Adversarial Multiple Domain Adaptation via Consistency on Graphs

Anonymous Author(s)
Affiliation
Address
email

Abstract

Unsupervised domain adaptation approaches have mainly focused on learning transferable representations from labeled source domains to an unlabeled target domain. Adversarial multiple domain adaptation can optimize domain-adaptive generalization bounds to learn invariant features with multiple source domains simultaneously. Despite the recent success, algorithms still suffer from generalization issues between the training and the testing distributions caused by the substantial domain shift, especially for the transferability of the top task-specific layers. Furthermore, relational information between labeled samples is largely ignored by prior works. In this paper, we propose a distance transduction mechanism via attentive message passing across established small graphs (minibatches) to enforce label consistency. The core training principle is to use meta-gradients as a bilevel learning scheme to enforce parameter adaptation. The empirical evidence shows that the proposed graph consistency criterion helps to achieve superior performance on different learning problems, including document sentiment analysis and digit classification, compared to the state-of-the-art domain adaptation methods.

1 Introduction

Adversarial Domain Adaptation (DA), which employs domain discriminators to learn domain-invariant features via minimizing an approximate domain discrepancy [11, 27, 36], has become a popular framework for transfer learning in recent years. By explicitly minimizing the errors of domain classifiers with gradient reversal, the upper bound of target risk is minimized. Although the general framework is theoretically sound, the problem of transferable feature learning and classification across multiple domains with unsupervised targets is still challenging, mainly due to the hardness in nonconvex training and the potential conditional shift during representation learning [49]. On one hand, unsupervised domain adaptation focuses on transferring knowledge for the unlabeled target domain given several fully labeled source domains. Hence the problem on how to utilize the supervision information effectively in adaptation with good generalization is the main goal in DA. On the other hand, most of the DA theories and algorithms focus on the single-source-single-target setting [4, 6, 7, 11, 25, 36]. In practice, however, the labeled data are usually collected from multiple labeled source domains with similar but different distributions. State-of-the-art adversarial DA systems contain no mechanism to tackle domain shift in the multiple unsupervised domains context.

Recently, theoretical analysis with a basic network architecture for multiple domain adversarial training has also been proposed [48]. Along with this direction, in this paper, we approach the key issue of domain shift among multiple source domains. Our key observation is that existing work only focuses on the pairwise relationship between each source domain and the target domain, while rich relationships among multiple source domains with labeled data are ignored. In general, the potential relations between data points matter a lot in quantifying instance-level domain shift. More specifically, discriminative features could be learned through propagation to unlabeled target instances through their interactions with labeled source instances. Hence it is necessary to encode the

Submitted to 33rd Conference on Neural Information Processing Systems (NeurIPS 2019). Do not distribute.

consistency of supervision simultaneously with the empirical risk error of source domains as well as the domain distribution discrepancy into the transferable embedding space. Recent advances in graph neural networks [15, 20, 28, 38] have led to massive success for graph-based analysis tasks on quantum chemistry [13, 18], social networks [43], relational reasoning [33], just to name a few. These approaches capture both the features of the data instances and the relations between them via a relational graph [2]. Apart from the natural structured data, instance interactions can also be regarded as a relational graph learning problem [2, 33, 44], which is more closely related to our goal. At a high level, we utilize a "self-attention"-style message passing algorithm to capture dependencies between data instances from multiple domains to preserve the labeling consistency along with domain-invariant feature learning.

Prior work [35, 46] on domain adaptation has already observed that the discrepancy not only exists in domain distributions but also in the learning procedure and the actual target inference procedure. During the inference phase, the gap among domains might be much larger than the one in the training phase because of the task-level shift for the held-out target data. Instead of manually fine-tuning the top layers in network architecture, our approach builds upon previous works about meta-learning to design an automatic parameter adaptation method to mitigate this issue. Basically, meta-learning [9, 30, 32] can be viewed as the process of discovering transferable representations among tasks such that the model could generalize to the new tasks sampled from the same distribution. Our episodic graph construction and propagation, with a set of the equal number of data points from multiple domains, makes adaptive parameter updating feasible. In this work, we seek to find the optimal domain transformation parameters, which avoids the more open-ended scenarios of unseen tasks in meta-learning research.

Our primary contributions can be summarized as follows. First, we develop a novel network architecture for adversarial multiple domain adaptation, which can provide a natural mechanism for modeling graph embedding consistency and its meta-learning algorithms. Second, our model is general – it combines adversarial learning and parameter adaptation phases into a unified framework, and requires essentially no manual network fine-tuning. Third, we perform empirical studies on tasks with different scales and characteristics, and demonstrate the superiority of the proposed architecture over competitive baseline methods. The source code of our work is available at `https://github.com/graph-mdan/graph-mdan`.

2 Backgrounds

In this section we first introduce the notations used throughout the paper and briefly describe the problem setup of multiple source domain adaptation.

