Curriculum Graph Co-Teaching for Multi-Target Domain Adaptation

Subhankar Roy\textsuperscript{1,2*}, Evgeny Krivosheev\textsuperscript{1*}, Zhun Zhong\textsuperscript{\dagger}, Nicu Sebe\textsuperscript{1}, Elisa Ricci\textsuperscript{1,2}

\textsuperscript{1}University of Trento, Italy \textsuperscript{2}Fondazione Bruno Kessler, Italy

Project Page: https://roysubhankar.github.io/graph-coteaching-adaptation

Abstract

In this paper we address multi-target domain adaptation (MTDA), where given one labeled source dataset and multiple unlabeled target datasets that differ in data distributions, the task is to learn a robust predictor for all the target domains. We identify two key aspects that can help to alleviate multiple domain-shifts in the MTDA: feature aggregation and curriculum learning. To this end, we propose Curriculum Graph Co-Teaching (CGCT) that uses a dual classifier head, with one of them being a graph convolutional network (GCN) which aggregates features from similar samples across the domains. To prevent the classifiers from over-fitting on its own noisy pseudo-labels we develop a co-teaching strategy with the dual classifier head that is assisted by curriculum learning to obtain more reliable pseudo-labels. Furthermore, when the domain labels are available, we propose Domain-aware Curriculum Learning (DCL), a sequential adaptation strategy that first adapts on the easier target domains, followed by the harder ones. We experimentally demonstrate the effectiveness of our proposed frameworks on several benchmarks and advance the state-of-the-art in the MTDA by large margins (e.g. +5.6\% on the DomainNet).

1. Introduction

Deep learning models suffer from the well known drawback of failing to generalize well when deployed in the real world. The gap in performance arises due to the difference in the distributions of the training (a.k.a source) and the test (a.k.a target) data, which is popularly referred to as domain-shift \cite{46}. Since, collecting labeled data for every new operating environment is prohibitive, a rich line of research, called Unsupervised Domain Adaptation (UDA), has evolved to tackle the task of leveraging the source data to learn a robust predictor on a desired target domain.

In the literature, UDA methods have predominantly been designed to adapt from a single source domain to a single target domain (STDA). Such methods include optimizing statistical moments \cite{48, 26, 45, 35, 4, 5, 6, 36}, adversarial training \cite{11, 47, 25}, generative modelling \cite{39, 17, 23}, to name a few. However, given the proliferation in unlabeled data acquisition, the need to adapt to just a single target domain has lost traction in the real world scenarios. As the number of target domains grows, the number of models that need to be trained also scales linearly. For this reason, the research focus has very recently been steered to address a more practical scenario of adapting simultaneously to multiple target domains from a single source domain. This adaptation setting is formally termed as Multi-target Domain Adaptation (MTDA). The goal of the MTDA is to learn more compact representations with a single predictor that can perform well in all the target domains. Straightforward application of the STDA methods for the MTDA may be sub-optimal due to the presence of multiple domain-shifts, thereby leading to negative transfer \cite{54, 9}. Thus, the desideratum to align multiple data distributions makes the MTDA considerably more challenging.

In this paper we build our framework for the MTDA pivoted around two key concepts: feature aggregation and curriculum learning. Firstly, we argue that given the intrinsic nature of the task, learning robust features in a unified space is a prerequisite for attaining minimum risk across multiple target domains. For this purpose we propose to represent the source and the target samples as a graph and then leverage Graph Convolutional Networks \cite{19} (GCN) to aggregate semantic information from similar samples in a neighborhood across different domains. For the GCN to be operative, partial relationships among the samples (nodes) in the graph must at least be known apriori in the form of class labels. However, this information is absent for the target samples. To this end, we design a co-teaching framework where we train two classifiers: a MLP classifier and a GCN classifier that provide target pseudo-labels to each other. On the one hand, the MLP classifier is utilized to make the GCN learn the pairwise similarity between two nodes in the graph. While, on the other hand, the GCN classifier, due to its feature aggregation property, provides better pseudo-labels to assist the training of the MLP classifier. Given that co-teaching works on the assumption that differ-
ent networks capture different aspects of learning [2], it is beneficial for suppressing noisy pseudo-labels. Its feature aggregation and/or co-teaching aspects are largely missing in existing MTDA methods [9, 12, 34, 53] (see Tab. 1).

Secondly, we make a crucial observation, very peculiar to the MTDA setting, i.e., during training as the network tries to adapt to multiple domain-shifts of varying degree, pseudo-labels obtained on-the-fly from the network for the target samples are very noisy. Self-training the network with unreliable pseudo-labeled target data further deteriorates the performance. To further combat the impact of noisy pseudo-labels, we propose to obtain pseudo-labels in an episodic fashion, and advocate the use of curriculum learning in the context of MTDA. In particular, when the domain labels of the target are silent, each episode of curriculum step consists of a fixed number of training iterations. Fairly consistent and reliable pseudo-labels are obtained from the GCN classifier at the end of each curriculum step. We call this proposed framework as Curriculum Graph Co-Teaching (CGCT) (see Fig. 1 (a)).

Furthermore, when the domain labels of the target are available, we propose an Easy-To-Hard Domain Selection (EHDS) strategy where the feature alignment process begins with the target domain that is closest to the source and then gradually progresses towards the hardest one. This makes adaptation to multiple targets smoother. In this case, each curriculum step involves adaptation with a single new target domain. The CGCT when combined with this proposed Domain-aware Curriculum Learning (DCL) (see Fig. 1 (b)) is referred to as D-CGCT. The Tab. 1 highlights the operating regimes of our frameworks versus the state-of-the-art MTDA methods. To summarize, the contributions of this work are threefold:

- We propose Curriculum Graph Co-Teaching (CGCT) for MTDA that exploits the co-teaching strategy with the dual classifier head, together with the curriculum learning, to learn more robust representations across multiple target domains.
- To better utilize the domain labels, we propose a Domain-aware Curriculum Learning (DCL) strategy to make the feature alignment process smoother.
- In the MTDA setting, we outperform the state-of-the-art for several UDA benchmarks by significant margins (including +5.6% on the large scale DomainNet [33]).

