DEAR: DECISION BOUNDARY-AWARE OPEN-SET GRAPH DOMAIN ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph domain adaptation, a subfield of graph transfer learning, aims to transfer knowledge from a label-rich source domain to a related but label-scarce target domain. In the Open-Set Graph Domain Adaptation (OSGDA) setting, the target domain contains novel classes that are absent in the source domain, which introduces significant difficulty by interfering with the recognition of known classes. How to effectively transfer domain-invariant knowledge while distinguishing unknown classes remains a major challenge. Existing OSGDA approaches primarily focus on separating unknown classes in the target domain, often overlooking the limitations imposed by the robustness of representation learning in the source domain itself. To address this, we propose a novel decision boundary-aware method called DEAR. Specifically, we introduce a virtual class to compress the decision space of known classes, thereby reserving space in the decision boundary for unknown classes in the target domain. Then, within the target domain, we exploit the virtual class to mine intra-class relationships and accurately identify the decision boundary between known and unknown classes. Finally, we perform cross-domain contrastive learning based on feature and structure perturbations for the known-class nodes to further align transferable knowledge. Extensive experiments on multiple benchmark datasets demonstrate that DEAR achieves state-of-the-art performance.

1 INTRODUCTION

Graph-structured data are ubiquitous in real-world scenarios, such as social networks, recommender systems, and biomolecular networks (Ju et al., 2024; Luo et al., 2026; Qin et al., 2026). Graph Neural Networks (GNNs), an emerging graph machine learning paradigm in recent years, leverage a message-passing mechanism (Gilmer et al., 2017) to aggregate information by jointly utilizing both the graph topology and node attributes. This approach has been successfully applied to various graph-related tasks (Shen et al., 2025; Ju et al., 2025b; Chen et al., 2025b). As GNNs demonstrate strong performance across diverse applications, effectively transferring knowledge across different graphs remains an important challenge in graph machine learning.

Graph transfer learning aims to transfer knowledge across different graphs to facilitate effective learning on target graphs, and graph domain adaptation (GDA) represents a prominent research direction in this field, focusing on knowledge transfer across related graph domains (Cai et al., 2024). Recent advances in GDA mainly follow two research directions: methods based on predefined criteria (e.g., graph subtree discrepancy, maximum mean discrepancy) (Shen et al., 2020; Gretton et al., 2012) and those driven by adversarial learning (Yang et al., 2026; Fang et al., 2026a). Despite substantial progress in GDA, most existing methods still rely on the closed-set assumption (CSGDA), where the label space is identical across domains and it often does not hold in real-world applications. In this paper, we instead explore a more realistic setting: Open-Set Graph Domain Adaptation (OSGDA), where the target graph contains novel classes that are absent from the source domain. As illustrated in Figure 1, the goal in the OSGDA task is to correctly classify samples from known classes in the target domain while effectively identifying and separating noisy samples from unknown classes.

In the OSGDA setting, we identify three major challenges that hinder effective knowledge transfer across graph domains: (1) **How to accurately detect unknown classes in the target domain.** Since the source domain contains no supervision for unknown categories, these nodes can be easily confused with known classes, thereby interfering with the classifica-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

tion of known nodes in target domain. Accurately detecting unknown nodes and separating them from known-node learning can effectively reduce the interference. (2) **How to perform effective cross-domain knowledge transfer under the interference of unknown classes.** Directly aligning the source and target domains may incorrectly align unknown nodes with known classes, leading to negative transfer and degraded adaptation performance. Therefore, it is crucial to exploit reliable known-class information for learning domain-invariant representations while suppressing the influence of unknown nodes. (3) **How to improve the robustness of representation learning in source domain.** Recent approaches such as SDA (Wang et al., 2024) and UAGA (Shen et al., 2025) mainly rely on standard cross-entropy loss for source-domain supervision. However, cross-entropy loss may produce ambiguous decision boundaries (Chen et al., 2018), leading to less robust representations of known classes. Such fragile boundaries become particularly problematic in OSGDA, where unknown-class nodes may easily fall into the decision regions of known classes and further deteriorate recognition performance.

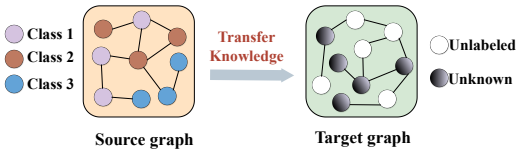


Figure 1: Illustration of the OSGDA task. It aims to transfer knowledge from a label-rich source graph to an unlabeled target graph that contains novel classes absent from the source graph.

To address the aforementioned challenges, we propose a novel **DE**cision bound**A**ry-awa**R**e open-set graph domain adaptation method (DEAR). Specifically, during the source domain training phase, we introduce a virtual class to encourage the model to compress the decision space of known classes and enhance inter-class discriminability. Then, we leverage the semantic information of the virtual class and model the target domain by a parametric probability distribution. This distributional modeling enables us to identify high-confidence known and unknown samples in the target domain, significantly reducing the risk of misclassification between unknown and known nodes. Finally, to exploit the semantic labels in the source domain and account for the inherent topological properties of graph data, we introduce perturbations to source node features and target domain edges, and conduct cross-domain contrastive learning, which facilitates more robust knowledge alignment in the representation space, achieving effective domain adaptation. In summary, our contributions are as follows:

- **Novel:** We introduce a decision boundary-aware OSGDA framework that leverages virtual-class learning to compact the decision regions of source-domain known classes and create a more discriminative decision space for distinguishing unseen target classes.
- **Robust:** We exploit virtual-class semantics to identify reliable known and unknown nodes, and perform perturbation-based cross-domain contrastive learning for robust knowledge transfer.
- **Theoretical:** We theoretically show that virtual-class learning enlarges classification margins and improves intra-class consistency, supporting robust boundary learning and reliable relation mining.
- **Empirical:** Extensive experiments demonstrate that our DEAR consistently achieves strong and competitive performance against state-of-the-art baselines across diverse OSGDA settings.

