OpenGDA: Graph Domain Adaptation Benchmark for Cross-network Learning

Boshen Shi
Data Intelligence System Research Center, Institute of Computing Technology, CAS
University of Chinese Academy of Sciences, Beijing, China
shiboshen19s@ict.ac.cn

Yongqing Wang*
Data Intelligence System Research Center, Institute of Computing Technology, CAS, Beijing, China
wangyongqing@ict.ac.cn

Fangda Guo*
Data Intelligence System Research Center, Institute of Computing Technology, CAS, Beijing, China
guofangda@ict.ac.cn

Jiangli Shao
Data Intelligence System Research Center, Institute of Computing Technology, CAS
University of Chinese Academy of Sciences, Beijing, China
shaojiangli19z@ict.ac.cn

Huawei Shen
Data Intelligence System Research Center, Institute of Computing Technology, CAS
University of Chinese Academy of Sciences, Beijing, China
shenhuawei@ict.ac.cn

Xueqi Cheng
Data Intelligence System Research Center, Institute of Computing Technology, CAS
University of Chinese Academy of Sciences, Beijing, China
cxq@ict.ac.cn

ABSTRACT

Graph domain adaptation models are widely adopted in cross-network learning tasks to transfer labeling or structural knowledge. Currently, there mainly exist two limitations in evaluating graph domain adaptation models. On one side, they are primarily tested for the specific cross-network node classification task, leaving tasks at edge-level and graph-level largely under-explored. Moreover, they are primarily examined in limited scenarios, such as social networks or citation networks, needing more validation in richer scenarios. As comprehensively assessing models could enhance model practicality in real-world applications, we propose a benchmark known as *OpenGDA*. It provides abundant pre-processed and unified datasets for different types of tasks (node, edge, graph). They originate from diverse scenarios, covering web information systems, urban systems and natural systems. Furthermore, it integrates state-of-the-art models with standardized and end-to-end pipelines. Overall, *OpenGDA* provides a user-friendly, scalable and reproducible benchmark for evaluating graph domain adaptation models. The benchmark experiments highlight the challenges of applying GDA models to real-world applications with consistent good performance, and they potentially provide insights to future research. As an emerging project, *OpenGDA* will be regularly updated with new datasets and models. It could be accessed from https://github.com/Skyorca/OpenGDA.

CCS CONCEPTS

• **Computing methodologies → Neural networks**.

*Corresponding authors.

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10.
https://doi.org/10.1145/3583780.3615127

KEYWORDS

Graph domain adaptation, Cross-network learning, Graph neural network, Transfer learning

ACM Reference Format:
Boshen Shi, Yongqing Wang, Fangda Guo, Jiangli Shao, Huawei Shen, and Xueqi Cheng. 2023. OpenGDA: Graph Domain Adaptation Benchmark for Cross-network Learning. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3583780.3615127

1 INTRODUCTION

Real-world graph data often faces the problem of limited labels and sparse structures, which will degrade the performance of graph models [3, 14]. To mitigate such problems and improve task performance, researchers establish cross-network learning tasks for leveraging relevant source graphs to transfer abundant labeling or structural knowledge to target graphs [6, 22]. As source and target graphs may originate from correlated yet distinct domains, such as road networks from multiple regions, there exist both node feature distribution shift and graph structure shift between them [24]. Therefore, graph domain adaptation(GDA) has been proposed to overcome distribution shifts and effectively transfer knowledge [4, 10, 21, 25, 29–31]. Inspired by conventional domain adaptation methods [1, 8, 23], GDA adapts such technique to graphs by taking the unique properties of graph-structured data into account. As an emerging area of research, GDA has the potential to boost graph learning tasks in various real-world applications [13, 15, 16, 19, 26, 28].

Experimentally assessing GDA models is essential to understand their competencies. However, previous studies mainly have two limitations in evaluation, that is, the limited type of task and the insufficient quantity of scenarios. On one side, most GDA models are only tested for the specific cross-network node classification task, leaving tasks at edge-level and graph-level largely under-explored. Furthermore, they are primarily tested in limited scenarios, such as social networks or citation networks, needing more validation of model's ability in diverse scenarios. These limitations largely stem from the challenges associated with collecting suitable datasets for different types of tasks, which meet the requisite criteria for GDA. These requirements dictate that each dataset should 1) comprises a group of relevant graphs originating from similar yet different domains and 2) ensures the feature spaces and label spaces of graphs are consistent with each other [18, 32]. As GDA is crucial to tackling cross-network learning tasks, it is necessary to comprehensively verify model capability by testing them in diverse scenarios for different types of tasks.

