Coupling Category Alignment for Graph Domain Adaptation

Nan Yin[1] , **Xiao Teng**[2] , **Zhiguang Cao**[3] , **Mengzhu Wang**[4]
[1]Hong Kong University of Science and Technology
[2]Naval University of Engineering
[3]Singapore Management University
[4]Hebei University of Technology
{yinnan8911,dreamkily}@gmail.com, t_x@nue.edu.cn, zhiguangcao@outlook.com,

Abstract

Graph domain adaptation (GDA), which transfers knowledge from a labeled source domain to an unlabeled target graph domain, attracts considerable attention in numerous fields. However, existing methods commonly employ message-passing neural networks (MPNNs) to learn domain-invariant representations by aligning the entire domain distribution, inadvertently neglecting category-level distribution alignment and potentially causing category confusion. To address the problem, we propose an effective framework named **Co**upling **C**ategory **A**lignment (CoCA) for GDA, which effectively addresses the category alignment issue with theoretical guarantees. CoCA incorporates a graph convolutional network branch and a graph kernel network branch, which explore graph topology in implicit and explicit manners. To mitigate category-level domain shifts, we leverage knowledge from both branches, iteratively filtering highly reliable samples from the target domain using one branch and fine-tuning the other accordingly. Furthermore, with these reliable target domain samples, we incorporate the coupled branches into a holistic contrastive learning framework. This framework includes multi-view contrastive learning to ensure consistent representations across the dual branches, as well as cross-domain contrastive learning to achieve category-level domain consistency. Theoretically, we establish a sharper generalization bound, which ensures the effectiveness of category alignment. Extensive experiments on benchmark datasets validate the superiority of the proposed CoCA compared with baselines.

1 Introduction

As a crucial problem in graph classification [Lin *et al.*, 2023; Luo *et al.*, 2023], Graph Domain Adaptation (GDA) has received substantial interest, particularly in the fields of temporally-evolved social analysis [Wang *et al.*, 2021], molecular biology [Zhu *et al.*, 2023; Yin *et al.*, 2023], and protein-protein interaction networks [Cho *et al.*, 2016]. GDA transfers graph representations learned from the source domain to the target domain, which is necessary in many applications. Domain adaptive learning is inherently challenging due to the distribution shift between source and target domains. This challenge is further amplified when handling graph-structured data, which often represent abstractions of varying natures [You *et al.*, 2022a].

Currently, various GDA methods have been proposed [Yin *et al.*, 2022; Yin *et al.*, 2023; You *et al.*, 2022b] by combining domain adaptation techniques with graphs. They usually assume the distribution invariance is limited [Garg *et al.*, 2020; Verma and Zhang, 2019] and directly employ adversarial training to align source and target distribution [Zhang *et al.*, 2019b; Wu *et al.*, 2020a]. However, the classifier still tends to favor source domain features and makes incorrect predictions on the target domain due to category-agnostic feature alignment [Zhang *et al.*, 2019a], as shown in Figure 1. To solve the issue and efficiently design the GDA framework, we still need to address the following challenges: (i) *How to fully exploit the features of the source and target domain for representation learning.* Previous approaches typically employ the MPNNs to capture implicit topological semantics. However, the absence of labels for the target domain poses challenges in obtaining sufficient topological semantics. (ii) *How to effectively align category-level distribution.* While there has been progress in matching the marginal distributions between two domains, they may not efficiently align the category distribution, leading to a degradation in classification performance. Certain methods have attempted to acquire pseudo-labels for the target graphs in target domain training [Yin *et al.*, 2023; Ding *et al.*, 2021; Zhu *et al.*, 2023], they are vulnerable to bias in cases of significant domain shift, leading to error accumulation in subsequent optimization. (iii) *How to design the GDA framework with the grounded theoretical foundation.* Theoretically, the generic domain adaptation (DA) bound is not specific to graph data and models [You *et al.*, 2023]. However, we can still design a more precise model tailored for graphs with the theoretical guarantee.

To tackle these challenges, we propose a framework named **Co**pling **C**ategory **A**lignment (CoCA) for unsupervised domain adaptive graph classification. Specifically, to fully exploit the features of both source and target graphs, CoCA incorporates an MPNN branch and a shortest path aggregation branch. The MPNN branch leverages neighborhood aggrega-

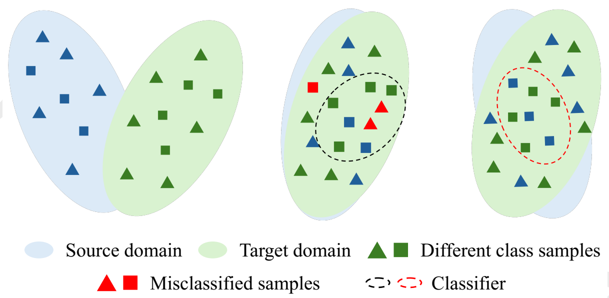


Figure 1: Left: The source and target graph domains. Middle: GDA methods that align the entire source and target domains, potentially confuse category distribution (see the red triangles and squares). Right: The proposed method, which aligns category-level distributions, alleviates the category-agnostic issue.

tion to implicitly learn topological semantics, while the shortest path aggregation branch generates paths for each node and utilizes position encoding to extract informative graph-level semantics. This shortest path aggregation branch provides explicit high-order structural semantics, serving as a complementary enhancement. To collaborate knowledge from two branches, we jointly train the branches by iteratively filtering highly reliable samples from the target domain using one branch and fine-tuning the other branch accordingly. Specifically, with the dual pre-trained branches, CoCA first fine-tunes the shortest path branch with the highly reliable samples filtered from the target domain with the MPNN branch and then optimizes the MPNN branch with the filtered target domain samples labeled by the shortest path branch. Theoretically, the interactive optimization of one branch with the support of the other one would gradually mitigate the category-level distribution shift. Furthermore, we embed the iterative learning process into a holistic contrastive learning framework, incorporating cross-domain contrastive learning to achieve category-level domain consistency, alongside multi-views contrastive learning to ensure consistent representations between branches. Overall, our approach emphasizes a unique focus on achieving category-level domain alignment. Specifically, our methodology is centered on Coupling Category Alignment (CoCA), which systematically iterates between branches to identify and select reliable samples. This process facilitates cross-branch adjustments, effectively mitigating potential domain shifts in an unsupervised manner. By iteratively refining the alignment process, our approach enhances the model's ability to achieve category-level alignment, supported by solid theoretical foundations, distinguishing it from prior methods. Extensive experiments conducted on various datasets with domain shifts for graph classification demonstrate the superiority of proposed CoCA.

In summary, the main contributions can be summarized as three-fold: (1) *Problem Formulation*: We present a novel problem in graph domain adaptation, which highlights the discrepancy in the distribution of graph categories between the source and target domains, posing significant challenges for accurate graph classification across domains. (2) *Methodology*: We propose a framework named CoCA, which utilizes two branches to explore structural semantics and integrates them into a category-level domain-invariant model. We provide theoretical proof demonstrating that CoCA is specifically designed to more accurately address the challenges of the graph domain. (3) *Experiments*: Extensive experiments conducted on various domain shift datasets for graph classification demonstrate the effectiveness of the proposed CoCA.