2 METHODOLOGY

2.1 PRELIMINARIES

Source & Target Graph. We denote the labeled source graph and the unlabeled target graph as $G^s = \{V^s, E^s, X^s, A^s, Y^s\}$ and $G^t = \{V^t, E^t, X^t, A^t\}$. V and E represent the sets of nodes and edges, $|V|$ and $|E|$ denote the numbers of nodes and edges. The node feature matrices are denoted as $X \in \mathbb{R}^{|V| \times F}$, where F denotes the dimensionality of the node features. The adjacency matrices are represented as $A \in \mathbb{R}^{|V| \times |V|}$. The label matrix for the source domain is denoted as $Y^s \in \mathbb{R}^{|V^s| \times |C_s|}$, where C_s is the label space in the source graph, and $|C_s|$ is the number of classes.

Problem Definition. In this paper, we focus on the task of the node-level classification under the OSGDA setting. Let C_s and C_t denote the true label spaces of the source graph and target graph. In the OSGDA setting, the target domain contains novel classes not present in the source domain, i.e., $C_s \subset C_t$. We define the set of known classes as $C_k := C_s \cap C_t$, and the set of unknown classes as $C_{uk} := C_t \setminus C_s$. The objective of OSGDA is to train a GNN model $\mathcal{G}$ on the label-rich source domain

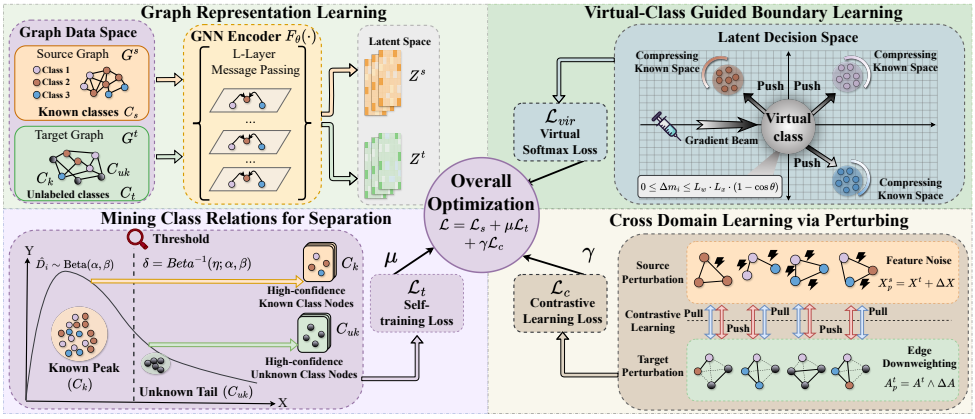


Figure 2: An overview of the proposed framework DEAR, which consists of three components: (1) Virtual-class guided boundary learning to enforce compact and discriminative decision boundaries; (2) class-relation mining to separate known and unknown nodes in the target domain; and (3) cross-domain alignment via both feature-level and structure-level perturbation and contrastive learning.

such that it can accurately classify nodes from C_k and effectively distinguish nodes belonging to C_{uk} in the target domain. Formally, the model outputs $\mathcal{G}(X^t) \to \hat{Y}^t \in \mathbb{R}^{|V^t| \times (|C_s|+1)}$, the first $|C_s|$ dimensions correspond to the known classes, and the rest $|C_s| + 1$-th dimension represents C_{uk}.

2.2 Overview of Framework

Our method consists of three key components: (1) virtual-class guided boundary learning; (2) mining class relations to separate C_k and C_{uk}; (3) cross-domain knowledge transfer via perturbation. We describe each component in detail below, with an overview shown in Figure 2.

2.3 Virtual-Class Guided Boundary Learning

Feature Extractor $F_\theta(\cdot)$. To comprehensively capture both node feature information and high-order topological structure on the graph, we project the graph from the data space into a high-dimensional embedding space using graph convolution (Kipf & Welling, 2017):

$$Z_l^* = \sigma(\tilde{D}^{*-\frac{1}{2}} \tilde{A}^* \tilde{D}^{*-\frac{1}{2}} Z_{(l-1)}^* W_l), \tag{1}$$

where $\tilde{A}^* = A^* + I$ denotes the adjacency matrix with self-loops added, and $\tilde{D}^*$ is the corresponding degree matrix. Z_l^* and W_l represent the node embeddings and filter weights at the l-th graph convolutional layer. By stacking L layers of convolution, we obtain an embedding matrix $Z^* = F_\theta(X^*) \in \mathbb{R}^{|V^*| \times D}, * \in \{s, t\}$.

Virtual-class oriented Classifier $\mathcal{C}_\omega(\cdot)$. To project node embeddings from the representation space into a label-related decision space for node classification, we introduce a classifier augmented with a virtual class. Unlike standard classifiers that use the softmax function, this classifier adopts the virtual softmax function (Chen et al., 2018), which incorporates a virtual class into the probability distribution during prediction. The formulation is as follows:

$$\mathcal{L}_{\text{vir}} = -\frac{1}{|V^s|} \sum_{i=1}^{|V^s|} \log \frac{e^{W_{y_i^s}^\top Z_i^s}}{\sum_{j=1}^{|C_s|} e^{W_j^\top Z_i^s} + e^{W_{vir}^\top Z_i^s}}, \tag{2}$$

where $W_j \in \mathbb{R}^{D \times 1}$ denotes the weight vector of the j-th row in the classifier $\mathcal{C}_\omega(\cdot)$, and y_i^s is the ground-truth label of the i-th instance. $W_{vir} = \frac{\|W_{y_i^s}\| Z_i^s}{\|Z_i^s\|}$ represents the virtual class we inject, where $\|\cdot\|$ denotes the L2 normalization. This virtual base is normalized and dynamically scaled based on each instance to better adapt to its label space. Introducing a virtual class significantly expands decision margins and compresses the intra-class space, thereby enhancing robustness for OSGDA. We now present the theoretical foundation demonstrating that the virtual class can effectively enlarge the model's decision margin as Theorem 1. And the proof is provided in the Appendix A.1.