2.1 Adversarial Multiple Domain Adaptation

Let *domain* correspond to a distribution $\mathcal{D}$ on the input space $\mathcal{X}$ and a labeling function $f : \mathcal{X} \to [0, 1]$. In the single source and single target adaptation setting, we use $\langle \mathcal{D}_{\mathcal{S}}, f_{\mathcal{S}} \rangle$ and $\langle \mathcal{D}_{\mathcal{T}}, f_{\mathcal{T}} \rangle$ to denote the source domain and the target domain, respectively. A hypothesis is a function $\eta : \mathcal{X} \to \{0, 1\}$. The *error* of a hypothesis η w.r.t. the labeling function f under distribution $\mathcal{D}_{\mathcal{S}}$ is defined as: $\varepsilon_{\mathcal{S}}(\eta, f) := \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{S}}} [|\eta(\mathbf{x}) - f(\mathbf{x})|]$. When f and η are binary classification functions, this definition reduces to the probability that η disagrees with f under $\mathcal{D}_{\mathcal{S}}$: $\mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\mathcal{S}}} [|\eta(\mathbf{x}) - f(\mathbf{x})|] = \Pr_{\mathbf{x}\sim\mathcal{D}_{\mathcal{S}}}(f(\mathbf{x}) \neq \eta(\mathbf{x}))$. Let the *risk* of hypothesis η be the error of η w.r.t the true labeling function under domain $\mathcal{D}_{\mathcal{S}}$, i.e., $\varepsilon_{\mathcal{S}}(\eta) := \varepsilon_{\mathcal{S}}(\eta, f_{\mathcal{S}})$. We denote $\varepsilon_{\mathcal{S}}$ and $\widehat{\varepsilon}_{\mathcal{S}}(\eta)$ as the true risk and the empirical risk of source domain. Then we denote $\varepsilon_{\mathcal{T}}(\eta)$ and $\widehat{\varepsilon}_{\mathcal{T}}(\eta)$ to be the true risk and the empirical risk on the target domain, respectively.

Our goal is to obtain a good generalization through learning from labeled samples of the source domain as well as the unlabeled samples of the target domain. In order to learn domain invariant representation, various distance measures are used to characterize the discrepancy between the source and target distributions, e.g., the *$\mathcal{H}$-divergence* [3–5, 11, 19], the *Maximum Mean Discrepancy* (MMD) [25, 26], the *Wasserstein distance* [7, 8], etc. In this paper, we build our architecture on the basis of the theoretical framework of $\mathcal{H}$-divergence which is defined as

$$d_{\mathcal{H}}(\mathcal{D}_{\mathcal{S}}, \mathcal{D}_{\mathcal{T}}) := 2 \sup_{\eta\in\mathcal{H}} \Big| \Pr_{\mathbf{x}\sim\mathcal{D}_{\mathcal{S}}} [\eta(\mathbf{x}) = 1] - \Pr_{\mathbf{x}\sim\mathcal{D}_{\mathcal{T}}} [\eta(\mathbf{x}) = 1] \Big|. \tag{1}$$

Based on $\mathcal{H}$-divergence, Ben-David et al. [3] and Blitzer et al. [5] introduced the generalization upper bound on the target risk with the source risk and the discrepancy between the single source domain and the target domain.

In the setting of multiple source domains, let $\{\mathcal{D}_{\mathcal{S}_i}\}_{i=1}^k$ and $\mathcal{D}_{\mathcal{T}}$ be k source domains and the target domain, respectively. With multiple source domains, we need to guarantee that a small training error $\widehat{\varepsilon}_{\mathcal{S}_i}(\eta)$ on multiple source domains can leads to a small test error $\varepsilon_{\mathcal{T}}(\eta)$ on target domain. Zhao et al. [48] propose a multisource domain adversarial networks (MDAN) and proved the following generalization bound for the target risk in terms of the average case source risks and the discrepancy between the multiple source domain and the target domain:

Theorem 1. Let $\mathcal{H}$ be a hypothesis class with $VCdim(\mathcal{H}) = d$. If $\{\widehat{\mathcal{D}}_{\mathcal{S}_i}\}_{i=1}^k$ are the empirical distributions generated with m $i.i.d.$ samples from each domain, and $\widehat{\mathcal{D}}_{\mathcal{T}}$ is the empirical distribution on the target domain generated from mk samples without labels, then, $\forall \nu \in \mathbb{R}_+^k$, $\sum_{i\in[k]} \nu_i = 1$, and for $0 < \delta < 1$, w.p.b. at least $1-\delta$, for all $\eta \in \mathcal{H}$, we have:

$$\varepsilon_{\mathcal{T}}(\eta) \leq \sum_{i\in[k]} \nu_i \left(\widehat{\varepsilon}_{\mathcal{S}_i}(\eta) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\widehat{\mathcal{D}}_{\mathcal{T}}, \widehat{\mathcal{D}}_{\mathcal{S}_i}) \right) + \lambda_\nu + \mathcal{O}\left(\sqrt{\frac{1}{km}\left(\log\frac{1}{\delta} + d\log\frac{km}{d} \right)} \right) \quad (2)$$

where λ_ν is the risk of the optimal hypothesis on the mixture source domain $\sum_{i\in[k]} \nu_i \mathcal{S}_i$ and $\mathcal{T}$.

