TRANSFERABLE FEATURE LEARNING ON GRAPHS ACROSS VISUAL DOMAINS

Anonymous authors
Paper under double-blind review

ABSTRACT

Unsupervised domain adaptation has attracted increasing attention in recent years, which adapts classifiers to an unlabeled target domain by exploiting a labeled source domain. To reduce discrepancy between source and target domains, adversarial learning methods are typically selected to seek domain-invariant representations by confusing the domain discriminator. However, classifiers may not be well adapted to such a domain-invariant representation space, as the sample-level and class-level data structures could be distorted during adversarial learning. In this paper, we propose a novel Transferable Feature Learning approach on Graphs (TFLG) for unsupervised adversarial domain adaptation, which jointly incorporates sample-level and class-level structure information across two domains. TFLG first constructs graphs for mini-batch samples, and identifies the class-wise correspondence across domains. A novel cross-domain graph convolutional operation is designed to jointly align the sample-level and class-level structures in two domains. Moreover, a memory bank is designed to further exploit the class-level information. Extensive experiments on benchmark datasets demonstrate the effectiveness of our approach, compared to the representative unsupervised domain adaptation methods.

1 INTRODUCTION

The successful development of visual learning systems in practice is largely dependent on abundant labeled data for model training. In many cases, however, it is difficult to have direct access to a large amount of labeled data for the task of interest, and data labeling is usually expensive and time consuming. Domain adaptation has been recognized as an appealing approach to tackle this challenge, which aims to adapt models from a source domain with abundant labeled data to a target domain with limited or even no labels (Wang & Deng, 2018). To this end, existing domain adaptation methods devise various criteria to mitigate the divergence between source and target domains, which is known as the *domain shift* problem. Unsupervised domain adaptation (UDA), as a special case of domain adaptation, has attracted increasing attention in recent years, owing to its relaxed assumption, i.e., label information is not available in the target domain (Kouw & Loog, 2019). Although UDA is promising in real-world applications, it is quite challenging to align source and target domains due to the lack of labels in target domain.

Existing UDA methods mainly focus on extracting domain-invariant features to reduce the discrepancy of two domains, such that the classifier trained on source domain can be gradually adapted to target domain during feature learning process. To characterize the distribution difference of two domains, the maximum mean discrepancy criterion has been extensively studied in traditional UDA methods (Long et al., 2015; Yan et al., 2017; Cao et al., 2018), such as the subspace learning based approaches (Ding et al., 2018). Some recent advancements in UDA have been benefited from adversarial learning, such as virtual adversarial training (Ganin et al., 2016; Shu et al., 2018; Saito et al., 2018; Zhang et al., 2019) and adversarial networks (Tzeng et al., 2017; Long et al., 2018; Sankaranarayanan et al., 2018; Chen et al., 2020; Yabin et al., 2019). These methods usually employ adversarial losses for domain adaptation, by maximizing the domain classification loss and minimizing the label prediction loss. Adversarial learning based UDA methods have achieved remarkable performance, but they still suffer some limitations. First, existing methods usually seek a shared embedding space for two domains via adversarial learning. However, the intrinsic sample-level and class-level neighborhood structures in two domains might be destructed during this process. A common assumption in UDA is that two domains share the same label space, and thus exploiting the class-level structures is critical for

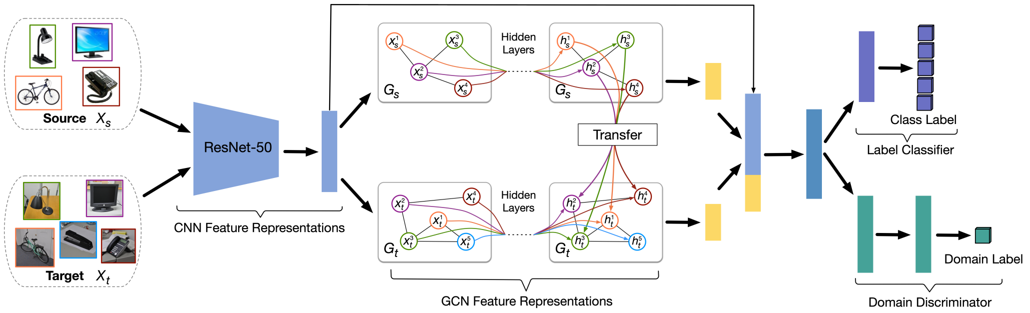


Figure 1: The architecture of our Transferable Feature Learning on Graphs (TFLG), which integrates cross-domain graph convolution with domain adversarial learning. TFLG adopts ResNet-50 as feature extractor, and exploits both sample-level and class-level structure information by using the proposed cross-domain graph convolutional operation. CNN features and graph ebmeddings are concatenated together, which are fed into label classifier and domain discriminator.

domain adaptation. Second, in existing methods, knowledge transfer across two domains is achieved through the domain-invariant feature learning, in which the domain discriminator plays a key role. However, when the divergence between two domains is large, the domain discriminator may not be easily fooled, and the domain shift issue still exists. Exploring additional pathways to bridge source and target domains is essential for robust and effective knowledge transfer.

In this paper, we propose a novel unsupervised domain adaptation approach by Transferable Feature Learning on Graphs (TFLG) to address the aforementioned problems. Building upon the adversarial learning framework, our approach incorporates sample-level and class-level structure information across two domains by designing a graph based feature propagation module. Graphs for mini-batches are constructed to capture the neighborhood structures in source and target domains. Furthermore, with the help of pseudo labels for target samples, we identify the class-wise correspondence between two domains, and achieve structure-aware feature refinement. Our graph based feature propagation module provides an additional knowledge transfer pathway across two domains, and it contributes to the domain-invariant feature learning. Moreover, the pseudo labels for samples in target domain will be gradually updated, leading to a joint optimization of class-wise structure preserving and classifier adaptation. Experimental results on benchmark datasets demonstrate the superiority of our approach over the representative UDA methods. Figure 1 illustrates the architecture of our approach.

The major contributions of our work are summarized as follows:

- We propose a transferable feature learning approach for unsupervised domain adaptation. Our approach seeks domain-invariant features with adversarial learning, and meanwhile preserves the sample-level structure information in each domain for feature refinement.
- We design a new graph convolution operator that exploits class-level structure across two domains. A memory bank is also incorporated to facilitate the matching of source and target domains at the class level.

2 Related Work

The key task in domain adaptation is to enhance the generalization ability of models trained in a source domain for a target domain. To deal with the issue of domain shift, numerous methods have been proposed, including the sample-based, feature-based, and inference-based methods (Kouw & Loog, 2019).