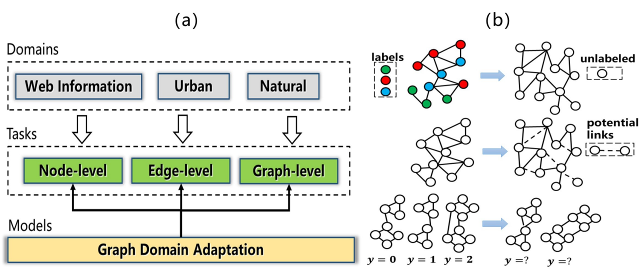

Figure 1: (a): OpenGDA provides diverse datasets, various types of tasks and standardized models. (b): Examples for cross-network learning tasks on node-level, edge-level and graph-level.

To address such limitations, we propose a systematic graph domain adaptation benchmark in this work, known as *OpenGDA*. As design principles, we strive to 1) provide abundant datasets from diverse scenarios with different cross-network learning tasks; and 2) integrate SOTA GDA models to encourage a fair and comprehensive comparison. Specifically, *OpenGDA* currently provides four node-level datasets, three edge-level datasets and two graph-level datasets covering diverse scenarios, including web information systems, urban systems and natural systems. These datasets are collected from open-source research projects. Overall, *OpenGDA* provides 70+ cross-network learning tasks. The details of datasets are summarized in Table 1. Apart from tasks and datasets, *OpenGDA* also pre-implements six SOTA GDA models [4, 24, 25, 29–31] and some baseline models based on graph neural network (GNN). It is worth mentioning that we adopt PyTorch and PyTorch Geometric(PyG) [7] to standardize the overall model pipeline. Consequently, *OpenGDA* is highly unified and customizable, supporting customizing new datasets or models. The experimental results based on *OpenGDA* highlight the challenges of applying GDA models to real-world applications with steady good performance and may provide insights to guide future research. *OpenGDA* will be regularly updated with new datasets and models. Generally, the contributions of this work are summarized as follows.

- To the best of our knowledge, *OpenGDA* is the first benchmark for evaluating GDA models comprehensively.
- *OpenGDA* provides abundant datasets, aiming to assess the capability of GDA models to handle different cross-network learning tasks in diverse scenarios. These datasets are collected, pre-processed and standardized for the convenience of researchers.
- *OpenGDA* provides standard pipelines for integrating new GDA models for validation. Overall, the benchmark experiments based on pre-integrated datasets and models provide insights for future research developments.

2 RELATED WORK

Graph domain adaptation utilizes both source and target graphs for training and tests model mainly on target graphs. Generally, both methods adopt deep graph models, such as GNN, to integrate node feature distribution shift and graph structure shift together as node embedding distribution shit. It could be classified into discrepancy-based methods [4, 10, 21, 24, 25, 29] and disentangle-based methods [2, 30]. Discrepancy-based methods compute such distribution shit via discrepancy measurement and gradually reduce it with supervision loss. For disentangle-based methods, they generally disentangle node embeddings into domain-invariant and domain-relevant parts. Consequently, the knowledge from source graphs could be transferred when both domain-invariant embeddings and source classification patterns are well learned. Although researchers have developed various SOTA GDA models, they have not been thoroughly compared with each other.

Currently, many benchmarks have been developed for graph learning tasks. Three main categories of them include 1) **Benchmarks for general graph machine learning**, such as Open Graph Benchmark [11] and Benchmarking-GNNs [5], 2) **Benchmarks for out-of-distribution on graphs**, such as GOOD [9], and 3) **Benchmarks for self-supervised learning on graphs**, such as DIG [27]. However, it is not trivial to directly establish cross-network learning tasks for GDA models from these benchmarks. Firstly, they mainly split train and test data from the same domain. Moreover, it is hard to guarantee that the provided datasets from similar domains share consistent feature-space and label-space. Therefore, it is necessary to establish a benchmark for evaluating GDA models for cross-network learning tasks.

