GDA-HIN: A Generalized Domain Adaptive Model across Heterogeneous Information Networks

Tiancheng Huang*
Zhejiang University
Westlake University
Westlake Institute for Advanced
Study, Hangzhou, China
huangtiancheng@westlake.edu.cn

Ke Xu*
Westlake University
Westlake Institute for Advanced
Study, Hangzhou, China
xuke@westlake.edu.cn

Donglin Wang†
Westlake University
Westlake Institute for Advanced
Study, Hangzhou, China
wangdonglin@westlake.edu.cn

ABSTRACT

Domain adaptation using graph-structured networks learns label-discriminative and network-invariant node embeddings by sharing graph parameters. Most existing works focus on domain adaptation of homogeneous networks. The few works that study heterogeneous cases only consider shared node types but ignore private node types in individual networks. However, for given source and target heterogeneous networks, they generally contain shared and private node types, where private types bring an extra challenge for graph domain adaptation. In this paper, we investigate Heterogeneous Information Networks (HINs) with both shared and private node types and propose a Generalized Domain Adaptive model across HINs (GDA-HIN) to handle the domain shift between them. GDA-HIN can not only align the distribution of identical-type nodes and edges in two HINs but also make full use of different-type nodes and edges to improve the performance of knowledge transfer. Extensive experiments on several datasets demonstrate that GDA-HIN can outperform state-of-the-art methods in various domain adaptation tasks across heterogeneous networks.

CCS CONCEPTS

• **Computing methodologies → Machine learning**.

KEYWORDS

Heterogeneous Information Networks, Transfer Learning, Domain Adaptation, Distribution Alignment

---

*Both authors contributed equally to this research.
†Corresponding author.

Figure 1: Challenges on DA across HINs: (a) feature and (b) structure discrepancies between source and target HINs.

1 INTRODUCTION

Domain Adaptation (DA) aims to learn transferable representations for problems those sample and label spaces remain unchanged, but the probability distribution is different [2, 7, 9, 20]. Recently, DA across graphs has been beginning to draw much attention [11, 12, 16, 19]. It aims to transfer the knowledge learned from a source network to a target network by learning label-discriminative and network-invariant node embeddings. However, existing methods mainly focus on the problem of homogeneous networks. In this paper, we study a generalized domain adaptation across Heterogeneous Information Networks (HINs), where nodes are fully labeled in the source network while completely unlabeled in the target network. We intend to minimize the domain shift and transfer the model trained on the source network to the target network.

However, this objective faces several challenges: 1) First, each HIN is composed of multiple different types of nodes and edges. For example, the source network DBLP [5] in Fig. 1(a) contains four types of nodes: papers (P), authors (A), venues (V), and terms (T), as well as different types of edges between them. This means that different types of nodes and edges are in different semantic spaces and have different data distributions, which results in three discrepancies between *pairwise node-type* distributions (See Fig. 1 (a.top)). 2) Second, the node and edge types in two heterogeneous

networks are not identical in general. For instance, DBLP and MAG [13] citation networks have some private-type nodes and edges in their networks (i.e., term nodes for DBLP, and field (F) nodes for MAG). Because private node types exist across the networks (See Fig. 1 (a.bottom)), a common semantic space needs to be built before they are aligned to capitalize on the information about such type nodes. For topological structures from two different domains, there is a discrepancy to be aligned for the center node embedding even after all distributions of node types have been aligned (See Fig. 1 (b)). However, previous work [18] only aligns the shared type node features but ignores that each network has its characteristics.

