Semantic-Specific Hierarchical Alignment Network for Heterogeneous Graph Adaptation

YuanXin Zhuang[1], Chuan Shi[1]✉, Cheng Yang[1], Fuzhen Zhuang[2,3], and Yangqiu Song[4]

[1] Beijing University of Posts and Telecommunications, Beijing 100876, China
[2] Institute of Artificial Intelligence, Beihang University, Beijing 100191, China
[3] SKLSDE, School of Computer Science, Beihang University, Beijing 100191, China
[4] Hong Kong University of Science and Technology, Hong Kong 999077, China
{zhuangyuanxin,shichuan,yangcheng}@bupt.edu.cn
zhuangfuzhen@buaa.edu.cn
yqsong@cse.ust.hk

Abstract. Node classification has been substantially improved with the advent of Heterogeneous Graph Neural Networks (HGNNs). However, collecting numerous labeled data is expensive and time-consuming in many applications. Domain Adaptation (DA) tackles this problem by transferring knowledge from a label-rich domain to a label-scarce one. However the heterogeneity and rich semantic information bring great challenges for adapting HGNN for DA. In this paper, we propose a novel semantic-specific hierarchical alignment network for heterogeneous graph adaptation, called HGA. HGA designs a sharing-parameters HGNN aggregating path-based neighbors and hierarchical domain alignment strategies with the MMD and L_1 normalization term. Extensive experiments on four datasets demonstrate that the proposed model can achieve remarkable results on node classification.

Keywords: Heterogeneous graph · Domain adaptation · Graph neural network.

1 Introduction

Graph Neural Networks (GNNs) have attracted much attention as it can be applied to many applications where the data can be represented as graphs [10, 22]. Heterogeneous Graphs (HGs), where nodes and edges can be categorized into multiple types, has also been proven to be effective to model many real-world applications, such as social networks and recommender systems [17, 1]. In order to learn representations of HG, there is a surge of Heterogeneous Graph Neural Networks (HGNNs) in the last few years, which employ graph neural network for heterogeneous graph to capture features from various types of nodes and relations. Different from traditional GNNs aggregating adjacent neighbors on homogeneous graph, HGNNs usually aggregates heterogeneous neighbors along meta-paths with a two-level attention mechanism [25, 2]. For example, HAN [25] designs node-level and semantic-level attention on meta-path based neighbors. MAGNN [2] employs the intra-metapath aggregation to incorporate intermediate semantic nodes and the inter-metapath aggregation to combine messages from multiple metapaths.

Recent advances in HGNNs have achieved remarkable results on node classification task which usually requires large amounts of labeled data to train a good network. However, In the real HGs, it is often expensive and laborsome to collect enough labeled data. A potential solution is to transfer knowledge from a related HG with rich labeled data (called source graph) to another HG with the shortage of labeled data (called target graph). Existing HGNNs are mostly developed for a single graph which has similar distribution in training and test data. However, different graphs generally have varied data distributions in real applications, which is usually called domain shift phenomenon [13]. Domain shift will undermine the generalization ability of learning models. Thus, those single graph based HGNNs which without addressing domain shift would fail to learn transferable representations.

Domain Adaptation (DA) [30] has shown promising advances for learning a discriminative model in the presence of the shift between the training and test data distributions. Given a target domain short of labels, DA aims to leverage the abundant labeled data from a source domain to help target domain learning, which has already attracted a lot of interests from the fields of Computer Vision [23, 13] and Natural Language Processing [14, 9]. The newest deep domain adaptation algorithms learn domain-invariant feature representations to mitigate domain shift with the Maximum Mean Discrepancy (MMD) metric [5, 12] or Generative Adversarial Net (GAN) [4, 21]. In recent years, there have been several attempts to apply domain adaptation to graph structure data. Some methods employ stacked autoencoders and MMD to learn network-invariant node representations [16, 15], while some methods apply graph convolutional network and adversarial learning to learn transferable embeddings [29, 26]. However, these techniques primarily focus on domain adaptation across homogeneous graphs, which cannot be directly applicable to heterogeneous graph. More recently, a heterogeneous graph domain adaptation method has been proposed to handle heterogeneity with multi-channel GCNs and two-level selection mechanisms [27]. But this method is not designed based on HGNN framework, which reduces its versatility. In addition, its performance improvement could be limited, because of lacking semantic-specific domain alignment mechanism to align the rich semantics of heterogeneous graphs separately.

Motivated by these observations, we make the first attempt to design a HGNN for DA, which is not a trivial task, due to the following two challenges: (1) How to adopt existing HGNNs to fully learn the knowledge of source graph and migrate to the target graph for the category-discriminative representations. We know that existing HGNNs are designed for single graph, we need to design an effective HGNN for knowledge transfer when adopting it for multiple graphs, (2) How to diminish the distribution discrepancy between source and target graphs to learn domain-invariant representations. Because of the domain shift among different semantics in source and target graphs, we need to design diverse domain alignment strategies to align distribution in source and target graphs intra- and inter-semantics.

In this paper, we propose a semantic-specific hierarchical alignment network for Heterogeneous Graph Adaptation (called HGA). The basic framework of HGA is a sharing-parameters HGNN which use hierarchical attentions to aggregate neighbor information via different meta-paths, to transfer knowledge from source graph to target graph. To be specific, HGA aggregates path-based neighbors with semantic-specific feature extractor and then classify and fuse these embeddings of different semantics with semantic-specific classifiers. In order to eliminate the distribution shift, a MMD normalization term is designed to align the feature distribution of nodes in source and target graph of every semantic path, and a L_1 normalization term is designed to align the class scores of nodes in target graph.