Theorem 1. *In the GNN node classification task, let $X_i \in \mathcal{R}^d$ denote the feature of node i, $F_\theta(\cdot)$ be the feature extractor, and $F_\theta(X_i)$ the embedding representation of node i. Let $\mathcal{C}_\omega(\cdot)$ denote the classifier with final-layer weights $W = [W_1, \cdots, W_c]$. The classification margin of node i is defined as: $m_i = Logits(X_i)_{y_i} - \max_{j \neq y_i} Logits(X_i)_j$. If the embeddings and weight vectors of all classes and instances satisfy the Lipschitz condition:*

$$\|W_c\| \leq L_w, \|F_\theta(X_i)\| \leq L_x, \left\|W_c^\top F_\theta(X_i)\right\| \leq M, \tag{3}$$

then for each instance X_i, after injecting the virtual class W_{vir}, the margin gain satisfies:

$$0 \leq \Delta m_i \leq L_w \cdot L_x \cdot (1 - \cos\Theta), \tag{4}$$

where Θ denotes the angle between the embedding vector of an instance and its corresponding class weight vector in the representation space.

2.4 Mining Class Relations for Separation

Accurately distinguishing C_k from C_{uk} is critical in OSGDA. Unlike existing methods that rely solely on target-specific information (e.g., entropy) (Wang et al., 2024), we incorporate label-rich source information to exploit inter-class relationships, thereby enhancing separation. Furthermore, we leverage virtual class semantics to mine cross-domain intra-class correlations. This mechanism facilitates the identification of high-confidence known nodes in the target domain, significantly bolstering model robustness.

Distinguish C_{uk} Nodes. To dynamically obtain source-domain information during training, we compute the mean prototype of the predictions for each class in source domain:

$$\tilde{\mathcal{P}}_c = \frac{\sum_{i=1}^{|V^s|} \Pi(y_i^s = c)\mathcal{C}_\omega(Z_i^s)}{\sum_{i=1}^{|V^s|} \Pi(y_i^s = c)}, \tag{5}$$

where $\Pi(\cdot)$ denotes the indicator function, which returns 1 if the condition inside holds, and 0 otherwise. To mine intra-class relationships between known-class nodes in the source domain and target domain, we compute the KL divergence between the prediction p_i^t of a target domain node and the corresponding source domain prototype $\tilde{\mathcal{P}}_c$:

$$D_{KL}(p_i^t || \tilde{\mathcal{P}}_c) = \sum_{k=1}^{|C_s|+1} p_i^t(k) \log \frac{p_i^t(k)}{\tilde{\mathcal{P}}_c(k)}, \quad c = \arg\max_c(p_i^t[c]). \tag{6}$$

Since the source domain prototype is derived from known classes, while the target domain includes both known and unknown classes, KL divergence values involving unknown classes tend to be significantly larger. This results in two underlying distributions: one for known classes and one for unknown classes. To better model this data distribution, we fit a Beta distribution to the KL divergence values:

$$\hat{D}_i = \frac{D_{KL}(p_i^t || \tilde{\mathcal{P}}_c)}{\max_j D_{KL}(p_j^t || \tilde{\mathcal{P}}_c)}, \quad \hat{D}_i \sim \text{Beta}(\alpha, \beta), \tag{7}$$

and we then use the η quantile of this distribution as a threshold to distinguish C_k and C_{uk} nodes:

$$\delta = \text{Beta}^{-1}(\eta; \alpha, \beta), \tag{8}$$

where η is a hyperparameter that strictly controls the threshold. Based on this threshold, we classify target domain nodes using the following decision function:

$$\Lambda(X_i^t) = \begin{cases} 1\,, & \text{if } \hat{D}_i > \delta \\ 0\,, & \text{otherwise} \end{cases}, \tag{9}$$

where $\Lambda(X_i^t) = 1$ indicates that the target node is identified as C_{uk}. Rather than relying on a manually specified fixed threshold, we allow the model to adaptively determine the threshold according to the underlying data distribution. This data-driven strategy enables more reliable identification of target domain nodes that are substantially dissimilar from the source-domain known classes, leading to more precise, flexible, and controllable filtering than conventional static thresholding.

Obtain High-Confidence C_k Nodes. To mitigate the potential risk of unknown-class nodes being mistakenly identified as known, we further explore the properties of the virtual class to mine inter-class relationships between nodes with the same label across source and target domains, thereby selecting high-quality known-class nodes. We now present a second theoretical foundation as Theorem 2 that reveals the intra-class consistency property of the virtual class. Theorem 2 demonstrates that the virtual logits obtained through virtual class injection exhibit more stable intra-class consistency than the original logits. Furthermore, the stability of these virtual logits can be strategically exploited to mine intrinsic cross-domain intra-class relationships, enabling the selection of source graph high-confidence C_k nodes. The proof of Theorem 2 is provided in the Appendix A.2.

Theorem 2. *Let the setup be the same as in Theorem 1. Let $\mathcal{X}_c = \{x_i | y_i = c\}$ denote the set of nodes belonging to a known class c. For each instance $x_i \in \mathcal{X}_c$, its virtual logit after injecting the virtual class is computed as $l_{vir} := \|W_c\| \cdot \|F_\theta(X_i)\|$. If the classifier satisfies the Lipschitz condition $\|W_c\| \leq L_w$, then the variance of l_{vir} is bounded by:*

$$\mathcal{V}(l_{vir}) \leq L_w^2 \cdot \mathcal{V}(\|F_\theta(X_i)\|), \tag{10}$$

and if $\|W_c\|$ remains roughly constant during training, then the variance of virtual logits is solely controlled by the magnitude of the embeddings and is significantly smaller than the variance of the original logits.

