Curriculum Graph Co-Teaching for Multi-Target Domain Adaptation

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

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* [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 [48, 26, 45, 35, 4, 5, 6, 36], adversarial training [11, 47, 25], generative modelling [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 [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 [19] (GCN) to aggregate semantic information from similar samples in a *neighbourhood* 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-

---

*Equal contribution
†Corresponding author

Method	Domain labels	Feature aggregation	Curriculum learning	Co-teaching
AMEAN [9]	✗	✗	✗	✗
DADA [34]	✗	✗	✗	✗
MTDA-ITA [12]	✓	✗	✗	✗
HGAN [53]	✓	✓	✗	✗
CGCT (**Ours**)	✗	✓	✓	✓
D-CGCT (**Ours**)	✓	✓	✓	✓

Table 1. Comparison with recent the state-of-the-art MTDA methods in terms of the operating regimes.

ent networks capture different aspects of learning [2], it is beneficial for suppressing noisy pseudo-labels. his 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 latent, each episode or *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 **C**urriculum **G**raph **C**o-**T**eaching (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 **D**omain-aware **C**urriculum **L**earning (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 can not 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-

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 $\mathcal{S} = \{(\mathbf{x}_{s,i}, y_{s,i})\}_{i=1}^{n_s}$, containing n_s labeled samples, and N unlabeled target datasets $\mathcal{T} = \{\mathcal{T}_j\}_{j=1}^{N}$, where $\mathcal{T}_j = \{\mathbf{x}_{t_j,k}\}_{k=1}^{n_j}$ 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 $\mathcal{S} \cup \{\mathcal{T}_j\}_{j=1}^{N}$.

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 $F_\theta : \mathbb{R}^{3 \times w \times h} \to \mathbb{R}^d$ be the feature extractor network, parameterized by θ, that outputs a feature $\mathbf{f} = F(\mathbf{x})$ for a given sample $\mathbf{x}$. The classifier network,

parameterized by ϕ, is denoted by $G_\phi : \mathbb{R}^d \to \mathbb{R}^{n_c}$, which takes as input a feature $\mathbf{f}$ and outputs class logits $\mathbf{g} = G(\mathbf{f})$. The discriminator network $D_\psi : \mathbb{R}^d \to \mathbb{R}^1$, parameterized by ψ, takes in the same feature $\mathbf{f}$ 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:

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

$$\text{where } \ell_{ce} = -\mathbb{E}_{(\mathbf{x}_{s,i}, y_{s,i}) \sim \mathcal{S}} \; \widetilde{y}_{s,i} \log G(F(\mathbf{x}_{s,i})),$$

$$\text{and } \ell_{adv} = -\mathbb{E}_{\mathbf{x}_{s,i} \sim \mathcal{S}} \; \log D(F(\mathbf{x}_{s,i}))$$
$$- \mathbb{E}_{x_{t,j} \sim \mathcal{T}} \; \log\left[1 - D(F(\mathbf{x}_{t,j}))\right].$$

$\widetilde{y}_{s,i}$ is the one-hot label for a source label $y_{s,i}$. The first term, ℓ_{ce}, in Eq. 1 is the cross-entropy loss computed on the source domain samples and minimized w.r.t. θ, ϕ. The second term, ℓ_{adv}, in Eq. 1 is the adversarial loss that is maximized w.r.t ψ but minimized w.r.t θ. λ_{adv} is the weighing factor for ℓ_{adv}. To capture the multi-modal nature of the distributions, CDAN [25] is proposed where D can be additionally conditioned on the classifier predictions $\mathbf{g}$. In CDAN [25], the D takes as input $\mathbf{h} = (\mathbf{f}, \mathbf{g})$, the joint variable of $\mathbf{f}$ and $\mathbf{g}$, instead of just $\mathbf{f}$. 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 $\Gamma = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ from all samples in mini-batch. In details, given a mini-batch of images, we represent each image $\mathbf{x}_i$ as a node $\mathbf{v}_i \in \mathcal{V}$ in the Γ. $e_{i,j} \in \mathcal{E}$ indicates an edge between nodes $\mathbf{v}_i$ and $\mathbf{v}_j$, and $a_{i,j}$ is the semantic similarity score for nodes $(\mathbf{v}_i, \mathbf{v}_j)$ 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 $(\mathbf{v}_i, \mathbf{v}_j) \in \mathcal{E}$:

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

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

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

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

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

$$\mathbf{v}_i^{(l)} = f_{node}^{(l)}\left([\mathbf{v}_i^{(l-1)}, \sum_{j \in \mathcal{B}} a_{i,j}^{(l)} \cdot \mathbf{v}_j^{(l-1)}]\right), \tag{4}$$

where $f_{node}^{(l)}$ is a non-linear function parameterized by φ', $\mathcal{B}$ is a set of samples in the mini-batch, and $[\cdot, \cdot]$ is the feature
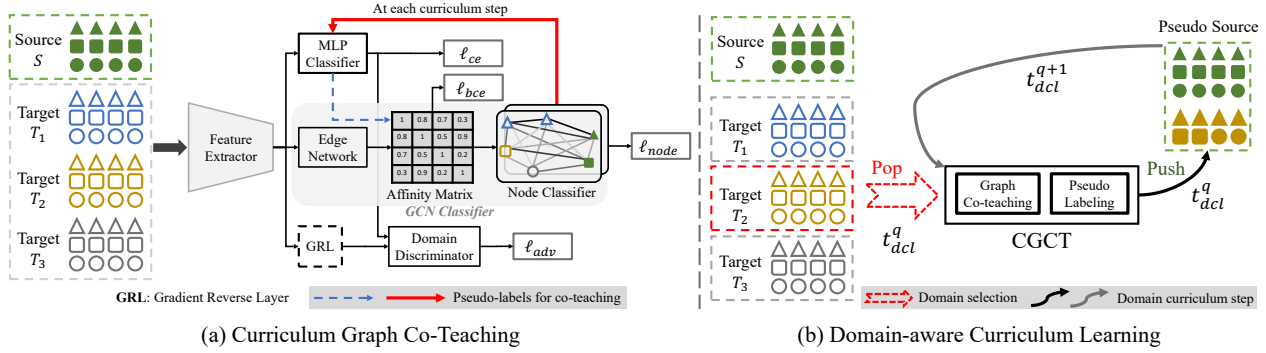
Figure 1. The pipeline of the proposed framework: a) CGCT: Curriculum Graph Co-Teaching and b) DCL: Domain-aware curriculum learning. (a) In the CGCT, the MLP Classifier provides pseudo-labels (PL) (--→ arrow) for the target samples to guide the Edge Network to learn the Affinity Matrix, whereas the Node Classifier of the GCN provides PL (bold → arrow) to the MLP Classifier at the end of each curriculum step, realizing the *co-teaching*. (b) In the DCL, the target domains are selected for adaptation, one at a time per domain curriculum step t_{dcl}^q, with the "easier" domains selected first and then the "harder" ones. After PL are obtained, the pseudo-labeled target dataset is added to the Pseudo Source dataset, which is then used in the next adaptation step.

concatenation function. The final $f_{node}^{(L)}$ layer is the output layer with n_c outputs. We slightly abuse the notations and drop the superscript l 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_{mlp} and a GCN classifier G_{gcn}. The G_{mlp} is a fully-connected output layer with n_c output logits. The G_{gcn} consists of an edge network f_{edge} and a node classifier f_{node}. The f_{node} aggregates the features of the samples in $\mathcal{B}$ by considering the learnt pairwise similarity in the affinity matrix $\hat{\mathcal{A}}$ produced by the f_{edge}. The G_{gcn} also outputs n_c logits. Since, the G_{mlp} and the G_{gcn} 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_{cur}^q, is an episode in which the network is trained for K training iterations. Each curriculum step t_{cur}^q is further decomposed into two stages: i) Adaptation stage and ii) Pseudo-labeling stage. Each stage in a t_{cur}^q 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_{cur}^0 we start with a source set $\hat{\mathcal{S}}^0 = \{\mathcal{S}\}$ and a target set $\mathcal{T}$. We sample mini-batches $\mathcal{B}^0 = \{\mathcal{B}_s^0, \mathcal{B}_t^0\} = \{\mathcal{B}_{s,i}^0, \mathcal{B}_{t,i}^0\}_{i=1}^B$ with size B such that $\mathcal{B}_{s,i}^0 \sim \hat{\mathcal{S}}$ and $\mathcal{B}_{t,i}^0 \sim \mathcal{T}$. Each mini-batch of images is first fed to the feature extractor F to obtain $\mathcal{F}^0 = \{\mathbf{f}_{s,i}^0, \mathbf{f}_{t,i}^0\}_{i=1}^B$ which are then simultaneously fed to both the G_{mlp} and G_{gcn}. When fed to