The contributions of this paper are summarized as follows:

- We study an important but seldom exploited problem of adopting DA to HGNN. The solution to this problem is crucial for label-absent HG representation.
- We design a novel heterogeneous graph adaptation method, called HGA, which employs a sharing-parameters HGNN with the MMD and L_1 normalization terms for domain-invariant and category-discriminative node representations.
- Experiments on eight transfer learning tasks show that the proposed HGA achieves significant performance improvements, compared to other state-of-the-art baselines.

2 Related Work

In this section, we briefly overview methods that are related to heterogeneous graph neural network and graph domain adaptation.

2.1 Heterogeneous Graph Neural Network

HGNN is designed to use GNN on heterogeneous graph, it can be divided into unsupervised and semi-supervised settings [24]. HetGNN [28] is the representative work of unsupervised HGNNs. It uses type specific RNNs to encode features for each type of neighbor vertices, followed by another RNN to aggregate the encoded neighbor representations of different types. semi-supervised HGNNs prefer to use attention mechanism to capture the most relevant structural and attribute information. There are a series of attention-based HGNNs was proposed [2, 25, 7]. HAN [25] uses a hierarchical attention mechanism to capture both node and semantic importance. MAGNN [2] extends HAN by considering both the meta-path based neighborhood and the nodes along the meta-path. HGT [7] uses each edge's meta relation to parameterize the Transformer-like self-attention architecture.

These HGNNs are designed for a single graph, and thus they can not be directly applied for knowledge transfer among multiple graphs.

2.2 Graph Domain Adaptation

There have been several attempts in the literature to apply domain adaptation to graph structure data. CDNE [16] incorporate MMD-based domain adaptation technique into deep network embedding to learn label-discriminative and network-invariant representations. ACDNE [15] integrate deep network embedding with the emerging adversarial domain adaptation technique to address cross-network node classification. DANE [29] applies graph convolutional network with constraints of adversarial learning regularization to learn transferable embeddings. UDA-GCN [26] used a dual graph convolutional networks to exploit both local and global relations of the graphs. However, these methods only consider knowledge transfer among homogeneous graphs. Recently, a heterogeneous graph domain adaptation method is proposed [27], which utilizes multi-channel GCNs to project nodes into multiple spaces, and proposes two-level selection mechanisms to choose the combination of channels and fuse the selected channels. Unfortunately, this method has limited performance improvement, due to lack semantic-specific domain alignment strategies.

3 Preliminaries

Definition 1. ***Heterogeneous Graph*** *[17]. A heterogeneous graph, denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consists of an object set $\mathcal{V}$ and a link set $\mathcal{E}$. Each node $v \in \mathcal{V}$ and each link $e \in \mathcal{E}$ are associated with their node type mapping function $\phi : \mathcal{V} \to \mathcal{A}$ and their link type mapping function $\psi : \mathcal{E} \to \mathcal{R}$. $\mathcal{A}$ and $\mathcal{R}$ denote the sets of predefined object types and link types, where $|\mathcal{A}| + |\mathcal{R}| > 2$.*

In heterogeneous graph, two objects can be connected via different semantic paths, which are called meta-paths.

Definition 2. ***Meta-path*** *[19]. A meta-path Φ is defined as a path in the form of $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_l} A_{l+1}$ (simplified to $A_1 A_2 \cdots A_{l+1}$), which describes a composite relation $R = R_1 \circ R_2 \circ \cdots \circ R_l$ between objects A_1 and A_{l+1}, where $\circ$ denotes the composition operator on relations.*

Definition 3. ***Domain Adaptation (DA)*** *[30]. Given a labeled source domain $\mathcal{D}_S$ and a unlabeled target domain $\mathcal{D}_T$, assume that their feature spaces and their class spaces are the same, i.e. $\mathcal{X}_S = \mathcal{X}_T$, $\mathcal{Y}_S = \mathcal{Y}_T$. The goal of domain adaptation is to use labeled data $\mathcal{D}_S$ to learn a classifier $f : \mathbf{x}_T \mapsto \mathbf{y}_T$ to predict the label $\mathbf{y}_T \in \mathcal{Y}_T$ of the target domain $\mathcal{D}_T$.*

Definition 4. ***Heterogeneous Graph Domain Adaptation***. *Given a source heterogeneous graph $\mathcal{G}_S = (\mathcal{V}_S, \mathcal{E}_S, \mathcal{X}_S, \mathcal{Y}_S)$, and a target heterogeneous graph $\mathcal{G}_T = (\mathcal{V}_T, \mathcal{E}_T, \mathcal{X}_T)$, where $\mathcal{A}_S \cap \mathcal{A}_T \neq \oslash$ and $\mathcal{R}_S \cap \mathcal{R}_T \neq \oslash$. $\mathcal{X}$ represents the* *features of $\mathcal{V}$, $\mathcal{Y}$ indicates the labels of $\mathcal{V}$. The goal of heterogeneous graph domain adaptation is to build a classifier f to predict the labels on $\mathcal{V}_T$ through reducing the domain shifts in different graphs and utilizing the structural information on both graphs, as well as $\mathcal{Y}_S$.*

Figure 1(a) demonstrates HGs on bibliographic data, where two authors can be connected via multiple meta-paths, e.g., Author-Paper-Author (APA) and Author-Paper-Conference-Paper-Author (APCPA). The meta-path APA depicts the co-author relation, whereas the APCPA depicts the co-conference relation. A task on heterogeneous graph domain adaptation is to predict the label of nodes in the target graph, with the help of the labeled source graph.

