Curriculum Graph Co-Teaching for Multi-Target Domain Adaptation (Supplementary Material)

Subhankar Roy[1,2*], Evgeny Krivosheev[1*], Zhun Zhong[1†], Nicu Sebe[1], Elisa Ricci[1,2]
[1]University of Trento, Italy [2]Fondazione Bruno Kessler, Italy

In this supplementary material, we provide more implementation details, discussion of the proposed CGCT, and additional experimental results. In details, we provide the pseudo-code algorithms of the proposed CGCT and D-CGCT in Sec. A. We highlight the key differences between CGCT and prior works in Sec. B. The details of datasets and implementation are provided in Sec. C and Sec. D, respectively. The additional experimental results are reported in Sec. E.

A. Algorithms

In this section we provide the pseudo-code algorithms for the proposed CGCT (see Sec. 3.2 of the main paper) and D-CGCT (see Sec. 3.3 of the main paper) in the Alg. 1 and Alg. 2, respectively. Note that the *adaptation stage* in the Alg. 2 can be replaced by any desired single-target domain adaptation (STDA) method of choice, thereby, making the proposed DCL flexible to a wide variety of STDA methods.

B. Discussion

Here we highlight the keys differences between the CGCT and PGL [10] as well as the dual classifier-based methods [4, 15]. The PGL [10] exploits the graph learning framework in an episodic fashion to obtain pseudo-labels for the unlabeled target samples, which are then used to bootstrap the model by training on the pseudo-labeled target data. While our proposed method is similar in spirit to the episodic training in [10], we do not solely rely on the GCN to obtain the pseudo-labels. We conjecture that due to the fully-connected nature of the graph and lack of target labels, the GCN will be prone to accumulate features of dissimilar neighbours, thereby, resulting in the erroneous label propagation. To address this peculiarity, we propose to resort to the co-teaching paradigm, where the G_{mlp} is exploited to train the f_{edge} network. As the two classifiers will capture different aspects of training [4], it will prevent the f_{edge} to be trained with the same erroneous pseudo-labels as the f_{node}. We validate this conjecture empirically, where a network with a single GCN classifier with pseudo-labels performs sub-optimally compared to CGCT (see Tab. 5 row 7 of the main paper). Finally, the dual classifier-based methods maintain two classifiers to identify and filter either harder target samples [15] or noisy samples [4]. Contrarily, we maintain G_{mlp} and G_{gcn} to provide feedback to each other by exploiting the key observation that each classifier learns different patterns during training. Furthermore, given the intrinsic design of the G_{gcn}, we also do away with an extra adhoc loss of keeping the weights of two networks different.

C. Datasets

Digits-five [19] is composed of five domains that are drawn from the: i) grayscale handwritten digits MNIST [6] (**mt**); ii) a coloured version of **mt**, called as MNIST-M [3] (**mm**); iii) USPS [2] (**up**), which is a lower resolution, 16×16, of the handwritten digits **mt**; iv) a real-world dataset of digits called SVHN [11] (**sv**); and v) a synthetically generated dataset *Synthetic Digits* [3] (**sy**). Following the protocol of [1], we sub-sample 25,000 and 9,000 samples from the training and test sets of **mt**, **mm**, **sv** and **sy** and use as train and test sets, respectively. For the **up** domain we use all the 7,348 training and 1,860 and test samples, for our experiments. All the images are re-scaled to a 28×28 resolution.

Office31 [14] is a standard visual DA dataset comprised of three domains: Amazon, DSLR and Webcam. The dataset consists of 31 distinct object categories with a total of 4,652 samples.

Office-Home [18] is a relatively newer DA benchmark that is larger than Office31 and is composed of four different visual domains: Art, Clipart, Product and Real. It consists of 65 object categories and has 15,500 images in total.

PACS [7] is another visual DA benchmark that also consists of four domains: Photo (P), Art Painting (A), Cartoon (C) and Sketch (S). This dataset is captured from 7 object categories and has 9,991 images in total.

