Unsupervised Multi-Target Domain Adaptation Through Knowledge Distillation

L.T. Nguyen-Meidine[α], A. Belal[β], M. Kiran[α], J. Dolz[α], L-A. Blais-Morin[γ], E. Granger[α]
[α] *LIVIA, École de technologie supérieure, Montreal, Canada*
[β] *Aligarh Muslim University, Aligarh, India*
[γ] *Genetec Inc., Montreal Canada*
le-thanh.nguyen-meidine.1@ens.etsmtl.ca, abelal@myamu.ac.in, mkiran@livia.etsmtl.ca,
{jose.dolz, eric.granger}@etsmtl.ca, lablaismorin@genetec.com

Abstract

Unsupervised domain adaptation (UDA) seeks to alleviate the problem of domain shift between the distribution of unlabeled data from the target domain w.r.t. labeled data from the source domain. While the single-target UDA scenario is well studied in the literature, Multi-Target Domain Adaptation (MTDA) remains largely unexplored despite its practical importance, e.g., in multi-camera video-surveillance applications. The MTDA problem can be addressed by adapting one specialized model per target domain, although this solution is too costly in many real-world applications. Blending multiple targets for MTDA has been proposed, yet this solution may lead to a reduction in model specificity and accuracy. In this paper, we propose a novel unsupervised MTDA approach to train a CNN that can generalize well across multiple target domains. Our Multi-Teacher MTDA (MT-MTDA) method relies on multi-teacher knowledge distillation (KD) to iteratively distill target domain knowledge from multiple teachers to a common student. The KD process is performed in a progressive manner, where the student is trained by each teacher on how to perform UDA for a specific target, instead of directly learning domain adapted features. Finally, instead of combining the knowledge from each teacher, MT-MTDA alternates between teachers that distill knowledge, thereby preserving the specificity of each target (teacher) when learning to adapt to the student. MT-MTDA is compared against state-of-the-art methods on several challenging UDA benchmarks, and empirical results show that our proposed model can provide a considerably higher level of accuracy across multiple target domains. Our code is available at: `https://github.com/LIVIAETS/MT-MTDA`.

1. Introduction

Deep Learning (DL) models, and in particular Convolutional Neural Networks (CNNs), have achieved state-of-the-art performance in many visual recognition applications such as image classification, detection and segmentation [10]. Despite their success, several factors limit their deployment in real-world industrial applications. Among these factors is the problem of domain shift, where the distribution of original training data (source domain) diverges w.r.t data from the operational environment (target domain). This problem often translates to a notable decline in performance once the DL model has been deployed in the target domain.

To address this problem, DL models for domain adaptation have been proposed to align a discriminant source model with the target domain using data captured from the target domain [6, 8, 19, 27]. In unsupervised domain adaptation (UDA), a large amount of unlabeled data is often assumed to have been collected from the target domain to avoid the costly task of annotating data. Currently, several conventional and DL models have been proposed for the single target domain adaptation (STDA) setting, using unlabeled data that is collected from a single target domain. These models rely on different approaches, ranging from the optimization of a statistical criterion to the integration of adversarial losses, in order to learn robust domain-invariant representations from source and target domain data. However, despite multi-target domain adaptation (MTDA) scenario, i.e. multiple unlabeled target domains, has many real-world applications, it remains virtually unexplored. For instance, in video-surveillance applications, each camera of a distributed network corresponds to a different non-overlapping viewpoint (target domain). A DL model for person re-identification [20] should normally be adapted to multiple different camera viewpoints.

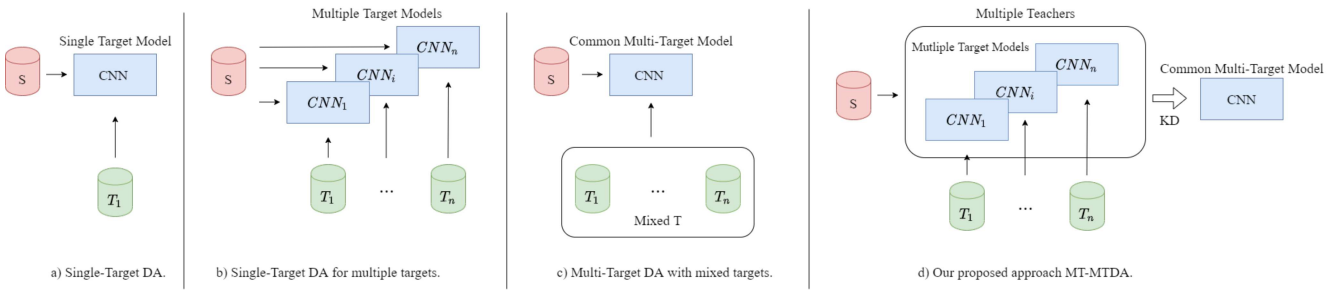

Figure 1. Illustration of different STDA and MTDA strategies for training CNNs across multiple target domains. S is the labelled source dataset, while T_i are the unlabelled target datasets for $i = 1, 2..., n$.