the G_{mlp} it outputs the logits $\hat{\mathcal{G}}^0 = \{\hat{\mathbf{g}}_{s,i}^0, \hat{\mathbf{g}}_{t,i}^0\}_{i=1}^B$. On the other hand, $\mathcal{F}^0$ are input to the f_{edge} to estimate the pairwise similarity of the samples in $\mathcal{B}^0$. Specifically, the f_{edge} outputs an affinity matrix $\hat{\mathcal{A}}$ following Eq. 2, where the entries $\hat{a}_{i,j}$ in $\hat{\mathcal{A}}$ 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_{node} aggregates the features in $\mathcal{F}^0$ based on the estimated $\hat{\mathcal{A}}$ such that for each node the most similar samples in the neighbourhood contribute more to its final representation. Subsequently, the f_{node} outputs its logits as $\bar{\mathcal{G}}^0 = \{\bar{\mathbf{g}}_{s,i}^0, \bar{\mathbf{g}}_{t,i}^0\}_{i=1}^B$. The elements in $\hat{\mathcal{G}}^0$ and $\bar{\mathcal{G}}^0$ are then passed through a softmax function to obtain the probabilities for each sample as $p(\hat{y} = c|\hat{\mathbf{g}}; c \in n_c)$ and $p(\bar{y} = c|\bar{\mathbf{g}}; c \in n_c)$, where $\hat{y}$ and $\bar{y}$ are the predictions, respectively.

To guide the f_{edge} to learn the pairwise similarity between the samples in $\mathcal{B}^0$ we propose the concept of co-teaching where the G_{mlp} provides feedback to the f_{edge}. Since, G_{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 $\hat{\mathcal{A}}^{tar}$ and enforce the predictions of f_{edge} to be as close as possible to the $\hat{\mathcal{A}}^{tar}$. Each entry $\hat{a}_{i,j}^{tar}$ in the $\hat{\mathcal{A}}_{tar}$ is given by:

$$\hat{a}_{i,j}^{tar} = \begin{cases} 1, & \text{if } y_i = y_j = c \\ 0, & \text{otherwise} \end{cases}, \quad (5)$$

where c is the class label. While the class labels of $\mathcal{B}_s^0$ are provided as ground truth, we do not have access to the labels of $\mathcal{B}_t^0$. Therefore, a target domain sample $\mathbf{x}_{t,j} \in \mathcal{B}_t^0$ is assigned a definitive pseudo-label $\hat{y}_{t,j} = c'$ where $c' = \text{argmax}_{c \in n_c} p(\hat{y}_{t,j} = c|\hat{\mathbf{g}}_{t,j})$ if the maximum likelihood $\max_{c \in n_c} p(\hat{y}_{t,j} = c|\hat{\mathbf{g}}_{t,j})$ is greater than a threshold τ. The

entries $\hat{a}_{i,j}^{tar}$ involving $\mathbf{x}_{t,j} \in \mathcal{B}_t^0$ not passing the τ are not optimized during training. We train the f_{edge} using a binary cross-entropy loss as:

$$\ell_{bce}^{edge} = \hat{a}_{i,j}^{tar} \log p(\hat{a}_{i,j}) + (1 - \hat{a}_{i,j}^{tar}) \log (1 - p(\hat{a}_{i,j})). \quad (6)$$