To address these problems, we propose a Generalized Domain Adaptive model across HINs (GDA-HIN). Several independent discriminators are used to align the embeddings in different semantic spaces. Specifically, our model firstly constructs auto-encoders and domain discriminators to align the node features. Secondly, we align the topological structure in graph embedding space using an automatically learned meta-path feature extractor (e.g., HGT [4]) and a domain discriminator. A low-rank matrix completion method is adopted to handle private node types. Our model combines the low-rank matrix completion method with the heterogeneity of HIN. In this way, GDA-HIN can get rid of the precondition of matrix completion method, that corresponding instances [17], or a few labeled target domain data [8]. The contributions of this work:

1) To the best of our knowledge, we are the first to investigate the generalized domain adaptation across HINs that contain both share and private node types. Thus, it is a more general situation on source and target HINs for real-world scenarios.

2) We systematically analyze the challenges of DA across HINs. Based on the above analyses, we propose a novel GDA-HIN by designing pairwise node-type distribution alignment and topological structure alignment to accomplish DA across HINs.

3) We conduct extensive experiments on the three citation networks and six groups of cross-network tasks for node classification to evaluate the performance of domain adaptation. The experiments show that GDA-HIN outperforms state-of-the-art baselines.

2 METHODOLOGY

2.1 Problem Definition

Given a fully-labeled source $HIN_S = \left(\mathcal{V}_S^l, \mathcal{E}_S, \mathcal{A}_S, \mathcal{R}_S\right)$ and a fully-unlabeled target $HIN_T = \left(\mathcal{V}_T^u, \mathcal{E}_T, \mathcal{A}_T, \mathcal{R}_T\right)$, where $\mathcal{V}$ and $\mathcal{E}$ are node and edge set, $\mathcal{A}$ and $\mathcal{R}$ denote the sets of node and edge types. $\mathcal{A}_S$ and $\mathcal{A}_T$ not only share same node types but also have their own private node types, also leading to the difference between $\mathcal{R}_S$ and $\mathcal{R}_T$. $\mathcal{V}_S^l$ represents a set of labeled nodes while $\mathcal{V}_T^u$ represents a set of unlabeled nodes. The transferable classification aims to predict the label on HIN_T with label information from HIN_S.

2.2 Pairwise Node-type Distribution Alignment

To align each semantic component independently between source and target domains, we construct *pairwise node-type* auto-encoders and domain discriminators before aggregation and updating:

1) Shared Node-type Distribution Alignment: Assuming there are K_1 shared node-type pairs between source and target domains, the *pairwise node-type* auto-encoder $AE^{k_1}, k_1 = 1, ..., K_1$ is used

to encode and decode the node features of k_1-th-type nodes. A reconstruction loss is used to constrain AE^{k_1}'s projection retaining semantic information:

$$\mathcal{L}_{\text{recon1}} = \sum MSE\left(\mathbf{X}^{k_1}, \hat{\mathbf{X}}^{k_1}\right), \tag{1}$$

where $\mathbf{X}^{k_1}$ is the node feature of the k_1-th shared node type for both domains, $\hat{\mathbf{X}}^{k_1}$ is the corresponding reconstructed feature, and *MSE* represents the mean square error. Then, we construct a *pairwise node-type* domain discriminator following with a Gradient Reversal Layer (GRL) [3] for the features of individual-type nodes separately. By minimizing the domain adversarial similarity loss, the encoder is trained to make similar its output processed from both domains while the domain discriminator learns to identify the domain of the encoder's output. The domain discriminator loss for all discriminators D^{k1} and auto-encoders AE^{k1}, is formulated as:

$$\mathcal{L}_{nda1} = \sum \mathbb{E}_{x \in \mathbf{X}_S^{k_1}} \left[\log\left(1 - D^{k_1}\left(AE^{k_1}(x)\right)\right)\right]$$
$$+ \mathbb{E}_{x \in \mathbf{X}_T^{k_1}} \left[\log D^{k_1}\left(AE^{k_1}(x)\right)\right], \tag{2}$$

where D^{k_1} is the discriminator for the k_1-th-type nodes.