Extension of STDA techniques to the MTDA setting is not straightforward, and they may perform poorly on multiple target domains. Although MTDA problems can be solved by producing one model per target domain, this approach becomes costly and impractical in applications with a growing number of target domains. In such cases, a MTDA approach should ideally yield a common DL model that is compact and has been adapted to perform accurately across all target domains. To adapt a common multi-target DL model, one recent MTDA approach considers the problem of MTDA without domain labels, and proposes an approach to blend all the target domains together, which may lead to a reduction in model specificity and accuracy [5]. While the current approach provides an interesting direction in adapting a common model to multiple target domains, we argue that directly adapting a model to multiple target domains can affect the performance since there are limitations on a model's capacity to learn and generalize in diverse target domains. Other works on MTDA have focused on the problem of unshared categories between target domains [30], nevertheless, this scenario has not been considered since it is outside the scope of this work.

In this paper, a novel MTDA learning strategy referred to as Multi-Teacher MTDA (MT-MTDA) is proposed to train a common CNN to perform well across multiple target domains. Our strategy relies on knowledge distillation to efficiently transfer information from several different target domains, each one associated with a specialized teacher, to a single common multi-target model. Figures 1(a)-(c) illustrate the different MTDA strategies from literature, evolving from strategies that adapt a single CNN per target domain, to strategies that adapt a common CNN across all target domains. Our novel MT-MTDA approach (illustrated in Figure 1(d)) is inspired by a common education scenario, where each teacher is responsible for a single subject (i.e. target domain), and these teachers sequentially educate a student to learn all the subjects.

In our MT-MTDA approach: (1) Since only the student performance is important after training, we can resort to complex architecture for the teacher model; (2) These complex teachers can provide a higher capacity to generalize toward a single target domain instead of having one model learning multiple target domains; (3) The student model learns compressed knowledge from teachers across target domains, instead of directly learning to generalize on multiple domains; and (4) MT-MTDA can benefit from different STDA algorithms since each teacher adapt to only one target.

We also propose an efficient alternative for the fusion of knowledge from multiple teachers. State-of-the-art techniques for multi-teacher knowledge distillation rely on average fusion (sum operations) to directly combine the information derived by teachers [24]. To preserve the specificity of individual teachers, we let our student model learn to adapt from each teacher separately and sequentially from teacher to teacher. We argue that having better preservation of target specificity leads to higher accuracy.

Finally, the proposed MT-MTDA is compared extensively to state-of-the-art strategies on widely used UDA benchmarks (OfficeHome, Office31, and Digits-5), and show that MT-MTDA consistently achieves a high level of accuracy across multiple target domains with different backbone network architectures.

2. Related Work:

Single Target Domain Adaptation. STDA is an unsupervised transfer learning task that focuses on adapting a model such that it can generalize well on an unlabeled target domain data while using a labeled source domain dataset. DL models for UDA seek to learn discriminant and domain-invariant representations from source and target data[26]. They are either based on either adversarial-[8], discrepancy-[17], or reconstruction-based approaches[7]. Taking advantage of adversarial training, several methods [8, 25, 18, 3] have been proposed using either gradient reversal[8] or a combination of feature extractor and domain classifier to encourage domain confusion. Discrepancy-based ap-

proaches [17, 14] rely on measures between source and target distributions that can be minimized to generalize on the target domain. In [17], authors minimize the Maximum Mean Discrepancy (MMD) between target and source features to find domain invariant features. On the other hand, [14] assumes that task knowledge is already learned and the domain adaptation is done on the batch normalization layer to correct the domain shift. Lastly, another set of domain adaptation techniques focuses on the mapping of the source domain to target domain data or vice versa [2, 13]. These techniques are often based on the use of Generative Adversarial Network (GAN) in order to find a mapping between source and target.

Knowledge Distillation (KD). KD techniques allow for model compression by transferring knowledge from a teacher model, usually complex, to a smaller compact student model. The two main approaches of transferring knowledge between teacher and student models consist in minimizing the difference between logits [12, 15], and between features maps [29, 23, 11]. Techniques from the first approach focus on measuring logits obtained from a temperature-based softmax and then minimize the distance between the logits of the teacher and the student [12]. More recently, techniques like [11] minimize the distance between the intermediate feature maps of the teacher and student using a partial L2 distance. In contrast with other techniques, these features are obtained using a margin ReLU that accounts for negative values of the feature map. KD has been also recently employed in STDA [24, 21]. For example, in [24], the authors use multiple teachers and employ a fusion scheme that sums the output of each teacher as distillation strategy. In a similar work [21], STDA is performed during compression using knowledge distillation. While their approach is limited to a single-target scenario, we extend this approach to an MTDA setting by leveraging multi-teacher distillation into a single common student.

Multi Target Domain Adaptation. MTDA is a set of domain adaptation techniques that improves upon the single target domain adaptation by adapting a single model to teacher target domains. Currently, MTDA still remains largely unexplored with many open research questions. The few existing MTDA approaches follow two main directions: MTDA either with target domain labels [9] or without target domain labels [16, 5, 22]. The work in [9] proposes an approach that can adapt a model to multiple target domains by maximizing the mutual information between domain labels and domain-specific features while minimizing the mutual information between the shared features. Recently, [5] proposed to blend multiple target domains together and minimize the discrepancy between the source and the blended targets. Additionally, the authors employ an unsupervised meta-learner in combination with a meta target domain discriminator in order to blend the target domains. In [16], authors use a curriculum domain adaptations strategy combined with an augmentation of the representation based on features from source domain to handle multiple-target domains. While these methods achieve good performance, they fail to take advantage of existing STDA techniques, which have been extensively studied. Another important common point to existing methods is that they try to capture the representation of all the target domains using a common feature extractor directly from the data, which can degrade the final accuracy because of the limited capacity of the common model. In our paper, we overcome this issue by performing UDA separately on different models, and then distilling compressed knowledge to a common model. In addition, our experiments show that current mixed-target approaches still struggle with blending target domains in the feature space. We can gain more by preserving each domain specificity using STDA on different models.