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

$$\ell_{ce}^{mlp} = -\frac{1}{|\mathcal{B}_s^0|} \sum_{i=1}^{|\mathcal{B}_s^0|} \tilde{y}_i \log \ p(\hat{y}_{s,i} | \hat{\mathbf{g}}_{s,i}^0), \quad (7)$$

$$\ell_{ce}^{node} = -\frac{1}{|\mathcal{B}_s^0|} \sum_{i=1}^{|\mathcal{B}_s^0|} \tilde{y}_i \log \ p(\bar{y}_{s,i} | \bar{\mathbf{g}}_{s,i}^0). \quad (8)$$

We feed the features $\{\hat{\mathbf{h}}_{s,i}^0, \hat{\mathbf{h}}_{t,i}^0\}_{i=1}^B = \{(\mathbf{f}_{s,i}^0, \hat{\mathbf{g}}_{s,i}^0), (\mathbf{f}_{t,i}^0, \hat{\mathbf{g}}_{t,i}^0)\}_{i=1}^B$, corresponding to $\mathcal{B}^0$, 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:

$$\max_{\psi} \min_{\theta,\phi,\varphi,\varphi'} \ell_{ce}^{mlp} + \lambda_{edge}\ell_{bce}^{edge} \\ + \lambda_{node}\ell_{ce}^{node} - \lambda_{adv}\ell_{adv}, \quad (9)$$

where λ_{edge}, λ_{node} and λ_{adv} are the weighing factors.

Pseudo-labelling stage. Upon completion of the adaptation stage in a curriculum step t_{cur}^q we put the network in inference mode and obtain pseudo-labels $\forall \mathbf{x}_{t,j} \in \mathcal{T}$. The G_{gcn} is employed for this task because, owing to its aggregating characteristics, it learns more robust features [53] 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_{node}.

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

$$\forall \mathbf{x}_{t,j} \in \mathcal{T}, w_j = \begin{cases} 1, & \text{if } \max_{c \in n_c} p(\bar{y}_{t,j} = c | \bar{\mathbf{g}}_{t,j}) > \tau \\ 0, & \text{otherwise} \end{cases},$$

$$(10)$$

where $w_j = 1$ signifies that $\mathbf{x}_{t,j}$ is selected with a pseudo-label $\bar{y}_{t,j} = c'$ where $c' = \text{argmax}_{c \in n_c} p(\bar{y}_{t,j} = c | \bar{\mathbf{g}}_{t,j})$, whereas $w_j = 0$ denotes no pseudo-label is assigned. After the pseudo-labeling stage in a t_{cur}^q we obtain a pseudo-labeled target set $\mathcal{D}_t^q = \{(\mathbf{x}_{t,j}, \bar{y}_{t,j})\}_{j=1}^{\bar{n}_t}$ where $\bar{n}_t$ is the number of recruited pseudo-labeled target samples. Post pseudo-labeling we update and prepare the source set for the succeeding step t_{cur}^{q+1} as:

$$\hat{\mathcal{S}}^{q+1} = \mathcal{S} \cup \mathcal{D}_t^q. \quad (11)$$

The update rule in Eq. 11 allows us to compute the supervised losses ℓ_{ce}^{node} and ℓ_{ce}^{mlp} from Eq. 9 for $\mathbf{x}_{t,j} \sim \mathcal{D}_t$. Note that we do not alter the domain labels in $\mathcal{D}_t^q$ and hence, the formulation for ℓ_{adv} remains unchanged.

At the culmination of Q curriculum steps, $\hat{\mathcal{S}}^Q$ 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.* $\mathcal{T} = \{\mathcal{T}_j\}_{j=1}^N$, 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 **D**omain-aware **C**urriculum **L**earning (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:

$$H(\mathcal{T}_j) = - \mathop{\mathbb{E}}_{\mathbf{x}_{t_j,k} \sim \mathcal{T}_j} \sum_{c=1}^{|n_c|} p(\hat{y}_{t_j,k,c} | \mathbf{x}_{t_j,k}) \log p(\hat{y}_{t_j,k,c} | \mathbf{x}_{t_j,k}).$$

$$(12)$$

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_{dcl}^q, 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_{dcl}^q 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_{dcl}^q is split into three stages and are described below:

Domain selection stage. Given a source-trained model $F_{\theta^*}(G_{\phi^*})$, where θ^* and ϕ^* are the trained parameters of

F and G, and initial source and target sets $\hat{\mathcal{S}}^0 = \{\mathcal{S}\}$ and $\hat{\mathcal{T}}^0 = \{\mathcal{T}_j\}_{j=1}^N$, the closest target domain is selected as:

$$\mathbb{D}^0 = \underset{j}{\arg\min}\{H_j(\mathcal{T}_j) \mid \forall \mathcal{T}_j \in \hat{\mathcal{T}}^0\}, \qquad (13)$$

where $\mathbb{D}^0$ is the target domain selected at step t_{dcl}^0 and is used for performing adaptation in the subsequent stage.

Adaptation stage. This stage is similar to the one in t_{cur}^q, described in Sec. 3.2, except the feature adaptation at any step t_{dcl}^q is performed using $\hat{\mathcal{S}}^q \cup \mathcal{T}_{\mathbb{D}^q}$, rather than the entire target set $\mathcal{T}$. 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 $\mathcal{D}_t^{\mathbb{D}^q}$. Consequently, the source and target set update changes as following:

$$\hat{\mathcal{S}}^{q+1} = \hat{\mathcal{S}}^q \cup \mathcal{D}_t^{\mathbb{D}^q}, \qquad (14)$$

$$\hat{\mathcal{T}}^{q+1} = \hat{\mathcal{T}}^q \setminus \mathcal{T}_{\mathbb{D}^q}. \qquad (15)$$

These three stages are repeated until all N domains have been exhausted. Then similarly, as in CGCT, the final model is fine-tuned with $\hat{\mathcal{S}}^Q$. 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.

Dataset	#domains	#classes	#images
Digits-five	5	10	$\sim$ 145K
PACS	4	7	9,991
Office-31	3	31	4,652
Office-Home	4	65	15,500
DomainNet	6	345	$\sim$ 0.6M

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, *i.e.*, *source→rest*, is given by averaging the accuracy on all the target domains, where *source* signifies the source domain and *rest* indicates all the unlabeled domains except the *source*. Importantly, in all our experiments we always report the final classification accuracy obtained with the G_{mlp} because the G_{gcn} 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 $\rightarrow$ *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.

Model	Co-teaching	Pseudo-labels from G_{mlp}	f_{edge}	f_{node}	**Avg(%)**
M1	✗	self	G_{mlp}	G_{mlp}	57.4
M2	✗	G_{gcn}	G_{gcn}	G_{gcn}	59.6
M3	✓	self	G_{mlp}, G_{gcn}	G_{mlp}	58.2
D-CGCT (Ours)	✓	G_{gcn}	G_{mlp}	G_{gcn}	**60.8**

Table 3. Ablation study of different co-teaching strategies on Office-Home. We reported the classification accuracy averaged across all the *source* $\rightarrow$ *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_{gcn} provide pseudo-labels to the each 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_{dcl}^q; ii) M2: a baseline similar to M1, except that the G_{gcn} provides the pseudo-labels; iii) M3: another baseline which is similar to M1 but with an exception that the G_{gcn} also provides pseudo-labels to f_{edge} for the current target domain in an ongoing t_{dcl}^q step.