Given a hypothesis class $\mathcal{H}$, its symmetric difference w.r.t. itself is defined as: $\mathcal{H}\Delta\mathcal{H} = \{\eta(\mathbf{x}) \oplus \eta'(\mathbf{x}) \mid \eta, \eta' \in \mathcal{H}\}$, where $\oplus$ is the XOR operation. Therefore, the MDAN aims to optimizes the upper bound in (2) in the following optimization problem:

$$\text{minimize} \quad \frac{1}{\rho} \log \sum_{i\in[k]} \exp\left(\rho(\widehat{\varepsilon}_{\mathcal{S}_i}(\eta) - \min_{\eta'\in\mathcal{H}\Delta\mathcal{H}} \widehat{\varepsilon}_{\mathcal{T},\mathcal{S}_i}(\eta')) \right) \quad (3)$$

where $\rho > 0$ is a constant. By combining all the labeled instances from k domains to one, convex combination from multiple domains to single domain is utilized in (2). That is, the gradient of MDAN in (3) is a convex combination of the gradients from all the domains.

2.2 Network Architecture

In unsupervised multisource domain adaptation, we are given a set of labeled instances $\{(\mathbf{x}_j^{\mathcal{S}_i}, y_j^{\mathcal{S}_i})\}_{j=1}^{n_i}$ sampled from source domains $\{\mathcal{D}_{\mathcal{S}_i}\}_{i=1}^k$, and a set of unlabeled instances $\{\mathbf{x}_j^{\mathcal{T}}\}_{j=1}^{n'}$ sampled from target domain $\mathcal{D}_{\mathcal{T}}$. To better understand the optimization problem defined in (3), note that each empirical risk term of source domain $\widehat{\varepsilon}_{\mathcal{S}_i}(\eta)$ is associated with a classification loss. Minimizing the source empirical loss for domain $\mathcal{D}_{S_i}$ can be generally written as:

$$\min_{\Theta} \quad \frac{1}{n_i} \sum_{j=1}^{n_i} \mathcal{L}_{cls}^j(\Phi_{cls}(\mathbf{x}_j^{\mathcal{S}_i}), y_j^{\mathcal{S}_i}; \Theta), \quad (4)$$

where $\mathcal{L}_{cls}^j$ can be chosen as the negative log likelihood function, and $\Phi_{cls}(\cdot)$ is neural networks where the input is an instance $\mathbf{x}_j^{\mathcal{S}_i}$ and the output indicates conditional probability assigning $\mathbf{x}_j^{\mathcal{S}_i}$ to label $y_j^{\mathcal{S}_i}$. For the empirical risk of discrepancy, adversarial modeling employs discriminators to distinguish samples from sources and the target so that the empirical risk $\widehat{\varepsilon}_{\mathcal{T},\mathcal{S}_i}(\eta')$ for one domain classifier has the following loss:

$$\min_{\Theta} \quad \frac{1}{n_i} \sum_{j=1}^{n_i} \mathcal{L}_{dsc}^j(\Phi_{dsc}^i(\mathbf{x}_j^{\mathcal{S}_i}), y_j'^{\mathcal{S}_i}; \Theta) + \frac{1}{n'} \sum_{j=1}^{n'} \mathcal{L}_{dsc}^j(\Phi_{dsc}^i(\mathbf{x}_j^{\mathcal{T}}), y_j'^{\mathcal{T}}; \Theta), \quad (5)$$

where $\mathcal{L}_{dsc}$ denotes loss of the domain discriminator, Φ_{dsc}^i is network layers involving in classifying samples $\{\mathbf{x}_j^{\mathcal{S}_i}, \mathbf{x}_j^{\mathcal{T}}\}$, and y_j' is a binary label indicating source or target point. The architectures for classifiers Φ_{cls} and Φ_{dsc}^i are generally feed-forward networks and can be readily extended depending on the task context [16, 21]. During backpropagation, the classification risks for k domains are combined through a gradient reversal layer (GRL) that changes the sign of gradients from discriminators [11].

We follow the domain-adversarial neural network (DANN) [11] for the design of network components in the setting of multiple sources: a feature extractor Φ_f, a class label predictor Φ_y, as well as domain classifiers $\{\Phi_d^i\}_{i=1}^k$. Then, the mapping functions become $\Phi_{cls} = \Phi_y(\Phi_f(\mathbf{x}))$ and $\Phi_{dsc}^i = \Phi_d^i(\Phi_f(\mathbf{x}))$. The bottom layers are the shared feature extractor that allows generic transferable feature learning and bridge the cross-domain discrepancy. Consequently, we have the associated parameters as θ_f, θ_y, and $\{\theta_d^i\}_{i=1}^k$, respectively.

3 Transferable Graph Embedding

3.1 Problem Overview

While the problem of multisource domain adaptation has been well approached in terms of both theoretical analysis and algorithm design, the learned domain-invariant embedding space cannot guarantee to obtain discriminative features for the target domain, especially when it comes to the case where the marginal label distributions differ between source and target domains [24, 49]. In this case, deep neural networks with large capacity might transform the target points into an embedding space that do not have desired prediction results with a small target error. In other words, domain-invariant representations will only lead to increasing joint error of both domains [49].