2 Related Work

Graph Classification. GNNs [Kipf and Welling, 2017a] have shown exceptional performance across a wide range of graph-based machine learning tasks, such as node classification [Kipf and Welling, 2017a], graph classification [Wu *et al.*, 2020b] and link prediction [Cai *et al.*, 2021]. The most prevalent GNNs follow the message-passing paradigm, which aggregates the neighbors for node update and applies graph pooling for graph representation. Nevertheless, MPNNs have limited capacity to capture high-order topological structures, such as paths and motifs [Ju *et al.*, 2022]. Therefore, numerous graph kernel methods have emerged to overcome this flaw [Long *et al.*, 2021]. However, these approaches typically require an ample supply of labeled annotations [Yin *et al.*, 2023] while this work delves into the realm of unsupervised graph domain adaptation and introduces a novel approach CoCA to tackle this challenge.

Unsupervised Domain Adaptation. Unsupervised domain adaptation is to learn domain-invariant representations that enable the transfer of a model from a source domain with abundant labels to a target domain with a scarcity of labels [Feng *et al.*, 2023]. The majority of technical routes can be broadly categorized into domain discrepancy-based methods and adversarial approaches. The former methods typically incorporate different distribution metrics like maximal mean discrepancy [Saito *et al.*, 2018] and Wasserstein distance [Shen *et al.*, 2018] to measure the discrepancy between different domains. Conversely, adversarial approaches involve a domain discriminator that is fused to implicitly reduce the domain discrepancy. However, these methods typically concentrate on Euclidean data such as images and texts, while graph domain adaptation has not been extensively explored. In this work, we explores graph semantics by utilizing dual perturbation branches for effective graph domain adaptation.

Graph Domain Adaption. Due to the potential economic value, graph domain adaptation [Lin *et al.*, 2023; Wu *et al.*, 2022; Luo *et al.*, 2023] is a crucial problem in the fields of social analysis and molecular biology [You *et al.*, 2022b; Zhu *et al.*, 2023]. Existing methods mainly focus on how to transfer information from source graphs to unlabeled target graphs to learn effective node-level [Dai *et al.*, 2022] and graph-level [Yin *et al.*, 2023; Ding *et al.*, 2021] representation. However, these approaches commonly merge GNNs with domain alignment [Luo *et al.*, 2023] methods, which overlook the alignment of category distributions in the presence of label scarcity and domain shift, consequently leading to a deterioration in classification performance. Towards this end, CoCA couples the dual branch in a variational optimization framework to address the issue.

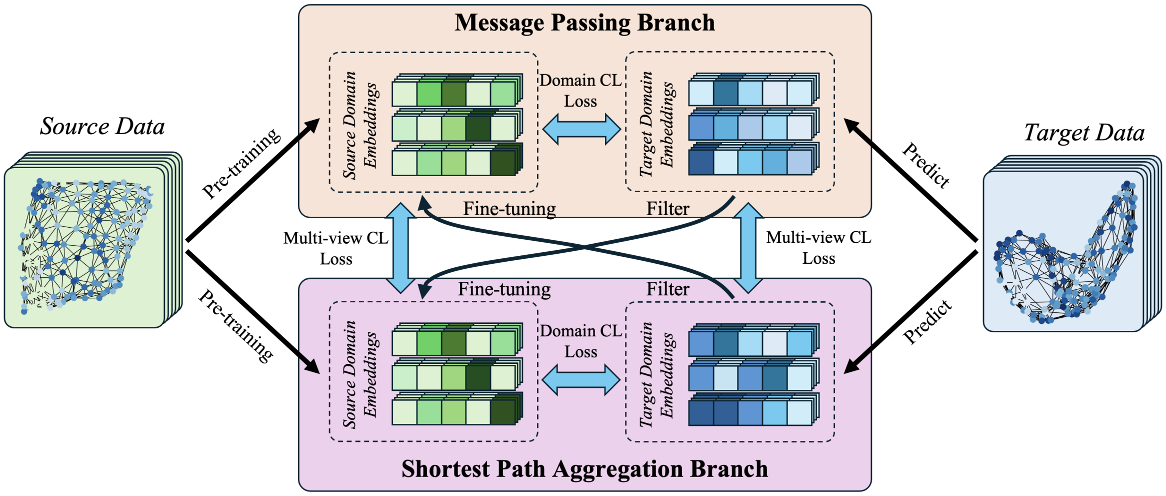


Figure 2: An overview of the proposed CoCA. CoCA contains a message passing branch and a shortest path aggregation branch. To align category-level distribution, we alternatively optimize each branch with highly dependable pseudo-labels learned from the other branch. CoCA incorporates the learning process in a multi-view and cross-domain contrastive learning framework.

3 Methodology

This work studies the unsupervised GDA problem [Yin *et al.*, 2023; Yehudai *et al.*, 2021] and proposes a new approach CoCA (see Figure 2). CoCA consists of two parts, the **dual graph branch** explores semantics from implicit and explicit perspectives; the **branch coupling module** interactively optimizes one branch with highly reliable sample filtered from the other branch to minimize the category distribution discrepancy. CoCA incorporates the iterative process into a learning framework and theoretical proof the designed method is more precisely tailored for the graph domain.

3.1 Dual Branch for Semantics Mining

Current graph transfer learning methods [Lin *et al.*, 2023; Wu *et al.*, 2022] typically rely on MPNNs to implicitly capture topological semantics through neighborhood aggregation for transfer learning. However, these approaches may be suffered under domain shift. To address this issue, we introduce a dual-branch architecture for graph representation learning, comprising a MPNNs branch for implicit topological semantics and a shortest path aggregation branch for explicit topological semantics derived from high-order structures.

Message Passing Branch. MPNNs extract graph semantics by aggregating neighborhood nodes to update each central node representation. We update the representation of node u at layer l in the massage passing branch $f_\theta^{MP}(\cdot)$ and summarize the node representations into graph-level as:

$$\mathbf{h}_u^l = \mathcal{C}_{MP}^l\left(\mathbf{h}_u^{l-1}, \mathcal{A}_{MP}^l\left(\left\{\mathbf{h}_v^{l-1}\right\}_{v\in\mathcal{N}(u)}\right)\right),$$

$$\mathbf{z}_G^{MP} = f_\theta^{MP}(G) = \text{READOUT}\left(\left\{\mathbf{h}_u^L\right\}_{u\in V}\right),$$

where $\mathcal{N}(u)$ is the neighbours of node u. $\mathcal{C}_{MP}^l$ and $\mathcal{A}_{MP}^l$ are combination and aggregation functions at layer l, and READOUT is the pooling function. In this way, the message passing branch learns the topological structure in an implicit manner under label supervision.

Shortest Path Aggregation Branch. However, the message passing branch merely extracts topological structural semantics in an implicit manner, which would be challenged under the circumstance of domain shift. Considering an alternative technical route, graph kernels [Shervashidze *et al.*, 2011] are capable of explicitly extracting high-order semantics. We introduce a shortest path aggregation branch that generates various shortest paths from local substructures to extract high-order semantics into graph-level representations. Thus, the representations alleviate the impact of structural shift across domains.