Setting	Model	Office-31				Office-Home				
		Amazon	DSLR	Webcam	**Avg(%)**	Art	Clipart	Product	Real	**Avg(%)**
w/o Target	Source train	68.6	70.0	66.5	68.4	47.6	42.6	44.2	51.3	46.4
Single-Target	DAN [26]	79.5	80.3	81.2	80.4	56.1	54.2	51.7	63.0	56.3
	RevGrad [11]	80.8	82.5	83.2	82.2	58.3	55.4	52.8	63.9	57.6
	JAN [27]	85.0	83.0	85.6	84.3	58.7	57.0	53.1	64.3	58.3
	CDAN [25]	**91.4**	84.1	84.0	86.6	64.2	62.9	59.9	68.1	63.8
	CGCT (ours)	89.6	**85.5**	**87.6**	**87.6**	**67.9**	**68.7**	**62.3**	**70.7**	**67.4**
Target-Combined	DAN [26]	78.0	64.4	66.7	69.7	55.6	56.6	48.5	56.7	54.4
	RevGrad [11]	78.2	72.2	69.8	73.4	58.4	58.1	52.9	62.1	57.9
	JAN [27]	84.2	74.4	72.0	76.9	58.3	60.5	52.2	57.5	57.1
	CDAN [25]	93.6	80.5	81.3	85.1	59.5	61.0	54.7	62.9	59.5
	AMEAN [9]	90.1	77.0	73.4	80.2	64.3	65.5	59.5	66.7	64.0
	CGCT (ours)	**93.9**	**85.1**	**85.6**	**88.2**	**67.4**	**68.1**	**61.6**	**68.7**	**66.5**
Multi-Target	MT-MTDA [32]	87.9	83.7	84.0	85.2	64.6	66.4	59.2	67.1	64.3
	HGAN [53]	88.0	84.4	84.9	85.8	-	-	-	-	-
	CDAN+DCL (ours)	92.6	82.5	84.7	86.6	63.0	66.3	60.0	67.0	64.1
	D-CGCT (ours)	**93.4**	**86.0**	**87.1**	**88.8**	**70.5**	**71.6**	**66.0**	**71.2**	**69.8**

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.

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 t_{dcl}^q 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

Model	Office-Home				
	Art	Clipart	Product	Real	**Avg(%)**
Source train	51.45	43.93	42.41	54.50	48.07
Baseline	50.70	50.78	47.95	57.63	51.77
Base.†	52.08	53.21	48.62	58.49	53.10
Base.†+PL	54.61	56.13	50.25	**61.04**	55.51
Base.† + **DCL**	**55.94**	**56.66**	**52.85**	60.18	**56.41**
Base.†+GCN‡	50.19	49.09	46.52	60.76	51.64
Base.†+GCN‡ + PL	54.52	57.60	53.20	65.49	57.70
CGCT	60.81	60.00	54.13	62.62	59.39
D-CGCT	**61.42**	**60.73**	**57.27**	**63.8**	**60.81**

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.

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.

Model	DomainNet						
	Cli.	Inf.	Pai.	Qui.	Rea.	Ske.	**Avg(%)**
Source train	25.6	16.8	25.8	9.2	20.6	22.3	20.1
SE [10]	21.3	8.5	14.5	13.8	16.0	19.7	15.6
MCD [41]	25.1	19.1	27.0	10.4	20.2	22.5	20.7
DADA [34]	26.1	20.0	26.5	12.9	20.7	22.8	21.5
CDAN [25]	31.6	27.1	31.8	12.5	33.2	35.8	28.7
MCC [18]	33.6	30.0	32.4	13.5	28.0	35.3	28.8
CDAN + DCL	35.1	31.4	37.0	**20.5**	35.4	**41.0**	33.4
CGCT	36.1	**33.3**	35.0	10.0	39.6	39.7	32.3
D-CGCT	**37.0**	32.2	**37.3**	19.3	**39.8**	40.8	**34.4**

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→rest* direction, with each *source* domain being indicated in the columns. All the reported numbers are evaluated on the multi-target setting.
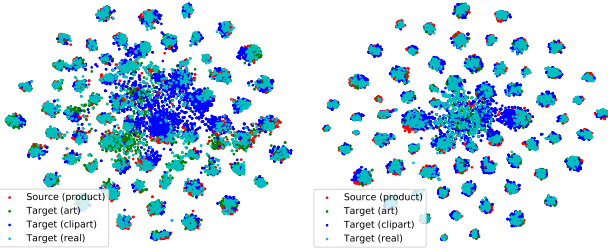

Figure 2. *t*-SNE plots of the feature embeddings with Product → *rest* in Office-Home. Left: CGCT. Right: D-CGCT.