In real-world applications, it might be difficult to annotate samples in target domain, leading to the problem of unsupervised domain adaptation (UDA). In particular, UDA aims to transfer knowledge from a source domain to a target domain without labels. Existing UDA methods can be roughly categorized into two groups, including the subspace learning methods and deep domain adaptation methods. The subspace learning based UDA methods assume that different domains contain domain-specific noises but share a common subspace. Some subspace alignment methods try to match two domains in the common subspace (Fernando et al., 2013; Sun & Saenko, 2015), while other methods rely on the manifold assumptions (Baktashmotlagh et al., 2014; Hoffman et al., 2014) or explore the domain-invariant spaces (Baktashmotlagh et al., 2013; 2016).

Deep domain adaptation methods leverage deep neural networks for feature learning. Marginalized denoising auto-encoders (Chen et al., 2012) have been proposed to reconstruct target data from the source data. In recent years, adversarial learning have been successfully applied to the UDA problem. Domain-adversarial neural networks (DANN) (Ganin et al., 2016) aim to find a representation space, in which the two domains cannot be distinguished but the samples from source domain can be correctly classified. To this end, DANN maximizes the domain classification loss and meanwhile minimizes the label classification loss. If the discrepancy between two domains is small, the classifier trained using source samples will be well adapted to classify the target samples. DIRT-T further improves DANN by seeking decision boundaries that do not cross the high-density data regions (Shu et al., 2018). Adversarial discriminative domain adaptation (ADDA) pretrains a classifier in source domain, and then it learns an encoder to align target and source domains with a domain-adversarial loss (Tzeng et al., 2017). To align domains with multimodal distributions, the conditional domain adversarial network (CDAN) (Long et al., 2018) is proposed, which exploits discriminative information provided by classifiers to help adversarial adaptation. In order to further explore fine-level structures of data, the discriminative clustering has been adopted in the cluster alignment with a teacher (CAT) method, which delivers a domain-invariant cluster-structure feature space (Deng et al., 2019). Graph Adaptive Knowledge Transfer (GAKT) (Ding et al., 2018) model jointly optimizes target labels and domain-free features in a unified framework, which doesn't involve graph convolutional operations or adversarial learning. GAKT adopts a very different technical approach and the performance is not comparable with the adversarial learning based methods. Most recently, the graph convolutional adversarial network (GCAN) (Ma et al., 2019) is proposed to jointly model the data structure, domain label and class label. The Graph Convolution Network (GCN) (Kipf & Welling, 2017) is applied to model data structure. Firstly, they use the Data Structure Analyzer (DSA) network to learn CNN features for mini-batch samples and utilize the learned features to construct a dense-connected instance graph based on the similarity of pairwise mini-batch samples. Secondly, they apply the GCN on instance graph to learn GCN features. Finally, they concatenate CNN (a CNN network different from DSA) and GCN features as the input for domain alignment and class centroid alignment modules.

Remark. The most relevant work to ours is the GCAN method. The major differences between our work and GCAN are four-fold. (1) Our method does not require a specific CNN network, e.g., DSA, to build the dense-connected instance mini-graph. Instead, we use the same CNN network to build mini-graph and extract CNN features. (2) GCAN directly uses the GCN to extract features for target samples, while we proposed the cross-domain graph convolutional operation to explore class-level and sample-level structure information by bridging the source and the target domains. Our ablation studies also demonstrate the importance of the cross-domain graph convolutional operation. (3) To better capture the class-level relationship and transfer relevant data-structural information across the source and target domains, we maintain a memory bank for source domain to store the most recent batches of source GCN features, which has been proved to be effective. (4) Our proposed cross-domain graph convolutional operation can be easily integrated with various adversarial learning based unsupervised domain adaptation approaches, such as DANN, CDAN and CDANE.

3 Our Approach

In this section, we first introduce the problem setting of unsupervised domain adaptation as well as notations, and present the adversarial domain adaptation framework. Then, we propose the cross-domain graph convolutional operation, and define the objective function for transferable feature learning on graphs. Finally we provide some remarks and discussions.

3.1 Unsupervised Adversarial Domain Adaptation

In unsupervised domain adaptation, let $\mathcal{D}_s = \{(x_s^i, y_s^i)\}_{i=1}^{n_s}$ denote a source domain, which contains n_s labeled samples with $x_s^i \in \mathcal{X}_s$ and $y_s^i \in \mathcal{Y}_s$. Meanwhile, a target domain $\mathcal{D}_t = \{x_t^i\}_{i=1}^{n_t}$ contains n_t unlabeled samples where $x_t^i \in \mathcal{X}_t$. Both the source domain $\mathcal{D}_s$ and target domain $\mathcal{D}_t$ have the same learning tasks, which implies that $\mathcal{Y}_s$ and $\mathcal{Y}_t$ share the same label space. The i.i.d assumption commonly used in traditional machine learning is violated in this scenario, as $\mathcal{X}_s$ and $\mathcal{X}_t$ are assumed to be different but related (Shimodaira, 2000). The goal of unsupervised domain adaptation is to classify the unlabeled target samples by leveraging the labeled source samples. Since samples in two domains are distributed in different feature spaces, the most challenging problems in unsupervised domain adaptation include: how to reduce the discrepancy between the source and target domains, and how to achieve discriminative knowledge transfer across two domains?

Inspired by the Generative Adversarial Networks (GANs) (Goodfellow et al., 2014), the adversarial domain adaptation framework has been extensively studied, which integrates adversarial learning and domain adaptation in a two-player game. In particular, a domain discriminator D is learned to discriminate the source from target domains, while a feature extractor G implemented by deep neural networks tries to learn a transferable feature to fool D. In this way, domain-invariant representations could be extracted from the source and target samples.

Mathematically, the adversarial domain adaptation can be formulated as a minimax optimization problem with two competitive loss terms: (a) $\mathcal{L}_{cls}$ on label classifier C, which is expected to minimize the classification error on the labeled source samples, and (b) $\mathcal{L}_{adv}$ on the domain discriminator D used to distinguish the source and target domains, which is minimized on D but maximized on G:

$$\mathcal{L}_{cls}(C) = \mathbb{E}_{(x_s^i, y_s^i) \sim \mathcal{D}_s} L(C(g_s^i), y_s^i), \tag{1}$$

$$\mathcal{L}_{adv}(D) = -\mathbb{E}_{x_s^i \sim \mathcal{D}_s} \log[D(g_s^i)] - \mathbb{E}_{x_t^i \sim \mathcal{D}_t} \log[1 - D(g_t^i)], \tag{2}$$

where $g = G(x)$ is the output of feature extractor G, and $L(\cdot)$ is the cross-entropy loss. Thus, a generic formulation for adversarial domain adaptation is written as follows:

$$\min_{C,G} \mathcal{L}_{cls}(C), \ \min_{D,G} \mathcal{L}_{adv}(D). \tag{3}$$

The adversarial domain adaptation frameworks described above are able to extract domain-invariant representations to align two domains, but they still have some limitations. For instance, they usually fail to model the structure information of samples during adversarial learning. As a result, the aligned but distorted distributions of two domains may not lead to a low-risk classifier for target samples.