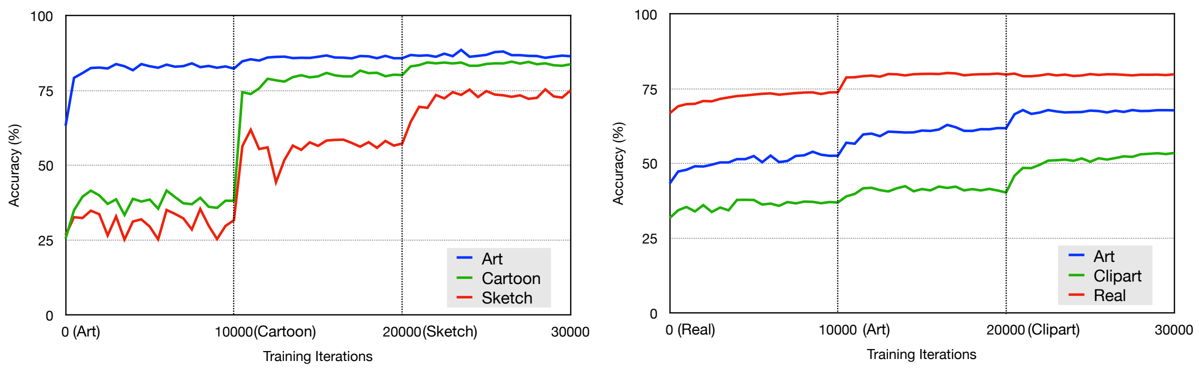
DomainNet [12] is the most challenging and very large scale DA benchmark, which has six different domains: Clipart (C), Infograph (I), Painting (P), Quickdraw (Q), Real

[*] Equal contribution
[†] Corresponding author

```
Algorithm 1: Training Procedure of Curriculum Graph Co-Teaching (CGCT)
require: number of target domains $N$, classes $n_c$
require: source dataset $\mathcal{S}$; combined target dataset $\mathcal{T}$
require: hyper-parameters $B$, $\tau$, $K$,
         $K'$, $\lambda_{edge}$, $\lambda_{node}$, $\lambda_{adv}$
require: networks $F$, $D$, $G_{mlp}$, $f_{edge}$, $f_{node}$ with
         parameters $\theta, \psi, \phi, \varphi, \varphi'$, respectively. The
         $f_{edge}$ and $f_{node}$ form the $G_{gcn}$.
Step 1: Pre-training on the source dataset
while $\ell_{ce}$ has not converged do
    $(\mathbf{x}_{s,i}, y_{s,i})_{i=1}^{B} \sim \mathcal{S}$
    update $\theta$, $\phi$ by $\min_{\theta,\phi} \ell_{ce}^{mlp}$
end
Step 2: Curriculum learning
$\hat{\mathcal{S}}^0 \leftarrow \mathcal{S}$
$Q \leftarrow N$                                  ▷ Total # curriculum steps
for $q$ in $(0 : Q-1)$ do
                                                  ▷ Curriculum step
    Stage 1: Adaptation stage
    for $k$ in $(1 : K)$ do
        $\hat{\mathcal{B}}_s^q \leftarrow (\mathbf{x}_{s,i}, y_{s,i})_{i=1}^{B} \sim \hat{\mathcal{S}}^q$
        $\hat{\mathcal{B}}_t^q \leftarrow (\mathbf{x}_{t,i})_{i=1}^{B} \sim \mathcal{T}$
        $\hat{y} \leftarrow \mathtt{softmax}(G_{mlp}(F(\mathbf{x})))$
        $\bar{y} \leftarrow \mathtt{softmax}(G_{gcn}(F(\mathbf{x})))$
        $\hat{d} \leftarrow \mathtt{sigmoid}(D(F(\mathbf{x})))$
        update $\psi$ by $\min_{\psi} \lambda_{adv}\ell_{adv}$
        update $\theta$, $\phi$ by $\min_{\theta,\phi} \ell_{ce}^{mlp} - \lambda_{adv}\ell_{adv}$
        update $\theta$, $\varphi, \varphi'$ by
         $\min_{\theta,\varphi,\varphi'} \lambda_{edge}\ell_{bce}^{edge} + \lambda_{node}\ell_{ce}^{node}$
    end
    Stage 2: Pseudo-labeling stage
    $\mathcal{D}_t^q \leftarrow \{\}$                         ▷ Empty list
    for $\mathbf{x}_{t,j} \in \mathcal{T}$ do
        $w_j \leftarrow \max_{c \in n_c} p(\bar{y}_{t,j} = c|\mathbf{x}_{t,j})$
        if $w_j > \tau$ then
            $\mathcal{D}_t^q \leftarrow \mathcal{D}_t^q || \{(\mathbf{x}_{t,j}, \mathrm{argmax}_{c \in n_c}\, p(\bar{y}_{t,j} =$
             $c|\mathbf{x}_{t,j}))\}$                         ▷ Append
        end
    end
    $\hat{\mathcal{S}}^{q+1} \leftarrow \mathcal{S} \cup \mathcal{D}_t^q$              ▷ Pseudo-source
end
Step 3: Fine-tuning on pseudo-source dataset
for $k'$ in $(1 : K')$ do
    $(\mathbf{x}_{s,i}, y_{s,i})_{i=1}^{B} \sim \hat{\mathcal{S}}^Q$
    update $\theta$, $\phi$ by $\min_{\theta,\phi} \ell_{ce}^{mlp}$
end
```