Intuitively, the learned representations via θ_f need to preserve the distance of source data points in the embedding space according to their labels. More specifically, data points with same labels should be closer than those with different labels in terms of embedding similarities, which we called as an *embedding consistency* property. To constrain the transferable representation space, we propose to optimize the mapping function Φ_f via constructing a loss for the feature extractor. The optimization problem can be mathematically formulated as:

$$\begin{aligned} &\min_{\theta_f} \mathcal{L}_{feat}(\Phi_f(\mathbf{x}), y''; \theta_f) \\ &s.t. \quad \min_{\Theta} \mathcal{L}_{da}(\Phi_y(\Phi_f(\mathbf{x})), \Phi_d(\Phi_f(\mathbf{x})), y, y'; \Theta) \end{aligned} \tag{6}$$

where $\mathcal{L}_{da} = \frac{1}{\rho} \log \sum_{i \in [k]} \exp\left(\rho(\mathcal{L}_{cls}(\theta_f, \theta_y) - \mu \mathcal{L}_{dsc}(\theta_f, \theta_d))\right)$, and $\Theta = \{\theta_f, \theta_y, \theta_d\}$. Here y'' is the newly introduced supervision for the consistency preserving, we will leave the exact form for the definition later. Arguably, the fundamental challenge of unsupervised domain adaptation is that the target domain has no label information. In light of this challenge, an intuitive method is to exploit transductive learning [10, 31] to retain the so-called embedding consistency and capture the dependencies among source-labeled and target-unlabeled instances. For this purpose, we separate the feature extraction as Stage-I and Stage-II, where instance embedding and transductive embedding are conducted, respectively. It gives rise to a new feature extractor $\Phi(\mathbf{x}) = \Phi_G(\Phi_f(\mathbf{x}))$ instead of $\Phi_f(\mathbf{x})$ in (6). Straightforwardly, relations between samples can be modeled by a *graph* for the embedding-level transduction. The nodes of the graph are associated with instances sampling from the source domains and the target domain during training, and the edges are built by identifying nearest neighbors based on the Stage-I embedding $\Phi_f(\mathbf{x})$. In the second stage, the relational graph learning Φ_G and the consistency objective $\mathcal{L}_{con}$ are introduced as $\mathcal{L}_{feat}$. The learning process of transferable graph embedding is illustrated in Fig. 1.

3.2 Propagation Model

To transit messages between data instances from source domains to the target domain, establishing a graph with sets of data instances is necessary. Instead of modeling the relation using a fully connected graph [12, 40], we enforce smooth property of embeddings on a k-NN graph. First, we connect k nearest neighbors for each target node according to the distance computed by $\|\Phi_f(\mathbf{x}^{\mathcal{T}}) - \Phi_f(\mathbf{x}^{\mathcal{S}_i})\|_2$, $i = 1, \cdots, k$. To effectively utilize the source labels, we also build edges between nodes and their neighbors in the same domain to encourage message passing within source domains. The learning goal is to propagate label information from labeled nodes towards the unlabeled ones.

Due to the powerful adaptive weight learning of self-attention mechanisms [17, 37], we explore the propagation model Φ_G around the graph attention networks [38]. Let the number of nodes be N, and the input of the l-th layer be $\mathbf{h}^l \in \mathbb{R}^{F^l}$, where the initial input $\mathbf{h}^0 = \Phi_f(\mathbf{x})$. The transformation function of the l-th layer are parameterized by $\theta_G^l = \{\mathbf{W}^l, \mathbf{a}^l\}$. In detail, $\mathbf{W}^l \in \mathbb{R}^{F^l \times F^{l+1}}$ is a

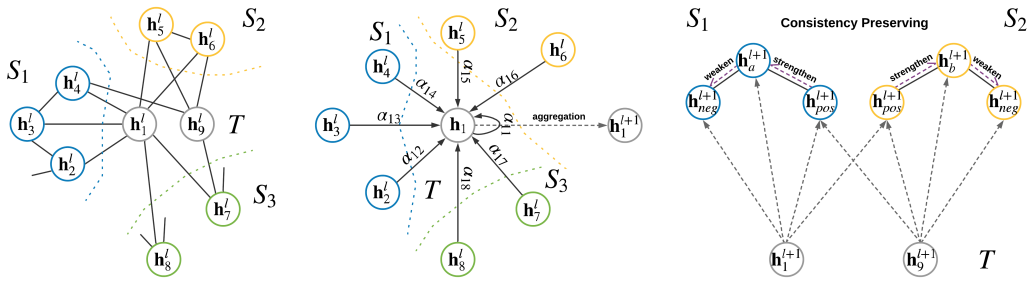

Figure 1: **Left:** Graph Construction. An example of k-NN graph built in a minibatch with three source domains and a target domain. Edges can be either directed or undirected. **Middle:** Transduction. Solid lines represent the assigned edge weights, and dashed lines represent aggregation by attention mechanism. For clarity nodes and edges without relations to $\mathbf{h}_1$ are omitted. **Right:** Supervision. Label consistency is preserved by a triplet loss that penalizes a small margin between positive neighbors and negative neighbors.

weight matrix of a linear mapping and $\mathbf{a}^l \in \mathbb{R}^{2F^{l+1}}$ is the parameter vector for the attention. With the above parameters, the weights between edges can be computed by:

$$\alpha_{pq}^l = \frac{\exp\left(\text{LeakyReLU}\left((\mathbf{a}^l)^T[\mathbf{W}^l\mathbf{h}_p^l||\mathbf{W}^l\mathbf{h}_q^l]\right)\right)}{\sum_{r\in\mathcal{N}_p}\exp\left(\text{LeakyReLU}\left((\mathbf{a}^l)^T[\mathbf{W}^l\mathbf{h}_p^l||\mathbf{W}^l\mathbf{h}_r^l]\right)\right)} \tag{7}$$

where $\cdot^T$ denotes matrix transpose and $||$ is the concatenation operation. The edge weights computed by (7) can be further used in feature aggregation of neighbors through $\mathbf{h}_p^{l+1} = \sigma(\sum_{q\in\mathcal{N}_p}\alpha_{pq}\mathbf{W}^l\mathbf{h}_q^l)$. The flexibility of masked attention makes it easy to define neighborhoods $\mathcal{N}_p$ for node p. Also, we applied the multi-head attention by averaging multi-head results. Therefore, higher-level embeddings with more expressive ability can be obtained for instances as well.