3.2 Cross-Domain Graph Convolutional Operation

In the theory of domain adaptation, the bounds of classification error ϵ_t in the target domain $\mathcal{D}_t$ is revealed in the following theorem.

Theorem 1. (Ben-David et al., 2010) Let h denote any classifier drawn from the hypothesis set $\mathcal{H}$. Given two domains $\mathcal{D}_s$ and $\mathcal{D}_t$, we have

$$\epsilon_t(h) \leq \epsilon_s(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_t) + \mathcal{C}, \tag{4}$$

where $\epsilon_s(h)$ is the expected error on source samples, and $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_t)$ denotes a discrepancy distance between the source and target domains. The last term $\mathcal{C} = \min_{h \in \mathcal{H}}(\epsilon_s(h, f_s) + \epsilon_t(h, f_t))$ denotes the difference in labeling functions across two domains, where f_s and f_t are labeling functions (Ben-David et al., 2010) for source domain and target domain.

With notable exceptions (Deng et al., 2019; Xie et al., 2018; Ma et al., 2019), existing methods mainly focus on minimizing the domain discrepancy term $\frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_t)$ in equation 4, by using strategies like maximum mean discrepancy or domain discriminator. However, if the third term $\mathcal{C}$ is large, we still cannot obtain a low expected error $\epsilon_t(h)$ for target samples. Recent studies (Xie et al., 2018) show that $\mathcal{C}$ in equation 4 can be further rewritten as:

$$\begin{aligned} \mathcal{C} = & \min_{h \in \mathcal{H}} \epsilon_s(h, f_s) + \epsilon_t(h, f_t) \\ \leq & \min_{h \in \mathcal{H}} \epsilon_s(h, f_s) + \epsilon_t(h, f_s) + \epsilon_t(f_s, f_t). \end{aligned} \tag{5}$$

The first two terms indicate the disagreement between h and the source labeling function f_s, which could be easily achieved because the source labels are available. The third term $\epsilon_t(f_s, f_t)$ implies the disagreement of labeling functions in the target domain. Even though the domain divergence is reduced as in existing work, samples in two domains may still have different class structure. As the result, the label functions f_s and f_t learned source and target domains still have certain disagreement which would lead to a large upper bound of $\epsilon_t(h)$ and result in dissatisfied performance on target domain (Xie et al., 2018). We argue that, by propagating sample and class structure information from source domain to target domain, the disagreement $\epsilon_t(f_s, f_t)$ could be reduced. As a result, we may achieve a lower expected error $\epsilon_t(h)$ in target domain.

In this work, our goal is to jointly exploit the sample-level and class-level structures to improve adversarial domain adaptation. As discussed above, adversarial domain adaptation optimizes the domain discriminator, label classifier, and feature extractor, simultaneously. In particular, the feature extractor is usually implemented by some popular convolutional neural network architectures like AlexNet or ResNet, which samples mini-batch data for efficient model training. Thus, the exploitation of data structure information shall be well aligned with the training of feature extractor and discriminator. To this end, we propose to construct mini-graphs on mini-batches to capture sample-level structure information, and then design a cross-domain graph convolutional operation to exploit both the sample and class structures.

First, we introduce the construction of mini-graphs. By using the feature extractor G, we obtain deep feature vectors for samples in the source and target domains. For every mini-batch, we construct a mini-graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to characterize the sample-level structure information in each domain, where $\mathcal{V}$ contains a set of mini-batch samples, and $\mathcal{E}$ represents edges of the mini-graph. Each sample is associated with a deep feature vector $\mathbf{g}^i$. Let A denote the adjacency matrix. The weight $A_{(i,j)}$ is the similarity between the i-th sample and j-th sample in the mini-batch:

$$A_{(i,j)} = \frac{\exp\left(sim(g^i, g^j)\right)}{\sum_j \exp\left(sim(g^i, g^j)\right)}, \tag{6}$$

where $sim(g^i, g^j)$ is the cosine similarity between i-th sample and j-th sample. The sample-level structural information in the representation space will be encoded in the mini-graphs $\mathcal{G}$. For mini-batches from source and target domains, we construct mini-graphs $\mathcal{G}_s$ and $\mathcal{G}_t$, and the corresponding adjacency matrices are denoted by $\tilde{A}_s$ and $\tilde{A}_t$.

Second, we design a cross-domain graph convolutional operation to jointly exploit the sample-level and class-level structure information across two domains. The sample-level structure in each domain could be learned by the graph convolutional networks (GCN) (Kipf & Welling, 2017). GCN allows the feature information to be propagated over graphs, by using an elegant layer-wise propagation rule based on neural networks. For mini-batches in the source domain, the feature propagation over neighborhood samples could be achieved by GCN:

$$Z_s = \sigma(\tilde{A}_s G(X_s) W_{gcn}), \tag{7}$$

where Z_s denotes the extracted GCN features of source samples. $G(X_s)$ denotes the CNN features of source samples. $\tilde{A}_s$ is the corresponding adjacency matrix. W_{gcn} is a trainable weight matrix and $\sigma(\cdot)$ is nonlinear function.

To attentively transfer relevant data structural information and class information from source domain to target domain, we propose a cross-domain feature propagation rule as follows:

$$Z_t = \sigma(\tilde{A}_t G(X_t) W_{gcn}) + \gamma\sigma(T Z_s W_{tra}), \tag{8}$$

where Z_t and $G(X_t)$ denote the extracted GCN features and CNN features of target samples, respectively. $\tilde{A}_t$ is the adjacency matrix. Source and target domains share the GCN parameters W_{gcn} to extract structural representations. $W_{tra} \in \mathbb{R}^{dim_s \times dim_t}$ denotes parameters of transfer layer. γ is a hyperparameter that controls the contribution of source samples for feature learning in the target domain.

In equation 8, $T \in \mathbb{R}^{b_{n_t} \times b_{n_s}}$ is a transfer matrix that indicates the class-wise correspondence across domains, b_{n_s} and b_{n_t} are batch size for source and target. Specifically, T is defined as the class label relationship between the source and target domains. If Z_t^i and Z_s^j have same label, $T^{i,j} = 1$. Otherwise, $T^{i,j} = 0$. As unsupervised domain adaptation assumes that the label information is not available in target domain, we infer pseudo labels for the target samples. In particular, the target pseudo labels are first initialized by a pre-trained classifier on labeled source data, and then the pseudo labels are updated along with the training process. These pseudo labels serve as a proxy of the missing true labels of target samples. If the discrepancy between two domains are minimized and meanwhile the class structures are well exploited and propagated, we expect that the pseudo labels will become very close to the true target labels. Accordingly, T would be able to encode reliable class-level structure information across two domains.