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 STDA 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 informa-

tion 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.
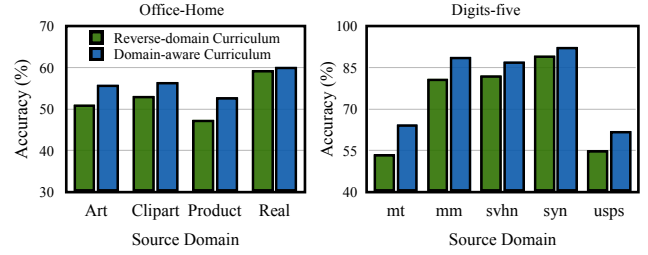

Figure 3. Comparison of the DCL with the *reverse*-domain curriculum model on Office-Home and Digits-Five. In the reverse-domain curriculum model the order of selection of target domains is exactly opposite to that of the DCL model.

Overcoming negative transfer. Careful inspection of the Tab. 4 tells us that the single-target DA methods always outperform the same STDA 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.

References

[1] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proc. ICML*, 2009. 2

[2] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proc. CLT*, 1998. 2

[3] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Michael I Jordan. Partial transfer learning with selective adversarial networks. In *Proc. CVPR*, 2018. 2

[4] Fabio Maria Carlucci, Lorenzo Porzi, Barbara Caputo, Elisa Ricci, and Samuel Rota Bulò. Autodial: Automatic domain alignment layers. In *Proc. ICCV*, 2017. 1, 2

[5] Fabio Maria Carlucci, Lorenzo Porzi, Barbara Caputo, Elisa Ricci, and Samuel Rota Bulo. Just dial: Domain alignment layers for unsupervised domain adaptation. In *Proc. ICIAP*, 2017. 1

[6] Woong-Gi Chang, Tackgeun You, Seonguk Seo, Suha Kwak, and Bohyung Han. Domain-specific batch normalization for unsupervised domain adaptation. In *Proc. CVPR*, 2019. 1, 2

[7] Chaoqi Chen, Weiping Xie, Wenbing Huang, Yu Rong, Xinghao Ding, Yue Huang, Tingyang Xu, and Junzhou Huang. Progressive feature alignment for unsupervised domain adaptation. In *Proc. CVPR*, 2019. 2

[8] Qingchao Chen, Yang Liu, Zhaowen Wang, Ian Wassell, and Kevin Chetty. Re-weighted adversarial adaptation network for unsupervised domain adaptation. In *Proc. CVPR*, 2018. 2

[9] Ziliang Chen, Jingyu Zhuang, Xiaodan Liang, and Liang Lin. Blending-target domain adaptation by adversarial meta-adaptation networks. In *Proc. CVPR*, 2019. 1, 2, 4, 6, 7, 8

[10] Geoff French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation. In *Proc. ICLR*, 2018. 8

[11] 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, 2, 3, 7

[12] Behnam Gholami, Pritish Sahu, Ognjen Rudovic, Konstantinos Bousmalis, and Vladimir Pavlovic. Unsupervised multi-target domain adaptation: An information theoretic approach. *TIP*, 2020. 2

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proc. NeurIPS*, 2014. 2

[14] Marco Gori, Gabriele Monfardini, and Franco Scarselli. A new model for learning in graph domains. In *Proc. IJNN*, 2005. 3

[15] 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. 3

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 6

[17] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *Proc. ICML*, 2017. 1, 2

[18] Ying Jin, Ximei Wang, Mingsheng Long, and Jianmin Wang. Minimum class confusion for versatile domain adaptation. In *Proc. ECCV*, 2020. 2, 8

[19] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proc. ICLR*, 2017. 1, 3

[20] Vinod Kumar Kurmi, Vipul Bajaj, Venkatesh K Subramanian, and Vinay P Namboodiri. Curriculum based dropout discriminator for domain adaptation. *arXiv*, 2019. 5

[21] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proc. ICCV*, 2017. 6

[22] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. In *Proc. ICLR-WS*, 2017. 2

[23] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *Proc. NeurIPS*, 2016. 1, 2

[24] Ziwei Liu, Zhongqi Miao, Xingang Pan, Xiaohang Zhan, Dahua Lin, Stella X Yu, and Boqing Gong. Open compound domain adaptation. In *Proc. CVPR*, 2020. 2

[25] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Proc. NeurIPS*, 2018. 1, 2, 3, 4, 7, 8