(R) and Sketch (S). It has around **0.6 million** images, including both train and test images, and has 345 different object categories. We use the official training and testing splits, as mentioned in [13], for our experiments.

```
Algorithm 2: Training Procedure of Domain-aware Curriculum Graph Co-Teaching (D-CGCT)
require: number of target domains $N$, classes $n_c$
require: source dataset $\mathcal{S}$; target dataset $\mathcal{T} = \{\mathcal{T}_j\}_{j=1}^{N}$
require: hyper-parameters $B$, $\tau$, $K$,
         $K'$, $\lambda_{edge}$, $\lambda_{node}$, $\lambda_{adv}$
require: networks $F$, $D$, $G_{mlp}$, $f_{edge}$, $f_{node}$ with
         parameters $\theta, \psi, \phi, \varphi, \varphi'$, respectively. The
         $f_{edge}$ and $f_{node}$ form the $G_{gcn}$.
Step 1: Pre-training on the source dataset
while $\ell_{ce}$ has not converged do
    $(\mathbf{x}_{s,i}, y_{s,i})_{i=1}^{B} \sim \mathcal{S}$
    update $\theta$, $\phi$ by $\min_{\theta,\phi} \ell_{ce}^{mlp}$
end
Step 2: Curriculum learning
$\hat{\mathcal{S}}^0 \leftarrow \mathcal{S}$ and $\hat{\mathcal{T}}^0 \leftarrow \{\mathcal{T}_j\}_{j=1}^{N}$
$Q \leftarrow N$                                  ▷ Total # curriculum steps
for $q$ in $(0 : Q-1)$ do
                                                  ▷ Curriculum step
    $\mathcal{H} \leftarrow \{\}$                              ▷ Empty list
    Stage 1: Domain selection stage
    for $\mathcal{T}_j$ in $\hat{\mathcal{T}}^q$ do
        compute $H(\mathcal{T}_j)$ as in Eqn. 12
        $\mathcal{H} \leftarrow \mathcal{H} \mid\mid H(\mathcal{T}_j)$                  ▷ Append
    end
    $\mathbb{D}^q \leftarrow \mathrm{argmin}_j\, \mathcal{H}$                    ▷ Chosen domain
    Stage 2: Adaptation stage
    for $k$ in $(1 : K)$ do
        $\hat{\mathcal{B}}_s^q \leftarrow (\mathbf{x}_{s,i}, y_{s,i})_{i=1}^{B} \sim \hat{\mathcal{S}}^q$
        $\hat{\mathcal{B}}_t^q \leftarrow (\mathbf{x}_{t,i})_{i=1}^{B} \sim \mathcal{T}_{\mathbb{D}^q}$
        $\hat{y} \leftarrow \mathtt{softmax}(G_{mlp}(F(\mathbf{x})))$
        $\bar{y} \leftarrow \mathtt{softmax}(G_{gcn}(F(\mathbf{x})))$
        $\hat{d} \leftarrow \mathtt{sigmoid}(D(F(\mathbf{x})))$
        update $\psi$ by $\min_{\psi} \lambda_{adv}\ell_{adv}$
        update $\theta$, $\phi$ by $\min_{\theta,\phi} \ell_{ce}^{mlp} - \lambda_{adv}\ell_{adv}$
        update $\theta$, $\varphi, \varphi'$ by
         $\min_{\theta,\varphi,\varphi'} \lambda_{edge}\ell_{bce}^{edge} + \lambda_{node}\ell_{ce}^{node}$
    end
    Stage 3: Pseudo-labeling stage
    $\mathcal{D}_t^{\mathbb{D}^q} \leftarrow \{\}$                        ▷ Empty list
    for $\mathbf{x}_{t,j} \in \mathcal{T}_{\mathbb{D}^q}$ do
        $w_j \leftarrow \max_{c \in n_c} p(\bar{y}_{t,j} = c|\mathbf{x}_{t,j})$
        if $w_j > \tau$ then
            $\mathcal{D}_t^{\mathbb{D}^q} \leftarrow \mathcal{D}_t^{\mathbb{D}^q} || \{(\mathbf{x}_{t,j}, \mathrm{argmax}_{c \in n_c}\, p(\bar{y}_{t,j} =$
             $c|\mathbf{x}_{t,j}))\}$                         ▷ Append
        end
    end
    $\hat{\mathcal{S}}^{q+1} \leftarrow \hat{\mathcal{S}}^q \cup \mathcal{D}_t^{\mathbb{D}^q}$           ▷ Pseudo-source
    $\hat{\mathcal{T}}^{q+1} = \hat{\mathcal{T}}^q \setminus \mathcal{T}_{\mathbb{D}^q}$
end
Step 3: Fine-tuning on pseudo-source dataset
for $k'$ in $(1 : K')$ do
    $(\mathbf{x}_{s,i}, y_{s,i})_{i=1}^{B} \sim \hat{\mathcal{S}}^Q$
    update $\theta$, $\phi$ by $\min_{\theta,\phi} \ell_{ce}^{mlp}$
end
```