In particular, denote $\mathcal{N}_k(u)$ as the set of nodes reachable from u through a shortest path of length k, and the representation of node u can be updated with $\mathcal{N}_1(u)\cup\cdots\cup\mathcal{N}_k(u)\cup\cdots\cup\mathcal{N}_K(u)$, K is the hyperparameter of largest length. We update the nodes on different path length k:

$$\mathbf{m}_{u,\mathcal{N}_k(u)}^l = \mathcal{C}_{SP}^l\left(\hat{\mathbf{m}}_u^{l-1}, \mathcal{A}_{SP}^l\left(\left\{\hat{\mathbf{m}}_v^{l-1}\right\}_{v\in\mathcal{N}_k(u)}\right)\right),$$

where $\mathcal{C}_{SP}^l$ and $\mathcal{A}_{SP}^l$ denotes combination and aggregation operators at layer l on shortest path branch. Thus, we obtain the embeddings from different path length, i.e., $\left\{\mathbf{m}_{u,\mathcal{N}_1(u)}^l,\cdots,\mathbf{m}_{u,\mathcal{N}_K(u)}^l\right\}$, and update the representation of u as follows:

$$\alpha^l = Atten\left(||_{k=1}^K \mathbf{m}_{u,\mathcal{N}_k(u)}^l\right),$$

$$\mathbf{m}_u^l = MLP\left((1+\epsilon)\mathbf{m}_u^{l-1} + \sum_{k=1}^K \alpha_k^l \mathbf{m}_{u,\mathcal{N}_k(u)}^l\right),$$

where $\epsilon \in \mathbb{R}$, $Atten$ is the self-attention mechanism, $||$ denotes the concatenation operation, and $||_{k=1}^K \mathbf{m}_{u,\mathcal{N}_k(u)}^l \in \mathbb{R}^{K\times d'}$, d' is the feature dimension of $\mathbf{m}_{u,\mathcal{N}_k(u)}^l$. MLP is the fully connected layer. After stacking L layers, we take the average of all nodes into a graph-level representation:

$$\mathbf{z}_G^{SP} = f_\phi^{SP}(G) = \text{READOUT}\left(\left\{\mathbf{m}_u^L\right\}_{u\in V}\right). \quad (1)$$

The $\mathrm{READOUT}$ function is similar to Eq. 1. The shortest path aggregation branch acquires topological semantics by focusing on paths of varying lengths, which help mitigate the impact of structural domain shifts, such as differences in density and graph size.

3.2 Coupling for Category Alignment

Recent work attempted to obtain target graph pseudo-labels for training [Yehudai *et al.*, 2021; Ding *et al.*, 2021; Zhu *et al.*, 2023]. However, due to discrepancies in the category distribution, they may suffer from error accumulation during subsequent optimization. To address this issue, we cleverly utilize the characteristics of the dual branch to obtain pseudo-labels and mitigate error accumulation.

With the message passing branch (MP branch) and the shortest path branch (SP branch), our target is to identify highly dependable pseudo-labels in the target domain and integrate them with the source domain to fine-tune the model. In this way, we can efficiently align the category distribution. Nevertheless, with category discrepancy and error accumulation challenges, we cannot achieve satisfactory pseudo-labels in a signal branch. To address the issue, we introduce the classifier $H^{MP}(\cdot)$ and $H^{SP}(\cdot)$ for the MP and SP branches. Specifically, we filter the highly dependable pseudo-labels with the threshold ζ in the MP branch:

$$\mathcal{G}^{MP} = \{G_i^t | H^{MP}(\mathbf{z}_{G_i^t}^{MP}) > \zeta,\ \forall G_i^t \in G^t\},$$

and then use those samples to help fine-tune the SP branch:

$$\mathcal{L}_1 = l(H^{SP}(\mathbf{z}_{G^s}^{SP}), y^s) + l_{G_i^t \in \mathcal{G}^{MP}}(H^{SP}(\mathbf{z}_{G_i^t}^{SP}), H^{MP}(\mathbf{z}_{G_i^t}^{MP})),$$

where $H^{MP}(\mathbf{z}_{G_i^t}^{MP})$ is the target graph pseudo-labels filtered from the MP branch, $l(\cdot)$ is the loss function. Similarly, we utilize the target samples filtered from the SP branch to support the fine-tune of the MP branch:

$$\mathcal{G}^{SP} = \{G_i^t | H^{SP}(\mathbf{z}_{G_i^t}^{SP}) > \zeta,\ \forall G_i^t \in G^t\},$$

and then use those samples to help fine-tune the MP branch:

$$\mathcal{L}_2 = l(H^{MP}(\mathbf{z}_{G^s}^{MP}), y^s) + l_{G_i^t \in \mathcal{G}^{SP}}(H^{MP}(\mathbf{z}_{G_i^t}^{MP}), H^{SP}(\mathbf{z}_{G_i^t}^{SP})),$$

where $H^{SP}(\mathbf{z}_{G_i^t}^{SP})$ is the target graph pseudo-labels filtered from the SP branch. The interactive optimization of the MP and SP branches offers two advantages. First, by incorporating highly confident target pseudo-labels into source domain training, we can effectively align the category distribution. Second, the pseudo-labels filtered from the other branch help mitigate the error accumulation issue caused by the single model.

Theoretical Analysis. Intuitively, incorporating training samples from target and source domains would effectively align the category distribution between domains. However, the theoretical basis for why iterative fine-tuning achieves category alignment still requires further investigation. Additionally, the graph category alignment bound remains agnostic. To address this, we present Theorem 1 demonstrating that employing a category distribution alignment module results in a lower bound on the empirical risk in the target domain than without this module.

Theorem 1. *Assume that there exists a small amount of high dependable i.i.d. samples with pseudo labels* $\{(G_n, Y_n)\}_{n=1}^{N_T'}$ *filter from the target distribution* $\mathbb{P}_T(G, Y)$ $(N_T' \ll N_S)$ *and bring in the conditional shift assumption that domains have different labeling function* $\hat{h}_S \neq \hat{h}_T$ *and* $\max_{G_1,G_2} \frac{|\hat{h}_D(G_1) - \hat{h}_D(G_2)|}{\eta(G_1,G_2)} = C_h \leq C_f C_g (D \in \{S, T\})$ *for some constant* C_h *and distance measure* η*. Let* $\mathcal{H} := \{h : \mathcal{G} \to \mathcal{Y}\}$ *be the set of bounded real-valued functions with the pseudo-dimension* $Pdim(\mathcal{H}) = d$*, with probability at least* $1 - \delta$ *the following inequality holds:*

$$\begin{aligned}\epsilon_T(h, \hat{h}_T) \leq & \frac{N_T'}{N_S + N_T'} \hat{\epsilon}_T(h, \hat{h}_T) + \frac{N_S}{N_S + N_T'} \Big(\hat{\epsilon}_S(h, \hat{h}_S) \\ & + \sqrt{\frac{4d}{N_S} \log(\frac{eN_S}{d}) + \frac{1}{N_S} \log(\frac{1}{\delta})} \\ & + 2C_f C_g W_1 \left(\mathbb{P}_S(G), \mathbb{P}_T(G)\right) + \omega' \Big),\end{aligned}$$

where $\omega = \min_{||g||_{Lip} \leq C_g, ||f||_{Lip} \leq C_f} \{\epsilon_S(h, \hat{h}_S) + \epsilon_T(h, \hat{h}_S)\}$ *and* $\omega' = \min(|\epsilon_S(h, \hat{h}_S) - \epsilon_S(h, \hat{h}_T)|, |\epsilon_T(h, \hat{h}_S) - \epsilon_T(h - \hat{h}_T)|)$.