3.3 Transferable Feature Learning on Graphs (TFLG)

We propose the transferable feature learning approach on graphs (TFLG) by integrating the proposed cross-domain graph convolutional operation with unsupervised adversarial domain adaptation method. TFLG jointly optimizes the domain discriminator loss $\mathcal{L}_{adv}(D)$ and label classification loss $\mathcal{L}_{cls}(C)$ defined in equation 3, and the cross-domain feature extractor in equation 8. For each sample, the final feature representation h is the concatenation of CNN features g and cross-domain graph embedding z, i.e., $h = [g, z]$. z is a graph embedding vector in Z_s or Z_t defined in equation 7 and equation 8.

The overall objective function of TFLG is formulated as:

$$\begin{aligned}\min_{C} \quad & \mathbb{E}_{(x_s^i, y_s^i)\sim\mathcal{D}_s}\mathcal{L}_{cls}(C(h_s^i), y_s^i) \\ & +\lambda\left(\mathbb{E}_{x_s^i\sim\mathcal{D}_s}\log\left[D(h_s^i)\right] + \mathbb{E}_{x_t^i\sim\mathcal{D}_t}\log\left[1 - D(h_t^i)\right]\right) \\ \max_{D} \quad & \mathbb{E}_{x_s^i\sim\mathcal{D}_s}\log\left[D(h_s^i)\right] + \mathbb{E}_{x_t^i\sim\mathcal{D}_t}\log\left[1 - D(h_t^i)\right],\end{aligned} \tag{9}$$

where λ is a hyper-parameter that balances the label classifier and conditional domain discriminator.

After model training with back propagation, the source and target domains are well aligned in the new representation space that is also aware of structure information. Thus, the trained classifier C can be used to predict the labels of target samples. At the test stage, the hyper-parameter γ in equation 8 will be set to 0, as source data is not used, the equation 8 will only include the first term $\sigma(\tilde{A}_t G(X_t) W_{gcn})$ during the test.

3.4 TFLG with Memory Bank

The transfer matrix $T \in \mathbb{R}^{b_{n_t} \times b_{n_s}}$ in equation 8 captures the class label relationship between source and target in the corresponding batches. In practice, batch size is usually limited by GPU memory. The limitation of batch size would hinder the training of TFLG on datasets with a large number of images and categories like Office-Home (Venkateswara et al., 2017) which has 15,500 images with 65 categories, since the batch data of source and target may not have common categories with a high probability. As a result, T fails to build the class-wise correspondence across domains. To address this issue, we further improve our TFLG approach by incorporating a memory bank (TFLGM). In particular, we maintain a GCN feature memory bank V for source domain, which stores the most recent n batches of GCN source features. The memory bank $V = \{z_s^i, z_s^{(i+1)}, ..., z_s^{(i+n*b_{n_s})}\}$ is built as a queue, with the current batch enqueued and the oldest batch dequeued. Compared with batch of source, memory bank of source includes n times samples which greatly increases the probability of common categories that shared by source and target training data at each iteration. equation 8 can be rewritten as follows:

$$Z_t = \sigma(\tilde{A}_t G(X_t) W_{gcn}) + \gamma\sigma(T_m V W_{tra}), \tag{10}$$

where $T_m \in \mathbb{R}^{b_{n_t} \times n*b_{n_s}}$. If Z_t^i and V^j have same label, ${T_m}^{i,j} = 1$. Otherwise, ${T_m}^{i,j} = 0$.

3.5 Discussions

Curriculum learning for cross-domain feature propagation. In equation 8, the second term $\gamma\sigma(T Z_s W_{tra})$ only contributes to the training stage. As source data is not used during testing stage, the GCN features Z_s and the transfer matrix$T$ cannot be constructed. Our experiments show that setting γ to 0.1 could already result in promising classification performance for target samples. We will also discuss the sensitivity of our approach to the values of γ in the range {0.05, 0.1, 0.2, 0.3, 0.4, 0.5}. Moreover, we notice that setting γ to a fixed value may lead to a gap between training and test in unsupervised domain adaptation. Inspired by the scheduled sampling (Bengio et al., 2015), a curriculum learning strategy could be adopted to bridge such a gap. In particular, γ can be adaptively range in $[0, \gamma]$ to explore how would the variation of γ influences the class structure information between source and target during training process. In the appendix, parameter sensitivity analysis and different curriculum learning strategies for setting γ are evaluated and discussed.

Table 1: Classification accuracy(%) on Office-31 dataset.

Method	A→W	D→W	W→D	A→D	D→A	W→A	Avg
ResNet-50 (He et al., 2016)	68.4±0.2	96.7±0.1	99.3±0.1	68.9±0.2	62.5±0.3	60.7±0.3	76.1
DANN (Ganin et al., 2016)	82.0±0.4	96.9±0.2	99.1±0.1	79.7±0.4	68.2±0.4	67.4±0.5	82.2
SAFN (Xu et al., 2019)	88.8±0.4	98.4±0.0	99.8±0.0	87.7±1.3	69.8±0.4	69.7±0.2	85.7
SAFN+ENT (Xu et al., 2019)	90.1±0.8	98.6±0.2	99.8±0.0	90.7±0.5	73.0±0.2	70.2±0.3	87.1
DMRL (Wu et al., 2020)	90.8±0.3	99.0±0.2	**100.0±0.0**	93.4±0.5	73.0±0.3	71.2±0.3	87.9
DSBN (Chang et al., 2019)	93.3	99.1	**100.0**	90.8	72.7	**73.9**	88.3
TAT (Liu et al., 2019)	92.5±0.3	**99.3±0.1**	**100.0±.0**	93.2±0.2	**73.1±0.3**	72.1±0.3	88.4
CDAN (Long et al., 2018)	93.1±0.2	98.2±0.2	**100.0±.0**	89.8±0.3	70.1±0.4	68.0±0.4	86.6
CDANE (Long et al., 2018)	94.1±0.1	98.6±0.1	**100.0±.0**	92.9±0.2	71.0±0.3	69.3±0.3	87.7
CDAN+TFLG	94.6±0.3	99.2±0.1	**100.0±.0**	**94.1±0.3**	72.4±0.3	71.6±0.4	88.6
CDAN+TFLGM	**95.3±0.3**	99.0±0.1	**100.0±.0**	**94.1±0.2**	72.5±0.2	71.5±0.1	**88.7**

Table 2: Classification accuray(%) on Office-Home dataset.