To achieve a unified optimization in (6), each k-NN graph is constructed for every minibatch so that the model is optimized in a minibatch updating manner. We keep samples from each domain with the same size m. The number of nodes N in one graph equals to $(k+1)m$. To cover the propagation in the second-order neighborhoods, we exploit a 2-layer graph attention network. Compared with conventional transductive learning on large-scale graphs such as citation [20] or social networks [43], we tackle a variety of small graphs with the shared transformation function Φ_G and the parameter set θ_G, whose training can alleviate the significant overhead and the prohibitively expensive computation of attention mechanism [23].

3.3 Consistency Preserving

With the propagation model, how to encode the discriminative information in source instances and enforce the label consistency via the path of target instances are crucial. Inspired by the large-margin nearest neighbor (LMNN) approach [41], we embed the distance learning on k-NN graphs with a triplet supervision [34, 42]. Specifically, given a source sample $\mathbf{h}_j$, consistency loss $\mathcal{L}_{con}$ on domain $\mathcal{D}_{\mathcal{S}_i}$ is defined as:

$$\sum_{\substack{\mathbf{x}_j\sim\mathcal{D}_{\mathcal{S}_i},\\ \mathbf{h}_j=\Phi_G(\Phi_f(\mathbf{x}_j))}} \left[\|\mathbf{h}_j - \mathbf{h}_j^{pos}\|_2^2 - \|\mathbf{h}_j - \mathbf{h}_j^{neg}\|_2^2 + \alpha\right]_+ . \tag{8}$$

where $\mathbf{h}_j$ is the output graph embedding for node j and $[t]_+ := \max\{t, 0\}$. $\mathbf{h}_j^{pos}$ and $\mathbf{h}_j^{neg}$ represent embeddings of neighbors with the same and different labels to node j, respectively. Hence, the supervision rule y'' can be given by triplets $\{\langle\mathbf{h}_j, \mathbf{h}_j^{pos}, \mathbf{h}_j^{neg}\rangle\}_j^{n_i}$ in source domain $\mathcal{D}_{\mathcal{S}_i}$, where the distance between positive and negative pairs is constrained by a margin α. For sampling, we generate triplets by selecting the most hard positive $\arg\max_{j|y_j=\hat{y}_j}\|\mathbf{h}_j - \hat{\mathbf{h}}_j\|_2^2$ and negative $\arg\min_{j|y_j\neq\hat{y}_j}\|\mathbf{h}_j - \hat{\mathbf{h}}_j\|_2^2$ samples within a minibatch, where $\hat{\mathbf{h}}_j$ is a neighborhood embedding of node j. Once we have Φ_G and $\mathcal{L}_{con}$, the entire problem defined by (6) becomes end-to-end differentiable so that all parameters can be learned jointly using gradient-based optimizations.

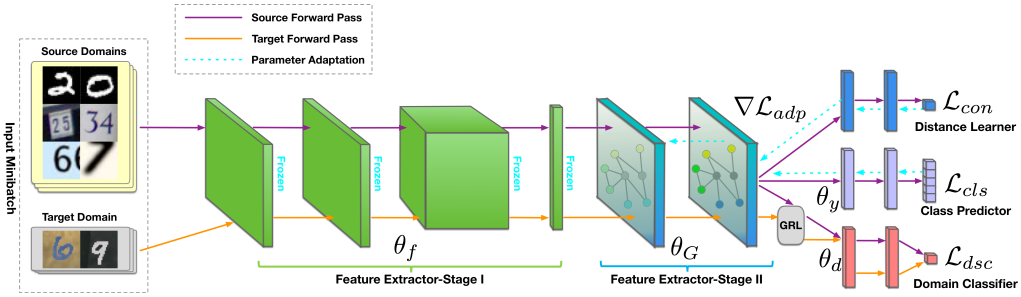

Figure 2: Illustration of the proposed model. Parameter adaptation with gradient pass is shown. Only one domain classifier is depicted for clarity. GRL is the abbreviation of Gradient Reversal Layer.

4 Domain-Adaptative Meta-Learning

4.1 Learning from Multiple Source Domains

Prior adversarial domain adaptation methods [11, 48] adopt the learned parameter for classifying target instances directly. In the inference phase, however, the domain discrepancy between source domains and the target domain might be larger because of the lack of parameter adaptation on the target domain. Meanwhile, transferability gap still exists in the top layer of the class predictor Φ_y, especially when the number of the fully connected layers grows [25, 45]. For this purpose, we incorporate meta-learning [1] as the parameter updating strategy in the context of multi-source adversarial domain adaptation.