3 BENCHMARK DESIGN

Generally, designing *OpenGDA* involves two key stages. The first is preparing abundant datasets from diverse scenarios, which satisfy the settings for GDA. Initially, we collect popular datasets in previous studies by task type. As most raw datasets have distinct properties and inconsistent formats, we pre-process and unify these datasets, providing user-friendly graph objects. They are compatible with PyTorch and PyG. In addition, we identify potential datasets for evaluating GDA models from other research areas, such as graph out-of-distribution. As they may not entirely meet GDA settings, we carry out additional pre-processing concerning node features and labels. The format of these datasets is also unified with graph objects.

After the preparation of datasets, we standardize the overall pipeline for each model, including data interface, model architecture and training/evaluation. Figure 2 illustrates the standardized pipeline. **Data interface**: As discussed above, datasets are provided with unified graph objects, each consisting of a node feature matrix,

Table 1: Details of tasks and currently-available datasets provided by OpenGDA. *#Domains* indicates the number of domains covered in the dataset, and *#Tasks* refers to the number of corresponding cross-network learning tasks built between domains. Dataset statistics, such as *#Feat*, illustrate the statistics for each domain, and we use '-' to connect minimum and maximum values when statistic varies across domains.

Task Level	Task Type	Dataset	#Domains	#Tasks	#Feat	#Label	#Nodes	#Edges	#Graphs
Node-level	node classification	Citation1 [21]	3	6	6775	5	5484 - 9360	8130-15602	1
		Twitch [20]	6	30	3170	1	1912-9498	31299 - 112667	1
		Blog [21]	2	2	8189	6	2300-2896	33471-53836	1
		Airport [24]	3	6	8	4	131-1190	1038-13599	1
Edge-level	link prediction	Amazon Review [24]	4	8	5000	1	8568 - 95248	51190-353942	1
		Citation2 [25]	2	2	7537	1	5578-7410	7341-11135	1
	link classification	PPI [29]	5	20	256	2	4286-8369	55668-207461	1
Graph-level	graph classification	IMDB-REDDIT [17]	2	2	136	1	19773-859254	96531-995508	1000-2000
		LetterHigh-LetterLow [17]	2	2	2	15	10507-10522	14092-20250	2250

adjacency matrix, label matrix and other related properties. They are loaded via a data-loader, which is shared across datasets and tasks. **Model architecture**: We establish models upon standard PyTorch modules and PyG GNN layers. In addition, they follow the same forward-propagation process, in which they take both source and target domains as input and then compute two primary losses (i.e., supervision loss and domain discrepancy loss). **Training/Evaluation**: We employ standard PyTorch backward propagation for training and define a suite of universal evaluation functions for diverse tasks. Consequently, *OpenGDA* is user-friendly and supports either conducting experiments on pre-implemented elements or integrating new datasets and models.

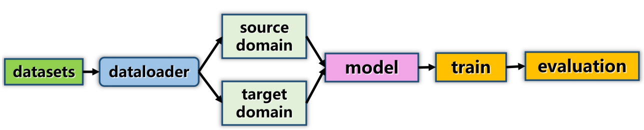

Figure 2: The standardized pipeline of OpenGDA.

The *OpenGDA* package is designed to make the pipeline of Figure 2 easily accessible to researchers, and the package framework is demonstrated in Figure 3. Firstly, the pre-processed datasets are organized in different paths depending on their task categories, yet they share a common data-loader. Besides, as different types of tasks may require different forward-propagation or training processes, each model M is implemented via three variants accordingly (i.e., M_n, M_l and M_g). For example, a unique property of edge-level task is that some datasets contain bipartite graphs, such as Amazon Review dataset. Therefore, the model should separately consider two categories of nodes for domain adaptation, leading to a different forward-propagation process compared to node-level and graph-level tasks. In general, the package is implemented with comprehensive file structures and code structures. It has a user-friendly workflow and could scale well with the addition of new models and datasets.