[26] Mingsheng Long and Jianmin Wang. Learning transferable features with deep adaptation networks. In *Proc. ICML*, 2015. 1, 2, 7

[27] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *Proc. ICML*, 2017. 2, 7

[28] Yadan Luo, Zijian Wang, Zi Huang, and Mahsa Baktashmotlagh. Progressive graph learning for open-set domain adaptation. *In Proc. ICML*, 2020. 3

[29] Xinhong Ma, Tianzhu Zhang, and Changsheng Xu. Gcan: Graph convolutional adversarial network for unsupervised domain adaptation. In *Proc. CVPR*, 2019. 3

[30] Massimiliano Mancini, Lorenzo Porzi, Samuel Rota Bulò, Barbara Caputo, and Elisa Ricci. Boosting domain adaptation by discovering latent domains. In *Proc. CVPR*, 2018. 2

[31] Pietro Morerio, Jacopo Cavazza, and Vittorio Murino. Minimal-entropy correlation alignment for unsupervised deep domain adaptation. In *Proc. ICLR*, 2018. 2

[32] Le Thanh Nguyen-Meidine, Madhu Kiran, Jose Dolz, Eric Granger, Atif Bela, and Louis-Antoine Blais-Morin. Unsupervised multi-target domain adaptation through knowledge distillation. In *Proc. WACV*, 2021. 7, 8

[33] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proc. ICCV*, 2019. 2, 6

[34] Xingchao Peng, Zijun Huang, Ximeng Sun, and Kate Saenko. Domain agnostic learning with disentangled representations. *arXiv*, 2019. 2, 6, 8

[35] Xingchao Peng and Kate Saenko. Synthetic to real adaptation with generative correlation alignment networks. In *Proc. WACV*, 2018. 1, 2

[36] Subhankar Roy, Aliaksandr Siarohin, Enver Sangineto, Samuel Rota Bulo, Nicu Sebe, and Elisa Ricci. Unsupervised domain adaptation using feature-whitening and consensus loss. *Proc. CVPR*, 2019. 1, 2

[37] Subhankar Roy, Aliaksandr Siarohin, Enver Sangineto, Nicu Sebe, and Elisa Ricci. Trigan: Image-to-image translation for multi-source domain adaptation. *Machine vision and applications*, 2021. 2

[38] Subhankar Roy, Aliaksandr Siarohin, and Nicu Sebe. Unsupervised domain adaptation using full-feature whitening and colouring. In *Proc. ICIAP*, 2019. 2

[39] Paolo Russo, Fabio Maria Carlucci, Tatiana Tommasi, and Barbara Caputo. From source to target and back: symmetric bi-directional adaptive gan. In *Proc. CVPR*, 2018. 1, 2

[40] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *Proc. ECCV*, 2010. 6

[41] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proc. CVPR*, 2018. 8

[42] Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *Proc. CVPR*, 2018. 2

[43] Rui Shu, Hung H Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. *arXiv*, 2018. 2, 6

[44] Yang Shu, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Transferable curriculum for weakly-supervised domain adaptation. In *Proc. AAAI*, 2019. 2

[45] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *Proc. ECCV*, 2016. 1, 2

[46] Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *Proc. CVPR*, 2011. 1

[47] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Adversarial discriminative domain adaptation. In *Proc. CVPR*, 2017. 1, 2

[48] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv*, 2014. 1, 2

[49] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proc. CVPR*, 2017. 2, 6

[50] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *TNNLS*, 2020. 3

[51] 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. 2, 6

[52] Luyu Yang, Yogesh Balaji, Ser-Nam Lim, and Abhinav Shrivastava. Curriculum manager for source selection in multi-source domain adaptation. *Proc. ECCV*, 2020. 2

[53] Xu Yang, Cheng Deng, Tongliang Liu, and Dacheng Tao. Heterogeneous graph attention network for unsupervised multiple-target domain adaptation. *TPAMI*, 2020. 2, 3, 5, 6, 7, 8

[54] Wen Zhang, Lingfei Deng, and Dongrui Wu. Overcoming negative transfer: A survey. *arXiv*, 2020. 1