Model-agnostic meta-learning (MAML) algorithm [9] aims to achieve a better generalization to new tasks with no or limited labels through effective learning of fast adaptation. This property allows for the ability to handle the domain shift between the test targets and the sources. Specifically, meta-learning for supervised learning assumes that there are a number of tasks $\mathcal{T}$, and data associated with each task $\mathcal{T}$ are randomly partitioned into two sets, $\mathcal{D}^{tr}$ and $\mathcal{D}^{val}$. The goal is to generalize from $\mathcal{D}^{tr}$ to $\mathcal{D}^{val}$, which is formally defined as minimizing the following objective:

$$\min_{\Theta} \sum_{\mathcal{T}\sim p(\mathcal{T})} \mathcal{L}(\mathcal{D}^{val}; \Theta - \alpha\nabla_{\Theta}\mathcal{L}(\mathcal{D}^{tr})) = \min_{\Theta} \sum_{\mathcal{T}\sim p(\mathcal{T})} \mathcal{L}(\mathcal{D}^{val}; \Theta^*) \tag{9}$$

where $p(\mathcal{T})$ is a distribution of task $\mathcal{T}$. Note that the above loss is based on $\mathcal{D}^{val}$, which means parameters are updated using gradient descent on $\mathcal{D}^{tr}$ to compute an optimal initial parameter for a good performance on $\mathcal{D}^{val}$. The basic mechanism is consistent with model generalization. The meta-gradient essentially leads to a bilevel learning scheme. Following the nomenclature in [30], we name *meta-train* and *meta-test* as training and testing procedure, respectively. Our final goal is to achieve a minimal error on target classification during meta-testing. To do so, fast adaptation between domains for parameters is conducted, which is to compute $\Theta - \alpha\nabla_{\Theta}\mathcal{L}(\mathcal{D}_{\mathcal{S}}; \Theta)$ iteratively in a few steps of the inner learning loop in meta-training. Since labels in source domains are available, it could be used for the losses of both inner loop and outer loop. This gives rise to a data partition in source domain, for instance, $\{\mathcal{D}_{\mathcal{S}_i}\}_{i=1}^{k-1}$ and $\mathcal{D}_{\mathcal{S}_k}$, to simulate the $\mathcal{D}^{tr}$ and $\mathcal{D}^{val}$. We keep the domain combinations in sources flexible for training a good parameter initialization. Next, we will focus on the specific meta-learn design of our network architecture.

The proposed network architecture is shown in Fig. 2. The labels y, y', and y'' of our unified objective given in (6) correspond to the class predictor in (4), domain classifiers in (5), and distance learner in (8). The unique challenge we are faced with is that the discriminator has no influence on the prediction result once the model is trained. That is, the parameters only needed to be adaptive to target samples are related to $\mathcal{L}_{cls}$ and $\mathcal{L}_{con}$. We use an adaptive objective function $\mathcal{L}_{adp}$ to represent the classification loss: $\mathcal{L}_{adp} = \mathcal{L}_{cls} + \gamma\nabla_{\theta_G}\mathcal{L}_{con}$. Here we omit ρ and regard $\mathcal{L}_{dsc}$ as a constant, with slightly abuse of notation. The meta-training phase can be summarized as the following optimization problem:

$$\min_{\theta_f,\theta_G,\theta_y,\theta_d} \mathcal{L}_{train}(\mathcal{D}_{\mathcal{T}}, \mathcal{D}_{\mathcal{S}_k}; \langle\theta_y, \theta_G\rangle - \alpha\nabla_{\langle\theta_y,\theta_G\rangle}\mathcal{L}_{adp}(\{\mathcal{D}_{\mathcal{S}_i}\}_{i=1}^{k-1}; \theta_f)) \tag{10}$$

where $\mathcal{L}_{train}$ combines all the three terms of the proposed algorithm. The associations with $\mathcal{L}_{adp}$ makes θ_G and θ_y updating for fast adaptation. The feature embedding parameters in Stage-I are frozen

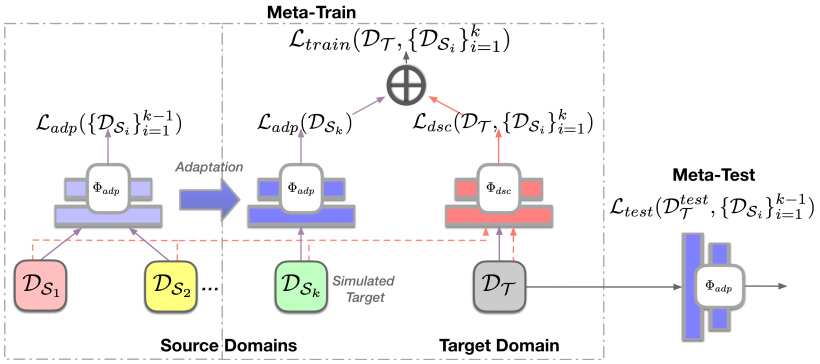

Figure 3: The workflow of meta-learning domain adaptation objective. The meta-training objective is to learn the parameters Φ_{adp} and Φ_{dsc} for minimizing source error and domain discrepancy, respectively. The meta-testing objective learns to generalize for a small target error.

during adaptation to keep them invariant for multiple domains. In contrast, all the parameters are meta-learned in the outer learning loop to fulfill the basic function of adversarial domain adaptation, formally defined as:

$$\min_{\theta_G, \theta_y} \mathcal{L}_{test}(\mathcal{D}_\mathcal{T}^{test}; \langle \theta_y, \theta_G \rangle - \alpha \nabla_{\langle \theta_y, \theta_G \rangle} \mathcal{L}_{adp}(\{\mathcal{D}_{\mathcal{S}_i}\}_{i=1}^{k-1}; \theta_f)) \tag{11}$$

where $\mathcal{L}_{test}$ is a learning procedure for $\mathcal{D}_{\mathcal{T}_{test}}$ composed of $\mathcal{L}_{cls}$ and $\mathcal{L}_{con}$ with a parameter tying to the counterpart updated via the split source domains $\{\mathcal{D}_{\mathcal{S}_i}\}_{i=1}^{k-1}$. The workflow is provided in Fig. 3. The algorithms on both meta-train and meta-test are shown in appendix due to space limit.