We first compute the mean prototype of the virtual class for each category in the source domain:

$$\tilde{\mathcal{P}}_{vir} = \frac{\sum_{i=1}^{|V^s|} \Pi(y_i^s = c) W_{i,vir}^s}{\sum_{i=1}^{|V^s|} \Pi(y_i^s = c)}, \tag{11}$$

where $W_{i,vir}^s$ denotes the virtual class vector injected for each instance in the source domain. Next, for target domain nodes with $\Lambda(X_i^t) = 0$, we compare them with the prototype virtual class to filter out high-confidence known-class nodes:

$$\Omega(X_i^t) = \begin{cases} 1\,, & \text{if } |W_{i,vir}^t - \tilde{\mathcal{P}}_{\hat{y}_i,vir}| < \epsilon, \\ 0\,, & \text{otherwise} \end{cases}, \tag{12}$$

where ϵ is set to 0.5 and $W_{i,vir}^t$ denotes the virtual class of the target node computed based on its pseudo label, $\hat{y}_i^t$ is the pseudo label of the target node, and $\tilde{\mathcal{P}}_{\hat{y}_i^t,\text{vir}}$ is the corresponding prototype virtual class. The operator $|\cdot|$ represents the L1 norm. If $\Omega(X_i^t) = 1$, the node is identified as a high-confidence known-class node. Finally, we apply self-supervised training on the selected high-confidence C_k and C_{uk} nodes from the target domain:

$$\mathcal{L}_{\text{t}} = -\frac{1}{|V_h^t|} \sum_{i \in V_h^t}^{|V_h^t|} \log \frac{e^{W_{\hat{y}_i^t}^\top Z_i^t}}{\sum_{j=1}^{|C_s|+1} e^{W_j^\top Z_i^t}}, \tag{13}$$

where V_h^t denotes the set of high-confidence C_k and C_{uk} nodes.

2.5 Cross Domain Learning via Perturbing

In GDA tasks, domain discrepancy often manifests not only in node features but also in graph topology. Therefore, to mitigate domain shifts and capture domain-invariant knowledge, it is essential to address both the node features and the structural properties of the graph. To improve cross-domain generalization, we introduce controlled perturbations to both node features in the source domain and graph structure in the target domain during training, which can be formulated as:

$$X_p^s = X^s + \Delta X, \quad A_p^t = A^t \wedge \Delta A. \tag{14}$$

Specifically, for the label-rich source domain, we perturb node features by adding Gaussian noise ΔX sampled from a standard normal distribution, encouraging the learned representations to remain invariant to small feature shifts and thus better generalize to known-class nodes in the target domain. Meanwhile, since the target domain contains both known and unknown classes, message passing may propagate noisy information from unknown nodes to known ones through inter-class edges. To

alleviate this, we apply edge perturbation by downweighting edges associated with nodes identified as unknown, where $\Delta A_{ij} = 0.2$ if $\Lambda(X_i^t) = 1$ or $\Lambda(X_j^t) = 1$. This strategy effectively suppresses harmful information propagation from unknown classes during message passing, thereby preserving discriminative representations for known classes in the target domain.

Finally, we perform cross-domain contrastive learning between the perturbed source domain and the perturbed target domain known-class nodes:

$$\mathcal{L}_{\mathrm{c}} = \sum_{v_i^t \in C_k} -\log \frac{e^{Z_{p,i}^t \cdot u_{p,\hat{y}_i^t}^s / \tau}}{\sum_{j=1}^{|C_s|} e^{Z_{p,i}^t \cdot u_{p,j}^s / \tau}}, \tag{15}$$

where $\mathcal{Z}_{p,i}^t$ denotes the perturbed embedding of the target node after edge perturbation, $u_{p,j}^s$ represents the class-wise prototype embedding of the perturbed source domain, and $\hat{y}_i^t$ is the pseudo-label of the target node identified as C_k. By strategically introducing both feature-level and structure-level perturbations and employing a contrastive learning paradigm, the model is effectively encouraged to learn domain-invariant representations that are inherently robust and highly generalizable, thereby significantly improving the overall performance of domain adaptation.

2.6 Overall Optimization

In our experiments, to further leverage the label information from the source domain for learning consistent knowledge, we additionally introduce a supervised learning objective without the virtual class in the source domain:

$$\mathcal{L}_{\mathrm{ce}} = -\frac{1}{|V^s|} \sum_{i=1}^{|V^s|} \log \frac{e^{W_{y_i^s}^\top Z_i^s}}{\sum_{j=1}^{|C_s|} e^{W_j^\top Z_i^s}}. \tag{16}$$

Based on the above formulations, we define the final overall optimization objective of our DEAR as:

$$\mathcal{L} = \mathcal{L}_{\mathrm{s}} + \mu \mathcal{L}_{\mathrm{t}} + \gamma \mathcal{L}_{\mathrm{c}}, \tag{17}$$

where $\mathcal{L}_{\mathrm{s}} = \mathcal{L}_{\mathrm{ce}} + \mathcal{L}_{\mathrm{vir}}$. The hyperparameter μ and hyperparameter γ control the contributions of the target domain self-supervised loss and the cross-domain contrastive learning loss in the total loss, respectively. And the complete algorithm of our proposed DEAR is provided in Appendix D.

3 Experiments

3.1 Experimental Setup

Task Setting. We evaluate node classification under OSGDA on three group of benchmark datasets. The first one consists of widely used citation networks, including ACMv9 (A), Citationv1 (C), and DBLPv7 (D) (Tang et al., 2008); The second is the large scale citation network ogbn-arxiv (Hu et al., 2020), which is partitioned into three temporal domains (Arx.I,II,III); And the last one is webpage graphs WebKB (Pei et al., 2020), including Cornell (Cor), Texas (Tex), and Wisconsin (Wis). Detailed dataset statistics and settings are provided in Appendix F.