4 The Proposed Model

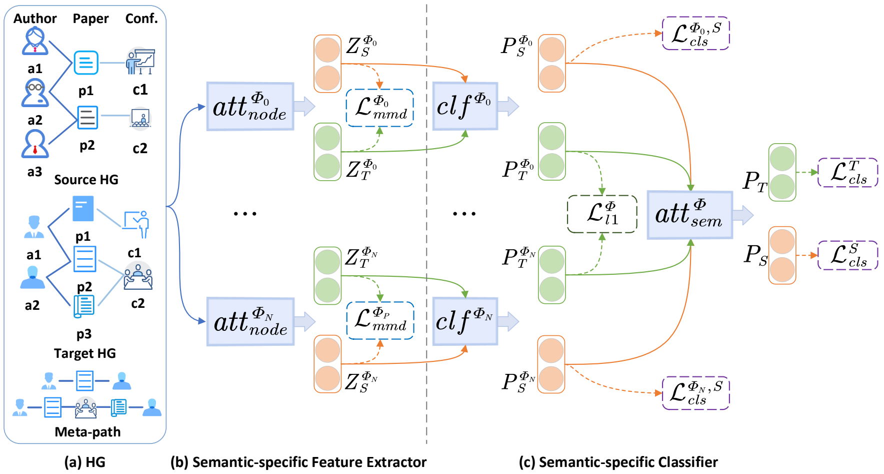


Fig. 1. An overview of the proposed hierarchical alignment network for Heterogeneous Graph domain Adaptation (HGA). HGA receives source graph instances with annotated ground truth and adapts to classifying the target samples. There are semantic-specific feature extractor and classifier for each meta-path.

In this paper, we propose a novel semantic-specific hierarchical alignment network for Heterogeneous Graph domain Adaptation (called HGA), whose basic idea is to adopt DA to HGNNs. As we know, existing HGNNs are designed for learning category-discriminative embeddings for node classification in single graph. That is, the learned embeddings can distinguish the category of nodes in a graph. Most HGNNs (e.g., HAN and MAGNN) employ node-level (also called intra-metapath) and semantic-level (also called inter-metapath) attention mechanism to aggregate node embeddings along different meta-paths. Unfortunately, these HGNNs cannot be directly applied to transfer knowledge among multiple graphs, because of domain shift.

In order to solve this obstacle, the proposed HGA adopts DA to HGNN with the goal of learning domain-invariant representations, as well as category-discriminative representations. HGA designs a shared parameters HGNN for source graph and target graph to aggregate path-based neighbors with semantic-specific feature extractor and then classify and fuse these embeddings of different meta-paths with semantic-specific classifiers. Furthermore, two normalized terms in HGA (i.e., mmd and l_1 terms) are proposed to hierarchically align the domain distribution of nodes intra- and inter-metapaths for domain-invariant representations. Concretely, the mmd term aligns the feature distribution of nodes in source and target graph of every semantic path, while the l_1 term aligns the class scores of nodes in target graph. The overall architecture of HGA is shown in Figure 1.

4.1 Semantic-specific GNN for DA

HGA adopts DA to a shared parameters HGNN for source graph and target graph, so the source graph can share the knowledge stored in the HGNN with target graph. Similar to typical HGNN architectures (e.g., HAN and MAGNN), HGA learns embeddings of nodes in source and target graphs through aggregating neighbors along a meta-path with a node-level attention in the semantic-specific feature extractor. However, different from existing HGNNs, the semantic-specific classifier in HGA first classifies these learned embeddings with linear classifiers to get class scores, and then fuse these scores with a semantic-level attention for node classification in source and target graphs. The classify-fuse mechanism in HGA has two benefits: (1) It makes full use of label information in source graph through constructing different node classification tasks for different meta-paths, which is helpful to learn category-discriminative representations. (2) It is convenient to align the class scores of nodes in target graph (i.e., the l_1 term).

Semantic-specific Feature Extractor Given a meta-path $\varPhi$, similar to typical HGNN architectures, the embedding of node i can aggregated from its meta-path based neighbors $\mathcal{N}_i^{\varPhi} = \{i\} \cup \{j | j \text{ connects with } i \text{ via the meta-path } \varPhi\}$ like HAN[25]:

$$\mathbf{z}_i^{\varPhi} = att_{node}^{\varPhi}\left(\mathbf{h}_j, j \in \mathcal{N}_i^{\varPhi}\right), \tag{1}$$

where $\mathbf{z}_i^{\varPhi}$ denotes the learned embedding of node i based on meta-path $\varPhi$, while $att_{node}^{\varPhi}$ is the feature extractor of meta-path $\varPhi$ which is a general component to aggregate neighbors. For example, $att_{node}^{\varPhi}$ can be the node-level attention in HAN which simply aggregates meta-path based neighbors, as well as the intra-metapath aggregation in MAGNN which also considers the nodes along the meta-path instances.

Semantic-specific Classifier Given an embedding $\mathbf{z}_i^{\varPhi}$ of node i based on meta-path $\varPhi$, the class scores $\mathbf{p}_i^{\varPhi}$ of node i in meta-path $\varPhi$ can be obtained by a classifier $clf^{\varPhi}$, such as linear classifier or softmax classifier:

$$\mathbf{p}_i^{\varPhi} = clf^{\varPhi}\left(\mathbf{z}_i^{\varPhi}\right). \tag{2}$$

As we know, semantic-specific embedding of nodes under a meta-path only reflect node characteristics from one aspect, while nodes contain multiple aspects of semantic information under different meta-paths. To learn a more comprehensive node embeddings, we need to fuse multiple semantics which can be revealed by meta-paths. To address the challenge of meta-path selection and semantic fusion in a heterogeneous graph, we adopt a semantic attention to automatically learn the importance of different meta-paths and fuse them for the specific task.

