STEM: An approach to Multi-source Domain Adaptation with Guarantees

Van-Anh Nguyen[1], Tuan Nguyen[2], Trung Le[2], Quan Hung Tran[3], Dinh Phung[2,4]
[1]VNU - University of Science, Vietnam
[2]Department of Data Science and AI, Monash University, Australia
[3]Adobe Research, San Jose, CA, USA
[4]VinAI Research, Vietnam
vananhnt57@gmail.com, {tuan.ng, trunglm}@monash.edu,
qtran@adobe.com, dinh.phung@monash.edu

Abstract

Multi-source Domain Adaptation (MSDA) is more practical but challenging than the conventional unsupervised domain adaptation due to the involvement of diverse multiple data sources. Two fundamental challenges of MSDA are: (i) how to deal with the diversity in the multiple source domains and (ii) how to cope with the data shift between the target domain and the source domains. In this paper, to address the first challenge, we propose a theoretical-guaranteed approach to combine domain experts locally trained on its own source domain to achieve a combined multi-source teacher that globally predicts well on the mixture of source domains. To address the second challenge, we propose to bridge the gap between the target domain and the mixture of source domains in the latent space via a generator or feature extractor. Together with bridging the gap in the latent space, we train a student to mimic the predictions of the teacher expert on both source and target examples. In addition, our approach is guaranteed with rigorous theory offered insightful justifications of how each component influences the transferring performance. Extensive experiments conducted on three benchmark datasets show that our proposed method achieves state-of-the-art performances to the best of our knowledge.

1. Introduction

Recent advances in deep learning have enjoyed great success in performing visual learning tasks under the collection of massive annotated data [26, 64, 50, 54, 3]. However, directly transferring knowledge of a learned model, which is trained on a source domain, to a novel target domain can undesirably degrade its performance due to the existence of *domain and label shifts* [49]. To address these issues, a diverse range of approaches in domain adaptation (DA) has been proposed from shallow domain adaptation [45, 16, 5, 6] to deep domain adaptation [13, 32, 51, 12, 55, 9, 29, 41, 40]. While the conventional DA aims to transfer knowledge from a labeled source domain to an unlabeled target domain, in many real-world contexts, labeled data are collected from multiple domains, for example, images taken under different conditions (e.g., weather, poses, lighting conditions, distinct backgrounds, and etc) [70]. This has arisen a very practical and useful setting for transfer learning named multi-source domain adaptation (MSDA) in which we need to transfer knowledge from multiple distinct source domains to a single unlabeled target domain.

For multi-source domain adaptation, there exist two fundamental challenges: (i) how to deal with the diversity in the labeled source domains and (ii) how to cope with the domain shift between the target domain and the source domains. The first challenge makes it harder to train a single model that is expected to work well on multiple source domains due to the requirement to resolve diverge data complexity imposed on model training. To overcome this challenge, inspired by [36, 23], we propose combining domain experts into a multi-source teacher by mixing the domain expert predictions using the coefficients learned by a domain discriminator. Our rigorous theory demonstrates that the performance of this multi-source teacher expert predicting globally on the mixture source domains is at least better than that of the worst domain expert predicting locally on its domain (see Theorem 1). Therefore, if we can train qualified domain experts, their combination leads to another qualified expert with significantly broader coverage.

To address the second challenge, as suggested by Theorem 3, we employ a joint feature extractor that maps the target domain and the mixture of source domains into the same latent space with the help of adversarial learning. Furthermore, together with closing the divergence of the target domain and mixture of source domains on the latent space,

we train a target-domain student to imitate the multi-source teacher on both source and target examples while enforcing the clustering assumption [4] on the target-domain student to strengthen the student's generalization ability.

- We propose an approach named ***S**tudent-**T**eacher **E**nsemble **M**ulti-source Domain Adaptation* (STEM) with theoretical guarantees for multi-source domain adaptation. Not only driving us in devising our STEM, the rigorous theory developed provides us an insightful understanding of how each model component really influences the transferring performance.
- We conduct extensive experiments on three benchmark datasets including Digits-five, Office-Caltech10, and DomainNet. Experimental results show that our STEM achieves state-of-the-art performances on those three benchmark datasets. More specifically, for Digits-five and Office-Caltech10 datasets, our STEM wins the baselines on all pairs and surpasses the runner-up baselines by 3.2% and 1.5% on average, while for DomainNet dataset, our STEM wins the runner-up baseline on 5 out of 6 pairs and surpasses the runner-up baseline by 6.0% on average.

2. Related Work

2.1. Unsupervised Domain Adaptation

A variety of unsupervised domain adaptation (UDA) approaches have been successfully applied to generalize a model learned from labeled source domain to unlabeled novel target domain. Several existing methods based on discrepancy-based alignment to minimize a different discrepancy metric to close the gap between source and target domain [32, 59, 56, 68, 31]. Another branch of UDA has leveraged adversarial learning wherein generative adversarial networks [18, 42, 22, 8, 28] were employed to align source and target domain on feature-level [13, 58, 33, 43] or pixel-level [15, 2, 53, 66]. On the category-level, some approaches utilized dual classifier [52, 31], or domain prototype [63, 46, 65] to investigate the category relations across domains.

2.2. Multi-Source Domain Adaptation

The aforementioned UDA methods mainly consider single-source domain adaptation, which is less practical than multi-source domain adaptation. The fundamental study in [7, 36, 1] has shed light upon the wide applications of MSDA, such as in [11, 67]. Based on the above works, Hoffman et al. [23] gave strong theoretical guarantees for cross-entropy and other similar losses, which is a normalized solution for MSDA problems. Recently, Zhao et al. [70] deployed domain adversarial networks to align the target domain to source domains. Xu et al. [67] proposed a new model to deal with the *category shift*, which is the case where sources may not completely share their categories. Peng et al. [47] introduced a model that aligned moments of source and target feature distributions in latent space. A multi-source distilling model was proposed in [71] to fine-tune generator and classifier separately and utilized domain weight to aggregate target prediction. Finally, the work in [61] deployed a graph convolutional network to conduct domain alignment on the category-level.

3. Our Proposed Framework

3.1. Problem Setting

In this paper, we address the problem of multi-source domain adaptation in which we have K source domains with collected data and labels, and a single target domain with only collected data. We wish to transfer a model learned on labeled source domains to an unlabeled target domain. Let us denote the collected data and labels for the source domains by $\mathbb{D}_k^S = \left\{\left(\boldsymbol{s}\mathbf{x}_i^k, y_i^k\right)\right\}_{i=1}^{N_k^S}$ where k is the index of a source domain and label $y_i^k \in \{1, 2, ..., M\}$ with the number of classes M, and collected data without labels for the target domain $\mathbb{D}^T = \{\boldsymbol{t}\mathbf{x}_i\}_{i=1}^{N^T}$.