### 2. Related Works

Single-source single-target DA (STDA) refers to the task of adapting a classifier from a single labeled source dataset to a single unlabeled target dataset. In the UDA literature, a plethora of STDA methods have been proposed, which can be broadly classified into three major categories based upon the adaptation strategy. The first category uses first (Maximum Mean Discrepancy [48, 26, 27, 49]) or second order (correlation alignment [45, 31, 35, 22, 4, 6, 30, 36, 38]) statistics of the source and target features to align the marginal feature distributions. The second category of STDA methods [11, 47, 3, 25, 8] adopts adversarial training strategy to align the marginal feature distributions of the two domains. Essentially, these methods use a gradient reversal layer [11] to make the feature extractor network agnostic to domain specific information. The final category of STDA methods [39, 17, 42, 23] resort to pixel-level adaptation by generating synthetic target-like source images or source-like target images with the help of generative adversarial network (GAN) [13]. However, practical applications go beyond the single-source and single-target setting and often involve multiple source [51, 37, 52] or target domains.

Multi-target DA aims to transfer knowledge from a single labeled source dataset to multiple unlabeled target datasets. While the research in STDA is quite mature, most STDA methods cannot be trivially extended to a multi-target setting. So far only a handful of methods [9, 34, 18, 24, 12, 53] for MTDA can be found in the literature. AMEAN [9] performs clustering on the blended target domain samples to obtain sub-targets and then learns domain-invariant features from the source and the obtained sub-targets using a STDA method [43]. The approaches introduced in [34, 18, 12] are derived from STDA and do not exploit any peculiarity of the MTDA setting. Conversely, our CGCT and D-CGCT are tailor-made for the multi-target setting as we propose to use feature aggregation of similar samples across multiple domains.

Curriculum for DA involves adopting an adaptive strategy that evolves over time to better address the adaptation across domains. Shu et. al [44] propose a strategy based on curriculum learning that exploits the loss of the network as weights to identify and eliminate unreliable source samples. An Easy-to-Hard Transfer Strategy (EHTS) is proposed in PFAN [7] that progressively selects the pseudo-labeled target samples which have higher cosine similarity to the per-category source prototypes. Similarly, our CGCT is inspired by the EHTS strategy except we progressively recruit the pseudo-labeled targets [1] from the robust GCN classification head to better train the MLP classifier, which in turn regularizes the GCN head (see Sec. 3.2). For the multi-source DA setting, CMSS [52] trains a separate network to weigh the most relevant samples across several source domains for adapting to a single target domain. However, dif-

<table>
<thead>
<tr>
<th>Method</th>
<th>Domain labels</th>
<th>Feature aggregation</th>
<th>Curriculum learning</th>
<th>Co-teaching</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMEAN [9]</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>DADA [34]</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>MTDATA [12]</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>HGAN [53]</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>CGCT (Ours)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>D-CGCT (Ours)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 1. Comparison with recent the state-of-the-art MTDA methods in terms of the operating regimes.
ferently from CMSS, our proposed DCL utilizes the domain information to adapt over time from the easiest to the hardest target domain in the MTDA setting (see Sec. 3.3).

**Graph Neural Networks** (GNN) are neural network models applied on graph-structured data that can capture the relationships between the objects (nodes) in a graph via message passing through the edges [14, 50]. Relevant to our work are GNN-derived Graph Convolutional Networks (GCN) [19] that have recently been applied for addressing DA [29, 28, 53]. For instance, Luo et al. [28] propose PGL for open-set DA to capture the relationship between the overlapping classes in the source and the target. Notably, Yang et al. [53] introduce heterogeneous Graph Attention Network (HGAN) for MTDA to learn the relationship of similar samples among multiple domains and then utilize the graph-based pseudo-labeled target samples to align their centroids with that of the source. Unlike [28, 53], we incorporate the idea of co-teaching [15] in a GCN framework for combating noisy pseudo-labels.

### 3. Method

In this section we present our proposed Curriculum Graph Co-Teaching (CGCT) and thereafter Domain Curriculum Learning (DCL) for the task of MTDA. We also discuss some preliminaries that are used to address the task.

**Problem Definition.** In the MTDA scenario, we are provided with a single source dataset $S = \{(x_{s,i}, y_{s,i})\}_{i=1}^{n_s}$, containing $n_s$ labeled samples, and $N$ unlabeled target datasets $T = \{T_j\}_{j=1}^{N}$, where $T_j = \{x_{j,k}\}_{k=1}^{n_j}$ with each containing $n_j$ unlabeled samples. As in any DA scenario, the fundamental assumption is that the underlying data distributions of the source and the targets are different from each other. It is also assumed that the label space of the source and targets are the same. Under these assumptions, the goal of the MTDA is to learn a single predictor for all the target domains by using the data in $S \cup \{T_j\}_{j=1}^{N}$.

#### 3.1. Preliminaries

**Baseline for Multi-target Domain Adaptation.** Domain Adversarial Network (DANN) [11], originally designed for STDA, aligns the feature distributions of the source and the target domains by using an adversarial training. DANN comprises of three networks: the feature extractor, the classifier and the domain discriminator. The classifier is responsible for classifying the features obtained from the feature extractor into $n_c$ classes. On the one hand, the domain discriminator distinguishes the source from the target features. While on the other hand, the feature extractor is trained to fool the discriminator and simultaneously learn good features for semantic classification.