2) Private Node-type Distribution Alignment: Unlike shared-node types, private-node types contain many unknown values. Reconstruction constraints applied to private-node types recover not only the observed values but also unknown parts. Meanwhile, the private-node types like term and field are semantically relevant [10], which makes the unobserved type's embedding contain plenty of linear dependent columns. Hence, the problem of recovery unobserved type's embedding turns into a low-rank matrix completion problem, which can be formulated to minimize the nuclear norm under the constraint of reconstruction loss [1]. For private node-type pairs, we recover the missing value and get a matrix $\hat{\mathbf{W}}$:

$$\mathbf{W} = \begin{bmatrix} \mathbf{X}_S & \mathbf{0} \\ \mathbf{0} & \mathbf{X}_T \end{bmatrix} \overset{recover}{\Longrightarrow} \hat{\mathbf{W}} = \begin{bmatrix} \hat{\mathbf{X}}_S & \hat{\mathbf{U}}_S \\ \hat{\mathbf{U}}_T & \hat{\mathbf{X}}_T \end{bmatrix}. \tag{3}$$

In $\hat{\mathbf{W}}$ above, $\hat{\mathbf{X}}_S$ and $\hat{\mathbf{X}}_T$ represent the recovered observed elements, while $\hat{\mathbf{U}}_S$ and $\hat{\mathbf{U}}_T$ represent the recovered unobserved elements. To recover $\hat{\mathbf{W}}$, we minimize the loss $\mathcal{L}_{\text{recon2}}$:

$$\mathcal{L}_{\text{recon2}} = \sum MSE\left(\mathbf{X}^{k_2}, \hat{\mathbf{X}}^{k_2}\right) + \delta\mathcal{R}\left(\hat{\mathbf{W}}\right), \tag{4}$$

where $\mathbf{X}^{k_2}$ is the node feature of the k_2-th private node type, $\hat{\mathbf{X}}^{k_2}$ is corresponding observed part of recovered embedding, K_2 is the number of the private-type pairs. $\mathcal{R}(\hat{\mathbf{W}})$ is a regularization term, where $\mathcal{R}(*)$ denotes nuclear norm operation, and $\delta > 0$ is a trade-off parameter. Under the reconstruction constraint, the encoder's output can retain enough semantic information for the private-type pairs. The loss of discriminators for k_1 shared- and k_2 private-type pairs is $\mathcal{L}_{nda} = \mathcal{L}_{nda1} + \mathcal{L}_{nda2}$, where $\mathcal{L}_{nda2}$ is similar as Eq. (2).

2.3 Topological Structure Alignment

In this subsection, we further elaborate on how to align the topological structure of source and target networks.

Representation of h-**hop:** We choose the advanced HIN model HGT [4] as feature extractor G, which can learn embeddings of h-hop structure ($\{\mathcal{N}_{(v)}^1, ..., \mathcal{N}_{(v)}^h\}$), that is h-hop embedding, to capture network topology information. In GDA-HIN, nodes from the

Figure 2: The framework of GDA-HIN. See more details in Methodology (§2).

source and target networks are encoded via a feature extractor with shared learnable parameters. However, there are still h-hop structure discrepancies between both domains. Then, we adopt the domain alignment on the output of HGT, which aligns the data distribution in the embedding space of h-hop structure.

Topological Domain Discriminator: After extracting each nodes' h-hop structure embedding by feature extractor G, which is a 2-layer HGT, another domain adversarial discriminator D^{tp} is implemented to minimize the topological structure discrepancy, and its loss $\mathcal{L}_{da}$ defined as:

$$\mathcal{L}_{da} = \mathbb{E}_{x \in H_S} \left[\log \left(1 - D^{tp} \left(G(x) \right) \right) \right] \qquad (5)$$
$$+ \mathbb{E}_{x \in H_T} \left[\log D^{tp} \left(G(x) \right) \right],$$

where H_S and H_T represent the outputs of encoders in source and target domains, respectively.