The proof is detailed in Appendix A. From Theorem 1, we observe that the bound of CoCA is lower than GDA by incorporating source and target samples during training, demonstrating that it is possible to design a more accurate model specifically tailored for graphs with a theoretical guarantee.

3.3 Learning Framework

Through the iterative learning process of the MP and SP branches, we integrate them into a unified contrastive learning framework. Specifically, this framework employs cross-domain contrastive learning to achieve category-level domain consistency, while multi-view contrastive learning ensures consistent representation between the branches [Yin *et al.*, 2023].

Multi-view Contrastive Learning. For each a graph G_i, we first obtain the embeddings from MP and SP branches, i.e., $\mathbf{z}_i^{MP}$ and $\mathbf{z}_i^{SP}$. Then, we introduce the InfoNCE loss to enhance the consistency representation cross coupled branches:

$$\mathcal{L}_{mv} = -\frac{1}{|\mathcal{D}^s| + |\mathcal{D}^t|} \sum_{G_i \in \mathcal{D}^s \cup \mathcal{D}^t} \log \frac{exp(\mathbf{z}_i^{MP} \cdot \mathbf{z}_i^{SP} / \tau)}{\sum_{G_j, j \neq i} exp(\mathbf{z}_i^{MP} \cdot \mathbf{z}_j^{SP} / \tau)},$$

where τ is set to 0.5 as default .

Cross-domain Contrastive Learning. To achieve the category-level domain consistency, we construe the cross-domain contrastive learning with the help of pseudo-labels from each branch. Taking the MP branch as an example, we first filter highly dependable samples with the threshold ζ as introduced in 3.2, and then calculate the loss between source and target domain with the same category.

$$\mathcal{L}_{cd} = -\sum_{j \in \Omega(j)} \frac{1}{|\Pi(j)|} \cdot \sum_{i \in \Pi(j)} \log \frac{exp\left(\mathbf{z}_j^{SP,t} \cdot \mathbf{z}_i^{SP,s} / \tau\right)}{\sum_{G_k \in \mathcal{D}^s, k \notin \Pi(i)} exp\left(\mathbf{z}_j^{SP,t} \cdot \mathbf{z}_k^{SP,s} / \tau\right)},$$

where $\Pi(j) = \{i | y_i^s = \hat{y}_j^t\}$ denotes the index of all positives in the source domain, $\Omega(i)$ is the index of filtered highly dependable samples from the target domain.

Methods	M0→M1	M1→M0	M0→M2	M2→M0	M0→M3	M3→M0	M1→M2	M2→M1	M1→M3	M3→M1	M2→M3	M3→M2	Avg.
WL subtree	74.9	74.8	67.3	69.9	57.8	57.9	73.7	80.2	60.0	57.9	70.2	73.1	68.1
GCN	73.0±1.7	68.7±1.5	66.8±3.5	69.2±0.9	53.9±3.4	53.4±2.7	69.3±0.8	74.0±1.1	55.1±1.3	42.6±1.9	55.5±3.5	57.9±2.9	61.6
GIN	74.1±1.8	73.4±3.4	65.4±1.5	70.4±2.9	58.9±2.7	61.2±1.1	73.2±3.8	77.7±3.0	63.1±3.7	63.9±2.4	67.4±2.3	73.2±1.9	68.5
GMT	69.0±4.0	67.4±3.8	60.3±4.2	66.5±3.8	54.9±1.6	54.8±3.6	65.6±4.2	70.4±3.2	64.0±2.3	56.8±4.3	64.7±1.5	61.1±3.5	63.0
CIN	68.5±2.1	65.1±2.6	65.4±1.3	63.6±2.8	57.3±3.4	59.0±3.1	59.3±1.5	68.3±1.3	58.1±2.4	71.1±3.1	60.7±1.7	61.7±2.4	63.2
CDAN	74.2±0.3	73.7±0.5	68.8±0.2	71.8±0.4	59.9±2.0	58.6±1.9	70.7±1.4	74.3±0.3	59.2±1.2	69.0±1.6	60.0±1.2	62.7±1.3	66.9
ToAlign	75.5±1.9	67.1±3.8	68.1±1.5	63.3±2.7	55.6±1.2	67.3±4.3	69.4±3.3	77.0±1.2	57.6±1.6	74.9±2.4	59.0±3.3	64.6±3.4	66.6
MetaAlign	74.5±0.9	73.8±0.6	69.4±1.2	72.6±1.3	59.8±1.8	70.7±2.7	72.0±0.5	75.6±0.6	62.4±2.1	72.3±1.9	62.2±1.1	72.0±1.2	69.7
DEAL	76.3±0.2	72.4±0.7	68.8±1.0	72.5±0.7	57.6±0.6	67.6±1.9	77.4±0.6	80.0±0.7	64.9±0.7	72.8±1.4	70.3±0.3	76.2±1.3	71.4
CoCo	77.5±0.4	75.7±1.3	68.3±3.7	74.9±0.5	65.1±2.1	74.0±0.4	76.9±0.6	77.4±3.4	66.4±1.5	71.2±2.7	62.8±4.2	77.1±0.6	72.2
SGDA	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
DGDA	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
A2GNN	55.3±0.3	54.9±0.6	55.8±0.4	55.1±0.8	54.2±1.0	57.1±1.2	56.1±0.5	55.2±0.7	57.9±1.5	56.3±0.6	54.4±0.5	58.1±1.5	55.8
PA-BOTH	56.3±0.5	57.7±0.9	56.9±0.6	56.2±1.0	55.7±0.8	56.5±0.9	57.8±1.2	56.9±2.1	56.5±1.5	56.2±1.8	56.8±1.4	57.4±0.7	56.8
CoCA	**82.4±1.5**	**80.8±1.2**	**74.5±1.7**	**79.6±2.1**	**74.8±2.2**	**79.2±0.7**	**83.4±0.9**	**85.7±0.6**	**73.9±0.8**	**81.3±1.5**	**77.8±0.7**	**83.3±1.4**	**79.7**

Table 1: The classification results (in %) on Mutagenicity under edge density domain shift (source→target). M0, M1, M2, and M3 denote the sub-datasets partitioned with edge density. **Bold** results indicate the best performance.

