w}_k / \sum \tilde{w}_k. \tag{3}$$

Note the only cross-domain information transfer in our framework is communicating the latent means and covariance matrices of the estimated GMMs plus the domain-specific model weights that

provide a warm start for adaptation. Throughout the whole pretraining and adaptation processes, data samples are never transferred between any two domains. As a result, our approach preserves data privacy for scenarios at which the source datasets are distributed across several entities. Additionally, the adaptation process for each source domain is performed independently. As a result, our approach can be used to incorporate new source domains as they become available over time without requiring end-to-end retraining from scratch. We will only require to update the normalized mixing weights via Equation 3, which takes negligible runtime compared to model training. Our proposed privacy preserving approach to address MUDA is presented in Algorithm 1 .

5 THEORETICAL ANALYSIS

We provide an analysis to demonstrate that our algorithm minimizes an upperbound for the target domain error. We adopt the framework developed by Redko et al. Redko & Sebban (2017) for single source UDA using Wasserstein distance to provide a theoretical justification for the Algorithm 1. Our analysis is performed in the latent embedding space. Let $\mathcal{H}$ represent the hypothesis space of all classifier subnetworks. Let $h_k(\cdot)$ denote the model learnt by each domain-specific model. We also set $e_{\mathcal{D}}(\cdot)$, where $\mathcal{D} \in \{\mathcal{S}_1 \dots \mathcal{S}_n, \mathcal{T}\}$, to be the true expected error returned by some model $h(\cdot) \in \mathcal{H}$ in the hypothesis space on the domain $\mathcal{D}$. Additionally, let $\hat{\mu}_{\mathcal{S}_k} = \frac{1}{n_k^s} \sum_{i=1}^{n_k^s} f(g(\boldsymbol{x}_{k,i}^s))$, $\hat{\mu}_{\mathcal{P}_k} = \frac{1}{n_k^a} \sum_{i=1}^{n_k^a} \boldsymbol{x}_{k,i}^a$, and $\hat{\mu}_{\mathcal{T}} = \frac{1}{n^t} \sum_{i=1}^{n_k^s} f(g(\boldsymbol{x}_i^t))$ denote the empirical distributions that are built using the samples for the source domain, the intermediate pseudo-domain, and the target domain in the latent space, respectively. Then the following theorem holds for the MUDA setting:

Algorithm 1 Secure Multi-source Unsupervised Domain Adaptation (SMUDA)

procedure SMUDA($\mathcal{S}_1 \dots \mathcal{S}_n, \mathcal{T}, L, \gamma$)
  for $k \leftarrow 1$ to n **do**
    $\mu_k, \Sigma_k, \theta_k = Train(\mathcal{S}_k)$
    Generate A_k based on μ_k, Σ_k
    Compute w_k via Equation 3
    $\theta_k^A = Adapt(\theta_k, A_k, \mathcal{T}, L, \gamma)$
  return $w_1 \dots w_n, \theta_1^A \dots \theta_n^A$
procedure TRAIN($\mathcal{S}_i$)
  Learn $\theta_k = (u_k, v_k)$ by minimizing $\mathcal{L}_{CE}(f_{\theta_k}(\mathcal{S}_k), \cdot)$
  Learn parameters μ_k, Σ_k following Equation 1
  return $\mu_k, \Sigma_k, \theta_k$
procedure ADAPT($\theta_k, A_k, \mathcal{T}, L, \gamma$)
  Initialize network with weights θ_k
  $\theta_k^A = \arg\min_\theta D(g_u(\mathcal{T}), A_k) + \gamma \mathcal{L}_{ent}(f_\theta(\mathcal{T}))$ via Equation 2
  return θ_k^A

Theorem 5.1. *Consider Algorithm 1 for MUDA under the explained conditions, then*

$$e_{\mathcal{T}}(h) \leq \sum_{k=1}^{n} w_k (e_{\mathcal{S}_k}(h_k) + D(\hat{\mu}_{\mathcal{T}}, \hat{\mu}_{\mathcal{P}_k}) + D(\hat{\mu}_{\mathcal{P}_k}, \hat{\mu}_{\mathcal{S}_k}) + \sqrt{\big(2\log(\frac{1}{\xi})/\zeta\big)}\big(\sqrt{\frac{1}{N_k}} + \sqrt{\frac{1}{M}}\big) + e_{\mathcal{C}_k}(h_k^*)) \tag{4}$$

where $\mathcal{C}_k$ is the combined error loss with respect to domain k, and h_k^* is the optimal model with respect to this loss when the model is trained jointly on annotated datasets from all domains.

Proof. due to space limitations, the complete proof is included in the Appendix.