Domain-invariant Classifier: The classifier C is used to to predict the label, and its loss $\mathcal{L}_{cls}$ is defined as:

$$\mathcal{L}_{cls} = -\frac{1}{N^l} \sum_{i=0}^{N^l} Y^i \log \left(\hat{Y}^i \right) + \zeta \, tr \left(H^\top L^g H \right), \qquad (6)$$

where ζ is a balance parameter, N^l represents the number of labeled source- and target-domain nodes, and $\hat{Y}^i$ denotes the i-th node's prediction. The regularization term in Eq.(6) is defined as $tr(H^\top L^g H)$, where H represents the hidden state of auto-encoders for all private-type nodes and L^g denotes the graph Laplacian matrix for all private-type nodes. In particular, the graph Laplacian matrix is formulated as $L^g = \begin{pmatrix} L_S & 0 \\ 0 & L_T \end{pmatrix}$, where L_S and L_T are the Laplacian matrices of source and target domains computed according to the adjacency matrices. In the matrix completion module, there are unobserved elements involved in the computation of private-type nodes' embeddings, which are not constrained by

reconstruction loss. We utilize graph Laplacian matrix to smooth their embedding over the graph, relying on the assumption that connected nodes in the graph are likely to share the same label [6].

2.4 Optimization

Phase I Training: To minimize the nuclear norm under the constraint of reconstruction loss Eq.(4), GDA-HIN trains a *Phase I Training model* on the shared node types of two networks to yeild pseudo labels, and the overall objective function is composed of the following four components:

$$\mathcal{L}_{p1} = \mathcal{L}_{cls} + \alpha \mathcal{L}_{recon1} + \beta \mathcal{L}_{nda1} + \gamma \mathcal{L}_{da}, \qquad (7)$$

where α, β and γ are hyper-parameters.

When there are only shared node types, Eq.(6) degenerates to cross-entropy loss, $\mathcal{L}_{cls} = -\frac{1}{N_S} \Sigma_{i=0}^{N_S} Y_S^i \log \left(\hat{Y}_S^i \right)$, where N_S denotes the number of source-domain nodes.

Phase II Training: For phase II training, we select some predictions by *Phase I Training model* as pseudo labels for nodes in target domain. Under the previous phase's guidance, GDA-HIN considers both shared- and private- node types from two networks, and the overall optimization objective is:

$$\mathcal{L}_{p2} = \mathcal{L}_{cls} + \alpha \left(\mathcal{L}_{recon1} + \mathcal{L}_{recon2} \right) + \beta \mathcal{L}_{nda} + \gamma \mathcal{L}_{da}. \qquad (8)$$

3 EXPERIMENTS

Table 1: Dataset statistics.

Dataset	Shared			Private	Shared		Private
	P	A	V	T/K/F	P-A	P-V	P-T/K/F
DBLP	14,328	4,057	20	2,517	19,645	14,328	8,647
Aminer	7,212	4,696	16	7,323	13,796	7,212	22,568
MAG	6,206	5,861	20	2,370	12,615	6,206	8,701

Table 2: Node classification accuracy (%) comparisons on six cross tasks. (D: DBLP; A: Aminer; M: MAG)

Methods	D→A	D→M	A→D	A→M	M→D	M→A
GCN+GRL	51.30	35.56	42.17	33.39	42.15	51.21
UDA-GCN	55.86	36.14	47.10	34.55	46.86	51.24
HAN+GRL	56.73	34.98	45.77	33.66	45.52	51.34
HGT (w/o DA)	34.11	28.83	30.74	29.14	27.24	31.35
GDA-HIN$^{w/o\ P}$	53.13	36.72	41.41	36.33	47.83	51.36
GDA-HIN$^{w/o\ T}$	54.10	32.42	44.73	35.00	48.02	46.29
GDA-HIN$^{w/\ S}$	58.18	37.71	47.77	36.80	48.66	52.87
GDA-HIN(ours)	**58.84**	**39.45**	**58.59**	**36.91**	**50.39**	**54.30**