We further equip source domains with data distributions $\mathbb{P}_{1:K}^S$ whose density functions are $p_{1:K}^S(\mathbf{x})$. Also, we define $p_{1:K}^S(y \mid \mathbf{x})$ as the conditional distributions that assign labels to each data example $\mathbf{x}$ for the source domains. Regarding the target domain, we define its data space as $\mathcal{X}^T$, data distribution and density function as $\mathbb{P}^T$ and $p^T(\mathbf{x})$, respectively. We further define the conditional distribution that assigns labels for the target domain as $p^T(y \mid \mathbf{x})$.

Furthermore, we denote $\mathbb{D}$ as a joint distribution with density function $p(\mathbf{x}, y)$ used to generate data-label pairs (i.e., $(\mathbf{x}, y) \sim \mathbb{D}$). Note that for the sake of notion simplification, we overload the notion $\mathbb{D}$ to denote both joint distribution for generating data-label pairs and a training set sampled from this distribution. Let h be a classifier in which $h(\mathbf{x}, y)$ specifies the probability to assign the data example $\mathbf{x}$ to a class $y \in \{1, ..., M\}$ and $h(\mathbf{x}) = [h(\mathbf{x}, y)]_{y=1}^M$ is the prediction probability vector w.r.t. $\mathbf{x}$. We consider the loss function $\ell(h(\mathbf{x}), y)$ and define the general loss w.r.t. the data-label joint distribution $\mathbb{D}$ as follows:

$$\begin{aligned}\mathcal{L}(h, \mathbb{D}) &:= \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{D}}\left[\ell\left(h(\mathbf{x}), y\right)\right] \\ &= \int \ell\left(h(\mathbf{x}), y\right) p(\mathbf{x}, y)\, d\mathbf{x}dy.\end{aligned}$$

Finally, given a discrete distribution $\boldsymbol{\pi}$ over $\{1, ..., K\}$, we define $\mathbb{P}_{\boldsymbol{\pi}}^S := \sum_{k=1}^K \pi_k \mathbb{P}_k^S$ which is a mixture of $\mathbb{P}_{1:K}^S$ with density function $p_{\boldsymbol{\pi}}^S(\mathbf{x}) = \sum_{k=1}^K \pi_k p_k^S(\mathbf{x})$ and $\mathbb{D}_{\boldsymbol{\pi}}^S := \sum_{k=1}^K \pi_k \mathbb{D}_k^S$ with density function $p_{\boldsymbol{\pi}}^S(\mathbf{x}, y) = \sum_{k=1}^K \pi_k p_k^S(\mathbf{x}, y)$. Moreover, the mixing proportion $\boldsymbol{\pi}$ can be the uniform distribution $[\frac{1}{K}, ..., \frac{1}{K}]$ or proportional to the

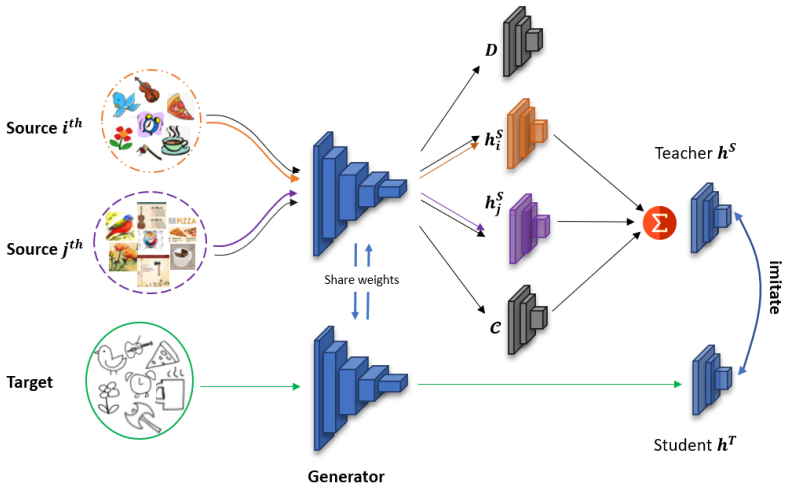

Figure 1. Overall framework of STEM for multi-source domain adaptation, which consists of cooperative agents, namely a multi-source teacher expert h^S and a target-domain student h^T. Our model is trained to implement simultaneously two tasks: (i) achieving the teacher expert h^S by first training to obtain domain experts $h^S_{1:K}$ using their labels (orange and purple arrows), and then output the teacher h^S using a weighted ensembling strategy (black arrows) and (ii) training the student h^T with the aim to mimic the prediction of its teacher expert h^S (green arrows) with the support of D to close the gap between the mixture of source data distributions and the target distribution on the latent space.

number of training examples in the source domains (i.e., $N^S_{1:K}$).

3.2. Overall Framework of STEM

Figure 1 illustrates the overall framework of our STEM. Source and target domains are mapped to a latent space via a shared generator or feature extractor G. On the latent space, we train the domain experts $h^S_{1:K}$ and a source domain discriminator $\mathcal{C}$ for which we can combine them to achieve a multi-source teacher expert h^S. Particularly, the source domain discriminator is trained to distinguish the source domains, hence rendering the probabilities to assign an example to the source domains. Therefore, given a source example, the domain experts more relevant to this example contribute more to the final decision. Furthermore, we develop a theory to demonstrate that the multi-source teacher expert h^S can predict well on the mixture of source domains with the performance at least better than the worst domain expert on its source domain. Note that to support the source domain discriminator $\mathcal{C}$ to do its task, the latent representations from the individual source domains are encouraged to be separate, hence increasing their coverage on the latent space. Meanwhile, with the assistance of adversarial learning framework [18], we train G with the support of a discriminator D to bridge the gap between the target distribution and the mixture of source distributions, which enables the multi-source teacher expert h^S to transfer its knowledge to predict well the target examples. Moreover, inspired by the principle of knowledge distillation [21] in which we can conduct a student to distill knowledge and outperform its teacher, we train an additional target-domain student h^T to mimic the predictions of the multi-source teacher expert h^S on the target and source examples. Finally, we develop a rigorous theory to quantify the loss in performance for this imitating.

3.3. Ensemble based Teacher Expert

In what follows, we present how to conduct the multi-source teacher expert h^S, an ensemble expert which leverages knowledge of domain experts. Particularly, using the labeled source training sets $\mathbb{D}^S_{1:K}$, we can train qualified domain expert classifiers $h^S_{1:K}$ with good generalization capacity (i.e., $\mathcal{L}\left(h^S_k, \mathbb{D}^S_k\right) \leq \epsilon$ for some small $\epsilon > 0$). The next arising question is how to combine those domain experts to achieve a multi-source teacher expert h^S that can work well on $\mathbb{D}^S_{\boldsymbol{\pi}}$ (i.e., $\mathcal{L}\left(h^S, \mathbb{D}^S_{\boldsymbol{\pi}}\right) \leq \epsilon$). Inspired by [36, 23], we leverage the domain experts to achieve a more powerful multi-source teacher expert by a weighted ensembling as follows:

$$h^S(\mathbf{x}, y) = \sum_{k=1}^{K} \frac{\pi_k p^S_k(\mathbf{x}, y)}{\sum_{j=1}^{K} \pi_j p^S_j(\mathbf{x}, y)} h^S_k(\mathbf{x}, y), \quad (1)$$

where $y \in \{1, 2, ..., M\}$, and $h^S_k(\mathbf{x}, y)$ and $h^S(\mathbf{x}, y)$ specify the y-th values of $h^S_k(\mathbf{x})$ and $h^S(\mathbf{x})$ respectively.