Method	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg
ResNet-50 (He et al., 2016)	34.0	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DANN (Ganin et al., 2016)	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
TAT (Liu et al., 2019)	51.6	69.5	75.4	59.4	69.5	68.6	59.5	50.5	76.8	70.9	56.6	81.6	65.8
SAFN (Xu et al., 2019)	**52.0**	71.7	76.3	**64.2**	69.9	71.9	**63.7**	51.4	77.1	70.9	57.1	81.5	67.4
CDAN (Long et al., 2018)	49.0	69.3	74.5	54.4	66.0	68.4	55.6	48.3	75.9	68.4	55.4	80.5	63.8
CDANE (Long et al., 2018)	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
CDAN+TFLG	49.5	68.7	74.7	57.7	67.8	68.8	55.2	48.5	75.6	68.4	55.8	80.6	64.3
CDAN+TFLGM	50.7	71.3	75.0	59.1	69.4	71.2	59.5	49.1	77.2	70.5	56.2	81.3	65.9
CDANE+TFLGM	51.4	**72.0**	77.2	61.7	71.9	72.2	60.0	51.7	78.8	**72.8**	58.9	82.0	**67.6**

Flexible feature refinement module. In this work, we propose the transferable feature learning approach on graphs (TFLG) for unsupervised adversarial domain adaptation. Furthermore, this cross-domain graph convolution can be considered as a general approach for feature refinement, by exploiting structure information resided in two domains. Therefore, the proposed graph convolution can be potentially integrated with many other domain adversarial networks like DANN (Ganin et al., 2016), CDAN and CDANE (Long et al., 2018). Intuitively, this feature refinement approach provides an additional pathway to bridge source and target domains, in addition to the adversarial learning. Incorporating TFLG will facilitate the robust and discriminative knowledge transfer from source domain to target domain.

4 Experiments

In this section, we evaluate the performance of our approach on benchmark datasets for domain adaptation and provide detailed ablation studies to demonstrate the effectiveness of our approach.

4.1 Datasets and Experimental Settings

In the experiments, we use two benchmark datasets that are widely used for domain adaptation, including the Office-31 dataset (Saenko et al., 2010) that contains 4,652 images from 31 classes in three domains, and the Office-Home (Venkateswara et al., 2017) dataset that contains 15,500 images from 65 classes in four domains. We compare our approach with the following representative and state-of-the-art unsupervised domain adaptation methods including DANN (Ganin et al., 2016), CDAN (Long et al., 2018) , CDANE (Long et al., 2018), TAT (Liu et al., 2019), SAFN (Xu et al., 2019), DSBN (Chang et al., 2019) and DMRL (Wu et al., 2020). We follow the standard evaluation protocols for unsupervised domain adaptation (Ganin & Lempitsky, 2015; Long et al., 2017). Our approach and all the baseline methods use the *ResNet-50* (He et al., 2016) to extract CNN features. Descriptions of datasets, details of experimental settings, and experiments on more datasets are provided in the appendix due to space limit.

4.2 Results and Discussions

Table 1 shows the results of our approach and baselines on the Office-31 dataset. ResNet-50 is a simple baseline without any consideration of domain adaptation. Among the recent unsupervised adversarial domain adaptation methods, CDAN, CDANE and TAT outperform traditional domain adversarial network DANN, as the former methods model the discriminative information beyond the standard adversarial learning. Our proposed TFLG approach combined with CDAN (CDAN+TFLG) outperforms these baselines methods in most cases, and CDAN+TFLGM achieves the best classifi-

Table 3: Key components and flexible module studies on Office-31 dataset.

Method	A→W	D→W	W→D	A→D	D→A	W→A	Avg
ResNet-50 (He et al., 2016)	68.4±0.2	96.7±0.1	99.3±0.1	68.9±0.2	62.5±0.3	60.7±0.3	76.1
DANN (Ganin et al., 2016)	82.0±0.4	96.9±0.2	99.1±0.1	79.7±0.4	68.2±0.4	67.4±0.5	82.2
DANN*	82.0±0.4	97.1±0.1	98.9±0.3	81.4±02	65.5±0.4	65.3±0.4	81.7
DANN+TFLG(w/o cg)	84.1±0.5	97.6±0.4	99.1±0.1	83.9±0.4	65.9±0.2	67.5±0.4	83.0
DANN+TFLG	85.5±0.3	98.0±0.4	99.2±0.2	85.2±0.3	66.9±0.3	67.6±0.2	83.7
CDAN (Long et al., 2018)	93.1±0.2	98.2±0.2	100.0±.0	89.8±0.3	70.1±0.4	68.0±0.4	86.6
CDAN+TFLG(w/o cg)	93.6±0.2	99.0±0.1	100.0±.0	92.6±0.2	70.5±0.4	70.3±0.3	87.7
CDAN+TFLG	94.6±0.3	99.2±0.1	100.0±.0	94.1±0.3	72.4±0.3	71.6±0.4	88.6

cation accuracy on average. It is noteworthy that CDAN+TFLG/TFLGM takes advantages of both the adversarial learning framework and the cross-domain graph convolutional operation. Compared with CDAN, CDAN+TFLG/TFLGM significantly improves the classification accuracy in all six domain adaptation tasks, especially on several hard cases, e.g. **A** → **D** and **W** → **A**, where source and target domains are quite different. Experimental results validate the effectiveness of propagating sample-level and class-level structures via the proposed cross-domain graph convolution.

Table 2 shows the results on Office-Home dataset. Compared with CDAN, CDAN+TFLG has small improvement. Since Office-Home has 65 categories and due to the limitation of GPU memory, there is a high probability that the batch data of source and target don't share common categories. With memory bank of TFLGM, it improves the probability that source and target share more common categories. As the result, CDANE+TFLGM outperforms the state-of-the art methods like TAT and SAFN, and gets similar result with CDANE+TransNorm. Although CDANE+TFLGM only has two best accuracy on twelve transfer tasks with comparative methods, it still gets best average result and the results of most transfer tasks are close to the best.

4.3 Ablation Study

Key Components and Flexible Module. We conduct key components and flexible module studies on the Office-31 dataset to investigate the effects of key components and flexible deploy of our proposed TFLG method. In our ablation study, we have a simple baseline that is ResNet-50 model fine-tuned in the source domain. To evaluate the improvement achieved by TFLG, we use DANN (Ganin et al., 2016)[1] and CDAN (Long et al., 2018) respectively as basic methods. To investigate how the mini-graph convolution and cross-domain graph convolutional operation help boost classification performance in the target domain, we remove the cross-domain graph convolutional operation from our objective function, and prepare a variant of our approach denoted as "TFLG(w/o cg)", which can be regraded as a simple combination of unsupervised adversarial domain adaptation and GCN. Results in Table 3 show that CDAN+TFLG (w/o cg) and DANN+TFLG(w/o cg) improve over CDAN and DANN* respectively, which demonstrates the efficacy of exploiting sample-level structure information by GCN in adversarial domain adaptation. CDAN+TFLG and DANN+TFLG significantly outperforms CDAN and DANN*, proving the importance of bridging source and target domains via our cross-domain graph convolution and the flexibility of TFLG.

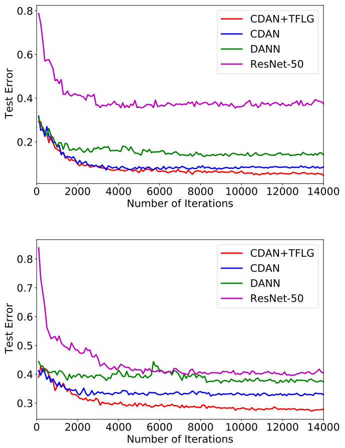


Figure 2: Convergence of ResNet-50, DANN, CDAN, CDAN+TFLG for transfer tasks: **A**→**W** and **W**→**A**.

