Supplementary Material

A. Experimental Methodology

A.1. Hyper-parameters

From Table 11, you can find all the hyper-parameters that was used for different datasets and backbones. We selected these hyper-parameters based on a standard cross-validation process. These hyper-parameters are selected based on the overall result in all the scenarios instead of each scenario.

A.2. Evaluation metrics

In the paper of [5], the authors first proposed an accuracy metrics that take in account the different sizes of each target domain in order to have a balanced accuracy score at the end. This accuracy is defined as:

$$Acc = \sum_{i=0}^{n} w_i Acc_i \tag{7}$$

With w_i calculated as $w_i = \frac{N_i}{\sum_{j=0}^{n} N_j}$. The problem with this accuracy is that it can hide the poor performance of a target domain that is small. The authors from the same paper proposed to use another accuracy which is the same one we used in our main paper where the same weight is used for each target domain. This is also referred as the equal-weight classification accuracy in the paper of [5]. This is accuracy is calculated as:

$$Acc^{EQ} = \frac{1}{n} \sum_{i=0}^{n} Acc_i \tag{8}$$

Additionally, we also give our result based on the Equation 7 and on each target domain in order to highlight where our algorithm can fail.

B. Results and Discussion

B.1. Digits-Four

In this comparison, we compare our results with both MTDA-ITA[9] and [16] on the same scenario as in [16] on Digits. For a fair comparison with [16], which is an open domain adaptation algorithm, we only takes in account the compound domains results of OCDA (close domain adaptation) since they are part of the closed domain of OCDA which is similar to our setting.

From Table12, we observe that, while our base techniques (with RevGrad) perform slightly worse than OCDA, it still performs better than most of the other techniques. As for our version that uses CDAN[18], our technique perform even better than state-of-the-art technique [16] by around 1%.

B.2. Comparison with DADA[22] on Office-Caltech 10

In this experiment, we provide an additional comparison with DADA[22] on the Office-Caltech10 dataset with AlexNet and ResNet101 as backbones similar to DADA[22]. For this, we use similar hyper-parameters as the Office31 dataset. From Table 13, we can see that both version of our techniques perform better than DADA[22]. Similar to previous comparison, we think that our technique achieves state-of-the-art performance by taking advantages of higher generalization capacity of teacher models and transfer it to a common student.

B.3. Further analysis on Digits-Five

As indicated in our results for Table 1, we analyzed the accuracy of each target domain in order to show where's our drop in accuracy.

From Table 14, we can see that the drop in performance in the scenario previously noted in the main paper is due to the decline of the domain adaptation on both **mt** $\rightarrow$ **mm** and **mt** $\rightarrow$ **sy**. Further analysis of these two domain adaptation shows that our common hyper-parameters do no work well for these two cases since we can get better performance using other hyper-parameters. However, these parameters would yield lower performance on other scenarios therefore we choose to remain on the same hyper-parameters as before. This indicates that in order to have better performance, it is best to have a different hyper-parameters set for each scenario and even each teacher.

B.4. Number of splits for mixed target domains

In this experiment, we evaluate the impact of the number of splits with our technique on the Office31 dataset with AlexNet as backbone. We gradually increase the number of splits starting from two splits, since one splits is the same as the scenario of single-teacher with a mixture of targets.

From Table 15, we noticed that an increase in the number of splits and the number of teachers can result in a slight increase in the overall accuracy. The results also show that our teacher model is capable of adapting to randomly split sub-mixture of target domains and then transfer the knowledge to a single student model, independently from the number of splits. In addition, these results also show the robustness of our algorithm since the sub-mixture target domains are always obtained randomly.

B.5. MT-MTDA using another STDA technique

In this experiment, we evaluate our algorithm with another domain adaptation technique, namely [18], in

order to show that our algorithm is agnostic w.r.t domain adaptation technique. We will also do the same for distillation, where we use [11] distillation.

From Table 16, we noticed that the version with CDAN[18], while it does not present the same performance gap as in the STDA case as shown in [18], it still performs better than the version with RevGrad of around 1%. Taking in account result from 12, we can see that our student model is reaching its limit in term of generalization across multiple domains for the Office31 dataset. Lastly, results in both of these tables also show the improvement our algorithm can achieve when employing state-of-the-art UDA technique.

B.6. Comparison with Other Fusion Methods.

To demonstrate the benefits of the proposed feature fusion strategy, we compare our alternative fusion scheme with other baselines fusion methods, e.g., the sum or the mean of the output. The hyper-parameters for all the cases remain the same to those of the main experiment, with the only difference being the output of all the teachers is summed/averaged and then distill to the student. Table 17 shows that the proposed alternative distillation works better than either fusion by sum or average. This means that the proposed alternative scheme transfers learned knowledge better than the baseline methods in the particular case of MTDA. In addition, this shows that the student does not need an explicit fusion scheme in order to learn target domain knowledge from multiple teachers.