Formally, let $F_\theta : \mathbb{R}^{3xwxh} \rightarrow \mathbb{R}^d$ be the feature extractor network, parameterized by $\theta$, that outputs a feature $f = F(x)$ for a given sample $x$. The classifier network, parameterized by $\phi$, is denoted by $G_\phi : \mathbb{R}^d \rightarrow \mathbb{R}^{n_c}$, which takes as input a feature $f$ and outputs class logits $g = G(f)$. The discriminator network $D_\psi : \mathbb{R}^d \rightarrow \mathbb{R}^1$, parameterized by $\psi$, takes in the same feature $f$ and outputs a single logit. By treating all the target domains as one combined target domain, the overall training objective of DANN for MTDA is given by:

$$
\max_{\psi} \min_{\theta, \phi} \ell_{ce} - \lambda_{adv} \ell_{adv}, \quad (1)
$$

where $\ell_{ce} = -E_{(x_{s,i}, y_{s,i}) \sim S} \log F(x_{s,i}))$, and $\ell_{adv} = -E_{x_{s,i} \sim S} \log D(F(x_{s,i})) - E_{x_{t,j} \sim T} \log [1 - D(F(x_{t,j}))]$.

$y_{s,i}$ is the one-hot label for a source label $y_{s,i}$. The first term, $\ell_{ce}$, in Eq. 1 is the cross-entropy loss computed on the source domain samples and minimized w.r.t. $\theta, \phi$. The second term, $\ell_{adv}$, in Eq. 1 is the adversarial loss that is maximized w.r.t $\psi$, but minimized w.r.t $\theta, \lambda_{adv}$ is the weighting factor for $\ell_{adv}$. To capture the multi-modal nature of the distributions, CDAN [25] is proposed where $D$ can be additionally conditioned on the classifier predictions $g$. In CDAN [25], the $D$ takes as input $h = (f, g)$, the joint variable of $f$ and $g$, instead of just $f$. In this work we use CDAN for aligning the feature distributions.

**Graph Convolutional Network.** For the GCN [19] classifier we construct an undirected and fully-connected graph $\Gamma = (V, E, \mathcal{A})$ from all samples in mini-batch. In details, given a mini-batch of images, we represent each image $x_i$ as a node $v_i \in V$ in the graph $\Gamma$. $e_{i,j} \in E$ indicates an edge between nodes $v_i$ and $v_j$, and $a_{i,j}$ is the semantic similarity score for nodes $\langle v_i, v_j \rangle$ forming an affinity matrix $\mathcal{A}$.

Following [28], we compute the semantic similarity scores $\hat{a}_{i,j}^{(l)}$ at the $l$-th layer for all pairs $(v_i, v_j) \in E$:

$$
\hat{a}_{i,j}^{(l)} = f_{\text{edge}}^{(l)}(\mathbf{v}_i^{(l-1)}, \mathbf{v}_j^{(l-1)}), \quad (2)
$$

where $f_{\text{edge}}^{(l)}$ is a non-linear similarity function parameterized by $\varphi$, and $\mathbf{v}_i^{(l-1)}$ is features at $l$-1 GCN layer of a sample $v_i$. The initial node features $\mathbf{v}_i$ are instantiated with $f_i$, the embedding obtained from $F$. Then, we add self-connections for nodes in the graph and normalize the obtained similarity scores as:

$$
\mathcal{A}^{(l)} = M^{-\frac{1}{2}}(\hat{A}^{(l)} + I)M^{-\frac{1}{2}}, \quad (3)
$$

where $M$ is the degree matrix, $I$ is the identity matrix, and $\hat{A}$ is the un-normalized affinity matrix.

Finally, given the affinity matrix $\mathcal{A}^{(l)}$, we update the node features with the following propagation rule:

$$
\mathbf{v}_i^{(l)} = f_{\text{node}}^{(l)} \left( \mathbf{v}_i^{(l-1)}, \sum_{j \in B} a_{i,j}^{(l)} \mathbf{v}_j^{(l-1)} \right), \quad (4)
$$

where $f_{\text{node}}^{(l)}$ is a non-linear function parameterized by $\varphi'$, $B$ is a set of samples in the mini-batch, and $[\cdot, \cdot]$ is the feature
concatenation function. The final $f^{(L)}_{\text{node}}$ layer is the output layer with $n_c$ outputs. We slightly abuse the notations and drop the superscript $i$ in our subsequent formulations for the sake of clarity.

### 3.2. Curriculum Graph Co-Teaching

In this work we introduce the Curriculum Graph Co-Teaching (CGCT) that employs feature aggregation with a GCN and uses curriculum learning for pseudo-labeling. In details, as shown in Fig. 1(a), it is composed of: a feature extractor $F$, a domain discriminator $D$, a MLP classifier $G_{\text{mlp}}$ and a GCN classifier $G_{\text{gen}}$. The $G_{\text{mlp}}$ is a fully-connected output layer with $n_c$ output logits. The $G_{\text{gen}}$ consists of an edge network $f_{\text{edge}}$ and a node classifier $f_{\text{node}}$. The $f_{\text{node}}$ aggregates the features of the samples in $\mathcal{B}$ by considering the learnt pairwise similarity in the affinity matrix $\hat{A}$ produced by the $f_{\text{edge}}$. The $G_{\text{gen}}$ also outputs $n_c$ logits. Since, the $G_{\text{mlp}}$ and the $G_{\text{gen}}$ capture different aspects of learning, they are exploited to provide feedback to each other in a co-teaching fashion. The CGCT is trained for $Q$ curriculum steps where a curriculum step, $t^q_{\text{cur}}$, is an episode in which the network is trained for $K$ training iterations. Each curriculum step $t^q_{\text{cur}}$ is further decomposed into two stages: i) Adaptation stage and ii) Pseudo-labeling stage. Each stage in a $t^q_{\text{cur}}$ is described below. Note that, as in [9], we assume that the domains labels of the target are latent and not observed during training.