Given a set of meta-paths $\{\varPhi_0, \varPhi_1, \cdots, \varPhi_N\}$, after feeding the feature of node i into semantic-specific feature extractors and semantic-specific classifiers, it has N semantic-specific node embeddings $\left\{\mathbf{p}_i^{\varPhi_0}, \mathbf{p}_i^{\varPhi_1}, \cdots, \mathbf{p}_i^{\varPhi_N}\right\}$. To effectively aggregate different semantic embeddings, we use a semantic fusion mechanism:

$$\mathbf{p}_i = att_{sem}\left(\mathbf{p}_i^{\varPhi_j}\right) = \sum_{j=1}^{N} \beta_j \cdot \mathbf{p}_i^{\varPhi_j}, \tag{3}$$

where

$$\beta_j = \frac{\exp\left(\frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}} \mathbf{q}^{\mathrm{T}} \cdot \tanh\left(\mathbf{M} \cdot \mathbf{p}_i^{\varPhi} + \mathbf{b}\right)\right)}{\sum_{i=1}^{N} \exp\left(\frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}} \mathbf{q}^{\mathrm{T}} \cdot \tanh\left(\mathbf{M} \cdot \mathbf{p}_i^{\varPhi} + \mathbf{b}\right)\right)} \tag{4}$$

can be interpreted as the contribution of meta-path $\varPhi_j$ for the specific task. Respectively, $\mathbf{q}$ is the semantic attention vector; $\mathbf{M}$ and $\mathbf{b}$ denote the weight matrix and bias vector; $\mathbf{p}_i$ denotes the final embedding of node i, and att_{sem} denotes the semantic aggregator which aggregates embeddings of different meta-paths. Then we can apply the final embeddings to specific tasks and design different loss functions.

In order to obtain category-discriminative representations and facilitate knowledge transfer between graphs, we optimize three different loss functions as follows to reduce the domain discrepancy and enable efficient domain adaptation, and thus our model can differentiate class labels in the source graph and target graph, respectively.

– Semantic-specific source classifier minimizes the cross-entropy loss for the source graph in a mate-path $\varPhi$:

$$\mathcal{L}_{cls}^{\varPhi,S}\left(\mathcal{P}_S^{\varPhi}, \mathcal{Y}_S\right) = -\frac{1}{N_S}\sum_{i=1}^{N_S} y_i^S \log\left(\hat{y}_i^S\right), \tag{5}$$

– Source classifier minimizes the cross-entropy loss for the source graph after semantic fusion:

$$\mathcal{L}_{cls}^{S}\left(\mathcal{P}_S, \mathcal{Y}_S\right) = -\frac{1}{N_S}\sum_{i=1}^{N_S} y_i^S \log\left(\hat{y}_i^S\right), \tag{6}$$

- Target classifier minimizes the entropy loss for target graph information absorption. Here we employ the predicted labels of target nodes obtained by the shared classifiers:

$$\mathcal{L}_{cls}^{T}\left(\mathcal{P}_T\right)=-\frac{1}{N_T}\sum_{i=1}^{N_T}\hat{y}_i^T\log\left(\hat{y}_i^T\right), \tag{7}$$

where y_i^S denotes the label of the i-th node in the source graph, $\hat{y}_i^S$ is the classification prediction for the i-th node in source graph, $\hat{y}_i^T$ is the classification prediction for the i-th node in target graph, N_S is the node number of source graph and N_T is the node number of target graph.

The total classification loss of HGA can be represented by Eq.8, which can learn category-discriminative embeddings for source and target graph.

$$\mathcal{L}_C\left(\mathcal{G}_S,\mathcal{G}_T\right)=\mathcal{L}_{cls}^{\Phi,S}\left(\mathcal{P}_S^{\Phi},\mathcal{Y}_S\right)+\mathcal{L}_{cls}^{S}\left(\mathcal{P}_S,\mathcal{Y}_S\right)+\mathcal{L}_{cls}^{T}\left(\mathcal{P}_T\right). \tag{8}$$

4.2 Hierarchical Domain Alignment

Although the target graph can share knowledge from source graph with the shared parameters HGNN, the above model cannot solve the domain shift problem in domain adaptation. In order to learn domain-invariant representations, we furtherly propose semantic-specific hierarchical alignment mechanism, which includes intra-semantic feature alignment and inter-semantic label alignment. The intra-semantic feature alignment aims to map each pair of semantic between source and target graph into multiple different feature spaces and align semantic-specific distributions to learn multiple semantic-invariant representations. Since the target samples near semantic-specific decision boundary predicted by different classifiers might get different labels, the inter-semantic label alignment is designed to align the classifiers' output for the target nodes.