Convergence. We testify the convergence of ResNet-50, DANN, CDAN and CDAN+TFLG, with the test errors on two domain adaptation tasks: **W**→**A** and **A**→**W** in the Office-31 dataset. As shown in Fig. 2, CDAN and CDAN+TFLG have faster convergence than DANN and ResNet-50 at the beginning. As training goes on, as CDAN+TFLG incorporates a cross-domain graph convolutional operation, it converges slightly slower than CDAN but gets lower test errors.

[1] DANN* is the result we get from the code (https://github.com/thuml/CDAN). For **D**→**A** and **W**→**A** tasks, the results are slightly different from the ones reported in the DANN paper.

5 Conclusions

In this paper, we proposed a transferable feature learning approach on graphs for unsupervised domain adaptation. Different from existing adversarial learning methods, our approach further exploits the sample-level and class-level structure information by designing a novel cross-domain graph convolutional operation. In this way, discriminative knowledge transfer across source and target domains could be achieved. Extensive experiments were conducted on two benchmark datasets, including Office-31 and Office-Home. Our approach outperforms the representative unsupervised domain adaptation methods in most cases. Moreover, ablation studies confirm the effectiveness of the joint class-wise and domain-wise alignment for unsupervised domain adaptation.

References

Mahsa Baktashmotlagh, Mehrtash T Harandi, Brian C Lovell, and Mathieu Salzmann. Unsupervised domain adaptation by domain invariant projection. In *ICCV*, pp. 769–776, 2013.

Mahsa Baktashmotlagh, Mehrtash T Harandi, Brian C Lovell, and Mathieu Salzmann. Domain adaptation on the statistical manifold. In *CVPR*, pp. 2481–2488, 2014.

Mahsa Baktashmotlagh, Mehrtash Harandi, and Mathieu Salzmann. Distribution-matching embedding for visual domain adaptation. *The Journal of Machine Learning Research*, 17(1):3760–3789, 2016.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1-2):151–175, 2010.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *NeurIPS*, pp. 1171–1179, 2015.

Yue Cao, Mingsheng Long, and Jianmin Wang. Unsupervised domain adaptation with distribution matching machines. In *AAAI*, 2018.

Woong-Gi Chang, Tackgeun You, Seonguk Seo, Suha Kwak, and Bohyung Han. Domain-specific batch normalization for unsupervised domain adaptation. In *CVPR*, pp. 7354–7362, 2019.

Minghao Chen, Shuai Zhao, Haifeng Liu, and Deng Cai. Adversarial-learned loss for domain adaptation. In *AAAI*, 2020.

Minmin Chen, Zhixiang Xu, Kilian Q Weinberger, and Fei Sha. Marginalized denoising autoencoders for domain adaptation. In *ICML*, pp. 1627–1634, 2012.

Zhijie Deng, Yucen Luo, and Jun Zhu. Cluster alignment with a teacher for unsupervised domain adaptation. In *ICCV*, 2019.

Zhengming Ding, Sheng Li, Ming Shao, and Yun Fu. Graph adaptive knowledge transfer for unsupervised domain adaptation. In *ECCV*, pp. 37–52, 2018.

Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *ICCV*, pp. 2960–2967, 2013.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, pp. 2672–2680, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

Judy Hoffman, Trevor Darrell, and Kate Saenko. Continuous manifold based adaptation for evolving visual domains. In *CVPR*, pp. 867–874, 2014.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

Wouter Marco Kouw and Marco Loog. A review of domain adaptation without target labels. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

Hong Liu, Mingsheng Long, Jianmin Wang, and Michael Jordan. Transferable adversarial training: A general approach to adapting deep classifiers. In *ICML*, pp. 4013–4022, 2019.

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

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *ICML*, pp. 2208–2217, 2017.

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *NeurIPS*, pp. 1640–1650, 2018.

Xinhong Ma, Tianzhu Zhang, and Changsheng Xu. Gcan: Graph convolutional adversarial network for unsupervised domain adaptation. In *CVPR*, pp. 8266–8276, 2019.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, pp. 213–226. Springer, 2010.

Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, pp. 3723–3732, 2018.

Swami Sankaranarayanan, Yogesh Balaji, Carlos D Castillo, and Rama Chellappa. Generate to adapt: Aligning domains using generative adversarial networks. In *CVPR*, pp. 8503–8512, 2018.

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

Rui Shu, Hung H Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. In *ICLR*, 2018.

Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert MÃžller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(May):985–1005, 2007.

Baochen Sun and Kate Saenko. Subspace distribution alignment for unsupervised domain adaptation. In *BMVC*, volume 4, pp. 24–1, 2015.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, pp. 7167–7176, 2017.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, pp. 5018–5027, 2017.

Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312: 135–153, 2018.

Yuan Wu, Inkpen Diana, and El-Roby Ahmed. Dual mixup regularized learning for adversarial domain adaptation. In *ECCV*, 2020.

Shaoan Xie, Zibin Zheng, Liang Chen, and Chuan Chen. Learning semantic representations for unsupervised domain adaptation. In *ICML*, pp. 5423–5432, 2018.

Ruijia Xu, Guanbin Li, Jihan Yang, and Liang Lin. Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation. In *ICCV*, pp. 1426–1435, 2019.

Zhang Yabin, Tang Hui, Jia Kui, and Mingkui Tan. Domain-symmetric networks for adversarial domain adaptation. In *CVPR*, pp. 5031–5040, 2019.

Hongliang Yan, Yukang Ding, Peihua Li, Qilong Wang, Yong Xu, and Wangmeng Zuo. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In *CVPR*, pp. 2272–2281, 2017.

Weichen Zhang, Dong Xu, Wanli Ouyang, and Wen Li. Self-paced collaborative and adversarial network for unsupervised domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

A Appendix

The appendix provides descriptions of datasets, details of experimental settings, additional experimental results, and justifications of the proposed transferable feature learning approach on graphs (TFLG) for unsupervised domain adaptation. In Section A.1, we introduce details about the experiment. Section A.2 presents the results on the ImageCLEF-DA dataset[2]. We conduct parameter sensitivity study on γ and evaluate different curriculum learning strategies by varying the parameter γ in Section A.3. The influence of different memory bank sizes for TFLGM is reported and discussed in Section A.4. We visualize the features of two domains learned by the ResNet50, CDAN and CDAN+TFLG in Section A.5.