**Adaptation stage.** In this stage we mainly perform the feature alignment using CDAN [25]. In details, initially at step $t^0_{\text{cur}}$, we start with a source set $\mathcal{S}^0 = \{S\}$ and a target set $T$. We sample mini-batches $\mathcal{B}^0 = \{\mathcal{B}^{0}_{s,i}, \mathcal{B}^{0}_{t,i}\}_{i=1}^T$ with size $B$ such that $\mathcal{B}^{0}_{s,i} \sim \mathcal{S}$ and $\mathcal{B}^{0}_{t,i} \sim \mathcal{T}$. Each mini-batch of images is first fed to the feature extractor $F$ to obtain $\mathcal{F}^0 = \{f^{0}_{s,i}, f^{0}_{t,i}\}_{i=1}^B$, which are then simultaneously fed to both the $G_{\text{mlp}}$ and $G_{\text{gen}}$. When fed to the $G_{\text{mlp}}$ it outputs the logits $\hat{G}^0 = \{\hat{g}^{0}_{s,i}, \hat{g}^{0}_{t,i}\}_{i=1}^B$. On the other hand, $\mathcal{F}^0$ are input to the $f_{\text{edge}}$ to estimate the pairwise similarity of the samples in $\mathcal{B}^0$. Specifically, the $f_{\text{edge}}$ outputs an affinity matrix $A$ following Eq. 2, where the entries $\hat{a}_{i,j}$ denote the strength of similarity between samples $i$ and $j$ in $\mathcal{B}^0$. Intuitively, higher the value of $\hat{a}_{i,j}$, higher is the likelihood of samples $i$ and $j$ belonging to the same semantic category. Finally, following Eq. 4, the $f_{\text{node}}$ aggregates the features in $\mathcal{F}^0$ based on the estimated $A$ such that for each node the most similar samples in the neighbourhood contribute more to its final representation. Subsequently, the $f_{\text{node}}$ outputs its logits as $\tilde{G}^0 = \{\tilde{g}^0_{s,i}, \tilde{g}^0_{t,i}\}_{i=1}^B$. The elements in $\hat{G}^0$ and $\tilde{G}^0$ are then passed through a softmax function to obtain the probabilities for each sample as $p(\hat{y}_{i} = c|\hat{g})$ and $p(\tilde{y}_{i} = c|\tilde{g})$ where $\hat{y}$ and $\tilde{y}$ are the predictions, respectively.

To guide the $f_{\text{edge}}$ to learn the pairwise similarity between the samples in $\mathcal{B}^0$ we propose the concept of co-teaching where the $G_{\text{mlp}}$ provides feedback to the $f_{\text{edge}}$. Since, $G_{\text{mlp}}$ makes instance-level independent predictions on the samples in $\mathcal{B}^0$, it is not susceptible to the accumulation of potential noise from the dissimilar neighbours. To this end, for a $\mathcal{B}^0$ we construct a "target" affinity matrix $\hat{A}^\text{tar}$ and enforce the predictions of $f_{\text{edge}}$ to be as close as possible to the $\hat{A}^\text{tar}$. Each entry $\hat{a}^\text{tar}_{i,j}$ in the $\hat{A}^\text{tar}$ is given by:

$$
\hat{a}^\text{tar}_{i,j} = \begin{cases} 1, & \text{if } y_i = y_j = c \\ 0, & \text{otherwise} \end{cases},
$$

where $c$ is the class label. While the class labels of $\mathcal{B}^0$ are provided as ground truth, we do not have access to the labels of $\mathcal{B}^1$. Therefore, a target domain sample $x_{n} \in \mathcal{B}^0$ is assigned a definitive pseudo-label $\hat{y}_{n,j} = c'$ where $c' = \arg\max_{c \in \mathcal{C}} p(\hat{y}_{n,j} = c|\hat{g}^{0}_{t,i})$ if the maximum likelihood $\max_{c \in \mathcal{C}} p(\hat{y}_{n,j} = c|\hat{g}^{0}_{t,i})$ is greater than a threshold $\tau$. The
entries $\hat{a}^\text{tar}_{t,j}$ involving $x_{t,j} \in B^0_t$ not passing the $\tau$ are not optimized during training. We train the $f_{\text{edge}}$ using a binary cross-entropy loss as:

$$
e_{\text{edge}} = \hat{a}^\text{tar}_{t,j} \log p(\hat{a}_{t,j}) + (1 - \hat{a}^\text{tar}_{t,j}) \log (1 - p(\hat{a}_{t,j})). (6)$$

Finally, for training the $G_{mlp}$ and the $f_{\text{node}}$ in the $G_{gen}$ we compute the standard cross-entropy loss with the samples in $B^0_t$ as:

$$
e_{\text{c}} = - \frac{1}{|B^0_t|} \sum_{i=1}^{|B^0_t|} \hat{y}_i \log p(\hat{y}_{s,i}|\tilde{g}^0_{s,i}), (7)$$

$$
e_{\text{node}} = - \frac{1}{|B^0_t|} \sum_{i=1}^{|B^0_t|} \hat{y}_i \log p(\hat{y}_{s,i}|\tilde{g}^0_{s,i}). (8)$$

We feed the features $\{(\hat{1}_t, \hat{0}_i): i = 1, \ldots, B\}$, corresponding to the $B^0_t$, to the domain discriminator $D$ and compute the conditional adversarial loss following Eq. 1. Thus, the final objective function for the CGCT can be written as:

$$\max_{\theta, \phi, \psi} \min_{\psi} e_{\text{c}} + \lambda_{\text{edge}} e_{\text{edge}} + \lambda_{\text{node}} e_{\text{node}} - \lambda_{\text{adv}} e_{\text{adv}}, (9)$$

where $\lambda_{\text{edge}}, \lambda_{\text{node}}$ and $\lambda_{\text{adv}}$ are the weighing factors.