3. Proposed Method

3.1. Domain Adaptation of Teachers:

In this paper, the RevGrad [8] technique is employed since it is the basis for many popular methods [1, 4], although it can be easily replaced by other STDA techniques. Let us define the source domain as $S = \{x_s, y_s\}$ where x_s is input pattern, and y_s its corresponding label. The set of target domains is defined as $T = \{T_1, T_2, ...T_n\}$, each one defined as $T_i = \{x_t^i\}$. For each target domain T_i, we define a teacher model Φ_i, and each of these teachers will be adapted to a corresponding target domain using the UDA technique proposed in [8]. The domain adaptation of the teacher relies on a domain classifier, a gradient reversal layer (GRL), and the domain confusion loss:

$$\mathcal{L}_{DC}(\phi_i, S, T_i) = \frac{1}{N_s + N_{ti}} \sum_{x \in S \cup T_i} \mathcal{L}_{CE}(D_i(\phi_i(x)), d_l) \quad (1)$$

where $\phi_i(x)$ is the output from the feature extractor of teacher network Φ_i, before the fully connected layers, D_i is the domain classifier for the corresponding teacher network, d_l the domain label (source or target), N_s is the number of samples in the source domain S, and N_{ti} is the number of samples in the target domain T_i.

The final domain adaptation loss is then defined as:

$$\mathcal{L}_{DA}(\Phi_i, S, T_i) = \frac{1}{N_s} \sum_{x_s, y_s \in S} \mathcal{L}_{CE}(\Phi_i(x_s), y_s) + \gamma \cdot \mathcal{L}_{DC}(\phi_i, T_i) \quad (2)$$

The first term (cross-entropy loss) allows the supervised training of the teacher model on the source domain that ensures the consistency of domain confusion. The second term is controlled by a hyper-parameter γ that regulates the importance of the domain confusion loss which is maximized using a gradient reversal layer. Figure 2 illustrates how GRL is applied for UDA.

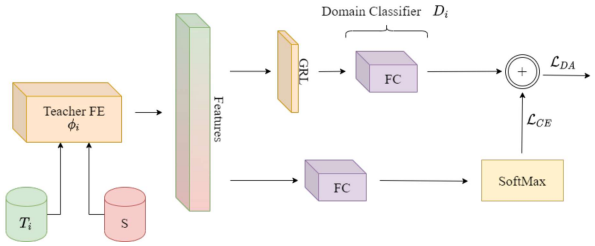

Figure 2. Illustration of GRL applied on a teacher model.

3.2. Teacher to Student Knowledge Distillation:

In this paper, we employ knowledge distillation based on logits as in [12][1]. The Figure 3 illustrates the overall process of distillation on both target and source domains. Logits from a teacher/student model are fed to a temperature-based softmax function, in combination with a KL divergence loss on both the teacher and student outputs:

$$\mathcal{L}_{KD}^{Source}(\Phi_i, \Theta, S) = \frac{1}{N_s} \sum_{x_s, y_s \in S} \mathcal{L}_{KL}(\Phi_i(x_s, \tau), \Theta(x_s, 1)) + \alpha \cdot \mathcal{L}_{CE}(\Theta(x_s, 1), y_s) \quad (3)$$

where Θ represents our student model with τ the temperature hyper-parameter the softmax, and α the hyper-parameter to regulate the importance of the cross-entropy term. Even though the second term of Eq. 3 may perform well with data from the source domain because it has labels, we add the domain confusion loss (Eq. 1) on the target domain to provide consistency during target distillation:

$$\mathcal{L}_{KD}^{Target}(\Phi_i, \Theta, T_i) = \frac{1}{N_{ti}} \sum_{x \in T_i} \mathcal{L}_{KL}(\Phi_i(x, \tau), \Theta(x, 1)) + \alpha \cdot \mathcal{L}_{DC}(\Theta, T_i) \quad (4)$$

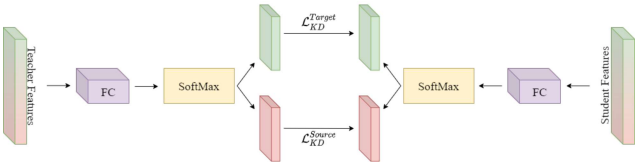

Figure 3. Illustration of proposed KD for domain adaptation.

[1]Note that our method can work with any other technique.

3.3. Multi-Teacher Multi-Target DA:

For progressive UDA of teacher models and transfer of knowledge from teacher to the student model, we adapt an exponential growing rate to gradually transfer the importance of UDA to KD in a similar way to [21]. The growth rate is defined as:

$$g = \frac{\log(f/s)}{N_e} \quad (5)$$

where s is the starting value, f the final value, and N_e the number of total epochs. This growth rate is used to calculate $\beta = s \cdot \exp\{g \cdot e\}$ with e the current epoch in the overall loss function for optimization of one teacher:

$$\mathcal{L}(\Phi_i, \Theta, S, T_i) = (1-\beta)\mathcal{L}_{DA}(\Phi_i, T_i) + \beta(\mathcal{L}_{KD}^{Source}(\Phi_i, \Theta, S) + \mathcal{L}_{KD}^{Target}(\Phi_i, \Theta, T_i)) \quad (6)$$

With β, the value that balances between the importance of the domain adaptation loss and the distillation loss. Our approach, MT-MTDA, instead of using deterministic fusion functions, such as average fusion, employs an alternative learning scheme for knowledge distillation from multiple teachers. This alternative scheme is performed by sequentially looping through each teacher at batch level (see Algorithm 1).