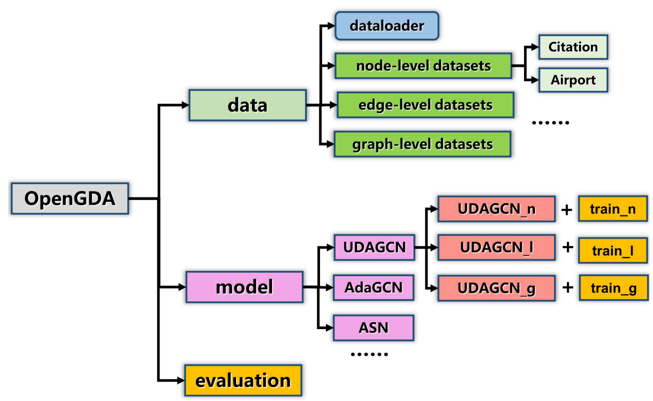

Figure 3: The framework of OpenGDA package. The *data* module includes a data-loader and three folders to store pre-processed datasets based on task type. The *model* module includes GDA models, each having three variants and the corresponding training manuscript. The *evaluation* module includes a suite of metric functions.

4 EXPERIMENTAL STUDIES

We conduct experiments to comprehensively evaluate GDA models based on *OpenGDA*. Our aim is to validate GDA models across various tasks and datasets and gain insights for future research in this field.

Without loss of generality, we select four widely adopted GDA models from two categories with naive GCN [12] baseline for evaluation in this work. Subsequently, we select tasks from node-level, edge-level and graph-level, each containing two datasets covering diverse scenarios. For node-level and edge-level tasks, one domain (source or target) comprises one graph, while it contains a group of graphs in graph-level tasks. We build cross-network learning tasks with one source domain and one target domain. In particular, we employ global mean pooling to compute graph embedding in graph-level tasks. During training, labeling information in target domain is not available in node-level and graph-level tasks. For evaluation,

Table 2: Node classification accuracy on Airport and Citation1 dataset. U, B and E are short for USA, Brazil and Europe in Airport dataset, while A, D and C are short for Acmv9, Dblpv7 and Citationv1 in Citation1 dataset.

	Airport						Citation1					
	U→B	U→E	B→U	B→E	E→U	E→B	A→D	D→A	A→C	C→A	C→D	D→C
GCN	0.427	0.436	0.454	0.481	0.458	0.465	0.623	0.578	0.675	0.635	0.666	0.654
UDAGCN	**0.607**	**0.488**	0.497	**0.510**	0.434	0.477	0.684	0.623	0.728	0.663	0.712	0.645
AdaGCN	0.466	0.434	**0.501**	0.486	0.456	0.561	0.687	0.663	0.701	0.643	0.709	0.702
ASN	0.519	0.469	0.498	0.494	**0.466**	**0.595**	**0.709**	**0.703**	0.732	0.658	**0.732**	**0.734**
GRADE	0.550	0.457	0.497	0.506	0.463	0.588	0.701	0.660	**0.736**	**0.687**	0.722	0.687

Table 3: Link prediction results on Amazon Review-nonoverlapping dataset, where users are disjoint between domains.

	cd→music			music→cd			book→movie			movie→book		
	Hits@10	MRR@10	NDCG@10	Hits@10	MRR@10	NDCG@10	Hits@10	MRR@10	NDCG@10	Hits@10	MRR@10	NDCG@10
GCN	0.158	0.052	0.076	0.273	0.119	0.153	0.339	0.150	0.192	0.094	0.044	0.056
UDAGCN	0.376	**0.155**	0.206	0.255	0.102	0.136	0.369	0.149	0.197	0.194	0.103	0.124
AdaGCN	0.274	0.137	0.170	0.280	0.125	0.159	0.337	0.131	0.177	0.184	0.097	0.118
ASN	**0.380**	**0.155**	**0.207**	**0.285**	**0.131**	**0.165**	0.348	0.157	0.199	**0.539**	**0.136**	**0.227**
GRADE	0.207	0.081	0.110	0.251	0.108	0.139	**0.406**	**0.204**	**0.247**	0.078	0.026	0.037

Table 4: Link Prediction results on Citation2 dataset

	acm→dblp			dblp→acm		
	Hits@10	MRR@10	NDCG@10	Hits@10	MRR@10	NDCG@10
GCN	0.818	0.661	0.699	0.184	0.073	0.098
UDAGCN	**0.853**	**0.675**	**0.716**	**0.258**	**0.108**	**0.141**
AdaGCN	0.533	0.401	0.432	0.145	0.045	0.068
ASN	0.849	0.672	0.712	0.245	0.103	0.134
GRADE	0.823	0.665	0.703	0.185	0.070	0.096

Table 5: Graph classification accuracy on IMDB-REDDIT and Letter dataset. I, R, L and H are short for domain names.