The following theorem shows that the multi-source domain teacher expert h^S can work well on the mixture joint distribution $\mathbb{D}^S_{\boldsymbol{\pi}}$. More specifically, it works better than the worst domain expert on its source domain, hence if each domain expert is an ϵ-qualified classifier (i.e., $\mathcal{L}\left(h^S_k, \mathbb{D}^S_k\right) \leq \epsilon$), the multi-source teacher expert h^S is also an ϵ-qualified classifier (i.e., $\mathcal{L}\left(h^S, \mathbb{D}^S_{\boldsymbol{\pi}}\right) \leq \epsilon$).

Theorem 1. *If ℓ is a convex function, the following statements hold true (the proof of this theorem is adapted from a proof in [36, 23]):*

i) $\mathcal{L}\left(h^S, \mathbb{D}^S_{\boldsymbol{\pi}}\right) \leq \max_{1\leq k\leq K} \mathcal{L}\left(h^S_k, \mathbb{D}^S_k\right)$.

ii) If each domain expert is an ϵ-qualified classifier (i.e., $\mathcal{L}\left(h^S_k, \mathbb{D}^S_k\right) \leq \epsilon$), the multi-source teacher expert h^S is also an ϵ-qualified classifier (i.e., $\mathcal{L}\left(h^S, \mathbb{D}^S_{\boldsymbol{\pi}}\right) \leq \epsilon$).

So far the question of how to weight the domain experts $h^S_{1:K}$ to form multi-source teacher expert h^S is still left unanswered. Moreover, [23] proposed using DC-programming (i.e., difference of convex) [10] for estimating weights. However, this approach seems to be overly complicated and there is not any convincing evidence of the effectiveness of this work for real-world datasets (i.e., the reported performance for the Office-31 dataset in the context of the standard multiple source setting without any transfer learning is only approximately 84.7%). In this paper, we propose a new approach to weight the domain experts, which is hinted from the following theoretical observation. Assume that we have K distributions $\mathbb{R}_{1:K}$ with density functions $r_{1:K}(\mathbf{z})$. We form a joint distribution $\mathbb{D}$ of a data instance $\mathbf{z}$ and label $t \in \{1, ..., K\}$ by sampling an index $t \sim \text{Cat}(\boldsymbol{\pi})$ (i.e., the categorical distribution w.r.t. $\boldsymbol{\pi}$), sampling $\mathbf{x} \sim \mathbb{R}_t$, and collecting $(\mathbf{z}, t)$ as a sample from $\mathbb{D}$. With this equipment, we have the following proposition.

Proposition 2. *If we train a source domain discriminator $\mathcal{C}$ to classify samples from the joint distribution $\mathcal{D}$ using the cross-entropy loss (i.e., $CE(\cdot,\cdot)$), the optimal source domain discriminator $\mathcal{C}^*$ defined as*

$$\mathcal{C}^* = argmin_{\mathcal{C}} \mathbb{E}_{(\mathbf{z},t)\sim\mathcal{D}}\left[CE\left(\mathcal{C}\left(\mathbf{z}\right), t\right)\right]$$

satisfies $\mathcal{C}^*\left(\mathbf{z}\right) = \left[\frac{\pi_k r_k(\mathbf{z})}{\sum_j \pi_j r_j(\mathbf{z})}\right]_{k=1}^{K}$.

Proposition 2 suggests us a way to compute the weights of the domain experts in Eq. (1) in which for a given $y = m$, the distributions $p^S_{1:K}(\mathbf{x}, y = m)$ play roles of $r_{1:K}(\mathbf{z})$ where $\mathbf{z} = (\mathbf{x}, y = m)$. More specifically, for each $m \in \{1, ..., M\}$, we sample $t \sim Cat(\boldsymbol{\pi})$, then sample $(\mathbf{x}, y = m)$ from $p^S_t(\mathbf{x}, y = m)$, and train a source domain discriminator $\mathcal{C}_m(\mathbf{x}, y = m)$ (i.e., only consider $(\mathbf{x}, y)$ in which $\mathbf{x}$ has label $y = m$) to distinguish the source domain of $(\mathbf{x}, y = m)$. We finally use $\mathcal{C}_m(\mathbf{x}, y = m)$ to estimate the weights of the domain experts. In addition, to conveniently train the source domain discriminators $\mathcal{C}_m$, we share their parameters, hence having an unique $\mathcal{C}$ that receives a pair $(\mathbf{x}, y)$ and predicts its source domain t. Therefore, we obtain the expert teacher

$$h^S(\mathbf{x}, y) = \sum_{k=1}^{K} \mathcal{C}(\mathbf{x}, y, k)\, h^S_k(\mathbf{x}, y). \tag{2}$$

To leverage the information of multiple source domains and encourage learning multiple-source domain-invariant representations for transfer learning in the sequel, we employ a feature extractor G to map multiple source domains and the target domain to a latent space. The domain experts $h^S_{1:K}$ and the source domain discriminator are trained on the latent space. The formula in Eq. (2) is rewritten as:

$$h^S(G(\mathbf{x}), y) = \sum_{k=1}^{K} \mathcal{C}(G(\mathbf{x}), y, k)\, h^S_k(G(\mathbf{x}), y).$$

At the outset, we want to emphasize that our principle to learn representations is different from that in some recent works in MSDA, typically [47]. In [47], the moment distance was used to force the representations of multiple source domains to be identical in the latent space, while ours encourages the representations of the individual source domains to be separate so that the source domain discriminator $\mathcal{C}$ can distinguish them more effectively. By this way, we increase the coverage of the representations from the multiple source domains, which makes the representations from the target domain more conveniently to adapt the source representation in the transfer learning phase.

3.4. Performance of The Multi-source Teacher Expert on the Target Domain

We have possessed a qualified multi-source teacher expert h^S that expects to predict well data examples sampled from $\mathcal{D}^S_{\boldsymbol{\pi}}$ (i.e., a mixture of $\mathcal{D}^S_{1:K}$) as indicated in Theorem 1. It is natural to ask the question of the factors that influence the performance of h^S when *predicting on the target joint distribution* $\mathbb{D}^T$. The following theorem answers this question.