4.2 Discussion

While meta-learning has been verified to be effective in many real-world applications [14, 46, 47], it works under a rather restrictive assumption that samples of training and test tasks should be drawn from the same distribution [12, 39]. As a comparison, our objective in (9) to minimize the target task loss based on the source data w.r.t learning Θ sequentially is particularly well suited to the multiple domain adaptation problems. Often the case, domain shift exist between multiple relevant sources, even if they are sampled from the same distribution. Hence, rather than adapting the network embedding and making it transferable, the meta-learning framework directly approaches the shifts between training (source) and testing (target) distributions. Another challenge is that meta-learning encourages randomly generating target tasks. In contrast, our graph-based embedding learning induces variance for the feature extraction, where the generated representation for a given data point would be different, depending on other points in the sets and the corresponding relations with neighbors. In addition, the consistency preserving term gives rise to a continuous task distribution $p(\mathcal{T})$ which defines the strength between data points and their positive neighbors. Therefore, the graph neural networks do benefits on data capacity.

5 Experiments

We evaluate the proposed Graph-MDAN and its meta-learning variant with state-of-the-art methods on two various real-world datasets: the Amazon benchmark dataset [6] for sentiment classification and a digit classification benchmark involves 4 datasets: MNIST [22], MNIST-M [11], SVHN [29], and SynthDigits [11].

5.1 Sentiment Classification

We keep the same data split used in [48], which includes 4 review domains on product categories: Books, DVDs, Electronics, and Kitchen appliances. 4 experiments are conducted: in each of them, we pick one product as target domain and the rest as source domains. The following 5 baselines are compared: **MLPNet**, marginalized stacked denoising autoencoders (**mSDA**) [6], **DANN** [11] and **MDAN** [48]. Since DANN is under a single source domain setting, we use two protocols for a fair comparison. **B-DANN**: we report the one achieving the best performance on the target

Table 1: Sentiment Classification Accuracy on Amazon Dataset.

Train/Test	MLPNet	mSDA	B-DANN	C-DANN	MDAN	Graph-MDAN	
						w/o ml	with ml
D+E+K/B	0.7655	0.7698	0.7650	0.7789	0.7863	0.7969	**0.8017**
B+E+K/D	0.7588	0.7861	0.7732	0.7886	0.8065	0.8081	**0.8141**
B+D+K/E	0.8460	0.8198	0.8381	0.8491	0.8534	0.8562	**0.8632**
B+D+E/K	0.8545	0.8426	0.8433	0.8639	0.8626	0.8656	**0.8728**

* D: DVDs; E: Electronics; K: Kitchen appliances; B: Books.

domain; **C-DANN**: we combine all the source domains into a single one and then train it using DANN. Table 1 shows the results. We can observe that, with the graph neural networks, our proposed Graph-MDAN clearly outperforms all the baselines constantly in given 4 source/target settings. Also, the comparisons between the full version of our model and the variant without meta-learning certify the effectiveness of parameter adaptation.

5.2 Digit Classification

We experimented on the same datasets MDAN[48] used and compare the proposed approach with 5 baselines: **B-source**: The results in B-source is from a basic network trained on each source domain (20, 000 images) without domain adaptation and tested on the target domain. Among the three models, we report the one achieves the best performance on the test set. **C-source**: The results are from the same network but trained on a combination of three source domain and tested on the target domain. **B-DANN**: We trained DANNs[11] on each source-target domain pair (20, 000 images for each source) and test it on target. The result reported is the best score among the three. **C-DANN**: C-DANN is trained on a combination of three source domains and tested on the target domain. **MDAN**: We trained a soft version of MDAN on three source domains based on the same basic baseline network structure and test it on the target domain. The details of our architecture are shown in the appendix.

The results show that Graph-MDAN with meta-learning training scheme outperforms all the baseline models in the first and third experiment and is comparable with the best performance baseline model in the other two. We think that this is because the dataset SVHN and SynthDigit are more dissimilar to other datasets. Our experiments results indicate that naively combining all the data sources will not always improve the performance. MDAN shows its superiority for multiple source adaptation, we extend the idea of MDAN and by constructing a graph and preserving consistency, we make information transition between data instance from source domains to target domains become possible, which make new features to be more informative. Our proposed model achieved better performance than most of the baseline models, and the results using meta-learning scheme achieved overall best performance compared to other baseline models, which indicates that proper weight initialization and further finetuning can help to achieve better performance.

Table 2: Digits Classification Accuracy.