We see the target domain error is upperbounded by the convex combination of the domain-specific adaptation errors. Algorithm 1 minimizes the right-hand side of Equation 4 as follows: for each source domain, our method minimizes the source expected error by training the models on each domain using ERM. The second term is minimized because the distance between the distributions of the intermediate pseudo-domain and the target domain is directly minimized in the latent space. The third term corresponds to how well the GMM distribution approximates the latent source samples. Our algorithm does not minimize this term but if the model performs well on the source domain (a prerequisite for domain adaptation) and a multi-modal distribution is formed in the embedding space (necessary for good performance), this term will be small. The second to last term is dependent on the number of available samples in the adaptation problem, and becomes negligible when sufficient number of samples are accessible. The final term measures the difficulty of the optimization, and is dependent only on the structure of the data. For related domains, this term will be small.

6 EXPERIMENTAL VALIDATION

Datasets We validate our algorithm on four standard domain adaptation benchmark datasets: *Office-31*, *Office-Home*, *Office-Caltech* and *Image-Clef*.

Office-31 (Saenko et al., 2010) is a dataset consisting of 4110 images from an office environment pertaining to three domains: Amazon, Webcam and DSLR. Domains differ in image quality, background, number of samples, class distributions etc. Images in all three domains are categorized under 31 different categories. **Office-Home** (Venkateswara et al., 2017) contains 30475 from four different domains: Art (stylized images), Clipart (clipart sketches), Product (images with no background) and Real-World (realistic images). Each domain contains images from 65 shared classes. **Office-Caltech** (Gong et al., 2012) contains 2533 office related images from four domains: Amazon, Webcam, DSLR, Caltech, falling under 10 categories. **Image-Clef** (Long et al., 2017a) contains 1800 images under 12 categories from three domains: Caltech, Imagenet and Pascal.

Preprocessing & Network structure: we follow the literature for a fair comparison. For each domain we re-scale images to a standard size of $(224, 224, 3)$. We use a ResNet50 (He et al., 2016) network as a backbone for the feature extractor, followed by fully connected layers. The network classification head consists of a linear layer, and source-training is performed using cross-entropy loss. The ResNet layers of the feature extractor are frozen during adaptation. We use classification accuracy for comparison and report average performance across five random runs. Experiments were run on a NVIDIA Titan Xp GPU. Our code is available at `redacted` (check Supplemntary).

To test the effectiveness of our privacy preserving approach for MUDA, we compare our method against state-of-the art SUDA and MUDA approaches. Benchmarks for single-best and source-combined performance are reported based on DAN (Long et al., 2015), D-CORAL (Sun & Saenko, 2016), RevGrad (Ganin & Lempitsky, 2015). We include most existing MUDA algorithms: DCTN (Xu et al., 2018), FADA (Peng et al., 2019b), MFSAN (Zhu et al., 2019), MDDA (Zhao et al., 2020), SimpAl (Venkat et al., 2020b), JAN (Long et al., 2017b), MEDA (Wang et al., 2018), MCD (Saito et al., 2018), M3SDA (Peng et al., 2019a), MDAN (Zhao et al., 2018), MDMN (Li et al., 2018), DARN (Wen et al., 2020a), DECISION (Ahmed et al., 2021), SHOT-Ens (Liang et al., 2021a; Ahmed et al., 2021). Note that we maintain full domain privacy throughout training and adaptation and hence most of the above works should be considered an **upperbound** for our performance as they address a less constrained problem by directly processing the source samples. Despite the privacy constraint, our results indicate our algorithm is competitive and at times outperforms the above approaches. We next present quantitative and qualitative analysis of our work.

6.1 PERFORMANCE RESULTS

Our performance results are presented in Table 1. In the case of **Office-31**, we observe state-of-the-art performance (SOTA) on the $\rightarrow A$ task with competitive performance on the other two tasks. Note that the domains *DSLR* and *Webcam* share similar distributions, as exemplified through the Source-Only performance, and hence there is small room to improve upon the Source-Only results. In the case of **Image-clef**, we obtain SOTA performance on the $\rightarrow C$ task and nearly SOTA on the $\rightarrow P$ task, and competitive performance on the last task. On the **Office-caltech** dataset, we obtain SOTA performance on the $\rightarrow A$ task, with close to SOTA performance on the three other tasks. Finally, we note the domains of the **Office-home** dataset have larger domain gaps with more classes, meaning this dataset is arguably the most challenging dataset of the four. Our approach obtains near SOTA performance on the $\rightarrow P$ and $\rightarrow R$ tasks and competitive performance on the remaining tasks. We reiterate most other MUDA algorithms should serve as upperbounds, as they either access source data directly, simultaneously use models from all sources for adaptation, or both. Our results across all tasks demonstrate that not only are we able to generate performance similar to these methods while preserving data privacy, but also obtain SOTA results on several of the tasks.