Algorithm 1: Multi-Teacher Multi-Target Domain adaptation (MT-MTDA)

input : A source domain dataset S, a set of target dataset $T_0, T_1, ...T_n$
output : A student model adapted to n targets
Initialize a set of teachers models $\Phi = \{\Phi_0, \Phi_1, ...\Phi_n\}$
Initialize a student model Θ
for $e \leftarrow 1$ **to** N_e **do**
  for $x_s \in S$ *and* $X_t \in \{T_0, ...T_n\}$ **do**
    Get the set of data of target domains X_t
    for $x_t^i \in X_t$ *and* $\Phi_i \in \Phi$ **do**
      Optimize $(1-\beta)\mathcal{L}_{DA}$ (2) for Φ_i using x_s, x_t^i
      Optimize the loss of equation $\beta\mathcal{L}_{KD}^{Source}$ (3) for Φ_i and Θ using x_s
      Optimize the loss of equation $\beta\mathcal{L}_{KD}^{Target}$ (4) for Φ_i and Θ using x_t^i
    end
    Update $\beta = s \cdot \exp^{g \cdot e}$
  end
  Evaluate the model
end

Figure 4 illustrates the overall pipeline for our MT-MTDA approach. While all teachers share the same source dataset, the figure shows that they each teacher has its own target dataset with their own domain adaptation loss.

4. Experiments

4.1. Datasets:

To the best of our knowledge, no specific dataset has been created for the for the MTDA task. For validation,

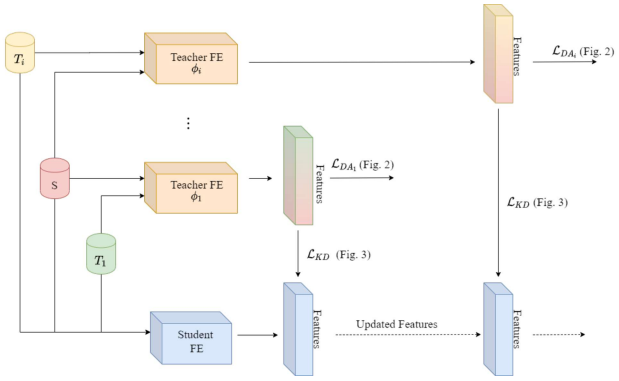

Figure 4. Illustration of the proposed learning technique.

we rely on datasets that are commonly used in other MTDA research [5, 9], which are described below.

1) Digits: This dataset regroups a set of digits datasets: MNIST (**mt**), MNIST-M (**mm**), SVHN (**sv**), USPS (**up**) and Synthetic Digits (**sy**). Each one has each 10 classes that represent all the digits. For the evaluation on this dataset, we follow the same protocol as in [5] for a fair comparison.

2) Office31: It has 3 subsets – Amazon (**A**), DSLR (**D**) and Webcam (**W**). These datasets all have 31 common classes. Images are taken respectively from the Amazon website, a DSLR camera and a webcam. We followed the standard evaluation protocol, a domain is chosen as a source, and the rest as targets.

3) OfficeHome: This dataset contains 4 subsets: Art (**Ar**), Clip Art (**Cl**), Real World (**Rw**) and Product (**Pr**). It has a total of 15,500 images for 65 object categories that are usually found in office or home settings. We follow the same evaluation protocol of Office31.

4) PACS: While this dataset is often used for domain generalization, [9] used it for MTDA. It contains 4 subsets: Art painting (**Ap**), Photo (**P**), Cartoon (**Cr**) and Sketch (**S**).

4.2. Implementation Details:

In MT-MTDA, we use the same number of optimizers as teacher models, which are responsible for the UDA of each teacher. Additionally, we add another optimizer for the knowledge distillation of the student. MT-MTDA is compared to 1) a lower bound, which is only trained on source and tested on target, 2) the current state-of-the-art in MTDA with domain labels such as MTDA-ITA[9] and 3) MTDA without domain labels such as AMEANS[5]. Additionally, we compare to baseline methods such as RevGrad[8] which is the basis of our MTDA method. We also use other baselines like DAN[17] or ADDA[25] in some cases for additional comparison, to show the advantages of MTDA algorithms. These baselines are domain adapted to directly to an ensemble of target domains similar to [5]. For the Digits-five dataset, we employ a LeNet backbone with ResNet50 as teacher. As for the comparison on Office31 and OfficeHome, we use AlexNet backbone with ResNet50 as teacher models, and as for the comparison on the ResNet50 backbone, we use a ResNext101 as teachers. Our backbone CNNs follows the choices in [5]. All these models start with pre-trained weights from ImageNet, except for LeNet.

When comparing with AMEANS[5] and DADA[22], we add **MT-MTDA Mixed** – our method when it employs mixed target domains without target domain labels. Specifically, we mix data from all the target domains, split them into subset of equal size without using domain labels, and then directly used them for UDA of respective teachers. For a fair comparison with AMEANS[5], we chose the same number of clusters, which corresponds to the number of target domains.