Intra-semantic Feature Alignment To learn domain-invariant representations, we need to match the distributions of source graph and target graph. In domain adaptation, the MMD [6] is a widely adopted nonparametric metric. We use the following term as the estimate of the discrepancy between source graph and target graph:

$$\mathcal{L}_{mmd}\left(\mathcal{G}_S,\mathcal{G}_T\right)=\left\|\frac{1}{N_S}\sum\phi\left(\mathbf{z}_S^{\Phi}\right)-\frac{1}{N_T}\sum\phi\left(\mathbf{z}_T^{\Phi}\right)\right\|_{\mathcal{H}}^2, \tag{9}$$

where $\phi\left(\cdot\right)$ denotes some feature projection function to map the original samples to reproducing kernel hilbert space. Through minimizing the Eq.9, the specific-semantic feature extractor could align the domain distributions between source domain and target domain under meta-path Φ.

Inter-semantic Label Alignment The classifiers are trained based on different meta-paths, hence they might have a disagreement on the prediction for target samples. Intuitively, the same target node predicted by different classifiers should get the same prediction. Hence, we need to minimize the classification discrepancy of nodes in target graph among all classifiers. Here we define the discrepancy loss as the differences of classification probability of nodes under different meta-paths with l_1 normalization.

$$\mathcal{L}_{l_1}\left(\mathcal{G}_T\right)=\frac{2}{N\times(N-1)}\sum_{j=1}^{N-1}\sum_{i=j+1}^{N}\mathbf{E}\left[|\mathbf{p}_T^{\Phi_i}-\mathbf{p}_T^{\Phi_j}|\right], \tag{10}$$

where N is the number of meta-path. By minimizing the Eq.10, the probabilistic outputs of all classifiers tend to be similar, which enforces the domain alignment under different semantic paths.

4.3 Optimization Objective

For HGA, a label prediction function f is trained by minimizing the overall objective as shown in Eq.11:

$$\mathcal{L}\left(\mathcal{G}_S,\mathcal{G}_T\right)=\mathcal{L}_C\left(\mathcal{G}_S,\mathcal{G}_T\right)+\lambda\left(\mathcal{L}_{mmd}\left(\mathcal{G}_S,\mathcal{G}_T\right)+\mathcal{L}_{l_1}\left(\mathcal{G}_T\right)\right), \tag{11}$$

where λ is the balance parameters. $\mathcal{L}_{mmd}$ and $\mathcal{L}_{l1}$ represent the intra-semantic feature alignment loss and the inter-semantic label alignment loss, respectively.

4.4 Discussion of the Proposed Model

Here we give the discussion of the proposed HGA as follows:

- From the optimization objective function Eq.11, we can find that HGA provides a general framework to adopt DA to HGNN. If we do not consider target graph $\mathcal{G}_T$, HGA degrades into tradition HGNNs for single graph. If we do not consider multiple meta-paths, HGA can be used for homogeneous graph domain adaptation. If ignoring the mmd and l_1 normalization terms, HGA becomes a simple DA-version of HGNN without considering domain shift. What's more, the $\mathcal{L}_{mmd}$ could be replaced by other adaptation methods, such as adversarial loss [3], coral loss [18]. And the $\mathcal{L}_{l1}$ could be replaced by other loss, such as l_2 regularization.
- Compared to traditional HGNNs, the additional complexity of HGA mainly lies on the mmd and l_1 normalization term. The complexity of mmd term is linear to the size of nodes in graphs, while the complexity of l_1 term is the square of the number of meta-paths, which is very small. And thus HGA has the same complexity with traditional HGNNs. Experiments also validate this point.

- The proposed HGA is highly efficient and can be easily parallelized. In the shared parameters HGNNs, the complexity is linear to the number of nodes and meta-path based node pairs. HGA can be easily parallelized, because att_{node}^{Φ} and att_{sem} can be parallelized across node pairs and meta-paths, respectively. The overall complexity is linear to the number of nodes and meta-path based node pairs.

5 Experiments

5.1 Datasets

We evaluate all the models on three academic attributed networks constructed from AMiner[20], DBLP[8] and ACM[11], and the detailed description is shown in Table 1. First, we adopt the constructed datasets from the work [27], i.e., ACM_A vs. ACM_B, DBLP_A vs. DBLP_B, AMiner_A vs. AMiner_B. For each pair of graphs, e.g., ACM_A vs. ACM_B, the density of meta-path edges is quite different between each other, which means they have domain discrepancy. (More statistics can be found in [27]).

Furthermore, we construct another pair of much larger graphs, i.e., ACM vs. DBLP. For ACM, we collected the papers published in SIGMOD, KDD, COLT and WWW, and divided them into four classes (*Database, data mining, machine learning, information retrieval*). The attributes of each paper in ACM are extracted from the paper title and abstract. For DBLP, we collected the papers published in ICDE, ICDM, PAKDD, PKDD, AAAI and SIGIR, and also divided them into the same classes. The attributes of each paper in DBLP are extracted from the paper title. Note that, DBLP has no overlapping nodes with ACM, and it is sparser than ACM. Finally, we have four pairs of datasets.

Dataset	# Nodes	# Meta-path Edges	Dataset	# Nodes	# Meta-path Edges
ACM_A	1,500	4,960	ACM_B	1,500	759
		6,691			3,996
		26,748			75,180
DBLP_A	1,496	2,602	DBLP_B	1,496	3,460
		673,730			744,994
		977,348			1,068,250
AMiner_A	1,500	4,360	AMiner_B	1,500	462
		554			3,740
		89,274			67,116
ACM	4,177	34,638	DBLP	4,154	38,966
		15,115,590			1,496,938

Table 1. Statistics of the experimental datasets.

5.2 Baselines and Implementation Details

Baselines In order to make a fair comparison and demonstrate the effectiveness of our proposed model, we compare our approach with both state-of-the-art single-domain methods as well as some domain adaptation methods on graphs.