Baselines. We compare DEAR with 12 baselines spanning five categories: classical GNNs (GCN, GraphSAGE) (Kipf & Welling, 2017; Hamilton et al., 2017); unsupervised domain adaptation (DANN, CDAN) (Ganin et al., 2016; Long et al., 2018); open-set domain adaptation (OSBP, DANCE) (Saito et al., 2018; 2020); closed-set graph domain adaptation (UDAGCN, ASN) (Wu et al., 2020; Zhang et al., 2021); and OSGDA (SDA, UAGA, ETA, GraphRTA) (Wang et al., 2024; Shen et al., 2025; Ju et al., 2025a; Zhang & He, 2025). More details on the baselines are provided in Appendix F.

Experimental Details. We use the widely adopted metrics in OSGDA, accuracy (ACC) and H-score (HS) (Fu et al., 2020) as our evaluation metrics. For the implementation, we use PyTorch (Paszke et al., 2019) and PyTorch Geometric (Fey & Lenssen, 2019), with a two-layer GCN backbone (Kipf & Welling, 2017) and a 512-dimensional embedding. The learning rate and weight decay are both 0.001, and the contrastive temperature is 0.5. We set $\eta = 0.9$ and $\mu = \gamma = 0.1$. For the citation datasets, we set two classes as unknown and train for 200 epochs under 10 possible unknown-class configurations and report the mean performance of DEAR and all baselines. For ogbn-arxiv and WebKB, we set ten classes and two classes as unknown and train 200 epochs, we report the mean and standard deviation of OSGDA baselines and DEAR over five runs per task.

Table 1: Performance evaluations of different methods on citation datasets with six OSGDA tasks (%). The notation A⇒D denotes the source domain is ACMv9 and the target domain is DBLPv7. The best and second-best results are **bold** and underlined, respectively.

Task	Metric	GCN	GraphSAGE	DANN	CDAN	OSBP	DANCE	UDAGCN	ASN	SDA	UAGA	ETA	GraphRTA	**DEAR**
A⇒D	ACC	45.10	48.26	33.30	31.13	28.56	60.54	36.20	56.40	61.60	58.44	61.87	61.24	**62.57**
	HS	41.80	46.22	28.07	21.65	11.27	25.99	26.59	37.55	49.22	56.28	60.96	61.34	**62.61**
D⇒A	ACC	38.95	43.14	34.58	29.03	26.20	53.27	31.90	47.49	51.36	52.97	54.21	55.48	**60.26**
	HS	39.52	42.84	36.53	27.76	12.91	39.53	12.31	43.93	50.86	47.40	53.76	56.52	**58.79**
A⇒C	ACC	46.36	50.60	39.64	30.99	29.32	63.23	37.44	59.88	64.47	64.13	62.76	65.43	**67.09**
	HS	43.91	49.04	41.22	26.00	11.15	39.15	32.01	49.82	55.27	53.66	62.57	65.68	**66.70**
C⇒A	ACC	44.14	48.13	34.47	31.72	27.80	60.44	35.64	57.51	61.67	55.17	61.82	61.24	**62.74**
	HS	43.66	46.36	34.42	30.91	7.34	35.88	22.76	47.87	55.89	54.17	58.17	62.04	**62.23**
C⇒D	ACC	48.45	51.72	36.92	35.69	33.81	64.29	41.88	56.65	**67.51**	62.18	64.78	64.82	65.37
	HS	44.61	48.66	41.88	30.47	18.89	28.98	36.48	45.62	55.35	61.17	63.50	**65.38**	64.85
D⇒C	ACC	42.26	47.20	35.20	28.62	28.63	57.62	35.50	56.97	57.74	62.58	59.96	59.63	**67.52**
	HS	41.25	46.70	35.46	21.82	14.16	39.50	25.09	46.19	54.72	58.00	55.90	61.56	**66.14**
Average	ACC	44.21	48.17	35.68	31.20	29.05	59.90	36.43	55.81	60.73	59.25	60.90	61.31	**64.26**
	HS	42.46	46.64	36.26	26.44	12.62	34.84	25.87	45.16	53.55	55.11	59.15	62.09	**63.55**

Table 2: Performance of OSGDA methods on ogbn-arxiv and WebKB (%), '–' indicates results unavailable from the released source code. We **bold** the best and underline the second-best results.

Method	Arx.I→Arx.II		Arx.I→Arx.III		Arx.II→Arx.III		Cor→Wis		Tex→Cor		Tex→Wis	
	ACC	HS	ACC	HS	ACC	HS	ACC	HS	ACC	HS	ACC	HS
SDA	39.77±0.34	42.60±0.15	36.37±0.21	39.44±0.21	41.53±0.20	46.03±0.18	–	–	–	–	–	–
UAGA	32.92±0.16	23.69±0.16	32.24±0.13	22.98±0.13	39.16±0.39	29.79±0.41	–	–	–	–	–	–
ETA	45.26±1.33	45.82±1.84	43.28±1.04	42.58±0.98	48.53±1.46	49.62±0.93	36.84±2.01	37.73±1.85	47.32±1.63	40.28±1.44	34.15±0.98	35.49±2.04
GraphRTA	47.70±1.39	50.79±2.79	45.52±2.00	46.25±0.40	52.37±1.49	48.42±1.94	33.46±4.43	34.36±1.76	**52.18±0.38**	35.64±0.83	29.34±1.21	30.08±2.96
DEAR	**53.25±0.63**	**53.88±1.88**	**51.49±1.04**	**54.09±3.82**	**57.03±0.45**	**61.05±0.84**	**47.97±3.12**	**48.77±2.96**	50.49±1.27	**51.56±1.09**	**47.25±3.49**	**47.95±3.85**

3.2 Analysis of Performance on OSGDA Tasks

The performance of our DEAR and the baselines on the OSGDA tasks is summarized in Table 1 and Table 2. On the citation networks, DEAR achieves the best ACC and HS on most transfer tasks and obtains the highest average performance in both metrics. These results indicate that DEAR not only improves the overall recognition accuracy, but also maintains a favorable balance between recognizing known classes and detecting unknown classes, which is particularly important for open-set graph domain adaptation. Notably, the improvements remain consistent across different transfer directions, even though different transfer pairs exhibit distinct domain discrepancies. This suggests that our DEAR does not rely on a specific domain configuration and can adapt effectively to varying distribution shifts. Moreover, our DEAR also achieves leading performance on most transfer tasks involving the large scale graph ogbn-arxiv and webpage graph WebKB, demonstrating that its effectiveness is not limited to relatively small citation networks. The consistent gains across citation networks, large scale graphs, and webpage graphs with different feature distributions and structural characteristics further demonstrate the general applicability of DEAR for diverse OSGDA scenarios.