We selected the models' hyper-parameters based on their overall result in cross-validation in all the scenarios, instead of having a set of dedicated hyper-parameters for each scenario. Details of our hyper-parameters can be found in the Suppl. Material. We report the average classification accuracy obtained by all implemented models over 3 replications, from all the target domains. For other baselines, we report their best published result for fair comparison. Additional results (with OCDA[16], DADA[22]) and ablation studies (fusion methods, number of splits, etc.) are shown and analysed, along with a weighted average accuracy version of MT-MTDA in the Supplementary Material.

4.3. Results and Discussion:

Comparison without domain labels Table 1 shows the average classification accuracy of the MT-MTDA versus baseline and state-of-the-art methods on the Digits-Five dataset. We observe that our technique provides a higher level of accuracy, on average than the other approaches. In the first scenario, where our method performs poorly, further analysis on separate target domains (in Suppl. Material) indicates that our teacher models did not adapt well to the *mm* and *sy* datasets. This is mainly due to our selection of hyper-parameters, which was based on the all-scenario setting instead of individual cases. This explains why, in our first scenario, the result lags behind current baselines. It is possible, however, to overcome this problem by optimizing each scenario with a different set of hyper-parameters, including each teacher. Comparing the performance between the MT-MTDA Mixed (without domain labels) and MT-MTDA (with domain labels), the former improved slightly which might be due to the

Table 1. Accuracy of MT-MTDA and reference methods on the Digits-Five dataset.

Models	mt → mm, sv, up, sy	mm → mt, sv, up, sy	sv → mt, mm, up, sy	sy → mt, mm up, sv	up → mt, sv, mm, sy	Avg
Lower-bound: Superv. (source only)	36.6	57.3	67.1	74.9	36.9	54.6
ADDA[25]	52.5	58.9	46.4	67.0	34.8	51.9
DAN[17]	38.8	53.5	55.1	65.8	27.0	48.0
RevGrad[8]	60.2	66.0	64.7	69.2	44.3	60.9
DADA[22]	39.4	61.1	**80.1**	**83.7**	47.2	62.3
AMEANS[5]	**61.2**	66.9	67.2	73.3	47.5	63.2
MT-MTDA Mixed (ours)	59.5	**71.5**	69.9	78.3	49.6	**65.8**
MT-MTDA (ours)	58.6	71.0	67.6	75.6	**51.0**	64.7
Upper-bound: Superv. (targets)	88.1	90.2	93.0	90.3	89.1	90.1

fact that the Mixed version is less susceptible to the sharing of hyper-parameters. Finally, an upper bound is included, i.e., trained and tested on target domain data, to show the gap between with a supervised model.

Table 2. Accuracy of MT-MTDA and reference methods on Alexnet and ResNet50 as backbone(student) on the Office31.

Models	A → D,W	D → A,W	W → A,D	Avg
Teacher: ResNet50 — Student: AlexNet				
Superv. (source only)	62.7	73.3	74.4	70.1
DAN[17]	68.2	71.4	73.2	70.9
RevGrad[8]	74.1	72.1	73.4	73.2
AMEANS[5]	74.9	74.9	76.3	75.4
MT-MTDA Mixed (Ours)	80.3	**76.3**	**78.0**	**78.2**
MT-MTDA (Ours)	**82.5**	74.9	77.6	**78.3**
Teacher: ResNext101 — Student: ResNet50				
Superv. (source only)	68.7	79.6	80.0	76.1
DAN[17]	77.9	75.0	80.0	77.6
RevGrad[8]	79.0	81.4	82.3	80.9
AMEANS[5]	**89.8**	**84.6**	**84.3**	**86.2**
MT-MTDA Mixed (Ours)	85.5	84.0	**84.4**	84.6
MT-MTDA (Ours)	87.9	83.7	84.0	85.2

Tables 2 and 3 present the average classification accuracy of the MT-MTDA versus baseline and state-of-the-art methods on Office31 and OfficeHome data, respectively. In both cases, we observe that MT-MTDA, Mixed or with target domain labels, typically outperform the current state-of-the-art methods. With the AlexNet backbone, the improvements are significant, which can be explained by the advantage of using KD from multiple complex teachers, leading to a better generalization on a single target domain. We can observe that on Office31, AMEANS performs slightly better that MT-MTDA with the larger ResNet50 backbone. We believe that this is due to the limitations of domain adaptation on teacher models with MT-MTDA. We further discuss this point in the ablation study where we compare and discuss the performance of teacher and student (Sec. 4.4). Using our Mixed version on both these datasets, we often achieve similar results to the version with target domain labels.

Comparison with domain labels From Table 4, we observe that, in a comparison with another MTDA technique that uses domain labels[9], our method can have an improvement up to 15-16%. Similar to other comparisons, the boost in our performance is due to the use of teacher models with higher capacity to generalize then distilled to the student.