Methods	F0→F1	F1→F0	F0→F2	F2→F0	F0→F3	F3→F0	F1→F2	F2→F1	F1→F3	F3→F1	F2→F3	F3→F2	Avg.
WL subtree	65.7	71.8	57.9	71.1	47.4	43.4	65.5	75.1	45.3	34.9	52.7	49.8	56.7
GCN	70.6±2.1	60.3±1.5	60.5±3.4	62.3±1.1	58.4±0.5	43.2±0.2	63.8±1.0	70.3±0.3	50.6±1.0	32.8±0.3	50.1±0.4	42.2±0.2	55.4
GIN	66.7±2.1	73.7±2.4	57.3±3.1	69.4±2.3	58.6±0.4	43.1±0.3	66.4±2.7	74.8±1.8	42.2±1.6	33.5±1.0	57.4±0.8	43.9±2.3	57.2
GMT	67.3±0.3	56.8±0.4	58.0±0.2	56.8±0.2	60.6±0.3	56.8±0.5	57.8±0.1	67.3±0.1	39.5±0.3	67.3±0.2	39.5±0.5	57.8±0.4	57.1
CIN	67.6±0.4	63.7±2.1	58.9±1.0	56.8±0.4	63.6±0.4	59.5±2.7	58.7±1.2	67.0±0.5	61.7±1.6	67.8±0.7	62.2±2.1	56.0±1.3	61.9
CDAN	72.9±0.4	72.7±0.4	65.4±0.3	72.9±0.1	61.2±0.3	70.3±0.2	65.7±0.4	72.7±0.1	61.0±0.1	72.1±1.2	60.7±0.2	65.3±0.6	67.7
ToAlign	32.7±2.0	43.2±0.1	42.2±1.3	43.2±0.9	60.5±0.7	43.2±1.2	42.2±0.4	32.7±1.2	60.5±0.9	32.7±0.3	60.5±0.7	42.2±0.4	44.7
MetaAlign	67.3±0.7	56.8±0.2	57.8±0.6	56.8±0.4	60.5±1.3	56.8±0.8	57.8±1.1	67.3±1.2	60.5±0.4	67.3±0.6	60.5±0.7	57.8±0.6	60.6
DEAL	75.0±0.9	76.3±2.4	65.9±1.8	77.5±2.7	60.3±4.5	69.7±3.2	67.2±1.5	75.3±1.7	57.4±4.1	71.1±2.2	65.7±2.7	66.4±1.6	69.0
CoCo	74.2±1.7	74.3±0.6	65.9±1.2	72.7±2.1	61.1±0.2	71.0±1.7	68.6±0.3	75.9±0.2	60.7±0.2	**73.9±0.4**	59.7±1.1	67.3±0.8	68.8
SGDA	55.9±0.6	57.1±0.5	56.1±0.4	54.6±0.8	55.8±1.1	57.7±0.6	54.3±0.7	53.6±1.3	59.1±0.8	56.7±0.6	55.4±1.2	53.8±0.5	55.9
DGDA	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
A2GNN	55.9±0.7	55.7±0.4	56.6±0.6	57.1±1.0	56.1±1.2	55.8±0.5	56.5±0.7	55.5±0.4	55.9±0.8	56.2±0.6	56.5±1.5	56.0±0.5	56.2
PA-BOTH	56.4±0.5	55.9±0.6	56.0±0.5	56.4±0.4	56.3±0.6	57.7±0.7	56.6±0.2	58.8±0.9	56.9±0.7	57.2±0.3	56.5±0.5	58.3±0.8	56.9
CoCA	**81.6±1.5**	**83.5±0.6**	**78.5±0.6**	**82.4±2.3**	**71.1±0.8**	**76.9±1.1**	**75.2±0.5**	**82.0±1.1**	**79.5±1.4**	**79.5±1.2**	**72.7±0.6**	**77.7±1.0**	**78.4**

Table 2: The graph classification results (in %) on FRANKENSTEIN under node domain shift (source→target). F0, F1, F2, and F3 denote the sub-datasets partitioned with node. **Bold** results indicate the best performance.

Iterative Optimization. As introduced in Section 3.2, we first filter the highly dependable samples in the MP branch, i.e., $\Omega(i) = \{i | H^{MP}(\mathbf{z}^{MP}_{G^t_i}) > \zeta\}$, and then optimize the objective function to update ϕ in the SP branch:

$$\mathcal{L} = \mathcal{L}_1 + \alpha \mathcal{L}_{mv} + \beta \mathcal{L}_{cd}. \tag{2}$$

After that, we utilize the updated SP branch to filter the highly dependable samples, i.e., $\Omega(i) == \{i | H^{SP}(\mathbf{z}^{SP}_{G^t_i}) > \zeta\}$, and update θ in the MP branch:

$$\mathcal{L} = \mathcal{L}_2 + \alpha \mathcal{L}_{mv} + \beta \mathcal{L}_{cd}, \tag{3}$$

where α and β are the hyper-parameters. Additionally, we analysis the model complexity, which is presented in Appendix B.

4 Experiments

4.1 Experimental Settings

Datasets. We use 4 graph classification benchmarks: Mutagenicity (M) [Kazius *et al.*, 2005], FRANKENSTEIN (F) [Orsini *et al.*, 2015], NCI1 (N) [Wale *et al.*, 2008], and PROTEINS (P) [Dobson and Doig, 2003], obtained from TUDataset [Morris *et al.*, 2020] to evaluate the effectiveness of the CoCA. The details are presented in Appendix C. To assess the domain shift in each dataset, we follow [Yin *et al.*, 2023] and partition each dataset into four sub-datasets ($D0$, $D1$, $D2$, and $D3$, where D represents the respective dataset) based on edge and node density and graph flux.

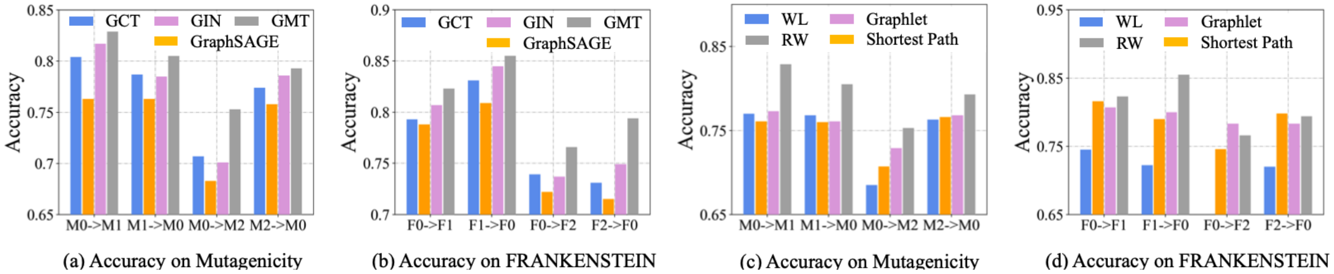
Baselines. We compare the proposed CoCA with various state-of-the-art methods, including the kernel-based approach: WL subtree [Shervashidze *et al.*, 2011], GNNs methods: GCN [Kipf and Welling, 2017b], GIN [Xu *et al.*, 2019a], GMT [Baek *et al.*, 2021], CIN [Bodnar *et al.*, 2021], and domain adaptation methods: CDAN [Long *et al.*, 2018], ToAlign [**?**], MetaAlign [Wei *et al.*, 2021], and GDA methods: DEAL [Yin *et al.*, 2022], CoCo [Yin *et al.*, 2023], SGDA [Qiao *et al.*, 2023], DGDA [Cai *et al.*, 2024], A2GNN [Liu *et al.*, 2024a] and PA-BOTH [Liu *et al.*, 2024b]. The details are introduced in Appendix E and the implementation details are proposed in Appendix D.