Theorem 3. *If ℓ is a convex function and upper-bounded by a positive constant L, the general loss $\mathcal{L}\left(h^S, \mathbb{D}^T\right)$ is upper-bounded by:*

i) $A\left[\max_k \mathcal{L}\left(h^S_k, \mathbb{D}^S_k\right) + L \max_k \mathbb{E}_{\mathbb{P}^S_k}\left[\left\|\Delta p_k\left(y \mid \mathbf{x}\right)\right\|_1\right]\right]^{\frac{\alpha-1}{\alpha}}$

where $A = \exp\left\{R^\alpha\left(\mathbb{P}^T \| \mathbb{P}^S_{\boldsymbol{\pi}}\right)\right\}^{\frac{\alpha-1}{\alpha}} L^{\frac{1}{\alpha}}$ *in which* $R^\alpha\left(\mathbb{P}^T \| \mathbb{P}^S_{\boldsymbol{\pi}}\right)$ *represents the Rényi divergence between those distributions and* $\Delta p_k\left(y \mid \mathbf{x}\right) := \left[\left|p^S_k\left(y = m \mid \mathbf{x}\right) - p^T\left(y = m \mid \mathbf{x}\right)\right|\right]_{m=1}^{M}$ *represents the label shift between the labeling assignment mechanisms of an individual source domain and target domain.*

ii) $A\left[\epsilon + L \max_k \mathbb{E}_{\mathbb{P}^S_k}\left[\left\|\Delta p_k\left(y \mid \mathbf{x}\right)\right\|_1\right]\right]^{\frac{\alpha-1}{\alpha}}$ *provided that* $\mathcal{L}\left(h^S_k, \mathbb{D}^S_k\right) \leq \epsilon, \forall k = 1, ..., K$.

We now interpret Theorem 3 which lays foundation for us to devise our STEM in the sequel. The general loss of interest $\mathcal{L}\left(h^S, \mathbb{D}^T\right)$ is upper-bounded by the construction of three terms, each of which has a specific meaning.

(i) The *expert-loss* term $\max_k \mathcal{L}\left(h_k^S, \mathbb{D}_k^S\right)$ represents the worst general loss of the domain experts $h_{1:K}^S$. Minimizing this term implies training the domain experts to work well on their domains.

(ii) The *label-shift* term $\mathbb{E}_{\mathbb{P}_k^S}\left[\left\|\Delta p_k\left(y \mid \mathbf{x}\right)\right\|_1\right]$ where $\Delta p_k\left(y \mid \mathbf{x}\right) := \left[\left|p_k^S\left(y=m \mid \mathbf{x}\right) - p^T\left(y=m \mid \mathbf{x}\right)\right|\right]_{m=1}^M$ specifies the label shift indicating the divergence of the ground-truth target labeling function and the ground-truth source labeling function on a source domain. This term is constant and reflects the characteristics of collected data.

(iii) The *domain-shift* term $R^\alpha\left(\mathbb{P}^T \| \mathbb{P}_{\boldsymbol{\pi}}^S\right)$ expresses the data shift between the mixture source distribution $\mathbb{P}_{\boldsymbol{\pi}}^S$ and the target distribution $\mathbb{P}^T$.

The observation in (iii) hints us using adversarial learning framework [18] to bridge the gap between the representations of the multiple source domains and the target domain on the latent space using an additional discriminator D (see Section 3.6.3).

3.5. Target-Domain Student

The multi-source teacher expert h^S is guaranteed to work well on the mixture of source data distributions $\mathbb{P}_{\boldsymbol{\pi}}^S$, while the generator G with the support of a discriminator D in adversarial learning framework [18] aims to close the discrepancy gap between the mixture of source data distributions $\mathbb{P}_{\boldsymbol{\pi}}^S$ and the target distribution $\mathbb{P}^T$ on the latent space. Therefore, the multi-source teacher expert h^S is expected to work well on the target domain. However, the ill-posed problem of GAN (e.g., the mode collapsing problem) could occur during training, so that using directly h^S to predict target samples in latent space is not the best solution, which motivates us to design the student network h^T. Particularly, in Figure 2a, GAN works perfectly, hence both h^S and h^T work equally well. In another case, since GAN does not mix up well the class 1 and 2 of source and target domains (Figure 2b), h^S predicts well on the source domain but unwell on the target one. By enforcing the clustering assumption on h^T [4] (i.e., h^T preserves clusters and is encouraged to give the same prediction for source and target data on the same cluster), the possible ill-posed training of GAN is mitigated. Additionally, inspired by the principle of knowledge distillation [21] in which we can conduct a student to distill knowledge and outperform its teacher, we propose to train h^T which aims to mimic the predictions of the teacher h^S on the mixture source and target domains. This also helps to mitigate the negative impact from possible ill-posed training of GAN, while offering us an opportunity to apply regularization techniques such as VAT [37] and label smoothing [38] to h^T. We note that in our framework, it is hard to apply those regularization techniques directly to the teacher h^S, but it is convenient to apply to h^T. Indeed, we decide to apply VAT to h^T(see Section 3.6.2) and observe its superiority to the teacher in terms of predictive performance (see Section 4.2.4).

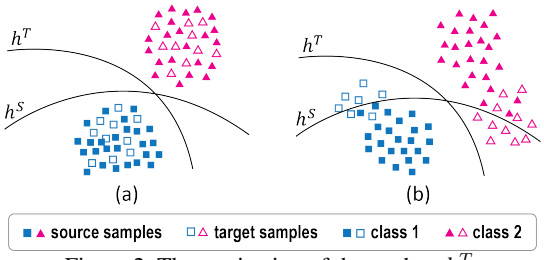

Figure 2. The motivation of the student h^T.

3.6. Training Procedure of Our STEM

3.6.1 Training Multi-Source Teacher Expert

To work out the multi-source teacher expert h^S, we simultaneously train domain experts $h_{1:K}^S$ on the labeled training sets $\mathcal{D}_{1:K}^S$ and the source domain discriminator $\mathcal{C}$ to offer the weights for leveraging the domain experts. We propose two workarounds to train $\mathcal{C}$ and ensemble the domain experts. Basically, we minimize: $\sum_{k=1}^K \mathcal{L}_k^{ie} + \alpha\mathcal{L}^{\mathcal{C}}$, where $\alpha > 0$ and consider two variants.

Theoretical oriented version. For the theoretical oriented version, we feed $\left(G\left(\mathbf{x}\right), y\right)$ to source domain discriminator $\mathcal{C}$ with the aim to predict the data source index of $\mathbf{x}$

$$\mathcal{L}_k^{ie} = \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{D}_k^s}\left[CE\left(h_k^S\left(G\left(\mathbf{x}\right)\right), y\right)\right],$$

$$\mathcal{L}^{\mathcal{C}} = \mathbb{E}_{(\mathbf{x},y,t)\sim\mathcal{D}}\left[CE\left(\mathcal{C}\left(G\left(\mathbf{x}\right), y\right), t\right)\right],$$

$$h^S\left(G\left(\mathbf{x}\right), y\right) = \sum_{k=1}^K \mathcal{C}\left(G\left(\mathbf{x}\right), y, k\right) h_k^S\left(G\left(\mathbf{x}\right), y\right),$$

where $\mathcal{D}$ is formed by sampling $t \sim Cat\left(\boldsymbol{\pi}\right)$ and $(\mathbf{x}, y) \sim \mathbb{D}_t^S$ and $CE\left(\cdot,\cdot\right)$ is the cross-entropy loss.