6.2 ABLATIVE STUDIES

We perform ablative experiments by investigating the effect of each of the two loss terms on performance in our combined loss in Eq. 2. Our ablative experiments are presented in Table 2. We observe that except for the *Office-caltech* dataset, combining the two terms yields improved performance for the rest of the datasets. Note however, the effect in the *Office-caltech* dataset is negligible. On the

	Method	→D	→W	→A	Avg.
SB	Source Only	99.3	96.7	62.5	86.2
	DAN	99.7	98.0	65.3	87.7
	D-CORAL	99.7	98.0	65.3	87.7
	RevGrad	99.1	96.9	66.2	87.5
SC	DAN	99.6	97.8	67.6	88.3
	D-CORAL	99.3	98.0	67.1	88.1
	RevGrad	99.7	98.1	67.6	88.5
MS	MDDA	99.2	97.1	56.2	84.2
	DCTN	99.3	98.2	64.2	87.2
	MFSAN	99.5	98.5	72.7	90.2
	SImpAl	$99.2^{\pm 0.2}$	$97.4^{\pm 0.1}$	$70.6^{\pm 0.6}$	89.0
	SHOT-Ens	**99.6**	94.9	75	89.3
	DECISION	**99.6**	**98.4**	75.4	91.1
	SMUDA (ours)	$99.4^{\pm 0.1}$	$97.3^{\pm 0.4}$	$\mathbf{76}^{\pm 0.6}$	90.9

(a) Office-31

	Method	→W	→D	→C	→ A	Avg.
SB	Source Only	99.0	98.3	87.8	86.1	92.8
	DAN	99.3	98.2	89.7	94.8	95.5
MS	FADA	88.1	87.1	88.7	84.2	87.1
	DAN	99.5	99.1	89.2	91.6	94.8
	DCTN	99.4	99.0	90.2	91.6	94.8
	JAN	99.4	99.4	91.2	91.8	95.5
	MEDA	99.3	99.2	91.4	92.9	95.7
	MCD	99.5	99.1	91.5	92.1	95.6
	M3SDA	99.4	99.2	91.5	94.1	96.1
	SImpAl	$99.3^{\pm 0.1}$	$99.8^{\pm 0.1}$	$92.2^{\pm 0.1}$	$95.3^{\pm 0.2}$	96.7
	SHOT-Ens	**99.6**	96.8	95.8	95.7	97.0
	DECISION	**99.6**	**100**	**95.9**	**95.9**	98.0
	SMUDA (ours)	$99.3^{\pm 0.3}$	$97.6^{\pm 0.3}$	$93.9^{\pm 0.1}$	$\mathbf{95.9}^{\pm 0.1}$	96.6

(b) Office-caltech

	Method	→P	→C	→I	Avg.
SB	Source Only	74.8	91.5	83.9	83.4
	DAN	75.0	93.3	86.2	84.8
	D-CORAL	76.9	93.6	88.5	86.3
	RevGrad	75.0	96.2	87.0	86.1
SC	DAN	77.6	93.3	92.2	87.7
	D-CORAL	77.1	93.6	91.7	87.5
	RevGrad	77.9	93.7	91.8	87.8
MS	DCTN	75.0	95.7	90.3	87.0
	MFSAN	**79.1**	95.4	**93.6**	89.4
	SImpAl	$77.5^{\pm 0.3}$	$93.3^{\pm 0.3}$	$91.0^{\pm 0.4}$	87.3
	SMUDA (ours)	$79^{\pm 0.5}$	$\mathbf{95.9}^{\pm 0.1}$	$91.8^{\pm 0.5}$	88.9

(c) Image-clef

	Method	→A	→C	→P	→ R	Avg.
SB	Source Only	65.3	49.6	79.7	75.4	67.5
	DAN	68.2	56.5	80.3	75.9	70.2
	D-CORAL	67.0	53.6	80.3	76.3	69.3
	RevGrad	67.9	55.9	80.4	75.8	70.0
SC	DAN	68.5	59.4	79.0	82.5	72.4
	D-CORAL	68.1	58.6	79.5	82.7	72.2
	RevGrad	68.4	59.1	79.5	82.7	72.4
MS	MFSAN	72.1	62.0	80.3	81.8	74.1
	M3SDA	$64.1^{\pm 0.6}$	$62.8^{\pm 0.4}$	$76.2^{\pm 0.3}$	$78.6^{\pm 0.2}$	70.4
	SImpAl	$70.8^{\pm 0.2}$	$56.3^{\pm 0.2}$	$80.2^{\pm 0.3}$	$81.5^{\pm 0.3}$	72.2
	MDAN	$68.1^{\pm 0.6}$	$67.0^{\pm 0.2}$	$81.0^{\pm 0.2}$	$82.8^{\pm 0.1}$	74.8
	MDMN	$68.7^{\pm 0.6}$	$67.6^{\pm 0.2}$	$81.4^{\pm 0.2}$	$83.3^{\pm 0.1}$	75.3
	DARN	$70.0^{\pm 0.4}$	$\mathbf{68.4}^{\pm 0.1}$	$82.8^{\pm 0.2}$	$\mathbf{83.9}^{\pm 0.2}$	76.26
	SHOT-Ens	72.2	59.3	82.8	82.9	74.3
	DECISION	**74.5**	59.4	**84.4**	83.6	75.5
	SMUDA (ours)	$69.1^{\pm 0.3}$	$61.5^{\pm 0.3}$	$83.5^{\pm 0.2}$	$83.4^{\pm 0.2}$	74.4