4.2 Performance Comparison

Table 1, 2 and 3 show the comparison performance of CoCA and baselines on Mutagenicity, FRANKENSTEIN and NCI1 datasets under different domain shift. More results are shown in Appendix F. From the results, we find that: (1) The GDA

Methods	N0→N1	N1→N0	N0→N2	N2→N0	N0→N3	N3→N0	N1→N2	N2→N1	N1→N3	N3→N1	N2→N3	N3→N2	Avg.
WL subtree	75.9	70.4	64.3	63.9	60.6	64.7	73.2	**78.9**	66.8	69.2	74.2	72.9	69.6
GCN	49.2±1.7	55.8±1.5	46.8±0.5	54.6±2.2	43.4±0.6	46.7±0.2	50.0±1.8	57.2±2.2	44.2±0.4	51.6±0.8	62.7±2.1	56.8±1.3	51.6
GIN	68.8±2.5	70.6±1.0	64.2±1.1	67.2±2.4	62.2±1.8	62.5±1.5	68.7±2.4	72.5±0.6	63.3±1.6	65.2±0.6	62.4±0.3	70.9±0.5	66.6
GMT	66.7±0.3	58.2±0.5	63.9±0.3	58.4±0.3	63.8±0.4	56.7±0.5	63.9±0.7	66.3±1.0	63.8±1.1	66.6±0.4	63.8±0.2	62.6±0.7	62.9
CIN	58.7±2.4	54.9±0.2	52.0±0.3	54.8±0.1	56.6±0.2	54.9±0.1	52.9±1.4	52.8±0.5	56.5±0.6	52.8±2.1	58.5±0.8	56.6±1.4	55.1
CDAN	64.0±1.1	68.1±0.3	60.1±0.5	64.0±1.3	60.9±0.2	57.8±1.0	64.3±1.6	61.2±0.2	66.3±0.7	59.0±0.5	68.9±0.3	63.7±0.6	63.2
ToAlign	52.8±0.5	54.8±0.2	48.2±1.1	54.8±1.5	44.0±0.8	54.8±2.0	48.2±1.7	52.8±0.6	44.0±0.2	52.8±0.3	44.0±1.0	48.2±1.2	50.0
MetaAlign	63.1±0.3	63.8±1.3	58.9±2.4	58.5±0.4	59.1±2.1	59.2±1.6	70.1±0.8	63.3±1.4	66.5±2.7	60.9±1.1	71.4±0.2	67.5±0.8	63.5
DEAL	70.7±0.9	72.3±0.2	69.9±0.8	68.9±0.7	64.1±0.6	65.6±0.9	71.9±0.4	69.9±1.7	70.6±0.4	66.5±0.3	71.6±0.7	69.9±0.5	69.3
CoCo	64.0±1.3	63.9±0.6	65.8±1.8	59.9±1.7	62.2±2.1	60.6±1.6	65.0±2.1	64.8±1.4	60.0±0.8	61.3±0.5	68.5±0.4	67.1±0.6	63.6
SGDA	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
DGDA	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
A2GNN	56.5±0.9	56.7±0.7	58.8±1.2	56.0±1.0	61.2±1.5	60.9±1.6	61.0±1.3	56.1±1.9	64.9±1.6	59.3±2.1	65.4±1.5	63.3±2.3	60.1
PA-BOTH	57.4±0.5	58.2±0.4	58.2±0.6	57.6±0.8	58.2±0.6	58.5±0.5	58.1±1.0	59.9±0.7	63.6±1.1	57.7±0.9	58.2±0.8	57.6±1.2	58.7
CoCA	**81.4**±0.9	**76.3**±1.5	**75.1**±0.7	**74.3**±1.2	**72.6**±1.5	**78.4**±0.8	**77.4**±2.3	73.2±2.0	**75.3**±0.5	**76.9**±1.0	**80.1**±1.3	**74.0**±0.3	**76.3**

Table 3: The graph classification results (in %) on NCI1 under graph flux domain shift (source→target). N0, N1, N2, and N3 denote the sub-datasets partitioned with graph flux. **Bold** results indicate the best performance. OOM means out of memory.



(a) Accuracy on Mutagenicity (b) Accuracy on FRANKENSTEIN (c) Accuracy on Mutagenicity (d) Accuracy on FRANKENSTEIN

Figure 3: The performance with different GNNs and kernels on different datasets. (a), (b) are the performance of different GNNs, (c), (d) are the performance of different graph kernels.

methods, including DEAL, CoCo, CoCA, etc., consistently outperform the kernel and GNN methods. This demonstrates that domain shift limits the expressive capability of traditional graph methods. Therefore, it is critical to design the domain invariant methods for GDA. (2) The GDA methods demonstrate competitive performance compared to traditional domain adaptation approaches. This achievement can be attributed to the challenges associated with obtaining high-quality graph representations, which make the direct application of domain adaptation techniques to graphs a demanding task. (3) The proposed CoCA outperforms recent GDA methods. We attribute this performance gain to two factors: (i) The dual branch approach for graph semantic extraction, which effectively leverages the complementary strengths of message passing and shortest path aggregation models. (ii) The architecture of the branch coupling module effectively aligns the category-level distribution, addressing the category-agnostic limitations typically encountered with GDA methods.

4.3 Flexibility of CoCA

To show the flexibility of CoCA, we replace the MP and SP branches with different GNNs and kernels. Specifically, we replace the MP branch with GCN [Kipf and Welling, 2017b], GIN [Xu *et al.*, 2019b] and Graphsage [Hamilton *et al.*, 2017], and the SP branch with Graph Sampling [Leskovec and Faloutsos, 2006], Random Walk [Kalofolias *et al.*, 2021] and WL kernel [Neumann *et al.*, 2016]. Figure 3 shows the performance of different GNNs and graph kernels on four datasets, and we have similar observations on other datasets. More results are shown in Appendix G. From the results, we observe that when compared to other GNNs and graph kernels, GMT and shortest path aggregation consistently achieve the best performance in most cases. This can be attributed to the powerful representation capabilities of GMT and the shortest path kernel. This observation further justifies our choice of GMT and shortest path aggregation to improve performance in our GDA task.

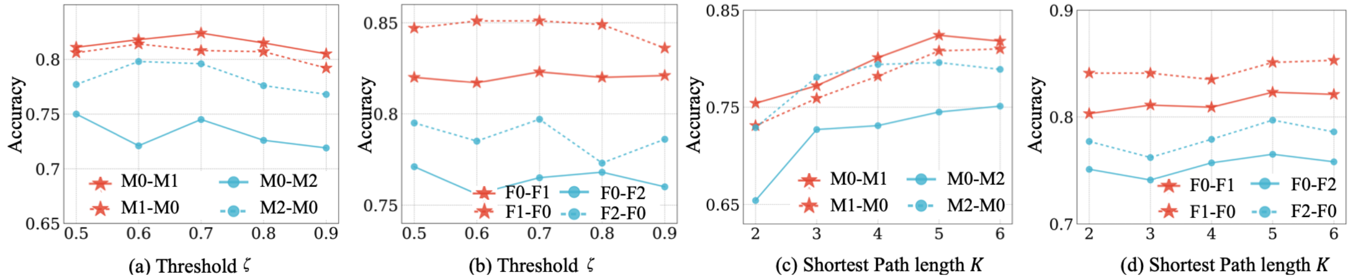
4.4 Ablation Study

To assess the impact of each module on CoCA, we conduct ablation experiments with various configurations: (1) CoCA-MP, where both branches exclusively use the message passing model; (2) CoCA-SP, where both branches exclusively use the path aggregation model; (3) CoCA/BC, removal of the branch coupling module; (4) CoCA/MV, removal of the multi-view contrastive learning module; (5) CoCA/CD, removal of the cross-domain contrastive learning module.