Simplified version. For the simplified version, instead of feeding $\left(G\left(\mathbf{x}\right), y\right)$ to the source domain discriminator $\mathcal{C}$, we only feed $G\left(\mathbf{x}\right)$ to this discriminator with the aim to predict the data source index of $\mathbf{x}$

$$\mathcal{L}_k^{ie} = \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{D}_k^s}\left[CE\left(h_k^S\left(G\left(\mathbf{x}\right)\right), y\right)\right],$$

$$\mathcal{L}^{\mathcal{C}} = \mathbb{E}_{(\mathbf{x},t)\sim\mathcal{D}}\left[CE\left(\mathcal{C}\left(G\left(\mathbf{x}\right)\right), t\right)\right],$$

$$h^S\left(G\left(\mathbf{x}\right), y\right) = \sum_{k=1}^K \mathcal{C}\left(G\left(\mathbf{x}\right), k\right) h_k^S\left(G\left(\mathbf{x}\right), y\right),$$

where $\mathcal{D}$ is formed by sampling $t \sim Cat\left(\boldsymbol{\pi}\right)$ and $\mathbf{x} \sim \mathbb{P}_t^S$. According to our ablation study in Section 4.2.2, the simplified version performs slightly better than the theoretical oriented version, while easier to train due to its simplicity. Therefore, we stick with the simplified version and detail the training of other components based on this version.

3.6.2 Training Target-Domain Student

We train the target domain student h^T to mimic the teacher h^S on the predictions for target and mixture of source examples using the following loss:

$$\mathcal{L}^m = \mathbb{E}_{\mathbb{P}^S_{\boldsymbol{\pi}}}\left[\ell\left(h^T\left(G\left(\mathbf{x}\right)\right), h^S\left(G\left(\mathbf{x}\right)\right)\right)\right] + \mathbb{E}_{\mathbb{P}^T}\left[\ell\left(h^T\left(G\left(\mathbf{x}\right)\right), h^S\left(G\left(\mathbf{x}\right)\right)\right)\right].$$

Moreover, Virtual adversarial training (VAT) [37] in conjunction with minimizing entropy of prediction [19] with the aim to ensuring the clustering assumption [4] has been applied successfully to UDA [55, 27, 44]. Inspired by this success, we propose minimizing

$$\mathcal{L}^{clus} = \mathcal{L}^{ent} + \mathcal{L}^{vat},$$

where $\mathbb{H}$ is the entropy,

$$\mathcal{L}^{ent} = \mathbb{E}_{\mathbb{P}^T}\left[\mathbb{H}\left(h^T\left(G\left(\mathbf{x}\right)\right)\right)\right],$$

$$\mathcal{L}^{vat} = \mathbb{E}_{\mathbf{x}\sim\mathbb{P}^T}\left[\max_{\mathbf{x}':\|\mathbf{x}'-\mathbf{x}\|<\theta} D_{KL}\left(h^T\left(G\left(\mathbf{x}\right)\right), h^T\left(G\left(\mathbf{x}'\right)\right)\right)\right]$$

with which D_{KL} represents a Kullback-Leibler divergence and θ is very small positive number. The total loss to train the student h^T is as follows:

$$\mathcal{L}^{stu} = \mathcal{L}^m + \beta\mathcal{L}^{clus},$$

where $\beta > 0$ is a parameter.

3.6.3 Training Discriminator

The discriminator D is employed to distinguish the examples from the mixture of source data distributions $\mathbb{P}^S_{\boldsymbol{\pi}}$ and the target distribution $\mathbb{P}^T$. The loss to train D is as follows:

$$\mathcal{L}^d = -\mathbb{E}_{\mathbb{P}^S_{\boldsymbol{\pi}}}\left[\log D(G(\mathbf{x}))\right] - \mathbb{E}_{\mathbb{P}^T}\left[\log\left(1 - D(G(\mathbf{x}))\right)\right].$$

3.6.4 Training Generator

We train the generator G to bring the target examples to the mixture of source examples and provide appropriate representations for learning h^S and h^T with the following loss:

$$\sum_{k=1}^{K}\mathcal{L}^{ie}_k + \alpha\mathcal{L}^{\mathcal{C}} + \mathcal{L}^{stu} - \gamma\mathcal{L}^d, \tag{3}$$

where $\gamma > 0$ is are parameters.

3.6.5 Overall Training

We simultaneously update $G, \mathcal{C}, h^S, h^T$ by minimizing:

$$\sum_{k=1}^{K}\mathcal{L}^{ie}_k + \alpha\mathcal{L}^{\mathcal{C}} + \mathcal{L}^m + \beta\mathcal{L}^{clus} - \gamma\mathcal{L}^d. \tag{4}$$

We alternatively update D by minimizing $\mathcal{L}^d$. In addition, the pseudocode of our STEM is presented in Algorithm 1.

Algorithm 1 Pseudocode for training our STEM.

Input: Sources $\mathbb{D}^S_k = \left\{\left(\boldsymbol{sx}^k_i, y^k_i\right)\right\}_{i=1}^{N^S_k}$, target $\mathbb{D}^T = \left\{\boldsymbol{tx}_i\right\}_{i=1}^{N^T}$.
Output: Classifiers h^S, h^T, source discriminator $\mathcal{C}$, generator G.

```
for epoch in epochs do
    for iter in iter_per_epoch do
        Sample minibatches of sources {(sx_i^k, y_i^k)}_{i=1}^m and target {tx_i}_{i=1}^m.
        Update G, C, h^S, h^T according to (4).
        Update D by minimizing L^d.
    end for
end for
```

4. Experiments

4.1. Experiments on Benchmark Datasets

This section describes our experiment settings. We compare our STEM with the state-of-the-art baselines for MSDA on three benchmark datasets: Digits-five, Office-Caltech10, and DomainNet to demonstrate its merits.

4.1.1 Experimental setup

Implementation detail. In the experiments, we use Adam optimizer ($\beta_1 = 0.5, \beta_2 = 0.999$) [25] with Polyak averaging [48] for Digits-five and Office-Caltech10, and the learning rate is set to 2×10^{-4} and 10^{-4}, respectively. For DomainNet, we apply Stochastic Gradient Decent (SGD) [57] (learning rate = 5×10^{-2}, momentum $= 0.9$, decay rate = 5×10^{-4}) to optimize the model.

For STEM, the trade-off hyper-parameter α is fixed to 1.0 in all experiments, while the parameters (β, γ) (with a recommended range of $[10^{-4}, 1]$ for each parameter) are set to $(0.1, 0.1)$ for Digit-five, $(0.01, 0.1)$ for Office-Caltech10, and $(10^{-4}, 10^{-4})$ for DomainNet.

Performance comparison. Following the previous work [61], we conduct the experiments to evaluate the model performance with the MSDA standards: (1) *Single best*: the highest classification accuracy among single-source domain adaptation results; (2) *Source combine*: the result on single-source domain adaptation where the source domain is a combination of multiple domains; (3) *Multi-source*: the evaluation of the adaptation from multiple source domains to the target domain.