(d) Office-home

Table 1: Main results, on the four considered datasets.

other hand, we see minimizing the SWD loss plays a more dominant role in performance improvements on the *Office-31* and *Office-home* datasets. In contrast, the conditional entropy contributes more on the *Image-clef* and *Office-caltech* datasets. Our insight is that the conditional entropy term performs better when the source trained models have higher performance on the target domain prior to the source-level adaptation, while the SWD term is more vital when there is a larger discrepancy between the source domains and the target domain. Ablative experiments conclude that both loss terms contribute in improving our performance.

Method	→D	→W	→A	Avg.
SWD only	95.8	95.3	72.6	87.9
$\mathcal{L}_{ent}$ only	99.6	97.6	63.7	87
SMUDA	99.4	97.3	76	90.9

(a) Office-31

Method	→W	→D	→C	→ A	Avg.
SWD only	98.1	97.8	92.1	95.5	95.9
$\mathcal{L}_{ent}$ only	99.4	97.7	94	96	96.8
SMUDA	99.3	97.6	93.9	95.9	96.6

(b) Office-caltech

Method	→P	→C	→I	Avg.
SWD only	78.1	94.6	90.8	87.8
$\mathcal{L}_{ent}$ only	78	95.6	91.3	88.3
SMUDA	79	95.9	91.8	88.9

(c) Image-clef

Method	→A	→C	→P	→ R	Avg.
SWD only	66.6	59.1	80.9	82.2	72.2
$\mathcal{L}_{ent}$ only	64.5	49.4	77.8	72.2	66
SMUDA	69.1	61.5	83.5	83.4	74.4

(d) Office-home

Table 2: Performance when using the SWD objective, the entropy objective or both (SMUDA).

6.3 Empirical Analysis

In our empirical analysis we study the effect of hyperparameters on the performance of our method to provide a better understanding of the algorithm.

We first validate empirically our choice for computing the mixing parameters w_k. We consider four mixing scenarios for combining the models' prediction: (i) Eq. 3, (ii) setting weights proportional to Sliced Wasserstein Distance between the intermediate and the target domains (a cross-domain measure of distributional similarity), (iii) using a uniform average, and (iv) assigning all mixing weight to the model with best target performance. Average performances for tasks of each dataset are reported in Table 3. We observe that our choice leads to maximum performance. We note the single

best performance is able to slightly outperform our method on two of the tasks, however suffers on datasets where domains have significant pairwise domain gap. This observation is expected, as using several domains is beneficial when they complement each other in terms of available information. Assigning weights proportional to $D(g(\mathcal{T}), A_k)$ may seem a reasonable choice, given that similarity between the pseudo-datasets and the target latent features indicates better classifier generalization. However, this method performs better only than the uniform averaging. We conclude that model reliability is a better criterion to combine models. The uniform averaging leads to poor generalization on the target domain because it treats all the models similarly. As a results, models with the least generalization ability on the target domain harm the collective performance.

Dataset	High confidence	W2	Uniform	Single Best
office-31	90.9	87.1	87.2	91
image-clef	88.9	88.8	88.8	88.4
office-caltech	96.6	96.6	96.6	97
office-home	74.4	74.2	74.2	72.8
total avg.	**87.6**	86.6	86.6	87.3

Table 3: Analytic experiments to study four strategies for combining the individual model predictions.

We study the effect of the SWD projection hyperparameter. SWD utilizes L random projections, as detailed in Equation 9 in the Appendix. We analyze the impact of this parameter on the adaptation performance using the *Office-31* dataset. In Figure 2, we reported performance results for $L \in \{1, 10, 50, 100, 200, 350, 500\}$. The SWD approximation becomes tighter with an increased number of projections, which we see translating on all three tasks.

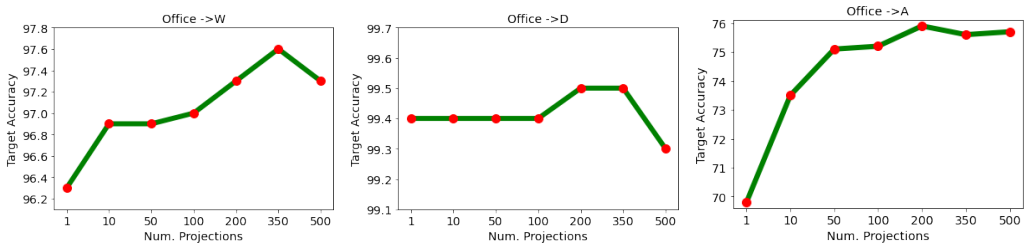

Figure 2: Performance of our algorithm under different numbers of latent projections used in the computation of the Sliced Wasserstein Distance. Results reported on the Office-31 tasks.