State-of-the-art single-domain methods:

- **GCN** [10]: a typical deep convolutional network designed for homogeneous graphs.
- **HAN** [25]: a heterogeneous graph embedding method uses meta-paths as edges to augment the graph, and maintains different weight matrices for each meta-path-defined edge. And uses semantic-level attention to differentiate and aggregate information from different meta-paths.
- **MAGNN** [2]: a heterogeneous graph embedding method uses intra-metapath aggregation to sample some meta-path instances surrounding the target node and use an attention layer to learn the importance of different instances. And uses inter-metapath aggregation to learn the importance of different meta-paths.

Domain adaptation methods on graphs:

- **UDAGCN** [26] : a homogeneous graph domain adaptation method uses a dual graph convolutional networks to exploit both local and global relations of the graphs. And uses a domain adversarial loss for domain discrimination.
- **MuSDAC** [27]: a heterogeneous graph domain adaptation method uses multi-channel shared weight GCNs and a Two-level Selection strategy to aggregate embedding spaces to ensure both domain similarity and distinguishability.
- **HAN+MMD**: The feature generator is a shared parameters HAN architecture [25] for source and target graph. And a MMD[5, 12] regularization term is added on the final embedding.
- **MAGNN+MMD**: The feature generator is a shared parameters MAGNN architecture [2] for source and target graph. And a MMD[5, 12] regularization term is added on the final embedding.
- **HGA-HAN**: The att_{node}^{Φ} and att_{sem} in HGA framework is using the node-level attention and semantic-level attention in HAN [25].
- **HGA-MAGNN**: The att_{node}^{Φ} and att_{sem} in HGA framework is using the Intra-metapath aggregation and Inter-metapath aggregation in MAGNN [2].

To further validate the effectiveness of mmd loss and l_1 loss, we also evaluate several variants of HGA: (1) HGA$_{\neg l_1}$, only considers mmd loss; (2) HGA$_{\neg mmd}$, only considers l_1 loss; (3) HGA$_{\neg mmd \wedge \neg l_1}$, only has the shared weight architecture of HGNN.

Implementation Details All deep learning algorithms are implemented in Pytorch and trained with Adam optimizer. In the experiment we employ linear classifier. The learning rate is using the following formula: $\eta_p = \frac{\eta_0}{(1+\alpha p)^{\beta}}$, where

p is the training progress linearly changing from 0 to 1, $\eta_0 = 0.01$, $\alpha = 10$ and $\beta = 0.75$, which is optimized to promote convergence and low error on the source domain. To suppress noisy activations at the early stages of training, instead of fixing the adaptation factor λ, we gradually change it from 0 to 1 by a progressive schedule: $\lambda_p = \frac{2}{\exp(-\theta p)} - 1$, and $\theta = 10$ is fixed throughout the experiments [3]. This progressive strategy significantly stabilizes parameter sensitivity and eases model selection for HGA. As for single-domain network methods, we take the data from source graph as training set and the one from target graph as test set. As for domain adaptation method which acts on homogeneous graph, we ignore multiple semantics in HGs.

5.3 Results

We compare HGA with the baselines on four pairs of datasets and the results are shown in Table 2. From these results, we have the following insightful observations:

Source Target	ACM DBLP	DBLP ACM	ACM_B ACM_A	ACM_A ACM_B	AMiner_B AMiner_A	AMiner_A AMiner_B	DBLP_B DBLP_A	DBLP_A DBLP_B	AVG
GCN	0.472	0.517	0.580	0.698	0.755	0.481	0.357	0.459	0.540
HAN	0.632	0.694	0.687	0.686	0.676	0.698	0.768	0.812	0.707
MAGNN	0.678	0.702	0.713	0.693	0.703	0.717	0.772	0.817	0.724
UDAGCN	0.673	0.696	0.654	0.687	0.792	0.712	0.693	0.723	0.704
MuSDAC	0.704	0.764	0.788	0.730	0.810	0.761	0.795	0.819	0.771
HAN+MMD	0.724	0.712	0.727	0.706	0.832	0.745	0.774	0.817	0.755
MAGNN+MMD	0.735	0.728	0.739	0.721	0.843	0.749	0.781	0.820	0.765
HGA-HAN	0.785	0.759	0.791	0.757	0.929	0.83	0.828	0.835	0.814
HGA-MAGNN	**0.793**	**0.771**	**0.798**	**0.765**	**0.937**	**0.838**	**0.833**	**0.840**	**0.822**

Table 2. Performance comparison on classification accuracy.

- Because MAGNN not only considers the meta-path based neighborhood, but also consider the nodes along the meta-path. So the effect of HGA-MAGNN is better than that of HGA-HAN.
- HGA-MAGNN (or HGA-HAN) outperforms all compared baseline methods over all tasks. These encouraging results indicate that the proposed intra-semantic feature alignment mechanism can learn semantic-invariant representations for each pair of source and target graphs effectively, and inter-semantic label alignment mechanism can control all the classifiers to learn a consensus label for each target node.
- HAN+MMD and MAGNN+MMD are the simplest way to apply DA to HGNN. By comparing HAN+MMD (or MAGNN+MMD) and HAN (or MAGNN), we can see that traditional HGNNs cannot deal with the problem of domain shift. By comparing HAN+MMD (or MAGNN+MMD) and HGA-MAGNN (or HGA-HAN), we can observe that HGA is more effectively

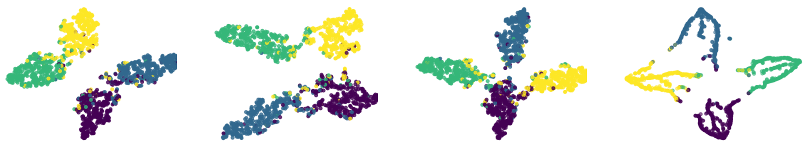
in transfering the knowledge of the source domain to the target domain by intra-semantic feature alignment mechanism and inter-semantic label alignment mechanism.