(a) Photo → *rest* in the PACS

(a) Product → *rest* in the Office-Home

Figure 1. The classification accuracy line plots with the D-CGCT using ResNet-50 as the backbone. At each indicated training iteration in the x-axis, a new target domain (shown in brackets) is selected for adaptation.

Layer	$k_{size}, C_{in}, C_{out}$, st, pad	IN/BN	Non-linearity	Dropout
Feature-extractor				
Conv1	(5, 3, 32, 1, 0)	IN/BN	ReLU	0.2
Maxpool2d	(2, -, -, 2, -)	-	-	-
Conv2	(5, 32, 64, 1, 0)	BN	ReLU	0.2
Maxpool2d	(2, -, -, 2, -)	-	-	-
FC3	(-, 64*4*4, 100, -, -)	BN	ReLU	0.2
FC4	(-, 100, 100, -. -)	BN	ReLU	-
Classifier				
FC_out	(-, 100, 10, -, -)	-	-	-
Domain-Discriminator				
D_FC1	(-, 100*10, 100, -, -)	-	ReLU	0.5
D_FC2	(-, 100, 100, -, -)	-	ReLU	0.5
D_FC3	(-, 100, 1, -, -)	-	-	-

Table 1. The network architecture for the baseline [8] used in the Digits-five experiments. Kernel size (k_{size}); in channels (C_{in}); out channels (C_{out}); stride (st); and padding (pad). IN stands for instance normalization. The input image resolution is 28 × 28 × 3.

D. Implementation Details

General Setting. To be fairly comparable with the state-of-the-art methods, we adopted comparable backbone feature extractors in the corresponding experiments and datasets. For Digits-five, we have used a small convolutional network as the backbone feature extractor (see Tab. 1), which is adapted from [1] and includes two *conv* layers and two *fc* layers. We trained the model using a Stochastic Gradient Descent (SGD) optimizer with an initial learning rate of 1e-3. For the rest of the datasets, we have adoptd ResNet [5] based feature extractors. Specifically, for the ablation studies on Office-Home, we have used ResNet-18 as the backbone network. For the state-of-the-art comparisons on Office31, PACS and Office-Home we have used ResNet-50. For the DomainNet, we have utilized ResNet-101 as used by the competitor methods. Similarly to the Digits-five, SGD optimizer is used with an initial learning rate of 1e-3 and is decayed exponentially. Each curriculum step consists of $K = 10,000$ training iterations for all the datasets, except the DomainNet, where $K = 50,000$ due to large size of the dataset. The final finetuning step is trained with $K' = 15,000$ iterations for all datasets.