3.3 Ablation Study

To further evaluate the contribution of each component in DEAR, we conduct ablation study with five components to be removed: • w/o $\mathcal{L}_{\text{vir}}$: removing the virtual loss associated with the injected virtual class; • w/o $\mathcal{L}_{\text{t}}$: removing the pseudo label based self-supervised loss on the target domain; • w/o $\mathcal{L}_{\text{c}}$: removing the contrastive learning loss based on perturbation; • w/o $\Delta_{X,A}$: removing the perturbations used in the contrastive learning process; • w/o v_h: removing the self-supervised loss that uses high-confidence known-class nodes identified through intra-class relationship mining. As shown in Figure 3, removing any component leads to a decline. The most significant drop occurs when removing $\mathcal{L}_{\text{t}}$, which is intuitive, since the unknown-class labels in the target domain rely entirely on self-supervised learning, removing it eliminates the model's ability to distinguish them. Removing the use of high-confidence

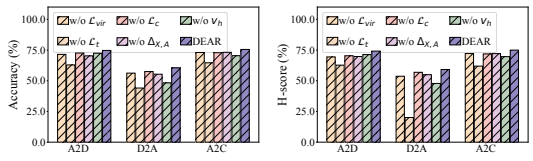


Figure 3: Illustration of the dynamic changes of the denoising masks during training.

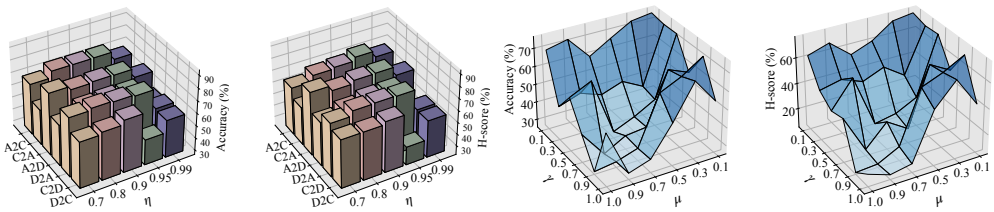
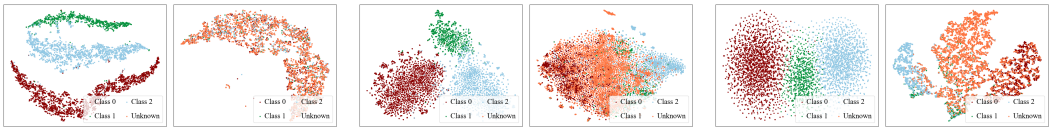


Figure 4: Visualization of the sensitivity analysis for hyperparameters η, μ, and γ. For η, the x-axis and y-axis represent the OSGDA tasks in citation datasets and the variant values of η, respectively. In the analyses of μ and γ, both the x- and y-axes denote the hyperparameter values. The z-axis consistently indicates the performance metrics (Accuracy and H-score).



(a) SDA (b) UAGA (c) DEAR

Figure 5: t-SNE visualization of node representation learned by baselines and our DEAR. Each pair corresponds to one method and for every pair the left one represents the source domain (ACMv9) and the right one represents the target domain (Citationv1).

known-class nodes also causes a substantial drop, highlighting their importance in ensuring reliable cross-domain knowledge transfer. The performance drops in other settings further validate the critical role of virtual class and perturbation based techniques.

3.4 Sensitivity Analysis

To further evaluate DEAR, we conduct hyperparameter analysis experiments to investigate the sensitivity of different parameter values on model performance, and the results are shown in Figure 4. **1) Sensitivity Analysis of η.** We evaluate how varying $\eta \in \{0.7, 0.8, 0.9, 0.95, 0.99\}$ affects model performance. As η increases, the performance initially improves and then declines. In DEAR, η controls the strictness of the threshold when distinguishing unknown nodes via inter-class relations. A small η leads to over identification of known class nodes as unknowns, hurting boundary learning. Conversely, a large η may be too conservative, failing to include enough unknown samples. Thus, we set $\eta = 0.9$ for DEAR to balance quantity and quality in unknown-class node selection for the target domain. **2) Sensitivity Analysis of μ and γ.** To evaluate the impact of loss weighting, we vary $\mu, \gamma \in \{0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$. We observe a consistent decline in performance as either parameter increases, underscoring the critical importance of balancing the source-domain supervised loss with the target-domain self-supervised and cross-domain contrastive losses. Given that the source domain provides reliable labeled data while pseudo-labels in the target domain are inherently noisy, reducing the weight of the latter is essential to mitigate the propagation of erroneous signals, improving overall robustness and stability. Consequently, we adopt the $\mu = \gamma = 0.1$ for DEAR to ensure robust and effective model training.

3.5 Visualization of Node Representation

To better understand how DEAR enhances GNN representation learning within OSGDA scenarios, we visualize the learned node embeddings (via t-SNE) of DEAR and baseline OSGDA methods in Figure 5. In the source domain, all methods exhibit relatively clear class separation. However, in the target domain, SDA and UAGA produce less discriminative representations, where unknown-class nodes are heavily entangled and poorly separated. In contrast, our DEAR yields more compact and clearly separated clusters for unknown classes, while also preserving clearer structures for known classes. This suggests that DEAR can better suppress the interference introduced by unknown nodes and maintain more discriminative target representations.