**Pseudo-labeling stage.** Upon completion of the adaptation stage in a curriculum step $t_{\text{cur}}$, we put the network in inference mode and obtain pseudo-labels $\forall x_{t,j} \in T$. The $G_{gen}$ is employed for this task because, owing to its aggregating characteristics, it learns more robust features than the $G_{mlp}$. This is the curriculum aspect of our proposed co-teaching training strategy in CGCT where the obtained pseudo-labeled target samples are then used to train the $G_{mlp}$, besides the $f_{\text{node}}$.

At any step $t_{\text{cur}}$, the criterion for pseudo-label selection is formally written as:

$$\forall x_{t,j} \in T, w_j = \begin{cases} 1, & \text{if } \max_{c \in \text{enc}} p(\hat{y}_{t,j} = c|\tilde{g}_{t,j}) > \tau, \\ 0, & \text{otherwise}, \end{cases}$$

(10)

where $w_j = 1$ signifies that $x_{t,j}$ is selected with a pseudo-label $\hat{y}_{t,j} = c'$ where $c' = \underset{c \in \text{enc}}{\arg\max} p(\hat{y}_{t,j} = c|\tilde{g}_{t,j})$, whereas $w_j = 0$ denotes no pseudo-label is assigned. After the pseudo-labeling stage in a $t_{\text{cur}}$, we obtain a pseudo-labeled target set $D^t_{\text{cur}} = \{(x_{t,j}, \hat{y}_{t,j})\}_{j=1}^{N_t}$, where $N_t$ is the number of recruited pseudo-labeled target samples. Post pseudo-labeling we update and prepare the source set for the succeeding step $t_{\text{cur}}^{t+1}$ as:

$$S^{t+1} = S \cup D^t_{\text{cur}}. (11)$$

The update rule in Eq. 11 allows us to compute the supervised losses $e_{\text{node}}$ and $e_{\text{c}}$ from Eq. 9 for $x_{t,j} \sim D_t$. Note that we do not alter the domain labels in $D^t_{\text{cur}}$ and hence, the formulation for $\ell_{\text{adv}}$ remains unchanged.

At the culmination of $Q$ curriculum steps, $S^Q$ is obtained using Eq. 11 and the network is fine-tuned with only the supervised losses in Eq. 9 for $K$ training iterations.

### 3.3. Domain-aware Curriculum Learning

Now we consider the case when the domain labels of the target are available, i.e. $T = \{T_j\}_{j=1}^N$, $N$ being the number of target domains. In principle, when the domain labels are available, one can either train $N$ domain discriminators or a $(N+1)$ way single domain discriminator. Apart from over-parameterization, it also suffers from limited gradients coming from the discriminator(s) due to single point estimates [20]. Thus, we propose Domain-aware Curriculum Learning (DCL) as an alternate learning paradigm to better utilize the target domain labels in the MTDA setting.

To this end we design the DCL that is based on our proposed Easy-to-Hard Domain Selection (EHDS) strategy. Our proposal for the DCL stems from the observation that different target domains exhibit different domain shifts from the source domain, where some domain shifts are larger than the others. Evidently, the network will find it easier to adapt to the closest target domain while performing sub-optimally on the domain with the largest domain shift. When adaptation is performed with $N$ domains at tandem then the large domain shifts of harder domains will interfere with the feature alignment on the easier target domains, thereby compromising the overall performance. To overcome this problem, in the EHDS strategy, as the name suggests, the network performs feature adaptation one domain at a time, starting from the easiest target domain and gradually moving towards the hardest. The "closeness" of a target domain from the source is measured by the uncertainty in the target predictions with a source-trained model. Lesser the uncertainty in predictions, closer the target from the source domain. Therefore, measuring the entropy on a target domain can serve as a good proxy for domain selection, and is defined as:

$$H(T_j) = - \sum_{x_{t,j} \sim T_j} p(\hat{y}_{t,j}, k, c|x_{t,j}, k) \log p(\hat{y}_{t,j}, k, c|x_{t,j}, k).$$

(12)

Due to this step-by-step adaptation through domain traversal, the intermediate target domains help in reducing large domain shifts by making the farthest domain shift considerably closer than that at the start. Differently from the CGCT, in the DCL, each curriculum step, defined as $t_{\text{cur}}^{t_{\text{cur}}}$, consist in learning over one target domain, with a total of $N$ steps. Since, the simulation of single-source and single-target adaptation inside the MTDA setup yields better domain-invariant features, at the end of each $t_{\text{cur}}^{t_{\text{cur}}}$ we also consider extracting pseudo-labels for the target samples from the classifier and add them to the source set (see Fig. 1(b)) for computing the supervised losses. This further reduces the domain gaps for the forthcoming harder domains. The $t_{\text{cur}}^{t_{\text{cur}}}$ is split into three stages and are described below:

**Domain selection stage.** Given a source-trained model $F_{\theta^*}(G_{\phi^*})$, where $\theta^*$ and $\phi^*$ are the trained parameters of...
\( F \) and \( G \), and initial source and target sets \( S^0 = \{ S \} \) and \( T^0 = \{ T \}_j \), the closest target domain is selected as:
\[
\mathbb{D}^0 = \arg\min_j \{ H_j(T_j) \mid \forall T_j \in \hat{T}^0 \},
\]
where \( \mathbb{D}^0 \) is the target domain selected at step \( t^0_{del} \) and is used for performing adaptation in the subsequent stage.

**Adaptation stage.** This stage is similar to the one in \( t^0_{cur} \), described in Sec. 3.2, except the feature adaptation at any step \( t^0_{del} \) is performed using \( S^0 \cup T^T_0 \), rather than the entire target set \( T \). The model is trained using the losses described in Eq. 9.