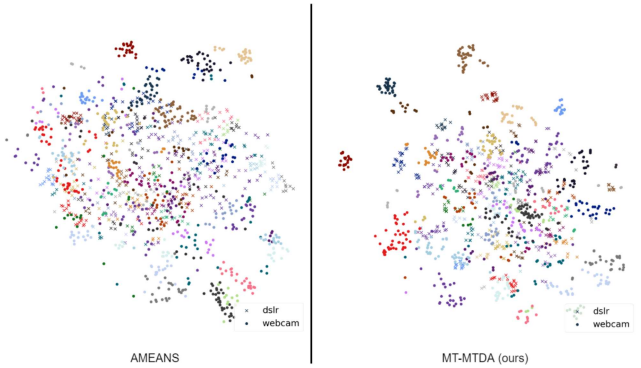

Figure 5. T-SNE visualization of Office31 data, where features are learned using MT-MTDA and AMEANS. Best viewed in color

Overall, our MT-MTDA technique outperforms both the baselines and state-of-the-art techniques. From Tables 1, 2, 3 and 4, we noticed that our model generally provides the highest accuracy on a compact backbone CNN, mainly because of the teacher's complexity and our knowledge distillation process. This is further confirmed by a comparison with the baseline RevGrad[8] technique adapted directly on multi-target domains. Additionally, the improvements in accuracy that our methods brings decrease when the complexity gap between the teachers and student is smaller. In this case, the performance bottleneck is the teacher and the distillation algorithm. We further discuss this point in the ablation study, when comparing between student and teacher models. Finally, the difference between our versions is very small, meaning that our technique can perform well without domain labels and still preserve a high level of accuracy. This indicates that our teacher models can generalize well on multiple sub-mixtures of target domains without reducing performance.

From Figure 5 [2], we observe that features learned with MT-MTDA better separates Office31 features, compared to other reference methods. Furthermore, MT-MTDA also separates class samples from different target domains in a better way than AMEANS. For comparison purposes, representations of other baselines are provided in the Suppl. Material. We noted that

[2]See a higher resolution image in Suppl. Material

Table 3. Accuracy of proposed and reference methods on OfficeHome dataset.

Models	Ar → Cl, Pr, Rw	Cl → Ar, Pr, Rw	Pr → Ar, Cl, Rw	Rw → Ar, Cl, Pr	Avg
Teacher: ResNet50 — Student: AlexNet					
Superv. (Source only)	33.4	35.3	30.6	37.9	34.3
DAN[17]	39.7	41.6	37.8	46.8	41.5
RevGrad[8]	42.2	43.8	39.9	47.7	43.4
AMEANS[5]	44.6	45.6	41.4	49.3	45.2
MT-MTDA Mixed (Ours)	48.6	46.6	41.1	52.1	47.1
MT-MTDA (Ours)	**48.8**	**48.7**	**42.9**	**55.8**	**49.1**
Teacher: ResNext101 — Student: ResNet50					
Superv. (Source only)	47.6	41.8	43.4	51.7	46.1
DAN[17]	55.6	55.1	47.8	56.6	53.8
RevGrad[8]	58.4	57.0	52.0	63.0	57.6
AMEANS[5]	64.3	64.2	59.0	66.4	63.5
MT-MTDA Mixed (Ours)	**64.8**	65.3	**60.5**	**67.5**	**64.5**
MT-MTDA (Ours)	64.6	**66.4**	59.2	67.1	**64.3**

Table 4. Comparison with [9] on PACS dataset.

LeNet	P → Ap, Cr, S				Ap → Cr, S, P			
	P → Ap	P → Cr	P → S	Avg	Ap → Cr	Ap → S	Ap → P	Avg
ADDA [25]	24.3	20.1	22.4	22.3	17.8	18.9	32.8	23.2
MTDA-ITA [9]	**31.4**	23.0	28.2	27.6	27.0	28.9	**35.7**	30.5
MT-MTDA (Ours)	24.6	**32.2**	**33.8**	**30.2**	**46.6**	**57.5**	**35.6**	**46.6**

in current state-of-the-art method, the target domains do not blend well with each other since the feature extractor can still differentiate them quite well based on the t-SNE.

4.4. Ablation Study

Detailed Comparison of Each Target Domain. For this experiment, we compare MT-MTDA in a setting where each target domain has a specific model. Our result on separate target domains is compared with RevGrad [8], but trained on a single target domain adaptation task. We also compared with the current best STDA algorithm, to our knowledge, in order to evaluate the effect of having a better STDA for the teacher model in our method.

Table 5. Accuracy of STDA methods for individual targets vs MTDA methods on Office31 dataset using AlexNet

AlexNet	A → D,W			D → A W			W → A D		
	A → D	A → W	Avg	D → A	D → W	Avg	W → A	W → D	Avg
Cosine Distance	0.23	0.21	0.22	0.21	0.24	0.22	0.21	0.24	0.22
RevGrad STDA	72.3	73.0	72.6	53.4	96.4	74.9	51.2	99.2	75.2
DM-ADA[28]	77.5	83.9	80.7	**64.6**	**99.8**	**82.2**	64.0	**99.9**	**81.9**
AMEANS	74.7[3]	74.6[2]	74.6[2]	-	-	74.9	-	-	76.3
MT-MTDA Mixed	81.9	78.7	80.3	56.2	96.5	76.3	56.3	99.8	78.0
MT-MTDA	**83.1**	**81.9**	**82.5**	52.5	97.3	74.9	55.5	**100**	77.6

Table 5 shows that while our algorithm does not perform as well as the state-of-the-art in STDA, it requires less computational power and memory since it only uses one model for all the target models instead of having a specific model for each target domain, i.e. two models in this case. This means that MTDA methods would typically scale better than current STDA methods since the number of models does not depend on the number of target models. Additionally, DM-ADA[28] shows that our algorithm can still be further improved since we can replace the STDA algorithm we are using on the teacher models (RevGrad[8]) with almost any other STDAs, i.e. CDAN[18]. The table also shows the cosine distance in order to quantify the domain shift between source and target features for each UDA problem. Results show that the UDA problems in Office31 have a similar level of difficulty.