3.1 Experiment Settings

Datasets: DBLP [5]. We extract a subset of DBLP which contains 14,328 papers (P), 4,057 authors (A), 20 venues (V), 2,517 terms (T), and the edges between nodes. **Aminer** [14]. We extract a subset of Aminer by selecting papers published from the year 2004 to 2008, which contains 7,212 papers, 4,696 authors, 16 venues, 7,323 keywords (K), and the edges between nodes. **MAG** [13]. Here we extract a subset of MAG with publish date between the year 2017 and 2019, which contains 6,206 papers, 5,861 authors, 20 venues, 2,370 fields (F), and the edges between nodes. We categorize authors according to their research areas: *Database*, *Data Mining*, *Artificial Intelligence*, and *Information Retrieval*. Summary statistics of the datasets are displayed in Table 1.

Baselines. We compare with the following baselines: **GCN+GRL**. The model adopts homogeneous graph-based methods GCN [6] as feature extractors and take GRL [3] as domain adaptation framework. **UDA-GCN**. The method [16] adopts a multi-channel GCN with a weight sharing strategy and takes GRL as the domain adaptation framework. It effectively maintains local consistency and global consistency of the graphs. **HAN+GRL**. This model adopts HIN-based method HAN [15] as feature extractor, and takes GRL as the domain adaptation framework.

3.2 Performance Comparison

Our model carries out six groups of cross-network tasks, and the node classification results are reported in Table 2. We have the following primary observations: (1) Compared with all baselines, the proposed GDA-HIN generally achieves the best performance. The results demonstrate the effectiveness of GDA-HIN. (2) All HIN-based methods exhibit better performance than homogeneous methods (i.e., GDA-HIN vs. GCN+GRL). This result reveals that considering heterogeneity helps the DA. In essence, it validates that for the domain adaptation problem of HINs, it is essential to handle the heterogeneity-caused multiple semantic spaces. (3) The GDA-HIN outperforms other heterogeneous graph-based methods, which demonstrates that our method can better address the discrepancies of both pairwise node-type distributions and topological structure.

3.3 Ablation Study

To gain insights about GDA-HIN, we study its variants: GDA-HIN$^{w/o\ P}$, GDA-HIN$^{w/o\ T}$ and GDA-HIN$^{w/\ S}$. Specifically, GDA-HIN$^{w/o\ P}$ & GDA-HIN$^{w/o\ T}$ are two variants of GDA-HIN with the pairwise node-type distribution alignment and the topological

(a) HGT (w/o DA) (b) GDA-HIN

Figure 3: The visualization: the data points of source and target domains are colored by red and blue color, respectively.

structure alignment *removed*, respectively. GDA-HIN$^{w/\ S}$ is only uses shared node types for *phase I training*, and without using private node types, to verify the effectiveness of private node types.

From Table 2, we observe that GDA-HIN is better than GDA-HIN$^{w/o\ P}$, demonstrating the effectiveness of the pairwise node-type distribution alignment for domain shift. Similarly, it is observed that GDA-HIN outperforms GDA-HIN$^{w/o\ T}$, verifying the effectiveness of the topological structure alignment. Moreover, GDA-HIN outperforms GDA-HIN$^{w/\ S}$ on all cross-domain tasks, indicating the effectiveness of considering private types in domain alignment. Note that GDA-HIN$^{w/\ S}$ outperforms other baselines, it indicates our model only using share- node types gains competitive results. In summary, GDA-HIN outperforms three variants, indicating that pairwise node-type alignment, which includes shared and private alignments, and topological structure alignment are indispensable.