**Pseudo-labeling stage.** The criterion for pseudo-label selection still remains the same, as described in Eq. 10, with the exception of target samples being drawn only from the current target domain \( \mathbb{D}^q \), yielding a pseudo-labeled target set \( \mathbb{D}^q_t \). Consequently, the source and target set update changes as following:
\[
\hat{S}^{q+1} = \hat{S}^q \cup \mathbb{D}^q_t,
\]
\[
\hat{T}^{q+1} = \hat{T}^q \setminus \mathbb{D}^q_t.
\]
These three stages are repeated until all \( N \) domains have been exhausted. Then similarly, as in CGCT, the final model is fine-tuned with \( S^q \). When CGCT is trained using the DCL strategy we refer to the model as D-CGCT. We would like to point that the DCL can also be realized with a single classifier model (see Sec. 4).

### 4. Experiments

#### 4.1 Dataset and Experimental Details

**Datasets.** We conduct experiments on five standard UDA benchmarks: Digits-five [51], Office-31 [40], PACS [21], Office-Home [49] and the very large scale DomainNet [33] (0.6 million images). The statistics of the datasets are summarized in Tab. 2. More details on the datasets can be found in the Supp. Mat.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#domains</th>
<th>#classes</th>
<th>#images</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digits-five</td>
<td>5</td>
<td>10</td>
<td>( \sim 145K )</td>
</tr>
<tr>
<td>PACS</td>
<td>4</td>
<td>7</td>
<td>9,991</td>
</tr>
<tr>
<td>Office-31</td>
<td>3</td>
<td>31</td>
<td>4,652</td>
</tr>
<tr>
<td>Office-Home</td>
<td>4</td>
<td>65</td>
<td>15,500</td>
</tr>
<tr>
<td>DomainNet</td>
<td>6</td>
<td>345</td>
<td>( \sim 0.6M )</td>
</tr>
</tbody>
</table>

Table 2. Dataset details for multi-target domain adaptation.

**Evaluation protocol.** We use the classification accuracy to evaluate the performance. The classification accuracy is computed for every possible combination of one source domain and the rest of the target domains. The performance for a given direction, \( \text{i.e., source} \rightarrow \text{rest} \), is given by averaging the accuracy on all the target domains, where \( \text{source} \) signifies the source domain and \( \text{rest} \) indicates all the unlabeled domains except the \( \text{source} \). Importantly, in all our experiments we always report the final classification accuracy obtained with the \( G_{mlp} \) because the \( G_{gen} \) always requires a mini-batch at inference, an assumption which is easily violated when deployed in the real world.

**Implementation details.** To be fairly comparable with the state-of-the-art methods, we adopt the backbone feature extractor networks used in [9, 53, 34] for the corresponding datasets. We train the networks by using a Stochastic Gradient Descent (SGD) optimizer having an initial learning rate of 1e-3 and decay exponentially. More details about the network architecture and experimental set-up can be found in the Supp. Mat.

**Hyperparameter selection.** In our final model we used only a single set of hyperparameters, which are \( \lambda_{edge} = 1, \lambda_{node} = 0.3, \lambda_{adv} = 1 \) and \( \tau = 0.7 \). Following the standard protocol in [43], we used a held-out validation set of 1000 samples for the MNIST \( \rightarrow \text{rest} \) direction to tune these hyper-parameters.

#### 4.2 Ablations

In this section we discuss the design choices of our proposed contributions and report the results of a thorough ablation study. Our ablation analysis highlights the importance of the graph co-teaching and the curriculum learning. We run the ablation experiments on Office-Home with ResNet-18 [16] as backbone network and on Digits-five with a network adopted from AMEAN [9]. We adopt the CDAN as a baseline for adaptation in Tab. 3 and Tab. 5.

<table>
<thead>
<tr>
<th>Model</th>
<th>Co-teaching</th>
<th>Pseudo-labels from</th>
<th>Avg(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>( \times )</td>
<td>self, ( G_{mlp} ), ( G_{mlp} )</td>
<td>57.4</td>
</tr>
<tr>
<td>M2</td>
<td>( \times )</td>
<td>( G_{gen} ), ( G_{gen} ), ( G_{gen} )</td>
<td>59.6</td>
</tr>
<tr>
<td>M3</td>
<td>( \checkmark )</td>
<td>self, ( G_{mlp} ), ( G_{gen} )</td>
<td>58.2</td>
</tr>
<tr>
<td>D-CGCT (Ours)</td>
<td>( \checkmark )</td>
<td>( G_{gen} ), ( G_{mlp} ), ( G_{gen} )</td>
<td><strong>60.8</strong></td>
</tr>
</tbody>
</table>

Table 3. Ablation study of different co-teaching strategies on Office-Home. We reported the classification accuracy averaged across all the source \( \rightarrow \text{rest} \) directions.

**Graph co-teaching.** The goal of this particular ablation study is to analyse why our proposed graph co-teaching is beneficial and the manner in which it should be realised in an adaptation framework. To this end, as shown in the Tab. 3, we design some baselines that can be distinguished in the manner in which the \( G_{mlp} \) and the \( G_{gen} \) provide pseudo-labels to the other (columns 3 to 5) and then compare it to our D-CGCT. In more details, the baseline models can be described as: i) M1: a baseline where the \( G_{mlp} \) provides pseudo-labels to itself, \( f_{edge} \) and \( f_{node} \) after each curriculum step \( t^q_{del} \); ii) M2: a baseline similar to M1, except that the \( G_{gen} \) provides the pseudo-labels; iii) M3: another baseline which is similar to M1 but with an exception that the \( G_{gen} \) also provides pseudo-labels to \( f_{edge} \) for the current target domain in an ongoing \( t^q_{del} \) step.
Unsurprisingly, M1 performs the worst of all the baselines because the pseudo-labels computed by the $G_{mlp}$ are less accurate due to $G_{mlp}$ not taking into account the feature aggregation from multiple domains. Contrarily, the baseline M2 performs better than the M1 due to the fact that M2 uses $G_{gcn}$ for pseudo-labeling, which are more accurate. This highlights the importance of feature aggregation in the MTDA setting. One other thing that separates D-CGCT from both M1 and M2 is the co-teaching, which is absent in the latter baselines. Since, the D-CGCT enables co-teaching, with the $G_{mlp}$ and the $G_{gcn}$ providing pseudo-labels to each other, it does not overfit on the same “incorrect” pseudo-label, thereby achieving more robust predictions. Contrarily, M3 uses co-teaching and yet it fails to achieve comparable performance. We speculate that, since the $f_{edge}$ is also trained with the pseudo-labels obtained from the $G_{gcn}$ for the current target domain in a $f_q^{th}$ step, it becomes susceptible to noise. Thus, in summary, the graph co-teaching is the most effective when the $G_{gcn}$ is exploited to provide pseudo-labels only after each curriculum step.