GCN architecture. We have implemented f_{node} network with 2 conv layers followed by a Batch Normalization (BN) layer and ReLU activation, except the final layer. The first layer takes as input image features concatenated with the context of the mini-batch, *i.e.*, the aggregated features of other images in a mini-batch (based on the affinity matrix estimated by the f_{edge}). The second conv layer outputs the logits that are equal to the number of classes n_c. We have used 1x1 convolution kernels in the f_{node}. Similarly, we have implemented the f_{edge} network with 3 conv layers and 1x1 kernels, where the first two layers are followed by the BN layers and ReLU activations, except the last. The third conv layer has a single channel as output, thus, representing the similarity scores between samples in a mini-batch.

E. Additional Experiments

E.1. Ablations

To explain why the step-by-step adaptation in the proposed DCL better addresses the alleviation of the larger domain-shifts in the MTDA setting, we plot the classification accuracy with the D-CGCT in Fig. 1. As can be observed from the Fig. 1 (a), for Photo → *rest* setting in the PACS, when the adaptation first begins with the Art as target, the performance of the model on the *unseen* Cartoon domain simultaneously improves in the first 10k iterations (or the 1st curriculum step), despite the network not seeing any sample from the Cartoon domain. This phenomenon

Setting	Model	Digits-five					
		mt $\rightarrow$ mm,sv,sy,up	mm $\rightarrow$ mt,sv,sy,up	sv $\rightarrow$ mm,mt,sy,up	sy $\rightarrow$ mm,sv,mt,up	up $\rightarrow$ mm,sv,sy,mt	**Avg** (%)
Target Combined	Source only	26.9	56.0	67.2	73.8	36.9	52.2
	ADDA [17]	43.7	55.9	40.4	66.1	34.8	48.2
	DAN [9]	31.3	53.1	48.7	63.3	27.0	44.7
	GTA [16]	44.6	54.5	60.3	74.5	41.3	55.0
	RevGrad [3]	52.4	64.0	65.3	66.6	44.3	58.5
	AMEAN [1]	**56.2**	65.2	67.3	71.3	47.5	61.5
	CDAN [8]	53.0	76.3	65.6	81.5	**56.2**	66.5
	CGCT	54.3	**85.5**	**83.8**	**87.8**	52.4	**72.8**
Multi-Target	CDAN [8]	53.7	76.2	64.4	80.3	46.2	64.2
	CDAN + DCL	62.0	87.8	87.8	92.3	**63.2**	78.6
	D-CGCT	**65.7**	**89.0**	**88.9**	**93.2**	62.9	**79.9**

Table 2. Comparison with the state-of-the-art methods on the Digits-five. "Target Combined" indicates methods are performed on one source to one combined target domain. "Multi-Target" indicates methods are performed on one source to multi-target setting. Our proposed models are highlighted in bold.

Setting	Model	PACS						
		A $\rightarrow$ S	A $\rightarrow$ C	A $\rightarrow$ P	P $\rightarrow$ S	P $\rightarrow$ C	P $\rightarrow$ A	**Avg** (%)
Target Combined	MSTN [21]	70.4	71.2	96.2	**55.9**	**49.1**	70.8	68.9
	ADDA [17]	65.3	68.0	96.0	48.8	47.1	67.3	65.4
	CDAN [8]	56.8	61.1	95.9	55.7	53.8	49.4	62.1
	CGCT	**70.5**	**75.4**	**98.3**	44.6	44.3	**81.7**	**69.1**
Multi-Target	CDAN [8]	75.9	81.9	95.4	51.3	61.7	65.0	71.9
	HGAN [20]	72.1	78.3	97.7	70.8	62.8	78.8	76.8
	CDAN + DCL	68.7	89.0	98.8	61.2	**82.9**	**89.8**	81.7
	D-CGCT	**84.6**	**90.2**	**99.4**	**76.5**	82.4	88.6	**87.0**

Table 3. Comparison with the state-of-the-art methods on the PACS. All methods use the ResNet-50 as the backbone. "Target Combined" indicates methods are performed on one source to one combined target domain. "Multi-Target" indicates methods are performed on one source to multi-target setting. Our proposed models are highlighted in bold.

is even vividly noticeable in the second curriculum step, where the performance on the unseen Sketch largely increases when the Cartoon is selected for adaptation. This in other words means that the domain-shift between the source (Photo) and the farthest target (Sketch) has already been considerably reduced by the time the Sketch enters the adaptation stage (from 20k iterations on wards). Thus, we empirically demonstrate the prime reason behind the DCL achieving superior performance over other state-of-the-art MTDA methods. Similar observations can also be noticed for the Office-Home. We depict the Product → *rest* setting in the Fig. 1 (b).