Teachers vs Student Performance. We now compare the performances of our teachers with the student in order to explore the impact of knowledge distillation. For this experiment, we select the accuracy of each separate teacher on their respective target domain and compare it with the result of our student. This comparison was performed on the Office31 dataset[4]. From Table 6, the gap in accuracy is small and the student is almost at the same level as the teachers, except for the case of **A → D,W**. This indicates that our student model has learned how to adapt to multi-target domains from each separate teacher without an explicit fusion scheme. The first scenario of **A → D,W** shows a particular case where knowledge distillation helps improving domain adaptation. This behavior is also found when using ResNet50 as backbone architecture and seems to happen when the gap between the teachers and student is very small. This also suggests that knowledge from other teachers also help increasing accuracy. Additionally, from the ResNet50 backbone, we can see that the bottleneck can be found on teacher models and its domain adaptation since the student is stuck with a very similar accuracy as the teachers.

Table 6. Comparison with of teachers accuracy vs students

	Models	A → D,W	D → A,W	W → A,D	Average
Student	AlexNet	**82.5**	74.9	77.6	78.3
Teachers	ResNet50	77.6	**79.9**	**80.0**	**79.2**
Student	ResNet50	**87.9**	**83.7**	84.0	**85.2**
Teachers	ResNext101	85.9	83.6	**84.3**	84.6

Consistency on Target Knowledge Distillation. In this section, we evaluate the impact of the consistency

[3]These results were obtained on the model provided in the authors's github, hence it is slightly different from the result reported in the original paper used in Table 2

[4]Additional results on OfficeHome can be found in the Supp. Material

Table 7. Comparison of student's accuracy on separate domains with gradual increase in domains. The order in the target domains indicates the order in which they were integrated into the training.

AlexNet	Ar → Cl	Ar → Pr	Ar → Rw	Ar → Pr, Cl	Ar → Pr, Rw	Ar → Rw, Cl	Ar → Cl, Pr, Rw
Acc on Cl	**34.0**			33.3		33.0	**34.1**
Acc on Pr		**55.3**		50.1	50.0		52.6
Acc on Rw			59.0		57.9	57.7	**59.7**

Table 8. Accuracy of each target domain and standard deviation between these accuracies

AlexNet	Ar → Cl,Pr,Rw	Ar → Cl,Rw,Pr	Ar → Pr,Cl,Rw	Ar → Pr,Rw,Cl	Ar → Rw,Cl,Pr	Ar → Rw,Pr,Cl	STDev
Acc on Cl	34.1	33.7	34.0	**34.7**	33.6	**34.6**	0.4
Acc on Pr	52.6	52.5	**53.1**	52.5	52.5	**53.1**	0.3
Acc on Rw	59.7	**60.6**	59.8	59.5	60.4	59.5	0.4
Average Acc	48.8	48.9	48.9	48.9	48.8	**49.1**	0.1 | 0.3

loss on the target knowledge distillation 4. To this end, we removed the second term of the target knowledge distillation, Eq. 4, completely and we run our algorithm with the same settings as before on the scenario with an AlexNet as backbone on Office31 dataset. From the Table 9, it seems that having a consistency term on the target distillation loss only brings a small boost in performance. This aligns with our main results since the hyper-parameter that controls this consistency term is set to a small value.

Table 9. Accuracy of proposed method with target distillation consistency vs without

Models	A → D,W	D → A,W	W → A,D	Average
MT-MTDA without CST	82.1	73.8	74.3	76.7
MT-MTDA with CST	**82.5**	**74.9**	**77.6**	**78.3**

Single Teacher vs Multiple Teachers. We compare the scenario of having multiple teachers, each in a different target domain versus a scenario with one teacher adapting on a mixed target domain similar to [21]. In this setting, we merge all the target domains into a single target domain, where a single teacher is then assigned. We run this study on the Office31 dataset with the same hyper-parameters as in the main experiment. From the results of Table 10, we can observe that the accuracy of a mixed target domain using similar algorithm to [21] is significantly lower than the results with a multi-teacher approach. This suggests that even with a complex teacher network, a good generalization on a mixed target domain is hard to achieve and a multi-teacher scenario is preferable.

Table 10. Accuracy of proposed method using a single teacher vs multiple teachers

Models	A → D,W	D → A,W	W → A,D	Average
Single Teacher MTDA	75.3	64.0	67.0	68.8
MT-MTDA	**82.5**	**74.9**	**77.6**	**78.3**

Impact of Number of Target Domains. We now investigate the impact of increasing the number of domains on the student model. This experiment starts with a STDA setting of our algorithm and slowly increases the number of domains until reaching the maximum. We decided to do this experiment on the scenario with **Ar** dataset as source in OfficeHome dataset since it has more than two target domains and the dataset is bigger than Digits-Five. From Table 7, we can see that while the performance degrades on the target domain **Pr**, we notice a slight increase in accuracy of the other cases. This means that, with our method, training multiple target domains together can boost the performance of some separate target domains. The decrease of performance in the case of **Pr** also indicates that there might be a saturation in learning capacity. In this case, we can say that the target domains **Cl** and **Rw** improved at the expense of **Pr**.