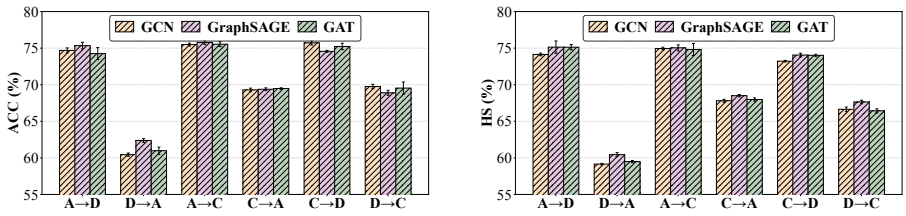
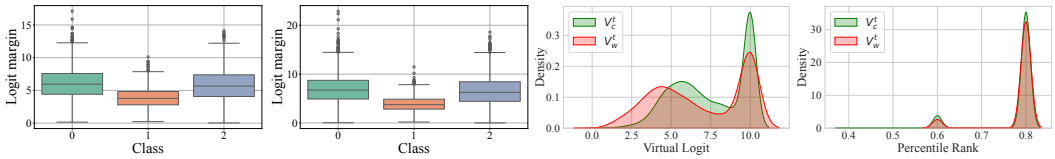


Figure 6: Backbone generalization of our proposed DEAR on six citation OSGDA tasks. Results are reported with three GNN backbones, including GCN, GraphSAGE, and GAT.



(a) Margin without W_{vir}. (b) Margin with W_{vir}. (c) Virtual logit value. (d) Percentile rank.

Figure 7: Effects of virtual class learning on A⇒C task. Figures (a) and (b) show logit margins of source nodes without and with virtual class injection, while figures (c) and (d) show virtual-logit value and percentile rank distributions for correctly classified (v_c^t) and misclassified (v_w^t) target nodes.

3.6 Backbone Generalization

To evaluate the backbone generalizability of our DEAR, we replace the default GCN encoder with GraphSAGE (Hamilton et al., 2017) and GAT (Veličković et al., 2018), which employ different neighborhood aggregation mechanisms. As shown in Figure 6, DEAR maintains consistently stable ACC and HS across all six transfer tasks under the three backbones, with only minor performance variations. Notably, GraphSAGE slightly improves performance on several tasks, while GAT remains comparable to the default GCN. These results indicate that the effectiveness of DEAR does not rely on a specific GNN architecture, demonstrating strong generalizability across different graph encoders.

3.7 The Investigation of Virtual Class

Following the theoretical analyses, we conduct experiments to empirically investigate the properties of the virtual class.**1) Margin Visualization**: We compute and visualize the decision margin (defined as the difference between the top-2 logits) for source-domain nodes trained both with and without virtual class injection. As illustrated in Figure 7a and Figure 7b, the logit margins are significantly larger when virtual classes are incorporated. This demonstrates that virtual class injection effectively increases the margin in the decision space and expands the decision boundary gaps. **2) Virtual Logit Stability**: We visualize the absolute value and percentage contribution of virtual logits for correctly and incorrectly classified target domain nodes (denoted as v_c^t and v_w^t). As shown in Figure 7c and Figure 7d, correctly classified nodes tend to have larger virtual logits, verifying the intra-class consistency property of virtual classes. This supports Theorem 2, showing that virtual logits serve as a reliable signal for high-confidence node identification.

4 Conclusion

In this paper, we further explore the problem of OSGDA. Motivated by the observation that existing methods often overlook the robustness of source domain representation learning, we propose a decision boundary-aware OSGDA framework named DEAR by injecting a virtual class into the representation learning process of the source domain, leading to state-of-the-art performance over several benchmark datasets. Our approach is validated not only through extensive experiments but also through theoretical analysis, which sheds light on the role of the virtual class in OSGDA and lays a solid foundation for future research in this area. Future work will explore adaptive unknown-class modeling and more challenging graph transfer scenarios.

AI USE STATEMENT

In this work, we used generative AI tools solely for language polishing of the manuscript and for no other purpose. All AI-polished content was reviewed and revised by the authors, and we take responsibility for the final content of this work.

REFERENCES

Ruichu Cai, Fengzhu Wu, Zijian Li, Pengfei Wei, Lingling Yi, and Kun Zhang. Graph domain adaptation: A generative view. *ACM Transactions on Knowledge Discovery from Data*, 18(3): 1–24, 2024.

Binghui Chen, Weihong Deng, and Haifeng Shen. Virtual class enhanced discriminative embedding learning. *Advances in Neural Information Processing Systems*, 31, 2018.

Jiuzhou Chen, Xiangyang Huang, and Shudong Zhang. Large-margin softmax loss using synthetic virtual class. *Neural Networks*, pp. 108068, 2025a.

Wei Chen, Guo Ye, Yakun Wang, Zhao Zhang, Libang Zhang, Daixin Wang, Zhiqiang Zhang, and Fuzhen Zhuang. Smoothness really matters: A simple yet effective approach for unsupervised graph domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 15875–15883, 2025b.

Wei Chen, Xingyu Guo, Shuang Li, Yan Zhong, Zhao Zhang, Fuzhen Zhuang, Hongrui Liu, Libang Zhang, Guo Ye, and HUIMEI HE. Learning structure-semantic evolution trajectories for graph domain adaptation. In *International Conference on Learning Representations*, volume 2026, pp. 109348–109368, 2026.

Ruiyi Fang, Changhong Chen, Qiang Wang, Charles Ling, Boyu Wang, and Hong Zhang. Transfer learning with graph neural networks to predict polymer solubility parameters. *npj Computational Materials*, 2026a.

Ruiyi Fang, Shuo Wang, Ruizhi Pu, Qiuhao Zeng, Hao Zheng, Ziyan Wang, Jiale Cai, Zhimin Mei, Song Tang, Charles Ling, et al. Graph domain adaptation via homophily-agnostic reconstructing structure. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 21047–21055, 2026b.