We conducted these experiments on the Mutagenicity dataset, and the results are presented in Table 4. From the results, we observe that: (1) CoCA consistently outperforms both CoCA-MP and CoCA-SP, emphasizing the critical importance of extracting graph semantics from both implicit and explicit perspectives in order to achieve superior performance. (2) The performance of CoCA is significantly superior to models that lack the branch coupling module (i.e., CoCA/BC), highlighting that by aligning the category-level distributions, the branch coupling module effectively resolves the category-agnostic issue that arises when attempting to

Methods	M0→M1	M1→M0	M0→M2	M2→M0	M0→M3	M3→M0	M1→M2	M2→M1	M1→M3	M3→M1	M2→M3	M3→M2	Avg.
CoCA-MP	77.3	70.1	70.8	71.6	68.3	71.1	77.2	82.8	68.3	77.6	67.8	76.2	73.2
CoCA-SP	80.6	74.1	68.8	70.8	65.7	72.5	78.3	83.3	67.2	78.5	69.6	81.3	74.2
CoCA/BC	76.0	76.3	69.3	74.4	67.4	64.9	78.8	82.2	68.5	73.2	69.4	77.5	73.2
CoCA/MV	77.3	77.2	71.8	76.4	70.2	75.7	79.3	83.1	71.4	79.5	74.2	78.5	76.2
CoCA/CD	78.1	75.4	70.4	76.7	72.7	77.2	78.8	**86.4**	72.3	80.1	73.4	79.3	76.7
CoCA	**82.9**	**80.5**	**75.3**	**79.3**	**74.4**	**79.2**	**83.1**	86.1	**74.7**	**81.3**	**78.5**	**82.6**	**79.8**

Table 4: The results of ablation studies on Mutagenicity (source→target). **Bold** results indicate the best performance.



(a) Threshold ζ

(b) Threshold ζ

(c) Shortest Path length K

(d) Shortest Path length K

Figure 4: Hyperparameter sensitivity of threshold ζ and shortest path length K on different datasets. (a), (b) are the performance of threshold ζ, (c), (d) are the performance of shortest path length K.

align the entire feature distribution, which could otherwise impair model performance. (3) CoCA/MV and CoCA/CD perform worse than CoCA, and we attribute this to the fact that by ignoring multi-view and cross-domain contrastive learning, CoCA is unable to efficiently learn consistent representations between the source and target domains, ultimately leading to a notable reduction in predictive performance.

4.5 Sensitivity Analysis

In this part, we investigate the influence of hyperparameters on the performance of the proposed CoCA. We specifically examine the effects of two key hyperparameters, including the threshold ζ in the branch coupling module for category alignment, and the shortest path length K in the SP branch. We report the results of ζ and K in Figure 4. ζ determines the number of reliable samples selected from each branch, and we vary ζ in the range from 0.5 to 0.9. The experimental results presented in Figure 4 (a), (b) indicate an initial increase followed by stability or a decreasing trend in performance as ζ increases. We attribute the reason to the fact that smaller values of ζ introduce low-confidence samples, which would detriment the performance of CoCA. Conversely, larger values of ζ introduce high-confidence samples for training. However, excessively high values of ζ may lead to fewer filtered samples, potentially resulting in a decline in model performance. Therefore, we set ζ to 0.7 as default. Additionally, the parameter K controls the number of shortest paths extracted in the SP branch, and we vary K in the range of $\{2, 3, 4, 5, 6\}$. The results are shown in Figure 4 (c) and (d). From the results, we observe that increasing K generally leads to improved performance when the value is small. This suggests that incorporating more shortest path aggregations can enhance the representation capability. However, when K becomes large, the performance stays stable. Considering the significant increase in algorithmic complexity associated with higher values of K, we set $K = 5$. Additionally, we examine the accuracy of filtered samples from MP and SP branches, and the results are shown in Appendix H.

5 Conclusion

In this paper, we address the practical challenge of unsupervised graph classification and introduce a novel approach called CoCA. We leverage a dual-branch architecture, consisting of a message passing branch and a shortest path aggregation branch, to capture graph semantics from both implicit and explicit perspectives. The incorporation of the branch coupling module ensures effective category-level alignment, thereby mitigating the category-agnostic issues commonly encountered in traditional graph domain adaptation methods. Additionally, our framework integrates cross-domain and multi-view contrastive learning, which enhances the consistency of representations across domains and branches. Theoretical analysis shows that CoCA achieves a tighter empirical risk bound compared to existing GDA methods. Extensive experiments across multiple datasets demonstrate the superior performance of CoCA in handling domain shifts. In future work, we plan to extend CoCA to more complex scenarios, including domain generalization and source-free GDA.

Contribution Statement

Nan Yin and Xiao Teng are contributing equally. Mengzhu Wang is the corresponding author.

Acknowledgments

This research was supported by the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 1 grant, the National Natural Science Foundation of China under Grants No. 62406100, Tianjin Natural Science Foundation under Grants No. 24JCQNJC00320, Beijing Postdoctoral Research Foundation and Beijing Natural Science Foundation No. 4254079.

References

[Abboud *et al.*, 2022] Ralph Abboud, Radoslav Dimitrov, and Ismail Ilkan Ceylan. Shortest path networks for graph property prediction. In *Learning on Graphs Conference*, pages 5–1, 2022.

[Baek *et al.*, 2021] Jinheon Baek, Minki Kang, and Sung Ju Hwang. Accurate learning of graph representations with graph multiset pooling. In *ICLR*, 2021.

[Bodnar *et al.*, 2021] Cristian Bodnar, Fabrizio Frasca, Nina Otter, Yu Guang Wang, Pietro Liò, Guido F Montufar, and Michael Bronstein. Weisfeiler and lehman go cellular: Cw networks. In *NeurIPS*, pages 2625–2640, 2021.

[Cai *et al.*, 2021] Lei Cai, Jundong Li, Jie Wang, and Shuiwang Ji. Line graph neural networks for link prediction. *TPAMI*, 44(9):5103–5113, 2021.

[Cai *et al.*, 2024] Ruichu Cai, Fengzhu Wu, Zijian Li, Pengfei Wei, Lingling Yi, and Kun Zhang. Graph domain adaptation: A generative view. *TKDD*, 18(3):1–24, 2024.

[Cho *et al.*, 2016] Hyunghoon Cho, Bonnie Berger, and Jian Peng. Compact integration of multi-network topology for functional analysis of genes. *Cell systems*, 3(6):540–548, 2016.

[Dai *et al.*, 2022] Quanyu Dai, Xiao-Ming Wu, Jiaren Xiao, Xiao Shen, and Dan Wang. Graph transfer learning via adversarial domain adaptation with graph convolution. *TKDE*, 35(5):4908–4922, 2022.

[Ding *et al.*, 2021] Mucong Ding, Kezhi Kong, Jiuhai Chen, John Kirchenbauer, Micah Goldblum, David Wipf, Furong Huang, and Tom Goldstein. A closer look at distribution shifts and out-of-distribution generalization on graphs. In *NeurIPSW*, 2021.

[Dobson and Doig, 2003] Paul D Dobson and Andrew J Doig. Distinguishing enzyme structures from non-enzymes without alignments. *Journal of molecular biology*, 330(4):771–783, 2003.

[Feng *et al.*, 2023] Kaituo Feng, Changsheng Li, Xiaolu Zhang, and Jun Zhou. Towards open temporal graph neural networks. *arxiv*, 2023.

[Garg *et al.*, 2020] Vikas Garg, Stefanie Jegelka, and Tommi Jaakkola. Generalization and representational limits of graph neural networks. In *ICML*, pages 3419–3430, 2020.