Next, we study the effect of the adaptation process using our algorithm on the target domain performance. Figure 3 presents the effect of adaptation process on an *Office-home* task. We note an increase in the target domain accuracy once adaptation commences, observation which is in line with the metrics reported so far. We also observe that MUDA performance using the three source domains outperforms the three SUDA performances, with the biggest discrepancy being observable for the *Clipart* trained model which is the most different domain from the target domain *Real World*.

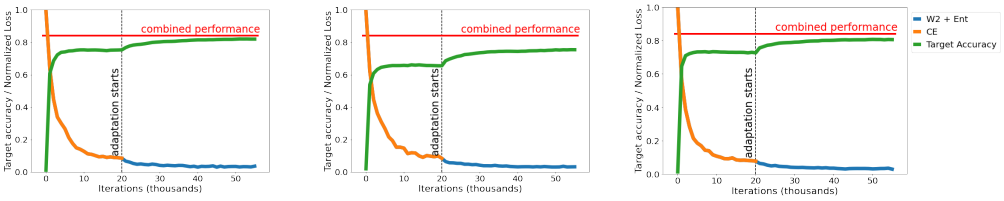

Figure 3: Effect of the adaptation process on the *Office-home* dataset: from left to right, we consider *Art, Clipart* and *Product* as the source domains, and *Real World* as the target domain.

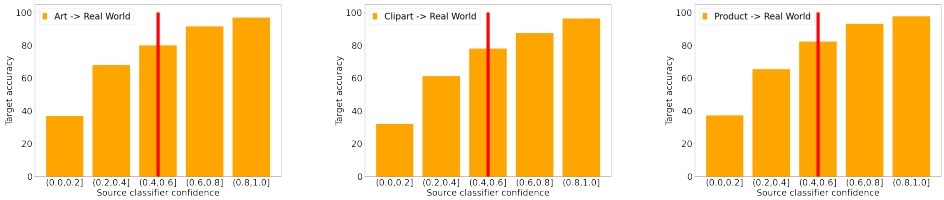

Figure 4: Prediction accuracy on the target domain for different levels of source model confidence, and our choice of λ. Tasks from the *Office-home* dataset are used in this experiments.

Another primary hyperparanter for our algorithm is the confidence parameter λ. Figure 4 provides the prediction accuracy for the high-confidence target domain samples based on the source-only models using the *Office-home* dataset. We observe low-confidence predictions offer poor accuracy for the target domain. For example, we see that when the confidence is less than 0.2, prediction accuracy is around 40%. On the other hand, for target samples with confidence-level above .6, we have an accuracy around 90% on all the three tasks of *Office-home*. This experiments supports our intuition that the mixing weights w_k can be determined based on the share of the high confidence target samples as a measure of reliability for the source-trained models.

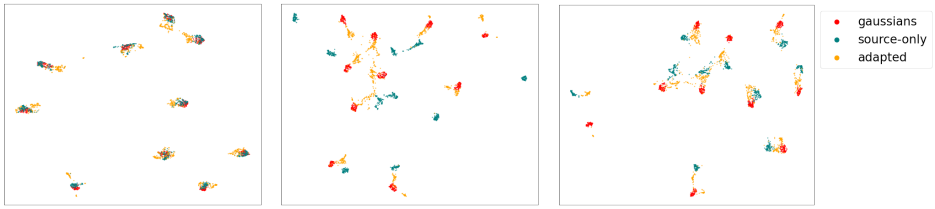

Figure 5: UMAP visualization of data representations in the embedding space for *Office-caltech* with *Amazon* as the target domain. From left to right: *Caltech, DSLR*, and *Webcam* as source domains.

We also assess the impact of domain adaptation on data representations in the latent embeddings in Figure 5. For data visualization, we reduced the data representation dimension to two using UMAP (McInnes et al., 2018). We display the GMM samples, the target latent embeddings before adaptation, and target latent embeddings post-adaptation in Figure 5. We observe that for each source domain, data representations for the target domains are shifted towards the GMM distribution throughout the adaptation process. This observation empirically validates the theoretical justification for our algorithm. Given the classification heads trained on the source domains are able to generalize well on the GMM samples as a result of pretraining, we conclude that source-specific domain alignment translate into an improved collective generalization performance.