4.1.2 Experiment Results on Digits-five

Digits-five contains five common digit-datasets: MNIST [30], Synthetic Digits [14], MNISTM [14], SVHN [39], and USPS [24]. This is a benchmark dataset in MSDA, with ten classes corresponding to the digits ranging from 0 to 9 in

Standard	Methods	→**mm**	→**mt**	→**up**	→**sv**	→**sy**	Avg
Single Best	Source-only	59.2	97.2	84.7	77.7	85.2	80.8
	DAN [32]	63.8	96.3	94.2	62.5	85.4	80.4
	CORAL [56]	62.5	97.2	93.5	64.4	82.8	80.1
	DANN [14]	71.3	97.6	92.3	63.5	85.4	82.0
	ADDA [58]	71.6	97.9	92.8	75.5	86.5	84.8
Source Combine	Source-only	63.4	90.5	88.7	63.5	82.4	77.7
	DAN [32]	67.9	97.5	93.5	67.8	86.9	82.7
	DANN [14]	70.8	97.9	93.5	68.5	87.4	83.6
	JAN [35]	65.9	97.2	95.4	75.3	86.6	84.1
	ADDA [58]	72.3	97.9	93.1	75.0	86.7	85.0
	MCD [52]	72.5	96.2	95.3	78.9	87.5	86.1
Multi-Source	MDAN [70]	69.5	98.0	92.4	69.2	87.4	83.3
	DCTN [67]	70.5	96.2	92.8	77.6	86.8	84.8
	M^3SDA [47]	72.8	98.4	96.1	81.3	89.6	87.7
	MDDA [71]	78.6	98.8	93.9	79.3	89.7	88.1
	CMSS [69]	75.3	99.0	97.7	88.4	93.7	90.8
	LtC-MSDA [61]	85.6	99.0	98.3	83.2	93.0	91.8
	STEM (ours)	**89.7**	**99.4**	**98.4**	**89.9**	**97.5**	**95.0**

Table 1. Classification accuracy (%) on Digits-five.

each domain. In each experiment on Digits-five, one domain will be chosen as the target domain and the rest as the source domains.

In Table 1, we report the performance of our STEM compared with the baselines. Our STEM outperforms the baselines on all transfer tasks. As far as we know, LtC-MSDA [61] is the current state-of-the-art on Digits-five. Compared to this baseline, our STEM significantly surpasses some transfer tasks, i.e., →**mm**, →**sv**, and →**sy** by sizeable margins of 4.1%, 6.7%, and 4.5% respectively and rank the first on average with a significant gap of 3.2%.

4.1.3 Experimental Results on Office-Caltech10

Office-Caltech10 [17] consists of four domains: Amazon (**A**), Caltech (**C**), DSLR (**D**), and Webcam (**W**). There are ten categories in each domain, and the total number of images is 2,533. In this experiment, we split the training and testing set with a ratio of 80% and 20%, respectively, and use ResNet-101 [20] pre-trained on ImageNet as a backbone.

In Table 2, we present the results of STEM and the baselines. Overall, it can be seen that our STEM surpasses the baselines in all four settings and achieves 98.2% on average. Since the baselines already achieve impressive performances on all adaptation tasks, it is hard to gain significant improvements. However, on two adaptation tasks (i.e., →**W** and →**D**), our model yields impressive performances with two perfect scores of 100%, while STEM also achieves remarkable improvements on the other tasks.

Standard	Methods	→**W**	→**D**	→**C**	→**A**	Avg
Source Combine	Source-only	99.0	98.3	87.8	86.1	92.8
	DAN [32]	99.3	98.2	89.7	94.8	95.5
Multi-Source	Source-only	99.1	98.2	85.4	88.7	92.9
	DAN [32]	99.5	99.1	89.2	91.6	94.8
	DCTN [67]	99.4	99.0	90.2	92.7	95.3
	JAN [35]	99.4	99.4	91.2	91.8	95.5
	MEDA [62]	99.3	99.2	91.4	92.9	95.7
	MCD [52]	99.5	99.1	91.5	92.1	95.6
	M^3SDA [47]	99.5	99.2	92.2	94.5	96.4
	CMSS [69]	99.6	99.3	93.7	96.6	97.2
	STEM (ours)	**100**	**100**	**94.2**	**98.4**	**98.2**

Table 2. Classification accuracy (%) on Office-Caltech10 dataset.

4.1.4 Experimental Results on DomainNet

DomainNet was first introduced in [47] and has become the most challenging dataset in MSDA. It consists of around 0.6 million images of 345 categories from 6 domains: *clipart* (clp), *infograph* (inf), *quickdraw* (qdr), *real* (rel) and *sketch* (skt). Prominently, the high number of classes and enormous noise in this dataset makes it challenging to gain satisfactory performances even when training and testing for supervised classification tasks in an individual domain, especially the *infograph* domain. Moreover, a significant difference in the distribution of each domain causes the domain shift problem when transferring knowledge. For all experiments on this dataset, we utilize ResNet-101 [20] pre-trained on ImageNet as the backbone.

We compare STEM with the current state-of-the-art method which is LtC-MSDA [61]. As shown in Table 3, our STEM exceeds LtC-MSDA on 5 out of 6 transfer tasks with significant improvements of 8.9% on →**clp** task, 9.4% on →**qdr** task, and 6.5% on →**rel** task. Averagely, STEM also yields an impressive improvement of 6.0%.

4.2. Ablation Study

4.2.1 Latent Space Visualization

The crucial factors for the success of our STEM include (i) the mix-up of target domain and the mixture of source domains in the latent space and (ii) the target examples are located in their matching classes in the source domains. To visually demonstrate why STEM can achieve good performances, we utilize t-SNE [60] to visualize the representations of target and source examples in the latent space. It is noticeable that in Figure 3, we visualize the case in which the target domain is USPS and the rest serves as source domains. As shown in Figure 3 (Left) wherein we visualize the mixture of source domains and the target domain when the model is trained with source domains only. In Figure 3 (Right), we show how accurately the target examples match the classes in the source domains when training the model with STEM approach. It is evident that our STEM forms