**Curriculum learning.** We also study the effect of domain-aware curriculum learning in isolation from co-teaching. For that purpose, as shown in the Tab. 5, we start with the baseline model CDAN by treating all the target domains as one single domain. When the domain labels of the target are available, the baseline improves by 1.33%, indicating that the domain labels can indeed improve the performance of an adaptation model. To show the benefit of the DCL without co-teaching, we train the Base$^{+}$ + DCL, and it yields an average accuracy that is higher than the Base$^{+}$ + PL counterpart. The advantage of using DCL is further amplified when coupled with the CGCT, where the D-CGCT outperforms all other baselines, including the CGCT. Due to the gradual adaptation, the D-CGCT also leads to the better cluster formation than the CGCT, as shown by the t-SNE visualization in the Fig. 2.

To demonstrate that the order of target domains selection in the DCL indeed makes a difference, we consider a reverse-domain curriculum learning where the hardest domain is selected first, followed by the less hard ones. To this end, we train two models: i) Baseline$^{+}$+DCL; and ii) Baseline$^{+}$+Rev-DCL and compare their performances in the Fig. 3. In both the datasets we observe the same phenomenon that the reverse-curriculum being detrimental. This once again re-establishes the importance of the proposed DCL in the MTDA setting.

### Table 4. Comparison with state-of-the-art methods on Office-31 and Office-Home. All methods use the ResNet-50 as the backbone. Single-Target indicates methods are performed on one source to one target setting. Target-Combined indicates methods are performed on one source to aggregated targets setting, while the Multi-Target indicates methods are performed on one source to multi-target setting.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Model</th>
<th>Office-31</th>
<th>Office-Home</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Amazon</td>
<td>Office-31</td>
</tr>
<tr>
<td>w/o Target</td>
<td>Source train</td>
<td>68.6</td>
<td>70.0</td>
</tr>
<tr>
<td>Single-Target</td>
<td>DAN [26]</td>
<td>79.5</td>
<td>80.3</td>
</tr>
<tr>
<td></td>
<td>RevGrad [11]</td>
<td>80.8</td>
<td>82.5</td>
</tr>
<tr>
<td></td>
<td>JAN [27]</td>
<td>85.0</td>
<td>83.0</td>
</tr>
<tr>
<td></td>
<td>CDAN [25]</td>
<td><strong>91.4</strong></td>
<td>84.1</td>
</tr>
<tr>
<td></td>
<td>CGCT (ours)</td>
<td>89.6</td>
<td><strong>85.5</strong></td>
</tr>
<tr>
<td>Target-Combined</td>
<td>DAN [26]</td>
<td>78.0</td>
<td>64.4</td>
</tr>
<tr>
<td></td>
<td>RevGrad [11]</td>
<td>78.2</td>
<td>72.2</td>
</tr>
<tr>
<td></td>
<td>JAN [27]</td>
<td>84.2</td>
<td>74.4</td>
</tr>
<tr>
<td></td>
<td>CDAN [25]</td>
<td>93.6</td>
<td>80.5</td>
</tr>
<tr>
<td></td>
<td>AMEAN [9]</td>
<td>90.1</td>
<td>77.0</td>
</tr>
<tr>
<td></td>
<td>CGCT (ours)</td>
<td><strong>93.9</strong></td>
<td><strong>85.1</strong></td>
</tr>
<tr>
<td>Multi-Target</td>
<td>MT-MTDA [32]</td>
<td>87.9</td>
<td>83.7</td>
</tr>
<tr>
<td></td>
<td>HGAN [53]</td>
<td>88.0</td>
<td>84.4</td>
</tr>
<tr>
<td></td>
<td>CDAN+DCL (ours)</td>
<td><strong>92.6</strong></td>
<td>82.5</td>
</tr>
<tr>
<td></td>
<td>D-CGCT (ours)</td>
<td><strong>93.4</strong></td>
<td><strong>86.0</strong></td>
</tr>
</tbody>
</table>

### Table 5. Ablation results of different baselines using ResNet-18 as backbone on Office-Home. **Baseline:** CDAN [25] model that combines all the target domains into a single target domain. “†” indicates the baseline models that use the domain labels of the target. GCN†: the baseline model with the GCN as the single classification head. PL: using pseudo-labels.