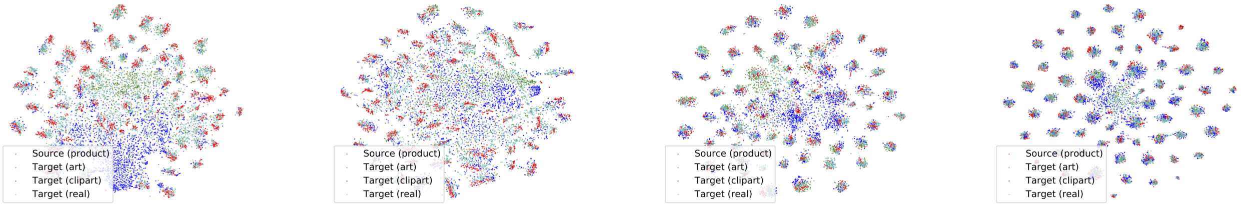
E.2. Comparison with the State-of-the-Art

In this section we compare with the state-of-the-art methods for the Digits-five and PACS. Since, the recent work of MTDA, HGAN [20], does not report results with all the domains available in the PACS and the DomainNet, we additionally report the results with those selected domains in this section for a fair comparison. In the Tab. 2, 3 and 4, we club the baselines into two distinct settings: target combined and multi-target. In the former setting, the domain labels of the targets are latent, and all the target domains are combined into a single target domain. Whereas in the latter, each target domain is treated separately. For both the settings, we just train one single model for a given *source* → *rest*, as in HGAN [20].

In the Tab. 2, we report the state-of-the-art comparison on the Digits-five. For a fair comparison, we compare with the baselines reported in [1] that use a backbone network similar to the one described in the Tab. 1. In both the target combined and multi-target settings, our proposed methods outperform all other baselines. For the PACS, reported in the Tab. 3, we notice that domain labels is very vital for mitigating multiple domain-shifts. For example, CDAN in the multi-target setting performs 9.8% better than its target combined counterpart. Similar trend can also be observed between our CGCT and D-CGCT, with the D-CGCT out-

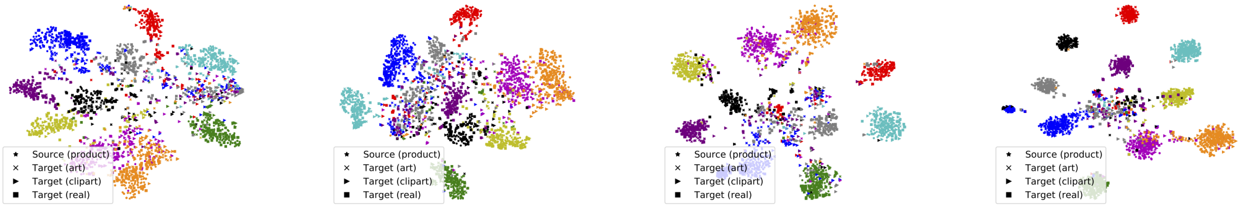
		DomainNet								
Setting	Model	R → S	R → C	R → I	R → P	P → S	P → R	P → C	P → I	**Avg** (%)
Target Combined	MSTN [21]	31.4	40.2	14.9	40.5	31.5	48.3	32.2	13.0	31.5
	ADDA [17]	27.5	33.9	12.7	35.0	26.2	41.7	26.9	10.7	26.8
	CDAN [8]	40.8	52.7	21.5	48.7	37.8	57.8	44.1	17.7	40.1
	CGCT	**48.9**	**60.3**	**26.9**	**57.1**	**43.4**	**58.8**	**48.5**	**21.7**	**45.7**
Multi-Target	CDAN [8]	40.7	51.9	22.5	49.0	39.6	57.9	44.6	18.4	40.6
	HGAN [20]	34.3	43.2	17.8	43.4	35.7	52.3	35.9	15.6	34.7
	CDAN + DCL	45.2	58.0	23.7	54.0	45.0	**61.5**	50.7	20.3	44.8
	D-CGCT	**48.4**	**59.6**	**25.3**	**55.6**	**45.3**	58.2	**51.0**	**21.7**	**45.6**

Table 4. Comparison with the state-of-the-art methods on the DomainNet. All methods use the ResNet-101 as the backbone. “Target Combined” indicates methods are performed on one source to one combined target domain. “Multi-Target” indicates methods are performed on one source to multi-target setting. Our proposed models are highlighted in bold.