Order of Target Domains. We evaluate whether the order of target domains impacts the performance of the final model. Similarly to the previous experiment, we used the scenario with **Ar** as the source domain in OfficeHome dataset since there are more than 2 target domains. Table 8 reports the results of individual target domains when their orders are different. These results indicate that even though the order of the domains leads to different average results, the difference between the configurations is marginal, of nearly 0.3%, with a standard deviation equal to 0.1. These results indicate that the order of target domains has little impact, if any, on the final result of the trained models.

5. Conclusion

In this paper, an avenue is unexplored for MTDA, relying on multiple teachers in order to distill knowledge from multiple target domains into a single student. Results from our experiment show that our method outperforms the current state-of-the-art, especially when using compact models, which can facilitate the use in numerous real-time applications. From our experiment, we identify several bottlenecks that can impede generalization of a compact model to multiple domains: 1) the STDA algorithm determines the accuracy of teacher models and 2) the transfer of target domain knowledge which needs to be improved when the student model is compact. Since STDA is a popular research area, our future work will focus on how to transfer target domain knowledge.

Acknowledgment

This research was supported in part by the NSERC, Compute Canada, and MITACS.

References

[1] Jan-Aike Bolte, Markus Kamp, Antonia Breuer, Silviu Homoceanu, Peter Schlicht, Fabian Huger, Daniel Lipinski, and Tim Fingscheidt. Unsupervised domain adaptation to improve image segmentation quality both in the source and target domain. In *CVPR Workshops*, June 2019.

[2] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. *CoRR*, abs/1612.05424, 2016.

[3] Minghao Chen, Shuai Zhao, Haifeng Liu, and Deng Cai. Adversarial-learned loss for fdomain adaptation. *arXiv preprint arXiv:2001.01046*, 2020.

[4] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *CVPR*, June 2018.

[5] Ziliang Chen, Jingyu Zhuang, Xiaodan Liang, and Liang Lin. Blending-target domain adaptation by adversarial meta-adaptation networks. In *CVPR*, June 2019.

[6] George Ekladious, Hugo Lemoine, Eric Granger, Kaveh Kamali, and Salim Moudache. Dual-triplet metric learning for unsupervised domain adaptation in video-based face recognition. *IJCNN*, 2020.

[7] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, Kun Zhang, and Dacheng Tao. Geometry-consistent generative adversarial networks for one-sided unsupervised domain mapping. In *CVPR*, June 2019.

[8] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation, 2014.

[9] Behnam Gholami, Pritish Sahu, Ognjen Rudovic, Konstantinos Bousmalis, and Vladimir Pavlovic. Unsupervised multi-target domain adaptation: An information theoretic approach. *CoRR*, abs/1810.11547, 2018.

[10] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[11] Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation. In *ICCV*, October 2019.

[12] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[13] Weixiang Hong, Zhenzhen Wang, Ming Yang, and Junsong Yuan. Conditional generative adversarial network for structured domain adaptation. In *CVPR*, 2018.

[14] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. *CoRR*, abs/1603.04779, 2016.

[15] Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Jia Li. Learning from noisy labels with distillation. *CoRR*, abs/1703.02391, 2017.

[16] Ziwei Liu, Zhongqi Miao, Xingang Pan, Xiaohang Zhan, Dahua Lin, Stella X. Yu, and Boqing Gong. Open compound domain adaptation. In *CVPR 2020*.

[17] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *ICML 2015*.

[18] Mingsheng Long, ZHANGJIE CAO, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *NIPS*, pages 1640–1650. 2018.

[19] Zimeng Luo, Jiani Hu, Weihong Deng, and Haifeng Shen. Deep unsupervised domain adaptation for face recognition. In *FG 2018*, 2018.

[20] Djebril Mekhazni, Amran Bhuiyan, George Ekladious, and Eric Granger. Unsupervised domain adaptation in the dissimilarity space for person re-identification. *ECCV*, 2020.

[21] Le Thanh Nguyen-Meidine, Eric Granger, Madhu Kiran, Jose Dolz, and Louis-Antoine Blais-Morin. Joint progressive knowledge distillation and unsupervised domain adaptation. *IJCNN*, 2020.

[22] Xingchao Peng, Zijun Huang, Ximeng Sun, and Kate Saenko. Domain agnostic learning with disentangled representations. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5102–5112, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[23] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

[24] Sebastian Ruder, Parsa Ghaffari, and John G. Breslin. Knowledge adaptation: Teaching to adapt. *CoRR*, abs/1702.02052, 2017.

[25] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, July 2017.

[26] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *CoRR*, abs/1802.03601, 2018.

[27] Ge Wen, Huaguan Chen, Deng Cai, and Xiaofei He. Improving face recognition with domain adaptation. *Neurocomputing*, 287:45–51, 2018.

[28] Minghao Xu, Jian Zhang, Bingbing Ni, Teng Li, Chengjie Wang, Qi Tian, and Wenjun Zhang. Adversarial domain adaptation with domain mixup. *arXiv preprint arXiv:1912.01805*, 2019.

[29] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *CVPR*, July 2017.

[30] Huanhuan Yu, Menglei Hu, and Songcan Chen. Multi-target unsupervised domain adaptation without exactly shared categories. *CoRR*, abs/1809.00852, 2018.