- Compared to HAN and MAGNN which do not consider the domain discrepancy between different graphs, HGA achieves better performance, especially on the pair of AMiner_A vs. AMiner_B where the density of meta-path edges is significantly different between them.
- For single-domain methods, HAN and MAGNN perform better GCN on most of tasks, which implies the superiority of considering heterogeneous graphs rather than homogeneous ones. The similar conclusion also can be concluded for domain adaptation methods.

Source Target	ACM DBLP	DBLP ACM	ACM_B ACM_A	ACM_A ACM_B	AMiner_B AMiner_A	AMiner_A AMiner_B	DBLP_B DBLP_A	DBLP_A DBLP_B	AVG
HGA-HAN$_{\neg mmd \wedge \neg l_1}$	0.667	0.676	0.752	0.746	0.835	0.814	0.813	0.824	0.766
HGA-HAN$_{\neg mmd}$	0.774	0.742	0.790	0.749	0.931	0.819	0.820	0.830	0.807
HGA-HAN$_{\neg l_1}$	0.765	0.739	0.784	0.751	0.920	0.822	0.826	0.833	0.805
HGA-HAN	**0.785**	**0.759**	**0.791**	**0.757**	**0.929**	**0.830**	**0.828**	**0.835**	**0.814**
HGA-MAGNN$_{\neg mmd \wedge \neg l_1}$	0.681	0.698	0.744	0.748	0.837	0.820	0.819	0.821	0.771
HGA-MAGNN$_{\neg mmd}$	0.782	0.764	0.788	0.752	0.935	0.827	0.825	0.832	0.813
HGA-MAGNN$_{\neg l_1}$	0.784	0.770	0.791	0.761	0.932	0.834	0.829	0.838	0.817
HGA-MAGNN	**0.793**	**0.771**	**0.798**	**0.765**	**0.937**	**0.838**	**0.833**	**0.840**	**0.822**

Table 3. Performance comparison on classification accuracy between HGA variants.

The ablation study results are shown in Table 3, From Table 3, we can easily observe that both HGA-MAGNN$_{\neg l_1}$ and HGA-MAGNN$_{\neg mmd}$ (or HGA-HAN$_{\neg l_1}$ and HGA-HAN$_{\neg mmd}$) outperform HGA-MAGNN$_{\neg mmd \wedge \neg l_1}$ (or HGA-HAN$_{\neg mmd \wedge \neg l_1}$), which verifies that on one hand the effectiveness of aligning the intra-semantic distributions of each pair of semantic in the source and target domains, and on the other hand the consideration of the inter-semantic label alignment to reduce the gap between all classifiers can help each classifier learn the knowledge from other classifiers.



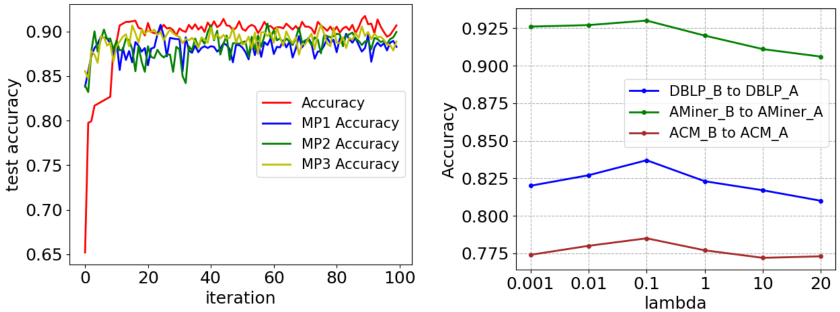
(a) MP 1 (b) MP 2 (c) MP 3 (d) Fusion

Fig. 2. The visualization of classifier's output and after fusion in target domain ('MP' means meta-path)

5.4 Analysis

Classification Output Visualization We visualize the outputs of each classifier and the output after meta-path fusion on the target domain of the task Aminer_B → Aminer_A with the model of HGA-HAN. From Figure 2, we can observe that the results in Figure 2(d) are better than the ones in Figure 2(a)(b)(c), which show that by fusing more information from meta-paths can lead to performance improvement. What's more, we can see that the target nodes near the class boundaries are more likely to be misclassified by the classifiers learned from single meta-path of source graph, while we can minimize the discrepancy among all classifiers by using inter-semantic label alignment.

Algorithm Convergence To investigate the convergence of our algorithm, we record the performance of target domain over all meta-path classifiers and the fusion one during the iterating on the task Aminer_B → Aminer_A. The results are shown in Figure 3(a). We can find that all algorithms can converge very fast, e.g., less than 20 iterations. Particularly, the fusion one is more stable with better accuracy, which illustrates the benefits of fusing multiple meta-paths again.



(a) Convergence

(b) Accuracy w.r.t λ

Fig. 3. Algorithm convergence and parameter sensitivity.