A.1 Experimental Details

A.1.1 Datasets

Office-31 (Saenko et al., 2010) is one of the most popular benchmark data sets for domain adaptation, with 4,652 samples in 31 categories collected from three domains: *Amazon* (**A**) which contains images downloaded from amazon.com, *Webcam* (**W**) and *DSLR* (**D**), which include images taken by web camera and digital SLR camera under different settings. We evaluate our method on six domain adaptation tasks $\mathbf{A} \rightarrow \mathbf{W}$, $\mathbf{D} \rightarrow \mathbf{W}$, $\mathbf{W} \rightarrow \mathbf{D}$, $\mathbf{A} \rightarrow \mathbf{D}$, $\mathbf{W} \rightarrow \mathbf{A}$ and $\mathbf{D} \rightarrow \mathbf{A}$. For instance, in the task of $\mathbf{A} \rightarrow \mathbf{W}$, the *Amazon* domain is considered as the source domain, while the *Webcam* is treated as the target domain.

ImageCLEF-DA is a benchmark data set for ImageCLEF 2014 domain adaptation challenge, organized by selecting the 12 common categories shared by *Caltech-256* (**C**), *ImageNet ILSVRC 2012* (**I**), and *Pascal VOC 2012* (**P**). Each domain includes 600 images with 50 images per category. We build six transfer tasks: $\mathbf{I} \rightarrow \mathbf{P}$, $\mathbf{P} \rightarrow \mathbf{I}$, $\mathbf{I} \rightarrow \mathbf{C}$, $\mathbf{C} \rightarrow \mathbf{I}$, $\mathbf{C} \rightarrow \mathbf{P}$, and $\mathbf{P} \rightarrow \mathbf{C}$.

Office-Home (Venkateswara et al., 2017) is a more challenging dataset than *Office31*, which includes 15,500 images from 65 categories in office and home circumstance, consisted by four particularly dissimilar domains: Artistic images (**Ar**), Clip Art (**CI**), Product images (**Pr**), and Real-World images (**Rw**). We use all domains to set up 12 transfer tasks.

A.1.2 Implementation Details

We follow the standard evaluation protocols for unsupervised domain adaptation (Ganin & Lempitsky, 2015; Long et al., 2017). All the labeled source samples and unlabeled target samples are used for model training. The average classification accuracy and standard error are calculated over three random experiments in each domain adaptation task. We also adopt the image random flipping and cropping strategies as in CDAN (Long et al., 2018). As mentioned above, our approach and all the baseline methods use **ResNet-50** (He et al., 2016) to extract CNN features. In particular, the output of layer $pool5$ of ResNet50 are used as features.

We implement our approach by using PyTorch. We fine-tune the ResNet-50 from ImageNet pretrained models (Russakovsky et al., 2015), and the dimensionality of CNN feature vector is 2048. For graph convolution operation, we only adopt one graph convolution layer and the dimensionality of GCN feature vector is 1024. After cross-domain graph convolutional operation, we concatenate the CNN feature and GCN feature, and pass a randomly initialized linear layer to form the final feature. The dimensionality of the final feature vector is 256. As for the domain classifier and label classifier, we use the same architecture as CDAN. The transfer ratio γ from source to target is simply set as 0.1. We employ mini-batch stochastic gradient descent (SGD) with batch size of 36, momentum of 0.9, and the annealing strategy for model training (Ganin et al., 2016). For TFLGM, we set the size of memory bank to 4 batches. The learning rate is adjusted by $\eta_p = \frac{\eta_0}{(1+\alpha\beta)^{\beta}}$, where p is the training progress range from 0 to 1. $\eta_0 = 0.01$, $\alpha = 10$, $\beta = 0.75$ are optimized by the importance-weighted cross-validation (Sugiyama et al., 2007). For the discriminator, we increase λ from 0 to 1 by multiplying to $\frac{1-\exp{(-p\zeta)}}{1+\exp{(-p\zeta)}}$ with $\zeta = 10$.

Table 4: Classification accuray(%) on ImageCLEF-DA dataset.

method	I→P	P→I	I→C	C→I	C→P	P→C	Avg
ResNet-50 (He et al., 2016)	74.8±0.3	83.9±0.1	91.5±0.3	78.0±0.2	65.5±0.3	91.2±0.3	80.7
DANN (Ganin et al., 2016)	75.0±0.6	86.0±0.3	96.2±0.4	87.0±0.5	74.3±0.5	91.5±0.6	85.0
DMRL Wu et al. (2020)	77.3±0.4	90.7±0.3	97.4±0.3	91.8±0.3	76.0±0.5	94.8±0.3	88.0
SAFN (Xu et al., 2019)	78.0±0.4	91.7±0.5	96.2±0.1	91.1±0.3	77.0±0.5	94.7±0.3	88.1
SAFN+ENT (Xu et al., 2019)	79.3±0.1	**93.3±0.4**	96.3±0.4 is	91.7±0.0	77.6±0.1	**95.3±0.1**	**88.9**
TAT (Liu et al., 2019)	78.8±0.2	92.0±0.2	97.5±0.2	92.0±0.3	**78.2±0.4**	94.7±0.4	**88.9**
CDAN (Long et al., 2018)	76.7±0.3	90.6±0.3	97.0±0.4	90.5±0.4	74.5±0.3	93.5±0.4	87.1
CDANE (Long et al., 2018)	77.7±0.3	90.7±0.2	97.7±0.3	91.3±0.3	74.2±0.2	94.3±0.3	87.7
CDAN+TFLG	79.0±0.3	92.3±0.2	**97.9±0.3**	92.0±0.2	76.6±0.2	94.7±0.2	88.7
CDAN+TFLGM	**79.1±0.3**	91.8±0.1	**97.9±0.1**	**92.4±0.2**	77.0±0.4	95.2±0.3	**88.9**

A.2 Results on ImageCLEF-DA

Table 4 shows the experimental results of our approach and baselines on the ImageCLEF-DA dataset. In general, the comparisons on this dataset are similar to those on the Office-31 dataset. Methods that consider discriminative information outperform the standard domain adversarial networks. CDAN+TFLGM approach achieves the best accuracy in four out of six domain adaptation tasks, which gets the same average result as TAT and SAFN+ENT. Compared with CDAN, our method improves the average accuracy by 1.8%.

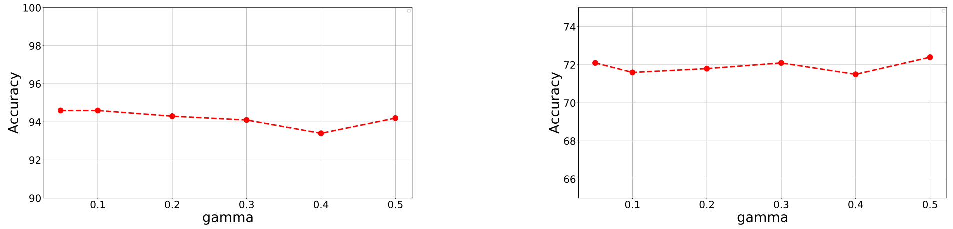
A.3 Parameter Sensitivity Study and Curriculum Learning for Cross-Domain Feature Propagation

The equation 11 below shows the proposed cross-domain feature propagation rule of our TFLG approach. In equation 11, the second term $\gamma\sigma(TZ_sW_{tra})$ controls how source sample-level and class-level structural information propagate to target.

$$Z_t = \sigma(\tilde{A}_t G(X_t) W_{gcn}) + \gamma\sigma(TZ_sW_{tra}). \tag{11}$$

A.3.1 Parameter Sensitivity Study

We discuss the sensitivity of parameter γ by evaluating it on **A**→**W** and **W**→**A** tasks in Office-31. The parameter γ are explored in the range $\{0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$. The experimental results are reported in Fig. 3(a) and 3(b). It can be observed that the domain adaptation performance is not very sensitive to the parameter γ.