[Hamilton *et al.*, 2017] William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, 2017.

[Ju *et al.*, 2022] Mingxuan Ju, Shifu Hou, Yujie Fan, Jianan Zhao, Yanfang Ye, and Liang Zhao. Adaptive kernel graph neural network. In *AAAI*, pages 7051–7058, 2022.

[Kalofolias *et al.*, 2021] Janis Kalofolias, Pascal Welke, and Jilles Vreeken. Susan: The structural similarity random walk kernel. In *SDM*, pages 298–306, 2021.

[Kazius *et al.*, 2005] Jeroen Kazius, Ross McGuire, and Roberta Bursi. Derivation and validation of toxicophores for mutagenicity prediction. *Journal of medicinal chemistry*, 48(1):312–320, 2005.

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

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

[Leskovec and Faloutsos, 2006] Jure Leskovec and Christos Faloutsos. Sampling from large graphs. In *KDD*, pages 631–636, 2006.

[Li *et al.*, 2021] Junnan Li, Caiming Xiong, and Steven CH Hoi. Learning from noisy data with robust representation learning. In *ICCV*, 2021.

[Lin *et al.*, 2023] Mingkai Lin, Wenzhong Li, Ding Li, Yizhou Chen, Guohao Li, and Sanglu Lu. Multi-domain generalized graph meta learning. In *AAAI*, pages 4479–4487, 2023.

[Liu *et al.*, 2024a] Meihan Liu, Zeyu Fang, Zhen Zhang, Ming Gu, Sheng Zhou, Xin Wang, and Jiajun Bu. Rethinking propagation for unsupervised graph domain adaptation. *AAAI*, pages 13963–13971, 2024.

[Liu *et al.*, 2024b] Shikun Liu, Deyu Zou, Han Zhao, and Pan Li. Pairwise alignment improves graph domain adaptation. *ICML*, 2024.

[Long *et al.*, 2018] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *NeurIPS*, 2018.

[Long *et al.*, 2021] Qingqing Long, Yilun Jin, Yi Wu, and Guojie Song. Theoretically improving graph neural networks via anonymous walk graph kernels. In *WWW*, pages 1204–1214, 2021.

[Luo *et al.*, 2023] Yadan Luo, Zijian Wang, Zhuoxiao Chen, Zi Huang, and Mahsa Baktashmotlagh. Source-free progressive graph learning for open-set domain adaptation. *TPAMI*, 2023.

[Mohri *et al.*, 2018] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[Morris *et al.*, 2020] Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. In *ICMLW*, 2020.

[Neumann *et al.*, 2016] Marion Neumann, Roman Garnett, Christian Bauckhage, and Kristian Kersting. Propagation kernels: efficient graph kernels from propagated information. *Machine Learning*, 102(2):209–245, 2016.

[Orsini *et al.*, 2015] Francesco Orsini, Paolo Frasconi, and Luc De Raedt. Graph invariant kernels. In *IJCAI*, pages 3756–3762, 2015.

[Qiao *et al.*, 2023] Ziyue Qiao, Xiao Luo, Meng Xiao, Hao Dong, Yuanchun Zhou, and Hui Xiong. Semi-supervised domain adaptation in graph transfer learning. In *IJCAI*, pages 2279–2287, 2023.

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

[Shen *et al.*, 2018] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. Wasserstein distance guided representation learning for domain adaptation. In *AAAI*, 2018.

[Shervashidze *et al.*, 2011] Nino Shervashidze, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. Weisfeiler-lehman graph kernels. *JMLR*, 12(9), 2011.

[Verma and Zhang, 2019] Saurabh Verma and Zhi-Li Zhang. Stability and generalization of graph convolutional neural networks. In *KDD*, pages 1539–1548, 2019.

[Wale *et al.*, 2008] Nikil Wale, Ian A Watson, and George Karypis. Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowledge and Information Systems*, 14:347–375, 2008.

[Wang *et al.*, 2021] Yanbang Wang, Yen-Yu Chang, Yunyu Liu, Jure Leskovec, and Pan Li. Inductive representation learning in temporal networks via causal anonymous walks. *arxiv*, 2021.

[Wei *et al.*, 2021] Guoqiang Wei, Cuiling Lan, Wenjun Zeng, and Zhibo Chen. Metaalign: Coordinating domain alignment and classification for unsupervised domain adaptation. In *CVPR*, pages 16643–16653, 2021.

[Wu *et al.*, 2020a] Man Wu, Shirui Pan, Chuan Zhou, Xiaojun Chang, and Xingquan Zhu. Unsupervised domain adaptive graph convolutional networks. In *WWW*, pages 1457–1467, 2020.

[Wu *et al.*, 2020b] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *TNNLS*, 32(1):4–24, 2020.

[Wu *et al.*, 2022] Qitian Wu, Hengrui Zhang, Junchi Yan, and David Wipf. Handling distribution shifts on graphs: An invariance perspective. *arxiv*, 2022.

[Xu *et al.*, 2019a] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2019.

[Xu *et al.*, 2019b] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2019.

[Yehudai *et al.*, 2021] Gilad Yehudai, Ethan Fetaya, Eli Meirom, Gal Chechik, and Haggai Maron. From local structures to size generalization in graph neural networks. In *ICML*, pages 11975–11986, 2021.

[Yin *et al.*, 2022] Nan Yin, Li Shen, Baopu Li, Mengzhu Wang, Xiao Luo, Chong Chen, Zhigang Luo, and Xian-Sheng Hua. Deal: An unsupervised domain adaptive framework for graph-level classification. In *ACMMM*, pages 3470–3479, 2022.

[Yin *et al.*, 2023] Nan Yin, Li Shen, Mengzhu Wang, Long Lan, Zeyu Ma, Chong Chen, Xian-Sheng Hua, and Xiao Luo. Coco: A coupled contrastive framework for unsupervised domain adaptive graph classification. In *ICML*, pages 40040–40053, 2023.

[You *et al.*, 2022a] Yuning You, Tianlong Chen, Zhangyang Wang, and Yang Shen. Bringing your own view: Graph contrastive learning without prefabricated data augmentations. In *WSDM*, pages 1300–1309, 2022.

[You *et al.*, 2022b] Yuning You, Tianlong Chen, Zhangyang Wang, and Yang Shen. Graph domain adaptation via theory-grounded spectral regularization. In *ICLR*, 2022.

[You *et al.*, 2023] Yuning You, Tianlong Chen, Zhangyang Wang, and Yang Shen. Graph domain adaptation via theory-grounded spectral regularization. In *ICLR*, 2023.

[Zhang *et al.*, 2019a] Qiming Zhang, Jing Zhang, Wei Liu, and Dacheng Tao. Category anchor-guided unsupervised domain adaptation for semantic segmentation. In *NeurIPS*, volume 32, 2019.

[Zhang *et al.*, 2019b] Yizhou Zhang, Guojie Song, Lun Du, Shuwen Yang, and Yilun Jin. Dane: domain adaptive network embedding. In *IJCAI*, page 4362–4368, 2019.

[Zhu *et al.*, 2023] Qi Zhu, Yizhu Jiao, Natalia Ponomareva, Jiawei Han, and Bryan Perozzi. Explaining and adapting graph conditional shift. *arxiv*, 2023.