Train/Test	B-source	C-source	B-DANN	C-DANN	MDAN	Graph-MDAN	
						w/o ml	with ml
S+M+D/T	0.974	0.947	0.967	0.923	0.979	0.973	**0.981**
S+T+M/D	**0.875**	0.804	0.761	0.781	0.827	0.838	0.841
S+T+D/M	0.518	0.596	0.591	0.651	0.687	0.673	**0.691**
M+T+D/S	**0.837**	0.685	0.818	0.776	0.816	0.819	0.825

* D: SynthDigits; M: MNIST–M; S: SVHN; T: MNIST.

6 Conclusion

We propose a novel graph-based network architecture for adversarial multiple domain adaptation Graph-MDAN, which can nicely incorporate adversarial training on domain adaptation with meta-learning on parameter adaptation. The empirical study shows our models can outperform competitors on a variety of multiple domain adaptation tasks.

References

[1] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. In *NIPS*, 2016.

[2] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[3] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in neural information processing systems*, pages 137–144, 2007.

[4] 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.

[5] John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman. Learning bounds for domain adaptation. In *Advances in neural information processing systems*, pages 129–136, 2008.

[6] Minmin Chen, Zhixiang Xu, Kilian Weinberger, and Fei Sha. Marginalized denoising autoencoders for domain adaptation. *arXiv preprint arXiv:1206.4683*, 2012.

[7] Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. In *Advances in Neural Information Processing Systems*, pages 3730–3739, 2017.

[8] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9): 1853–1865, 2017.

[9] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.

[10] Alexander Gammerman, Volodya Vovk, and Vladimir Vapnik. Learning by transduction. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 148–155. Morgan Kaufmann Publishers Inc., 1998.

[11] 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.

[12] Victor Garcia and Joan Bruna. Few-shot learning with graph neural networks. *arXiv preprint arXiv:1711.04043*, 2017.

[13] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1263–1272. JMLR. org, 2017.

[14] Jiatao Gu, Yong Wang, Yun Chen, Kyunghyun Cho, and Victor OK Li. Meta-learning for low-resource neural machine translation. *arXiv preprint arXiv:1808.08437*, 2018.

[15] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017.

[16] Judy Hoffman, Sergio Guadarrama, Eric S Tzeng, Ronghang Hu, Jeff Donahue, Ross Girshick, Trevor Darrell, and Kate Saenko. Lsda: Large scale detection through adaptation. In *Advances in Neural Information Processing Systems*, pages 3536–3544, 2014.

[17] Yedid Hoshen. Vain: Attentional multi-agent predictive modeling. In *Advances in Neural Information Processing Systems*, pages 2701–2711, 2017.

[18] Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design*, 30(8):595–608, 2016.

[19] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting change in data streams. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages

180–191. VLDB Endowment, 2004.

[20] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[22] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[23] John Boaz Lee, Ryan A Rossi, Sungchul Kim, Nesreen K Ahmed, and Eunyee Koh. Attention models in graphs: A survey. *arXiv preprint arXiv:1807.07984*, 2018.

[24] Zachary C Lipton, Yu-Xiang Wang, and Alex Smola. Detecting and correcting for label shift with black box predictors. *arXiv preprint arXiv:1802.03916*, 2018.

[25] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015.

[26] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, pages 136–144, 2016.

[27] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, pages 1640–1650, 2018.

[28] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5115–5124, 2017.

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

[30] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.

[31] Marcus Rohrbach, Sandra Ebert, and Bernt Schiele. Transfer learning in a transductive setting. In *Advances in neural information processing systems*, pages 46–54, 2013.

[32] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850, 2016.

[33] Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In *Advances in neural information processing systems*, pages 4967–4976, 2017.

[34] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[35] Ozan Sener, Hyun Oh Song, Ashutosh Saxena, and Silvio Savarese. Learning transferrable representations for unsupervised domain adaptation. In *Advances in Neural Information Processing Systems*, pages 2110–2118, 2016.

[36] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7167–7176, 2017.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[38] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[39] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016.

[40] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.

[41] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(Feb):207–244, 2009.

[42] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2840–2848, 2017.

[43] Felix Wu, Tianyi Zhang, Amauri Holanda de Souza Jr, Christopher Fifty, Tao Yu, and Kilian Q Weinberger. Simplifying graph convolutional networks. *arXiv preprint arXiv:1902.07153*, 2019.

[44] Zhilin Yang, Bhuwan Dhingra, Kaiming He, William W Cohen, Ruslan Salakhutdinov, Yann LeCun, et al. Glomo: Unsupervisedly learned relational graphs as transferable representations. *arXiv preprint arXiv:1806.05662*, 2018.

[45] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

[46] Tianhe Yu, Chelsea Finn, Annie Xie, Sudeep Dasari, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot imitation from observing humans via domain-adaptive meta-learning. *arXiv preprint arXiv:1802.01557*, 2018.

[47] Xi Sheryl Zhang, Fengyi Tang, Hiroko Dodge, Jiayu Zhou, and Fei Wang. Metapred: Meta-learning for clinical risk prediction with limited patient electronic health records. *arXiv preprint arXiv:1905.03218*, 2019.

[48] Han Zhao, Shanghang Zhang, Guanhang Wu, José MF Moura, Joao P Costeira, and Geoffrey J Gordon. Adversarial multiple source domain adaptation. In *Advances in Neural Information Processing Systems*, pages 8559–8570, 2018.

[49] Han Zhao, Remi Tachet des Combes, Kun Zhang, and Geoffrey J Gordon. On learning invariant representation for domain adaptation. *arXiv preprint arXiv:1901.09453*, 2019.