Standard	Methods	→clp	→inf	→pnt	→qdr	→rel	→skt	Avg
Single Best	Source-only	39.6	8.2	33.9	11.8	41.6	23.1	26.4
	DAN [32]	39.1	11.4	33.3	16.2	42.1	29.7	28.6
	RTN [34]	35.3	10.7	31.7	13.1	40.6	26.5	26.3
	JAN [35]	35.3	9.1	32.5	14.3	43.1	25.7	26.7
	DANN [14]	37.9	11.4	33.9	13.7	41.5	28.6	27.8
	ADDA [58]	39.5	14.5	29.1	14.9	41.9	30.7	28.4
	MCD [52]	42.6	19.6	42.6	3.8	50.5	33.8	32.2
Source Combine	Source-only	47.6	13.0	38.1	13.3	51.9	33.7	32.9
	DAN [32]	45.4	12.8	36.2	15.3	48.6	34.0	32.1
	RTN [34]	44.2	12.6	35.3	14.6	48.4	31.7	31.1
	JAN [35]	40.9	11.1	35.4	12.1	45.8	32.3	29.6
	DANN [14]	45.5	13.1	37.0	13.2	48.9	31.8	32.6
	ADDA [58]	47.5	11.4	36.7	14.7	49.1	33.5	32.2
	MCD [52]	54.3	22.1	45.7	7.6	58.4	43.5	38.5
Multi-Source	MDAN [70]	52.4	21.3	46.9	8.6	54.9	46.5	38.4
	DCTN [67]	48.6	23.5	48.8	7.2	53.5	47.3	38.2
	M^{3}SDA [47]	58.6	26.0	52.3	6.3	62.7	49.5	42.6
	MDDA [71]	59.4	23.8	53.2	12.5	61.8	48.6	43.2
	CMSS [69]	64.2	28.0	53.6	16.0	63.4	53.8	46.5
	LtC-MSDA [61]	63.1	**28.7**	56.1	16.3	66.1	53.8	47.4
	STEM (ours)	**72.0**	28.2	**61.5**	**25.7**	**72.6**	**60.2**	**53.4**

Table 3. Classification accuracy (%) on DomainNet dataset.

Figure 3. The t-SNE visualization of the transfer task →**up** with label and domain information in two settings: *Source-only* (left) and our *STEM* (right). Each color denotes a class while the circle and triangle markers represent the mixture of source and target data respectively.

source domains and target domain into the same clusters and the target examples can find their matching classes in the source domains, hence the label shift is mitigated. This explains the qualified performances of our STEM.

4.2.2 Simplified and Theoretical-Oriented Domain Discriminator $\mathcal{C}$

We conduct an ablation study to compare two variants of the domain discriminator $\mathcal{C}$: theoretical oriented and simplified versions (see Section 3.6.1). As shown in Table 4, the simplified variant performs better than the theoretical oriented one. We conjecture that this is because the simplified variant still keeps the principal spirit of the theoretically oriented one, while much easier to train due to its simplicity. Therefore, we select the simplified variant in all experiments.

Method	→mm	→mt
Theoretical $\mathcal{C}$	86.8	99.1
Simplified $\mathcal{C}$	89.7	99.4

Table 4. Comparison of the theoretical oriented and simplified version of the proposed method

$\mathcal{L}^{vat}$	$\mathcal{L}^{ent}$	→mm	→up
		83.04	96.86
✓		86.25	96.11
	✓	86.82	97.11
✓	✓	89.71	98.42

Table 5. Ablation study for the affection of VAT and entropy term.

4.2.3 Clustering Assumption Effect

We now speculate the effect of VAT and conditional entropy terms on our model performance. According to Table 5, adding $\mathcal{L}^{vat}$ (first row) or $\mathcal{L}^{ent}$ (second row) alone improves the performance, while combining these two losses (third row) even boosts the performance further.

Component	Digit-five	Office-Caltech10	DomainNet
h^S	92.7	97.9	51.6
h^T	95.0	97.9	53.4

Table 6. The comparison of teacher and student performance.

4.2.4 Teacher and Student Performances

We observe that the performance of the student h^T totally depends on that of the teacher h^S. In what follows, we compare the performance of the teacher and student on the target domain. We report the average of the *teacher* and *student*'s accuracy scores for all transfer tasks regarding each dataset. As shown in Table 6, the student outperforms its teacher except for Office-Caltech10 dataset. This totally makes sense because the student not only strictly imitates its teacher, but also is strengthened the generalization ability by enforcing the clustering assumption (see Section 3.6.2).

5. Conclusion

In this paper, we propose Student-Teacher Ensemble Multi-source Domain Adaptation (STEM) for multi-source domain adaptation. Our approach gives strong theoretical guarantees and provides an insightful understanding of how each model component really influences the transferring performance. Experiments conducted on three benchmark datasets, including Digits-five, Office-Caltech10, and DomainNet, demonstrate that our STEM achieves state-of-the-art performances to the best of our knowledge.

Acknowledgements

This work was supported by the US Air Force grants FA2386-19-1-4040 and FA9550-19-S-0003.

References

[1] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Mach. Learn.*, 79(1-2):151–175, May 2010. 2.2

[2] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3722–3731, 2017. 2.1

[3] Z. Cao, T. Simon, S. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1

[4] O. Chapelle and A. Zien. Semi-supervised classification by low density separation. In *AISTATS*, volume 2005, pages 57–64. Citeseer, 2005. 1, 3.5, 3.6.2

[5] N. Courty, R. Flamary, A. Habrard, and A. Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. In *Advances in Neural Information Processing Systems*, pages 3730–3739, 2017. 1

[6] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1853–1865, 2017. 1

[7] K. Crammer, M. Kearns, and J. Wortman. Learning from multiple sources. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 321–328. MIT Press, 2007. 2.2

[8] N. Dam, Q. Hoang, T. Le, T. D. Nguyen, H. Bui, and D. Phung. Three-player wasserstein gan via amortised duality. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 2202–2208. International Joint Conferences on Artificial Intelligence Organization, 7 2019. 2.1

[9] B. B. Damodaran, B. Kellenberger, R. Flamary, D. Tuia, and N. Courty. Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In *Computer Vision - ECCV 2018*, pages 467–483. Springer, 2018. 1

[10] T. P. Dinh and T. H. A. Le. Convex analysis approach to d.c. programming: Theory, algorithm and applications. 1997. 3.3

[11] L. Duan, D. Xu, and S. Chang. Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1338–1345, 2012. 2.2

[12] G. French, M. Mackiewicz, and M. Fisher. Self-ensembling for visual domain adaptation. In *International Conference on Learning Representations*, 2018. 1

[13] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, 2015. 1, 2.1

[14] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17(1):2096–2030, jan 2016. 4.1.2, 4.1.4

[15] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *European Conference on Computer Vision*, pages 597–613. Springer, 2016. 2.1

[16] B. Gong, K. Grauman, and F. Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *Proceedings of the 30th International Conference on Machine Learning*, pages 222–230, 17–19 Jun 2013. 1

[17] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. pages 2066–2073, 06 2012. 4.1.3

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2.1, 3.2, 3.4, 3.5

[19] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 529–536. MIT Press, 2005. 3.6.2

[20] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 4.1.3, 4.1.4

[21] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. 3.2, 3.5

[22] Q. Hoang, T. D. Nguyen, T. Le, and D. Phung. Multi-generator gernerative adversarial nets. *arXiv preprint arXiv:1708.02556*, 2017. 2.1

[23] J. Hoffman, M. Mohri, and N. Zhang. Algorithms and theory for multiple-source adaptation. In *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., 2018. 1, 2.2, 3.3, 1, 3.3

[24] J. J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994. 4.1.2

[25] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 4.1.1

[26] A. Krizhevsky, I. Sutskever, and G. E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1

[27] A. Kumar, P. Sattigeri, K. Wadhawan, L. Karlinsky, R. Feris, B. Freeman, and G. Wornell. Co-regularized alignment for unsupervised domain adaptation. In *Advances in Neural Information Processing Systems 31*, pages 9345–9356. Curran Associates, Inc., 2018. 3.6.2