<table>
<thead>
<tr>
<th>Model</th>
<th>Office-Home</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source train</td>
<td><strong>51.45</strong></td>
</tr>
<tr>
<td>Baseline</td>
<td>50.70</td>
</tr>
<tr>
<td>Base$^{†}$</td>
<td>52.08</td>
</tr>
<tr>
<td>Base$^{†}$+PL</td>
<td>54.61</td>
</tr>
<tr>
<td>Base$^{†}$+DCL</td>
<td><strong>55.94</strong></td>
</tr>
<tr>
<td>Base$^{†}$+GCN†</td>
<td>50.19</td>
</tr>
<tr>
<td>Base$^{†}$+GCN†+PL</td>
<td>54.52</td>
</tr>
<tr>
<td>CGCT</td>
<td>60.81</td>
</tr>
<tr>
<td>D-CGCT</td>
<td><strong>61.42</strong></td>
</tr>
</tbody>
</table>

5357
Table 6. Comparison with the state-of-the-art methods on DomainNet. All methods use the ResNet-101 as the backbone. The classification accuracy are reported for each source $\rightarrow$ rest direction, with each source domain being indicated in the columns. All the reported numbers are evaluated on the multi-target setting.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Source train</td>
<td>25.6</td>
<td>16.8</td>
<td>25.8</td>
<td>9.2</td>
<td>20.6</td>
<td>22.3</td>
<td>20.1</td>
</tr>
<tr>
<td>SE [10]</td>
<td>21.3</td>
<td>8.5</td>
<td>14.5</td>
<td>13.8</td>
<td>16.0</td>
<td>19.7</td>
<td>15.6</td>
</tr>
<tr>
<td>MCD [41]</td>
<td>25.1</td>
<td>19.1</td>
<td>27.0</td>
<td>10.4</td>
<td>20.2</td>
<td>22.5</td>
<td>20.7</td>
</tr>
<tr>
<td>DADA [34]</td>
<td>26.1</td>
<td>20.0</td>
<td>26.5</td>
<td>12.9</td>
<td>20.7</td>
<td>22.8</td>
<td>21.5</td>
</tr>
<tr>
<td>CDAN [25]</td>
<td>31.6</td>
<td>27.1</td>
<td>31.8</td>
<td>12.5</td>
<td>33.2</td>
<td>35.8</td>
<td>28.7</td>
</tr>
<tr>
<td>MCC [18]</td>
<td>33.6</td>
<td>30.0</td>
<td>32.4</td>
<td>13.5</td>
<td>28.0</td>
<td>35.3</td>
<td>28.8</td>
</tr>
<tr>
<td><strong>CDAN + DCL</strong></td>
<td>35.1</td>
<td>31.4</td>
<td>37.0</td>
<td>20.5</td>
<td>35.4</td>
<td>41.0</td>
<td>33.4</td>
</tr>
<tr>
<td><strong>CGCT</strong></td>
<td>36.1</td>
<td>33.3</td>
<td>35.0</td>
<td>10.0</td>
<td>39.6</td>
<td>39.7</td>
<td>32.3</td>
</tr>
<tr>
<td><strong>D-CGCT</strong></td>
<td>37.0</td>
<td>32.2</td>
<td>37.3</td>
<td>19.3</td>
<td>39.8</td>
<td>40.8</td>
<td>34.4</td>
</tr>
</tbody>
</table>

Figure 2. t-SNE plots of the feature embeddings with Product $\rightarrow$ rest in Office-Home. Left: CGCT. Right: D-CGCT.

4.3. Comparison with State-of-The-Art

We compare our proposed method and its variants with several state-of-the-art methods that are designed exclusively for the MTDA as well as the STD methods that can be extended and used in the MTDA setting. In the main paper we only report the results for the Office-31, Office-Home and DomainNet experiments. Due to lack of space we report the numbers for Digits-five and PACS in the Supp. Mat.

In Tab. 4 we report the numbers for Office-31 and Office-Home for single-target, target-combined and multi-target setting. The single-target setting denotes training single-source to single-target models, the target-combined means treating all the target domains as one aggregated target, while the multi-target setting comprise of training a single model for single-source to multiple-targets. As can be observed, in all the settings our proposed CGCT and D-CGCT outperform all the state-of-the-art methods. Specifically, for the Office-31, our CGCT without using domain labels is already 2.4% better than the HGAN [53], which is a MTDA method exploiting domain labels for feature aggregation with a single GCN classifier besides pseudo-labeling. This highlights the importance of having a co-teaching strategy with two classifiers and curriculum learning for counteracting the impact of noisy pseudo-labels in the GCN framework. We also observed that incorporating domain information following the proposed DCL strategy improves the performance in the Office-Home, with the D-CGCT achieving 5.5% improvement over MT-MTDA [32], a MTDA method that also utilizes domain labels. Finally, as can be seen from the Tab. 6, the D-CGCT advances the state-of-the-art results for the very challenging DomainNet dataset by a non-trivial margin of 5.6%. This further verifies the effectiveness of our proposed methods for addressing the MTDA.

Overcoming negative transfer. Careful inspection of the Tab. 4 tells us that the single-target DA methods always outperform the same STD method when applied in the multi-target setting. For e.g., CDAN is 4.3% better in the single-target than in the multi-target setting. The drop in performance for the multi-target setting clearly hints at the fact that negative transfer [34, 9] is quite prevalent in the MTDA, despite having access to more data. Contrarily, our proposed CGCT when applied to both the settings fares equally well for the Office-Home and outperforms the single-target counterpart by 0.6% for the Office-31. This once again shows that the design choices made in our CGCT and D-CGCT lead to learning more robust domain-invariant features and provide resilience against negative transfer.

5. Conclusion

To address multi-target domain adaptation (MTDA), we proposed Curriculum Graph Co-Teaching (CGCT) that uses a graph convolutional network to perform robust feature aggregation across multiple domains, which is then trained with a co-teaching and curriculum learning strategy. To better exploit domain labels of the target we presented a Domain-aware curriculum (DCL) learning strategy that adapts easier target domains first and harder later, enabling a smoother feature alignment. Through extensive experiments we demonstrate that our proposed contributions handsomely outperform the state-of-the-art in the MTDA.

Acknowledgements This work is supported by the EU H2020 SPRING No. 871245 and AI4Media No. 951911 projects; the Italy-China collaboration project TALENT:2018YFE0118400; and the Caritro Deep Learning Lab of the ProM Facility of Rovereto.
References