	GCN	UDAGCN	AdaGCN	ASN	GRADE
I→R	0.537	0.564	0.568	**0.589**	0.565
R→I	0.519	0.526	0.503	**0.552**	0.524
L→H	**0.121**	0.106	0.107	0.097	0.089
H→L	0.092	0.093	**0.136**	0.124	0.132

we adopt hit ratio (Hits@k), mean reciprocal rank (MRR@k), and normalized discounted cumulative gain (NDCG@k) where k equals 10 in edge-level tasks and classification accuracy for other tasks. All models follow the same hyper-parameter settings which could be referred from Github repo. The corresponding experimental results are shown in Table 2-5.

Overall, we observe two forms of inconsistency from numerical results. 1) **Scenario-inconsistency**, which indicates the model may fail to consistently perform well across datasets from various scenarios in specific tasks. For example, in edge-level tasks, UDAGCN tends to perform worse than ASN on Amazon Review dataset, but the situation is reversed on Citation2 dataset. 2) **Task-inconsistency**, which means the model may fail to consistently perform well across different types of tasks. For example, AdaGCN performs averagely in egde-level tasks, but its performance improves significantly in node-level and graph-level tasks. These inconsistencies make it difficult to predict the performance of GDA models in real-world applications. Moreover, GDA models only slightly outperform GCN baseline in some tasks and scenarios. Such observations underscore the necessity for further enhancement of the practical applicability of GDA models. In addition to establishing benchmarks for comprehensive evaluation, such as *OpenGDA*, more theoretical studies should be conducted for thoroughly understanding and utilizing intrinsic graph structural properties and domain adaptation mechanisms.

5 CONCLUSION

To enable a comprehensive evaluation of graph domain adaptation models for cross-network learning tasks, we develop a graph domain adaptation benchmark, *OpenGDA*. Currently, it provides standard datasets from various scenarios and tasks covering different types, together with standardized model pipelines. Generally, it offers user-friendly access to researchers for conducting experiments with pre-implemented elements and customizing additional datasets or models. The experimental studies from real-world scenarios highlight the difficulties in achieving consistent good performance for existing GDA models. Altogether, *OpenGDA* presents fruitful opportunities for future research, including digging into GDA theory and continuing to improve practicality. It is worth mentioning that *OpenGDA* is a growing project. We plan to integrate more datasets, tasks and methods as GDA is an emerging line of research. We also expect to keep improving package scalability and enhancing user experience.

ACKNOWLEDGMENTS

This work was funded by the National Key R&D Program Young Scientists Project under grant number 2022YFB3102200 and the National Natural Science Foundation of China under grant number U21B2046. This work is also supported by the fellowship of China Postdoctoral Science Foundation 2022M713206.

REFERENCES

[1] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. 2016. Domain separation networks. *Advances in neural information processing systems* 29 (2016).

[2] Ruichu Cai, Fengzhu Wu, Zijian Li, Pengfei Wei, Lingling Yi, and Kun Zhang. 2021. Graph domain adaptation: A generative view. *arXiv preprint arXiv:2106.07482* (2021).

[3] Enyan Dai, Wei Jin, Hui Liu, and Suhang Wang. 2022. Towards robust graph neural networks for noisy graphs with sparse labels. In *WSDM 2022 - Proceedings of the 15th ACM International Conference on Web Search and Data Mining*. 181–191.

[4] Quanyu Dai, Xiao-Ming Wu, Jiaren Xiao, Xiao Shen, and Dan Wang. 2022. Graph Transfer Learning via Adversarial Domain Adaptation with Graph Convolution. *IEEE Transactions on Knowledge and Data Engineering* (2022).