[28] T. Le, Q. Hoang, H. Vu, T. D. Nguyen, H. Bui, and D. Phung. Learning generative adversarial networks from multiple data sources. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 2823–2829. International Joint Conferences on Artificial Intelligence Organization, 7 2019. 2.1

[29] T. Le, T. Nguyen, N. Ho, H. Bui, and D. Phung. Lamda: Label matching deep domain adaptation. In *ICML*, 2021. 1

[30] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 4.1.2

[31] C. Lee, T. Batra, M. H. Baig, and D. Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 10285–10295. Computer Vision Foundation / IEEE, 2019. 2.1

[32] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 97–105, 2015. 1, 2.1, 4.1.2, 4.1.3, 4.1.4

[33] M. Long, Z. Cao, J. Wang, and M. I. Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems 31*, pages 1640–1650. Curran Associates, Inc., 2018. 2.1

[34] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems 29*, pages 136–144. 2016. 4.1.4

[35] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 2208–2217, 2017. 4.1.2, 4.1.3, 4.1.4

[36] Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation with multiple sources. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1041–1048. 2009. 1, 2.2, 3.3, 1

[37] T. Miyato, S. Maeda, M. Koyama, and S. Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1979–1993, Aug 2019. 3.5, 3.6.2

[38] R. Müller, S. Kornblith, and G. E. Hinton. When does label smoothing help? In *Advances in Neural Information Processing Systems*, pages 4694–4703, 2019. 3.5

[39] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. 4.1.2

[40] T. Nguyen, T. Le, H. Zhao, H. Q. Tran, T. Nguyen, and D. Phung. Most: Multi-source domain adaptation via optimal transport for student-teacher learning. In *UAI*, 2021. 1

[41] T. Nguyen, T. Le, H. Zhao, H. Q. Tran, T. Nguyen, and D. Phung. Tidot: A teacher imitation learning approach for domain adaptation with optimal transport. In *IJCAI*, 2021. 1

[42] T. D. Nguyen, T. Le, H. Vu, and D. Phung. Dual discriminator generative adversarial nets. In *Advances in Neural Information Processing Systems 29 (NIPS)*, 2017. 2.1

[43] V. Nguyen, T. Le, O. De Vel, P. Montague, J. Grundy, and D. Phung. Dual-component deep domain adaptation: A new approach for cross project software vulnerability detection. In *PAKDD*, 2020. 2.1

[44] V. Nguyen, T. Le, T. Le, K. Nguyen, O. De Vel, P. Montague, L. Qu, and D. Phung. Deep domain adaptation for vulnerable code function identification. In *IJCNN*, 2019. 3.6.2

[45] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. In *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, IJCAI'09, pages 1187–1192, 2009. 1

[46] Y. Pan, T. Yao, Y. Li, Y. Wang, C. Ngo, and T. Mei. Transferrable prototypical networks for unsupervised domain adaptation. In *CVPR*, pages 2234–2242, 2019. 2.1

[47] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1406–1415, 2019. 2.2, 3.3, 4.1.2, 4.1.3, 4.1.4

[48] B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM J. Control Optim.*, 30(4):838–855, July 1992. 4.1.1

[49] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009. 1

[50] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc., 2015. 1

[51] K. Saito, Y. Ushiku, and T. Harada. Asymmetric tri-training for unsupervised domain adaptation. In *Proceedings of the 34th International Conference on Machine Learning*, pages 2988–2997. JMLR. org, 2017. 1

[52] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, June 2018. 2.1, 4.1.2, 4.1.3, 4.1.4

[53] S. Sankaranarayanan, Y. Balaji, Carlos D. Castillo, and R. Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8503–8512, 2018. 2.1

[54] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):640–651, April 2017. 1

[55] R. Shu, H. H. Bui, H. Narui, and S. Ermon. A DIRT-t approach to unsupervised domain adaptation. In *ICLR*, 2018. 1, 3.6.2

[56] B. Sun and K. Saenko. Deep coral: Correlation alignment for deep domain adaptation. In Gang Hua and Hervé Jégou, editors, *Computer Vision – ECCV 2016 Workshops*, pages 443–450, Cham, 2016. Springer International Publishing. 2.1, 4.1.2

[57] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. volume 28 of *Proceedings of Machine Learning Research*, pages 1139–1147, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. 4.1.1

[58] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2962–2971, 2017. 2.1, 4.1.2, 4.1.4

[59] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. *CoRR*, abs/1412.3474, 2014. 2.1

[60] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008. 4.2.1

[61] H. Wang, M. Xu, B. Ni, and W. Zhang. Learning to combine: Knowledge aggregation for multi-source domain adaptation. In *Computer Vision – ECCV*, 2020. 2.2, 4.1.1, 4.1.2, 4.1.4

[62] J. Wang, W. Feng, Y. Chen, H. Yu, M. Huang, and P. S. Yu. Visual domain adaptation with manifold embedded distribution alignment. *Proceedings of the 26th ACM international conference on Multimedia*, 2018. 4.1.3

[63] S. Xie, Z. Zheng, L. Chen, and C. Chen. Learning semantic representations for unsupervised domain adaptation. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 5423–5432. PMLR, 10–15 Jul 2018. 2.1

[64] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. volume 37 of *Proceedings of Machine Learning Research*, pages 2048–2057, Lille, France, 07–09 Jul 2015. PMLR. 1

[65] M. Xu, H. Wang, B. Ni, Q. Tian, and W. Zhang. Cross-domain detection via graph-induced prototype alignment. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 12352–12361. IEEE, 2020. 2.1

[66] M. Xu, J. Zhang, B. Ni, T. Li, C. Wang, Q. Tian, and W. Zhang. Adversarial domain adaptation with domain mixup. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 6502–6509. AAAI Press, 2020. 2.1

[67] R. Xu, Z. Chen, W. Zuo, J. Yan, and L. Lin. Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3964–3973, 2018. 2.2, 4.1.2, 4.1.3, 4.1.4

[68] H. Yan, Yukang Ding, P. Li, Qilong Wang, Yong Xu, and W. Zuo. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 945–954, 2017. 2.1

[69] L. Yang, Y. Balaji, S. Lim, and A. Shrivastava. Curriculum manager for source selection in multi-source domain adaptation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 608–624, Cham, 2020. 4.1.2, 4.1.3, 4.1.4

[70] H. Zhao, S. Zhang, G. Wu, J. M. F. Moura, J. P. Costeira, and G. J Gordon. Adversarial multiple source domain adaptation. In *Advances in Neural Information Processing Systems 31*, pages 8559–8570. Curran Associates, Inc., 2018. 1, 2.2, 4.1.2, 4.1.4

[71] S. Zhao, G. Wang, S. Zhang, Y. Gu, Y. Li, Z. Song, P. Xu, R. Hu, H. Chai, and K. Keutzer. Multi-source distilling domain adaptation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07):12975–12983, Apr. 2020. 2.2, 4.1.2, 4.1.4