B.7. Weighted Accuracy

In this section, we present our average accuracy using Equation 7. We compare with the weighted accuracy reported in the paper of [5] for a fair comparison.

From Tables 18, 19, 20, our weighted results are still consistent with our equal-weight results in the main paper. Our method performs better than current state-of-the-art method in all cases except on Office31 with ResNet50. These results show that our method does not improve upon state-of-the-art by having a good accuracy on an easy case of domain adaptation with huge amount of data but it improves in more general manner.

B.8. Additional Comparison on Each Target

As mentioned in the main paper, we present more results on each separate target domain comparing to a standard STDA baseline [8] on OfficeHome using AlexNet.

From Table 21, we can draw a similar conclusion of the main paper. Our method performs in average better than multiple STDA on different target domains. This shows that we can have one model handling different target domains without sacrificing computational power or memory.

B.9. TSNE Visualization

In this section, we add the TSNE of RevGrad[8] and DAN[17] and provide a higher resolution of the previous TSNE. From Figure 6, we can see that features between different target domains can be mixed together even when there's a blending mechanism like in [5].

Table 11. Hyper-parameters for our algorithms for each backbone and dataset

Hyper parameters	**Digits-Five LeNet**	**Office31 Alexnet**	**OfficeHome Alexnet**	**Office31 ResNet50**	**OfficeHome Resnet50**
N_e	100	100	200	100	200
batch size	64	16	8	16	8
τ	20	20	20	20	20
α	0.5	0.3	0.5	0.3	0.5
s	0.1	0.1	0.1	0.1	0.1
f	0.8	0.8	0.5	0.8	0.5
γ	0.5	0.5	0.5	0.5	0.5
UDA Learning Rate	0.0005	0.001	0.0001	0.001	0.0001
KD Learning Rate	0.0005	0.01	0.001	0.01	0.001
weight decay	0.0005	0.0005	0.0005	0.0005	0.0005

Table 12. Accuracy of proposed and reference methods on Digits-Four dataset

LeNet	**Source → Targets sv → mt, mm, up**			
	sv → mt	**sv → mm**	**sv → up**	**Average**
ADDA [25]	80.1	56.8	64.8	67.2
MTDA-ITA [9]	84.6	65.3	70.0	73.3
AMEANS [5]	85.2	65.7	74.3	75.1
OCDA [16]	90.9	65.7	83.4	80.0
MD-MTDA Mixed (Ours)	87.5	65.4	84.7	79.2
MD-MTDA (Ours)	86.9	65.2	84.3	78.8
MD-MTDA CDAN Mixed (Ours)	**92.8**	67.3	85.1	81.7
MD-MTDA CDAN (Ours)	92.0	**71.1**	**88.9**	**84.0**

Table 13. Accuracy of MT-MTDA and reference methods on Alexnet and Resnet101 as backbone(student) on the Office-Caltech

Models	**A → C,D,W**	**C → A,D,W**	**D → A,C,W**	**W → A,C,D**	**Average**
Teacher: ResNet50 — Student: AlexNet					
Source only	83.1	88.9	86.7	82.2	85.2
RevGrad[8]	85.9	90.5	88.6	90.4	88.9
DADA[22]	86.3	91.7	89.9	91.3	89.8
MT-MTDA Mixed (Ours)	92.8	93.4	89.2	90.8	91.6
MT-MTDA (Ours)	**93.3**	**93.9**	**90.1**	**91.2**	**92.1**
Teacher: ResNext101 — Student: ResNet101					
Source only	90.5	94.3	88.7	82.5	89.0
RevGrad[8]	91.5	94.3	90.5	86.3	90.6
DADA[22]	92.0	95.1	91.3	93.1	92.9
MT-MTDA Mixed (Ours)	94.9	97.9	94.7	95.3	95.7
MT-MTDA (Ours)	**96.1**	**98.1**	**96.3**	**96.4**	**96.7**

Table 14. Accuracy of each target domain on Digits-Five dataset with LeNet as Backbone.