[5] Vijay Prakash Dwivedi, Chaitanya K Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. 2020. Benchmarking Graph Neural Networks. *arXiv preprint arXiv:2003.00982* (2020).

[6] Meng Fang, Jie Yin, and Xingquan Zhu. 2013. Transfer learning across networks for collective classification. In *2013 IEEE 13th International Conference on Data Mining*. IEEE, 161–170.

[7] Matthias Fey and Jan Eric Lenssen. 2019. Fast graph representation learning with PyTorch Geometric. *arXiv preprint arXiv:1903.02428* (2019).

[8] 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.

[9] Shurui Gui, Xiner Li, Limei Wang, and Shuiwang Ji. 2022. Good: A graph out-of-distribution benchmark. *arXiv preprint arXiv:2206.08452* (2022).

[10] Gaoyang Guo, Chaokun Wang, Bencheng Yan, Yunkai Lou, Hao Feng, Junchao Zhu, Jun Chen, Fei He, and Philip Yu. 2022. Learning adaptive node embeddings across graphs. *IEEE Transactions on Knowledge and Data Engineering* (2022).

[11] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. 2020. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems* 33 (2020), 22118–22133.

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

[13] Haoran Li, Hanghang Tong, and Yang Weng. 2022. Domain Adaptation in Physical Systems via Graph Kernel. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 868–876.

[14] Zemin Liu, Trung-Kien Nguyen, and Yuan Fang. 2021. Tail-gnn: Tail-node graph neural networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1109–1119.

[15] Bin Lu, Xiaoying Gan, Weinan Zhang, Huaxiu Yao, Luoyi Fu, and Xinbing Wang. 2022. Spatio-Temporal Graph Few-Shot Learning with Cross-City Knowledge Transfer. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1162–1172.

[16] Jianxin Ma, Peng Cui, Kun Kuang, Xin Wang, and Wenwu Zhu. 2019. Disentangled graph convolutional networks. In *International conference on machine learning*. PMLR, 4212–4221.

[17] Christopher Morris, Nils M Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. 2020. Tudataset: A collection of benchmark datasets for learning with graphs. *arXiv preprint arXiv:2007.08663* (2020).

[18] Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22, 10 (2009), 1345–1359.

[19] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. 2021. Multi-scale attributed node embedding. *Journal of Complex Networks* 9, 2 (2021), cnab014.

[20] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. 2021. Multi-scale attributed node embedding. *Journal of Complex Networks* 9, 2 (2021), 1–22.

[21] 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.

[22] 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.

[23] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. 2014. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474* (2014).

[24] Jun Wu, Jingrui He, and Elizabeth Ainsworth. 2022. Non-IID Transfer Learning on Graphs. *arXiv preprint arXiv:2212.08174* (2022).

[25] 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.

[26] Qitian Wu, Hengrui Zhang, Junchi Yan, and David Wipf. 2022. Handling Distribution Shifts on Graphs: An Invariance Perspective. In *International Conference on Learning Representations*.

[27] Yaochen Xie, Zhao Xu, Jingtun Zhang, Zhengyang Wang, and Shuiwang Ji. 2022. Self-supervised learning of graph neural networks: A unified review. *IEEE transactions on pattern analysis and machine intelligence* 45, 2 (2022), 2412–2429.

[28] Haotian Xue, Kaixiong Zhou, Tianlong Chen, Kai Guo, Xia Hu, Yi Chang, and Xin Wang. 2021. CAP: Co-Adversarial Perturbation on Weights and Features for Improving Generalization of Graph Neural Networks. *arXiv preprint arXiv:2110.14855* (2021).

[29] Yuning You, Tianlong Chen, Zhangyang Wang, and Yang Shen. 2023. Graph Domain Adaptation via Theory-Grounded Spectral Regularization. In *The Eleventh International Conference on Learning Representations*.

[30] Xiaowen Zhang, Yuntao Du, Rongbiao Xie, and Chongjun Wang. 2021. Adversarial Separation Network for Cross-Network Node Classification. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2618–2626.

[31] Yizhou Zhang, Guojie Song, Lun Du, Shuwen Yang, and Yilun Jin. 2019. DANE: Domain Adaptive Network Embedding. In *IJCAI*.

[32] 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.