3.4 Visualization

Due to the space limit, we report the t-SNE visualization of models' embeddings on Aminer→DBLP as an illustration. The results are shown in Fig. 3, where the source and target domains are colored according to their own domains. From Fig. 3 (a) ~ (b), we can find that the result of (a) HGT contains a tremendous distribution discrepancy due to w/o DA. Apparently, the visualization of our (b) GDA-HIN performs best, where data points of source and target domains are evenly mixed and hardly separated, which means that the domain shift has been successfully minimized, and the node embeddings learned by GDA-HIN are actually domain-invariant. The domain-invariant representations are beneficial to transferring from source domain to target domain.

4 CONCLUSION

In GDA-HIN, we achieve a better domain adaptation performance by jointly considering the pairwise node-type distribution alignment and topology structure alignment. We adopt the matrix completion method to effectively deal with private-type nodes by projecting each private-type pair into a new common feature space. The proposed scheme has been proven effective by experiments.

REFERENCES

[1] Emmanuel J Candès and Benjamin Recht. 2009. Exact matrix completion via convex optimization. *Foundations of Computational mathematics* 9, 6 (2009), 717–772.

[2] Lisheng Fu, Thien Huu Nguyen, Bonan Min, and Ralph Grishman. 2017. Domain adaptation for relation extraction with domain adversarial neural network. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. 425–429.

[3] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The journal of machine learning research* 17, 1 (2016), 2096–2030.

[4] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *Proceedings of The Web Conference 2020*. 2704–2710.

[5] Ming Ji, Yizhou Sun, Marina Danilevsky, Jiawei Han, and Jing Gao. 2010. Graph regularized transductive classification on heterogeneous information networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 570–586.

[6] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. *The International Conference on Learning Representations* (2017).

[7] Wouter M Kouw and Marco Loog. 2018. An introduction to domain adaptation and transfer learning. *arXiv preprint arXiv:1812.11806* (2018).

[8] Haoliang Li, Sinno Jialin Pan, Renjie Wan, and Alex C Kot. 2019. Heterogeneous transfer learning via deep matrix completion with adversarial kernel embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 8602–8609.

[9] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. 2015. Learning transferable features with deep adaptation networks. In *International conference on machine learning*. PMLR, 97–105.

[10] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[11] Xiao Shen, Quanyu Dai, Fu-lai Chung, Wei Lu, and Kup-Sze Choi. 2020. Adversarial deep network embedding for cross-network node classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 2991–2999.

[12] Xiao Shen, Quanyu Dai, Sitong Mao, Fu-lai Chung, and Kup-Sze Choi. 2020. Network together: Node classification via cross-network deep network embedding. *IEEE Transactions on Neural Networks and Learning Systems* 32, 5 (2020), 1935–1948.

[13] Arnab Sinha, Zhihong Shen, Yang Song, Hao Ma, Darrin Eide, Bo-June Hsu, and Kuansan Wang. 2015. An overview of microsoft academic service (mas) and applications. In *Proceedings of the 24th international conference on world wide web*. 243–246.

[14] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. 2008. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 990–998.

[15] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *The World Wide Web Conference*. 2022–2032.

[16] Man Wu, Shirui Pan, Chuan Zhou, Xiaojun Chang, and Xingquan Zhu. 2020. Unsupervised domain adaptive graph convolutional networks. In *Proceedings of The Web Conference 2020*. 1457–1467.

[17] Min Xiao and Yuhong Guo. 2013. A novel two-step method for cross language representation learning. *Advances in Neural Information Processing Systems* 26 (2013), 1259–1267.

[18] Shuwen Yang, Guojie Song, Yilun Jin, and Lun Du. 2020. Domain adaptive classification on heterogeneous information networks. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*. 1410.

[19] Yizhou Zhang, Guojie Song, Lun Du, Shuwen Yang, and Yilun Jin. 2019. Dane: Domain adaptive network embedding. In *Proceedings of the Twenty-Eight International Joint Conference on Artificial Intelligence*.

[20] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2020. A comprehensive survey on transfer learning. *Proc. IEEE* 109, 1 (2020), 43–76.