Lenet	**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**
Student Acc on mt	-	96.3	69.1	85.8	87.0
Student Acc on mm	46.6	-	48.1	55.5	40.3
Student Acc on sv	53.8	43.9	-	75.3	30.7
Student Acc on sy	57.7	82.9	83.7	-	48.4
Student Acc on up	77.3	61.1	69.4	85.8	-
Average	58.85	71.05	67.575	75.6	51.6

Table 15. Accuracy of MT-MTDA Mixed on different number of splits of sub-targets

AlexNet	**A → D,W**	**D → A,W**	**W → A,D**	Average
MT-MTDA Mixed 2	80.3	76.3	78.0	78.2
MT-MTDA Mixed 3	81.2	76.2	78.2	78.5
MT-MTDA Mixed 4	**82.1**	76.6	**78.8**	**79.2**
MT-MTDA Mixed 10	81.3	**76.9**	78.5	78.9

Table 16. Accuracy of MT-MTDA with RevGrad vs CDAN

Models	**A → D,W**	**D → A,W**	**W → A,D**	**Average**
Teacher: ResNet50 — Student: AlexNet				
MT-MTDA Mixed (Ours)	80.3	76.3	**78.0**	78.2
MT-MTDA (Ours)	82.5	74.9	77.6	78.3
MT-MTDA CDAN Mixed (Ours)	84.3	75.4	77.8	79.2
MT-MTDA CDAN (Ours)	**84.5**	**76.9**	**78.0**	**79.8**

Table 17. Accuracy of proposed method with different fusions

Models	**A → D,W**	**D → A,W**	**W → A,D**	**Average**
MT-MTDA Mean	75.1	65.4	67.1	69.2
MT-MTDA Sum	78.3	66.9	69.8	71.6
MT-MTDA	**82.5**	**74.9**	**77.6**	**78.3**

Table 18. Weighted accuracy of proposed and baseline methods on Digits-Five dataset with AlexNet as Backbone.

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**	**Average**
Source only	26.9	56.0	67.2	73.8	36.9	52.2
ADDA	43.7	55.9	40.4	66.1	34.8	48.2
DAN	31.3	53.1	48.7	63.3	27.0	44.7
RevGrad	52.4	64.0	65.3	66.6	44.3	58.5
AMEANS	56.2	65.2	67.3	71.3	47.5	61.5
MT-MTDA Mixed (ours)	51.6	69.2	**79.7**	**76.0**	61.5	**67.6**
MT-MTDA (ours)	**54.3**	**73.4**	67.1	73.1	**64.0**	66.4

Table 19. Weighted accuracy of proposed and baseline methods on Office31 dataset.

Models	**A → D,W**	**D → A,W**	**W → A,D**	**Average**
Teacher: ResNet50 — Student: AlexNet				
Source only	62.4	60.8	57.2	60.1
DAN[17]	68.2	58.7	55.6	60.8
RevGrad[8]	74.1	58.6	55.0	62.6
AMEANS[5]	74.5	62.8	59.7	65.7
MT-MTDA Mixed (ours)	79.9	**65.1**	**62.8**	**69.3**
MT-MTDA (ours)	**82.4**	62.4	61.9	68.9
Teacher: ResNext101 — Student: ResNet50				
Source only	68.6	70.0	66.5	68.4
DAN[17]	78.0	64.4	66.7	69.7
RevGrad[8]	78.2	72.2	69.8	73.4
AMEANS[5]	**90.1**	**77.0**	73.4	**80.2**
MT-MTDA Mixed (ours)	85.2	75.8	**73.6**	78.2
MT-MTDA (ours)	87.8	75.4	72.8	78.7

Table 20. Weighted accuracy of proposed and baseline methods on OfficeHome dataset.

Models	Ar ->Cl, Pr, Rw	Cl ->Ar, Pr, Rw	Pr ->Ar, Cl, Rw	Rw ->Ar, Cl, Pr	Average
Teacher: ResNet50 — Student: AlexNet					
Source only	33.4	37.6	32.4	39.3	35.7
DAN	39.7	43.2	39.4	47.8	42.5
RevGrad	42.1	45.1	41.1	48.4	44.2
AMEANS	44.6	47.6	42.8	50.2	46.3
MT-MTDA Mixed (ours)	48.6	48.1	42.3	53.0	48.0
MT-MTDA (ours)	**48.8**	**50.1**	**44.0**	**56.0**	**49.7**
Teacher: ResNext101 — Student: ResNet50					
Source only	47.6	42.6	44.2	51.3	46.4
DAN	55.6	56.6	48.5	56.7	54.3
RevGrad	58.4	58.1	52.9	62.1	57.9
AMEANS	64.3	65.5	59.5	66.7	64.0
MT-MTDA Mixed (ours)	**64.9**	66.3	**60.2**	**66.9**	**64.6**
MT-MTDA (ours)	64.6	**67.1**	59.0	66.4	64.3

Table 21. Average accuracy of proposed and baseline STDA methods for individual and overall target datasets on OfficeHome dataset using AlexNet