Finally, we investigate the representation quality of the intermediate GMM distribution as a surrogate for the source data distribution, which is the backbone of our method. In Figure 6, we have visualized the data representations for the estimated GMMs and the source domain distributions for the *Image-clef* dataset. We note that for both source domains, their latent space distributions after pretraining are multi-modal distributions with 12 modes, each corresponding to one of the 12 classes. This observation confirms that we can approximate the source domain distribution with a GMM. We also observe that for both source domains the estimated GMM distribution approximates the source domain distribution in the embedding space with high accuracy. This experiment validates empirically that the third term in Eq. 4 is small in practice.

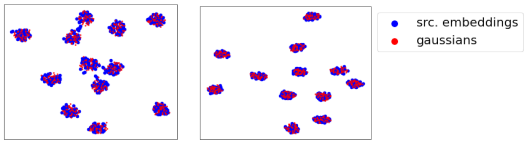

Figure 6: Source and GMM embeddings for the *Image-clef* dataset with *Imagenet* as the target and *Pascal* and *Caltech* as sources.

7 Conclusion

We develop a novel privacy-preserving MUDA algorithm. Our approach is based on the assumption that an input distribution is mapped into a multi-modal distribution in an embedding space as a result of supervised learning. We achieve privacy between each source domain and the target domain by minimizing the SWD loss between an intermediate GMM distribution and the target domain distribution in the latent embedding space. We then combine the source-specific models according to their reliability. We provide theoretical analysis to justify our algorithm. Our experimental results demonstrate that our algorithm performs favorably against SOTA MUDA algorithms on four standard benchmark datasets while preserving privacy. Future direction includes considering setting where the target domain shares different classes with each of the source domains.

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, 2021.

Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 447–463, 2018.

Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and Radon Wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, 2015.

Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3722–3731, 2017.

Chaoqi Chen, Weiping Xie, Wenbing Huang, Yu Rong, Xinghao Ding, Yue Huang, Tingyang Xu, and Junzhou Huang. Progressive feature alignment for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 627–636, 2019.

Sofien Dhouib, Ievgen Redko, and Carole Lartizien. Margin-aware adversarial domain adaptation with optimal transport. In *Thirty-seventh International Conference on Machine Learning*, 2020.

Qi Dou, Cheng Ouyang, Cheng Chen, Hao Chen, Ben Glocker, Xiahai Zhuang, and Pheng-Ann Heng. Pnp-adanet: Plug-and-play adversarial domain adaptation network at unpaired cross-modality cardiac segmentation. *IEEE Access*, 7:99065–99076, 2019.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of International Conference on Machine Learning*, pp. 1180–1189, 2015.

B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 2066–2073. IEEE, 2012.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems*, volume 17, pp. 529–536, 2004.

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, pp. 7830–7838, 2020.

Jiang Guo, Darsh Shah, and Regina Barzilay. Multi-source domain adaptation with mixture of experts. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4694–4703, 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*, pp. 770–778, 2016.

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*, pp. 1989–1998. PMLR, 2018.

Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10285–10295, 2019.

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*, pp. 9641–9650, 2020.

Yitong Li, michael Murias, geraldine Dawson, and David E Carlson. Extracting relationships by multi-domain matching. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/2fd0fd3efa7c4cfb034317b21f3c2d93-Paper.pdf.

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*, pp. 6028–6039. PMLR, 2020.

Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation, 2021a.

Jian Liang, Dapeng Hu, Yunbo Wang, Ran He, and Jiashi Feng. Source data-absent unsupervised domain adaptation through hypothesis transfer and labeling transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021b.

Chuang Lin, Sicheng Zhao, Lei Meng, and Tat-Seng Chua. Multi-source domain adaptation for visual sentiment classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 2661–2668, 2020.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of International Conference on Machine Learning*, pp. 97–105, 2015.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, pp. 136–144, 2016.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2208–2217. JMLR. org, 2017a.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Deep transfer learning with joint adaptation networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2208–2217. PMLR, 06–11 Aug 2017b. URL http://proceedings.mlr.press/v70/long17a.html.

Pauline Luc, Camille Couprie, Soumith Chintala, and Jakob Verbeek. Semantic segmentation using adversarial networks. In *NIPS Workshop on Adversarial Training*, 2016.

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. UMAP: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861, 2018.

Pietro Morerio, Jacopo Cavazza, and Vittorio Murino. Minimal-entropy correlation alignment for unsupervised deep domain adaptation. In *ICLR*, 2018.

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*, pp. 1406–1415, 2019a.

Xingchao Peng, Zijun Huang, Yizhe Zhu, and Kate Saenko. Federated adversarial domain adaptation, 2019b.

J. Rabin, G. Peyré, J. Delon, and M. Bernot. Wasserstein barycenter and its application to texture mixing. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pp. 435–446. Springer, 2011.

A. Redko, I.and Habrard and M. Sebban. Theoretical analysis of domain adaptation with optimal transport. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 737–753. Springer, 2017.

Ievgen Redko, Nicolas Courty, Rémi Flamary, and Devis Tuia. Optimal transport for multi-source domain adaptation under target shift. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 849–858. PMLR, 2019.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *European Conference on Computer Vision*, pp. 213–226. Springer, 2010.