(a) **A**→**W**

(b) **W**→**A**

Figure 3: Accuracy w.r.t. different γ on **A**→**W** and **W**→**A**.

A.3.2 Curriculum Learning Study

Inspired by the scheduled sampling (Bengio et al., 2015), a curriculum learning strategy could be adopted. In particular, γ can be adaptively set in the range $[0, \gamma]$. We explore the following four strategies for setting γ and present empirical evaluations on the Office-31 dataset.

- set γ to 0.1, denoted as "TFLG (γ=0.1)";
- decrease γ from 0.1 to 0 during training, denoted as "TFLG (γ=0.1→0)";
- increase γ from 0 to 0.1 during training, denoted as "TFLG (γ=0→0.1)";

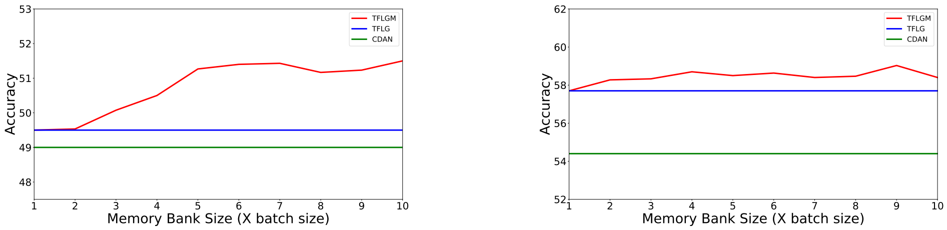
[2] http://imageclef.org/2014/adaptation

Table 5: Classification accuracy(%) on Office-31 dataset with different strategies for γ.

Method	A→W	D→W	W→D	A→D	D→A	W→A	Avg
ResNet-50 (He et al., 2016)	68.4±0.2	96.7±0.1	99.3±0.1	68.9±0.2	62.5±0.3	60.7±0.3	76.1
DANN (Ganin et al., 2016)	82.0±0.4	96.9±0.2	99.1±0.1	79.7±0.4	68.2±0.4	67.4±0.5	82.2
CDAN (Long et al., 2018)	93.1±0.2	98.2±0.2	100.0±.0	89.8±0.3	70.1±0.4	68.0±0.4	86.6
TFLG (γ=0→0.1→0)	94.7±0.4	**99.3±0.2**	**100.0±.0**	92.5±0.1	71.8±0.2	71.5±0.3	88.3
TFLG (γ=0.1→0)	**94.8±0.5**	99.1±0.2	**100.0±.0**	93.1±0.1	**72.6±0.5**	71.2±0.2	88.5
TFLG (γ=0.1)	94.6±0.3	99.2±0.1	**100.0±.0**	94.1±0.3	72.4±0.3	**71.6±0.4**	88.6
TFLG (γ=0→0.1)	94.7±0.3	99.1±0.2	**100.0±.0**	**94.4±0.5**	72.4±0.2	71.2±0.2	**88.7**

- increase γ from 0 to 0.1 and then decrease it to 0 during training, denoted as "TFLG (γ=0→0.1→0)".

Table 5 shows that all the strategies get better results than the baseline methods ResNet-50 (He et al., 2016) and recent adversarial domain adaptation methods, DANN (Ganin et al., 2016) and CDAN (Long et al., 2018). Results confirm that the performance of our TFLG approach is relatively robust to the different curriculum learning strategies. Among all the strategies, TFLG (γ=0→0.1) achieves the best average accuracy, which gradually incorporates source information to target feature learning. Furthermore, the good performance of TFLG (γ=0.1) and TFLG (γ=0→0.1) verify the importance of consistently bridging source and target domain via graph convolution as the training process goes on.



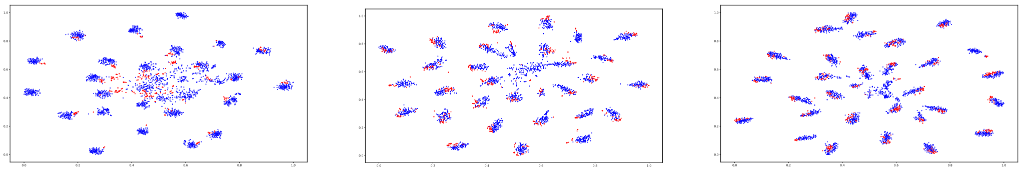
(a) **Ar**→**Cl**

(b) **Cl**→**Ar**

Figure 4: Accuracy w.r.t. different memory bank sizes on **Ar**→**Cl** and **Cl**→**Ar**.

A.4 Study on Memory Bank Size

We conduct empirical evaluations by setting different memory bank sizes on the Office-Home dataset with **Ar**→**Cl** and **Cl**→**Ar** transfer tasks. We run TFLGM with different memory bank sizes, and the results are shown in Fig. 4(a) and 4(b). With memory bank size increasing, compared with TFLG and DANN, TFLGM performs better on the **Ar**→**Cl** and **Cl**→**Ar** transfer tasks. Although larger memory bank sizes increase the probability of common categories that shared by source and target data at each training iteration, features in memory bank are not consistent because they are from recent training iterations. We can observe from Fig. 4(b) that, when memory bank size equals to 10 times batch size, the performance degradation is quite obvious.



(a) ResNet-50

(b) CDAN

(c) CDAN+TFLG

Figure 5: t-SNE feature visualization of (a) ResNet-50, (b) DANN, (c) TFLG for transfer task **A**→**W** (blue dot: **A**; red dot: **W**).

A.5 VISUALIZATION

We use t-SNE (Maaten & Hinton, 2008) to visualize the feature extracted by ResNet-50, CDAN and CDAN+TFLG for transfer task **A**→**W** in the Office-31 dataset. In Fig. 5(a), by only using ResNet-50, feature distributions of source and target domains are not well aligned. In Fig. 5(b), although CDAN has the ability to merge source and target domains into the unified feature distribution, there are some points scattered in the inter-class gap. Fig. 5(c) shows that CDAN+TFLG makes target clusters closely match with their corresponding source clusters, which proves the benefit of exploiting both sample-level and class-level structure information by using the proposed cross-domain graph convolutional operation.