Parameter Sensitivity To study the sensitivity of λ, which controls the importance of mmd loss and l_1 loss. We sample the values in {0.001, 0.01, 0.1, 1, 10, 20}, and perform the experiments on tasks DBLP_B → DBLP_A, AMiner_B → AMiner_A, and ACM_B → ACM_A. All the results are shown in Figure 3(b), and we find that the accuracy first increases and then decreases, and displays as a bell-shaped curve. The results further illustrate the necessity of proper constraint of domain alignments. Finally, we set $\lambda = 0.1$ to achieve good performance.

6 Conclusion

Most previous heterogeneous graph neural networks focus on a single graph and fail to consider the knowledge transfer across graphs. In this paper, we study the problem of HGNN for domain adaptation, and propose a semantic-specific hierarchical alignment network for heterogeneous graph adaptation, called HGA. The HGA employs a shared parameters HGNN with the *mmd* and $l1$ normalization terms for domain-invariant and category-discriminative node representations. Experiments on eight transfer learning tasks validate the effectiveness of the proposed HGA.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (No. U20B2045, 61772082, 61702296, 62002029). It is also supported by "The Fundamental Research Funds for the Central Universities 2021RC28".

References

1. Fan, S., Zhu, J., Han, X., Shi, C., Hu, L., Ma, B., Li, Y.: Metapath-guided heterogeneous graph neural network for intent recommendation. In: KDD. pp. 2478–2486 (2019)
2. Fu, X., Zhang, J., Meng, Z., King, I.: Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In: WWW. pp. 2331–2341 (2020)
3. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: ICML. pp. 1180–1189 (2015)
4. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems **27**, 2672–2680 (2014)
5. Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., Smola, A.: A kernel method for the two-sample-problem. Advances in neural information processing systems **19**, 513–520 (2006)
6. Gretton, A., Borgwardt, K.M., Rasch, M.J., Schölkopf, B., Smola, A.: A kernel two-sample test. The Journal of Machine Learning Research **13**(1), 723–773 (2012)
7. Hu, Z., Dong, Y., Wang, K., Sun, Y.: Heterogeneous graph transformer. In: WWW. pp. 2704–2710 (2020)
8. Ji, M., Sun, Y., Danilevsky, M., Han, J., Gao, J.: Graph regularized transductive classification on heterogeneous information networks. In: ECML-PKDD. pp. 570–586. Springer (2010)
9. Jiang, J., Zhai, C.: Instance weighting for domain adaptation in nlp. In: ACL (2007)
10. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
11. Kong, X., Yu, P.S., Ding, Y., Wild, D.J.: Meta path-based collective classification in heterogeneous information networks. In: Chen, X., Lebanon, G., Wang, H., Zaki, M.J. (eds.) CIKM. pp. 1567–1571. ACM (2012)

12. Long, M., Zhu, H., Wang, J., Jordan, M.I.: Deep transfer learning with joint adaptation networks. In: ICML. vol. 70, pp. 2208–2217 (2017)
13. Luo, Y., Zheng, L., Guan, T., Yu, J., Yang, Y.: Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In: CVPR. pp. 2507–2516 (2019)
14. Ramponi, A., Plank, B.: Neural unsupervised domain adaptation in nlp—a survey. arXiv preprint arXiv:2006.00632 (2020)
15. Shen, X., Dai, Q., Chung, F.l., Lu, W., Choi, K.S.: Adversarial deep network embedding for cross-network node classification. In: AAAI. pp. 2991–2999 (2020)
16. Shen, X., Dai, Q., Mao, S., Chung, F.l., Choi, K.S.: Network together: Node classification via cross-network deep network embedding. TNNLS (2020)
17. Shi, C., Li, Y., Zhang, J., Sun, Y., Philip, S.Y.: A survey of heterogeneous information network analysis. TKDE **29**(1), 17–37 (2016)
18. Sun, B., Saenko, K.: Deep coral: Correlation alignment for deep domain adaptation. In: ECCV. pp. 443–450. Springer (2016)
19. Sun, Y., Han, J., Yan, X., Yu, P.S., Wu, T.: Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. Proceedings of the Vldb Endowment **4**(11), 992–1003 (2011)
20. Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., Su, Z.: Arnetminer: extraction and mining of academic social networks. In: Li, Y., Liu, B., Sarawagi, S. (eds.) KDD. pp. 990–998. ACM (2008)
21. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: CVPR. pp. 2962–2971 (2017)
22. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
23. Wang, M., Deng, W.: Deep visual domain adaptation: A survey. Neurocomputing **312**, 135–153 (2018)
24. Wang, X., Bo, D., Shi, C., Fan, S., Ye, Y., Yu, P.S.: A survey on heterogeneous graph embedding: Methods, techniques, applications and sources. arXiv preprint arXiv:2011.14867 (2020)
25. Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., Yu, P.S.: Heterogeneous graph attention network. In: WWW. pp. 2022–2032 (2019)
26. Wu, M., Pan, S., Zhou, C., Chang, X., Zhu, X.: Unsupervised domain adaptive graph convolutional networks. In: WWW. pp. 1457–1467 (2020)
27. Yang, S., Song, G., Jin, Y., Du, L.: Domain adaptive classification on heterogeneous information networks. In: IJCAI. pp. 1410–1416 (2020)
28. Zhang, C., Song, D., Huang, C., Swami, A., Chawla, N.V.: Heterogeneous graph neural network. In: KDD. pp. 793–803 (2019)
29. Zhang, Y., Song, G., Du, L., Yang, S., Jin, Y.: Dane: Domain adaptive network embedding. arXiv preprint arXiv:1906.00684 (2019)
30. Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., He, Q.: A comprehensive survey on transfer learning. CoRR **abs/1911.02685** (2019), http://arxiv.org/abs/1911.02685