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*, pp. 3723–3732, 2018.

S. Sankaranarayanan, Y. Balaji, C. D Castillo, and R. Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *CVPR*, 2018.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pp. 443–450. Springer, 2016.

Baochen Sun, Jiashi Feng, and Kate Saenko. Correlation alignment for unsupervised domain adaptation. In *Domain Adaptation in Computer Vision Applications*, pp. 153–171. Springer, 2017.

Onur Tasar, Yuliya Tarabalka, Alain Giros, Pierre Alliez, and Sébastien Clerc. Standardgan: Multi-source domain adaptation for semantic segmentation of very high resolution satellite images by data standardization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 192–193, 2020.

A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 1521–1528. IEEE, 2011.

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*, pp. 7167–7176, 2017.

Naveen Venkat, Jogendra Nath Kundu, Durgesh Kumar Singh, Ambareesh Revanur, and R Venkatesh Babu. Your classifier can secretly suffice multi-source domain adaptation. In *NeurIPS*, 2020a.

Naveen Venkat, Jogendra Nath Kundu, Durgesh Kumar Singh, Ambareesh Revanur, and Venkatesh Babu R. Your classifier can secretly suffice multi-source domain adaptation. In *NeurIPS*, 2020b. URL https://proceedings.neurips.cc/paper/2020/hash/3181d59d19e76e902666df5c7821259a-Abstract.html.

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*, pp. 5018–5027, 2017.

Jindong Wang, Wenjie Feng, Yiqiang Chen, Han Yu, Meiyu Huang, and Philip S Yu. Visual domain adaptation with manifold embedded distribution alignment. In *Proceedings of the 26th ACM international conference on Multimedia*, pp. 402–410, 2018.

Junfeng Wen, Russell Greiner, and Dale Schuurmans. Domain aggregation networks for multi-source domain adaptation. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10214–10224. PMLR, 13–18 Jul 2020a. URL http://proceedings.mlr.press/v119/wen20b.html.

Junfeng Wen, Russell Greiner, and Dale Schuurmans. Domain aggregation networks for multi-source domain adaptation. In *International Conference on Machine Learning*, pp. 10214–10224. PMLR, 2020b.

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*, pp. 3964–3973, 2018.

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, pp. 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, pp. 5989–5996, 2019.

A APPENDIX

A.1 PROOF OF THEOREM 5.1

We offer a proof for Theorem 5.1 from the main paper. Consider the following results.

Theorem A.1. *Theorem 2 from Redko & Sebban (2017)*

Let h be the hypothesis learnt by our model, and h^* the hypothesis that minimizes $e_{\mathcal{S}} + e_{\mathcal{T}}$. Under the assumptions described in our framework, consider the existence of N source samples and M target samples, with empirical source and target distributions $\hat{\mu}_{\mathcal{S}}$ and $\hat{\mu}_{\mathcal{T}}$ in $\mathbb{R}^d$. Then, for any $d' > d$ and $\zeta < \sqrt{2}$, there exists a constant number N_0 depending on d' such that for any $\xi > 0$ and $\min(N, M) \geq N_0 \max(\xi^{-(d'+2)}, 1)$ with probability at least $1 - \xi$, the following holds:

$$e_{\mathcal{T}}(h) \leq e_{\mathcal{S}}(h) + W(\hat{\mu}_{\mathcal{T}}, \hat{\mu}_{\mathcal{S}}) + \sqrt{\big(2\log(\frac{1}{\xi})/\zeta\big)}\big(\sqrt{\frac{1}{N}} + \sqrt{\frac{1}{M}}\big) + e_{\mathcal{C}}(h^*) \tag{5}$$

The above theorem provides an upper bound on the target error with respect to the source error, the distance between source and target domains, a term that is minimized based on the number of samples, and a constant $e_{\mathcal{C}} = e_{\mathcal{S}}(h^*) + e_{\mathcal{T}}(h^*)$ describing the performance of an optimal hypothesis on the present set of samples.

We adapt the result in Theorem A.1 to provide an upper bound in our multi-source setting. Consider the following two results.

Lemma A.2. *Under the definitions of Theorem A.1*

$$W(\hat{\mu}_{\mathcal{S}}, \hat{\mu}_{\mathcal{T}}) \leq W(\hat{\mu}_{\mathcal{S}}, \hat{\mu}_{\mathcal{P}}) + W(\hat{\mu}_{\mathcal{P}}, \hat{\mu}_{\mathcal{T}}) \tag{6}$$

where $\hat{\mu}_{\mathcal{P}}$ is the GMM distribution learnt for source domain $\mathcal{S}$.