(a) CDAN [8] (b) CDAN (**w/** domain labels) (c) CGCT (d) D-CGCT

Figure 2. *t*-SNE plots of the feature embeddings for the Product → *rest* of the Office-Home. All the models use ResNet-50 as backbone. Each colour indicates a different domain.



(a) CDAN [8] (b) CDAN (**w/** domain labels) (c) CGCT (d) D-CGCT

Figure 3. *t*-SNE plots of the feature embeddings for the Product → *rest* of the Office-Home depicting only 10 randomly sampled classes. All the methods use ResNet-50 as backbone. Each colour indicates a different class while each shape represents a different domain.

performing the former by a large margin. Finally, we re-evaluate our methods on the 5 domains of the DomainNet, by leaving out the Quickdraw domain as in [20]. Results are reported in the Tab. 4. We produce state-of-the-art performance in the DomainNet for both the settings by non-trivial margins. This further shows that our proposed feature aggregation and training strategy are much more effective than the HGAN.

E.3. Visualization

In this section we visualize the features learned by our models and compare them with the baseline methods. The Fig. 2 depicts the *t*-SNE plots of the feature embeddings computed by feature extractor network (ResNet-50) for the direction Product → *rest* of the Office-Home. The plots in the Fig. 2 (c) and (d) demonstrate that the proposed CGCT and D-CGCT result in well clustered and discriminative features compared to CDAN baselines (see Fig. 2 (a) and (b)).

To better visualize the decision boundaries in the latent feature space, we select 10 classes, randomly from the Office-Home, and depict the *t*-SNE plots of the feature embeddings in the Fig. 3. It is can be seen that our models learn features that can be easily separated by a linear classifier, much easier than the CDAN models. In particular, the CDAN when using domain labels (see Fig. 3 (b)) produces more overlapping classes than our D-CGCT (see Fig. 3 (d)). Thus, when the domain labels are leveraged with our DCL strategy, the model produces features that are more discriminative, thereby leading to an improved performance in the MTDA.

References

[1] Ziliang Chen, Jingyu Zhuang, Xiaodan Liang, and Liang Lin. Blending-target domain adaptation by adversarial meta-adaptation networks. In *Proc. CVPR*, 2019. 1, 3, 4

[2] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*. Springer series in statistics New York, 2001. 1

[3] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 2016. 1, 4

[4] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Proc. NeurIPS*, 2018. 1

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 3

[6] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 1998. 1

[7] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proc. ICCV*, 2017. 1

[8] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Proc. NeurIPS*, 2018. 3, 4, 5

[9] Mingsheng Long and Jianmin Wang. Learning transferable features with deep adaptation networks. In *Proc. ICML*, 2015. 4

[10] Yadan Luo, Zijian Wang, Zi Huang, and Mahsa Baktashmotlagh. Progressive graph learning for open-set domain adaptation. *In Proc. ICML*, 2020. 1

[11] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 1

[12] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proc. ICCV*, 2019. 1

[13] Xingchao Peng, Zijun Huang, Ximeng Sun, and Kate Saenko. Domain agnostic learning with disentangled representations. *arXiv*, 2019. 2

[14] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Proc. ECCV*, 2010. 1

[15] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proc. CVPR*, 2018. 1

[16] Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *Proc. CVPR*, 2018. 4

[17] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Adversarial discriminative domain adaptation. In *Proc. CVPR*, 2017. 4, 5

[18] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proc. CVPR*, 2017. 1

[19] Ruijia Xu, Ziliang Chen, Wangmeng Zuo, Junjie Yan, and Liang Lin. Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. In *Proc. CVPR*, 2018. 1

[20] Xu Yang, Cheng Deng, Tongliang Liu, and Dacheng Tao. Heterogeneous graph attention network for unsupervised multiple-target domain adaptation. *TPAMI*, 2020. 4, 5

[21] Xu Zhang, Felix Xinnan Yu, Shih-Fu Chang, and Shengjin Wang. Deep transfer network: Unsupervised domain adaptation. *arXiv*, 2015. 4, 5