Alexnet	Ar → Cl, Pr, Rw				Cl → Ar, Pr, Rw				Pr → Ar, Cl, Rw				Rw → Ar, Cl, Pr			
	Ar → Cl	Ar → Pr	Ar → Rw	Avg	Cl → Ar	Cl → Pr	Cl → Rw	Avg	Pr → Ar	Pr → Cl	Pr → Rw	Avg	Rw → Ar	Rw → Cl	Rw → Pr	Avg
RevGrad STDA	**36.4**	45.2	54.7	45.4	35.2	51.8	**55.1**	47.4	31.6	**39.7**	**59.3**	**43.5**	45.7	**46.4**	65.9	52.6
AMEANS	-	-	-	44.6	-	-	-	45.6	-	-	-	41.4	-	-	-	49.3
MT-MTDA	34.1	**52.6**	**59.7**	**48.8**	**40.7**	**52.0**	53.5	**48.7**	**36.5**	33.7	58.6	42.9	**55.0**	42.0	**70.3**	**55.7**

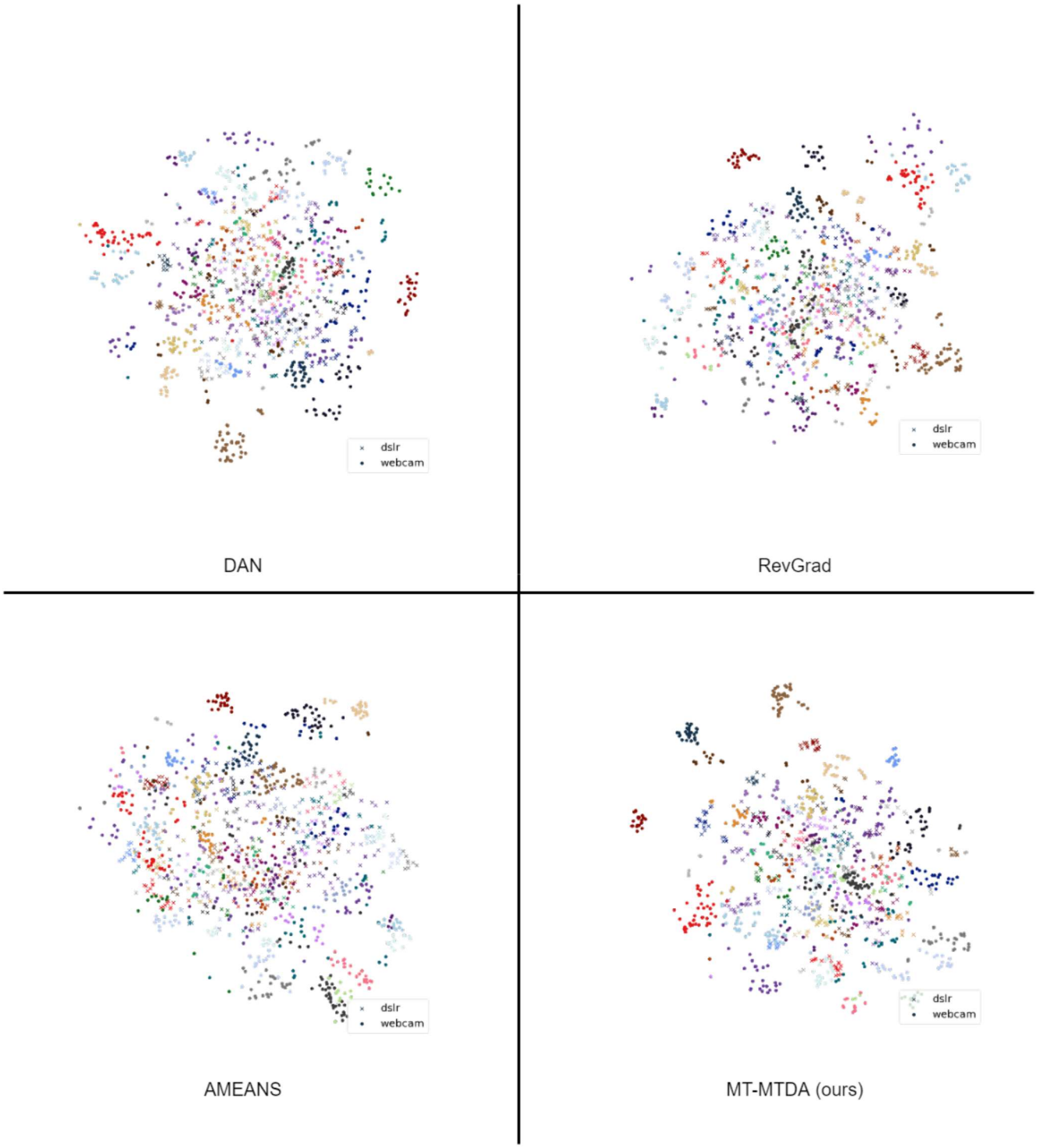

Figure 6. T-SNE visualization of all baselines methods versus MT-MTDA (ours)