Proof. As W is a distance metric, the proof is an immediate application of the triangle inequality. □

Lemma A.3. *Let h be the hypothesis describing the multi-source model, and let h_k be the hypothesis learnt for a source domain k. If $e_{\mathcal{T}}(h)$ is the error function for hypothesis h on domain $\mathcal{T}$, then*

$$e_{\mathcal{T}}(h) \leq \sum_{k=1}^{n} w_k e_{\mathcal{T}}(h_k) \tag{7}$$

Proof. Let $p(X) = \sum_{k=1}^{n} w_k f_k(X)$ with $\sum w_k = 1, w_k > 0$ be the probabilistic estimate returned by our model for some input X, and let y be the label associated with this input. The proof for the Lemma proceeds as follows

$$\begin{aligned}
e_{\mathcal{T}}(h) = \mathbb{E}_{(X,y)\sim\mathcal{T}}\mathcal{L}_{ce}(p(X), \mathbb{1}_y) &= \mathbb{E}_{(X,y)\sim\mathcal{T}} - \log p(X)[y] \\
&= \mathbb{E}_{(X,y)\sim\mathcal{T}} - \log(\sum_{k=1}^{n} w_k f_k(X)[y]) \\
&\leq \mathbb{E}_{(X,y)\sim\mathcal{T}} \sum_{k=1}^{n} w_k(-\log f_k(X)[y]) \text{ By Jensen's Inequality} \\
&= \sum_{k=1}^{n} w_k \mathbb{E}_{(X,y)\sim\mathcal{T}}\mathcal{L}_{ce}(f_k(x), \mathbb{1}_y) \\
&= \sum_{k=1}^{n} w_k e_{\mathcal{T}}(h_k)
\end{aligned}$$

□

We now extend Theorem A.1 as follows

Theorem A.4. *Multi-Source unsupervised error bound (Theorem 5.1 from the main paper)*

Under the assumptions of our framework and using the definitions from Theorem A.1

$$e_{\mathcal{T}}(h) \leq \sum_{k=1}^{n} w_k(e_{\mathcal{S}_k}(h_k) + W(\hat{\mu}_{\mathcal{T}}, \hat{\mu}_{\mathcal{P}_k}) + W(\hat{\mu}_{\mathcal{P}_k}, \hat{\mu}_{\mathcal{S}_k}) + \sqrt{\big(2\log(\frac{1}{\xi})/\zeta\big)}\big(\sqrt{\frac{1}{N_k}} + \sqrt{\frac{1}{M}}\big) + e_{\mathcal{C}_k}(h_k^*)) \tag{8}$$

where $\mathcal{P}_k$ is the sample GMM distribution learnt for source domain k, N_K is the sample size of domain k, $\mathcal{C}_k$ is the combined error loss with respect to domain k, and h_k^* is the optimal model with respect to this loss.

Proof.

$$\begin{aligned}
e_{\mathcal{T}}(h) &\leq \sum_{k=1}^{n} w_k e_{\mathcal{T}}(h_k) \text{ From Lemma A.3} \\
&\leq \sum_{k=1}^{n} w_k(e_{\mathcal{S}_k}(h_k) + W(\hat{\mu}_{\mathcal{T}}, \hat{\mu}_{\mathcal{S}_k}) + \sqrt{\big(2\log(\frac{1}{\xi})/\zeta\big)}\big(\sqrt{\frac{1}{N_k}} + \sqrt{\frac{1}{M}}\big) + e_{\mathcal{C}_k}(h_k^*)) \text{ by Theorem A.1} \\
&\leq \sum_{k=1}^{n} w_k(e_{\mathcal{S}_k}(h_k) + W(\hat{\mu}_{\mathcal{T}}, \hat{\mu}_{\mathcal{P}_k}) + W(\hat{\mu}_{\mathcal{P}_k}, \hat{\mu}_{\mathcal{S}_k}) + \\
&\quad \sqrt{\big(2\log(\frac{1}{\xi})/\zeta\big)}\big(\sqrt{\frac{1}{N_k}} + \sqrt{\frac{1}{M}}\big) + e_{\mathcal{C}_k}(h_k^*)) \text{ by Lemma A.2}
\end{aligned}$$

□

A.2 Sliced Wasserstein Distance

As mentioned in the main body of the manuscript, the Sliced Wasserstein Distance is an approximation of optimal transport. Following the results in Rabin et al. (2011), the SWD acts as an estimate for the quadratic Wasserstein Distance (WD) between two distributions, by aggregating the tractable $1-$dimensional WD of L projections onto the unit hypersphere. In the context of our algorithm, the discrepancy measure $D(\cdot,\cdot)$ can be written in the form of SWD as follows:

$$D(g(\mathcal{T}), A_k) = \frac{1}{L}\sum_{l=1}^{L} |\langle g(X_{i_l}^t), \gamma_l\rangle - \langle X_{j_l}^a, \gamma_l\rangle|^2 \approx W_2(g(\mathcal{T}), A_k) \tag{9}$$