Zhen Fang, Jie Lu, Feng Liu, Junyu Xuan, and Guangquan Zhang. Open set domain adaptation: Theoretical bound and algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, 32(10):4309–4322, 2020.

Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.

Bo Fu, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Learning to detect open classes for universal domain adaptation. In *European Conference on Computer Vision*, pp. 567–583. Springer, 2020.

Xingbo Fu, Zihan Chen, Yinhan He, Song Wang, Binchi Zhang, Chen Chen, and Jundong Li. Virtual nodes can help: Tackling distribution shifts in federated graph learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 16657–16665, 2025.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pp. 1263–1272. Pmlr, 2017.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*, 30, 2017.

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

Clark M Ju, William Shiao, Zhichun Guo, Yanfang Ye, Yozen Liu, Neil Shah, and Tong Zhao. How does message passing improve collaborative filtering? *Advances in neural information processing systems*, 37:8760–8784, 2024.

Wei Ju, Xin Ma, Yifan Wang, Siyu Yi, Yusheng Zhao, Junyu Luo, Chenwei Tang, Ziyue Qiao, and Jiancheng Lv. Eta: Dual evidence-aware uncertainty learning for open-set graph domain adaptation. 2025a.

Wei Ju, Zhengyang Mao, Siyu Yi, Yifang Qin, Yiyang Gu, Zhiping Xiao, Jianhao Shen, Ziyue Qiao, and Ming Zhang. Cluster-guided contrastive class-imbalanced graph classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 11924–11932, 2025b.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

P. Langley. Crafting papers on machine learning. In Pat Langley (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Omri Lifshitz and Lior Wolf. Sample selection for universal domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 8592–8600, 2021.

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. *Advances in Neural Information Processing Systems*, 31, 2018.

Renqiang Luo, Huafei Huang, Tao Tang, Jing Ren, Ziqi Xu, Mingliang Hou, Enyan Dai, and Feng Xia. Fairge: Fairness-aware graph encoding in incomplete social networks. In *Proceedings of the ACM Web Conference 2026*, pp. 4541–4552, 2026.

Xudong Mao, Yun Ma, Zhenguo Yang, Yangbin Chen, and Qing Li. Virtual mixup training for unsupervised domain adaptation. *arXiv preprint arXiv:1905.04215*, 2019.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. *arXiv preprint arXiv:2002.05287*, 2020.

Jiayu Qin, Zhengquan Luo, Guy Tadmor, Changyou Chen, David Zeevi, and Zhiqiang Xu. Kgot: Unified knowledge graph and optimal transport pseudo-labeling for molecule-protein interaction prediction. In *International Conference on Learning Representations*, volume 2026, pp. 91277–91295, 2026.

Kuniaki Saito, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. Open set domain adaptation by backpropagation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 153–168, 2018.

Kuniaki Saito, Donghyun Kim, Stan Sclaroff, and Kate Saenko. Universal domain adaptation through self supervision. *Advances in Neural Information Processing Systems*, 33:16282–16292, 2020.

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

Xiao Shen, Zhihao Chen, Shirui Pan, Shuang Zhou, Laurence T Yang, and Xi Zhou. Open-set cross-network node classification via unknown-excluded adversarial graph domain alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20398–20408, 2025.

Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 990–998, 2008.

Yujun Tong, Dongliang Chang, Da Li, Xinran Wang, Kongming Liang, Zhongjiang He, Yi-Zhe Song, and Zhanyu Ma. Reserve to adapt: Mining inter-class relations for open-set domain adaptation. *IEEE Transactions on Image Processing*, 2025.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Yu Wang, Ronghang Zhu, Pengsheng Ji, and Sheng Li. Open-set graph domain adaptation via separate domain alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 9142–9150, 2024.

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

Liang Yang, Hui Ning, Jiaming Zhuo, Ziyi Ma, Chuan Wang, Wenning Wu, and Zhen Wang. Source-free graph foundation model adaptation via pseudo-source reconstruction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 27556–27564, 2026.

Nan Yin, Mengzhu Wang, Zhenghan Chen, Li Shen, Huan Xiong, Bin Gu, and Xiao Luo. Dream: Dual structured exploration with mixup for open-set graph domain adaption. In *International conference on learning representations*, volume 2024, pp. 19776–19797, 2024.

Kaichao You, Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Universal domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2720–2729, 2019.

Jing Zhang, Chao Wu, Chunyang Fan, and Jie Cui. Intrusion detection system for open network scenarios: A known/unknown attack detection method based on inter-class relationships. *IEEE Transactions on Information Forensics and Security*, 2026.

Xiaowen Zhang, Yuntao Du, Rongbiao Xie, and Chongjun Wang. Adversarial separation network for cross-network node classification. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 2618–2626, 2021.

Yuelin Zhang, Jiacheng Cen, Jiaqi Han, Zhiqiang Zhang, Jun Zhou, and Wenbing Huang. Improving equivariant graph neural networks on large geometric graphs via virtual nodes learning. In *Forty-first International Conference on Machine Learning*, 2024.

Zhen Zhang and Bingsheng He. Towards unsupervised open-set graph domain adaptation via dual reprogramming. In *Advances in Neural Information Processing Systems*, volume 38, pp. 53320–53343, 2025. doi: 10.52202/085713-1776.

APPENDIX: SUPPLEMENTARY OF DEAR

A PROOF OF THEOREM

In this section, we provide the derivations and proofs for Theorem 1 and Theorem 2.

A.1 PROOF OF THEOREM 1

According to the setting in Theorem 1, the classification margin of a node instance i is defined as:

$$\begin{aligned} m_i &:= \text{Logits}(X_i)_{y_i} - \max_{j \neq y_i} \text{Logits}(X_i)_j \\ &= W_{y_i}^\top F_\theta(X_i) - \max_{j \neq y_i} W_j^\top F_\theta(X_i), \end{aligned} \tag{1}$$

where $F_\theta(X_i)$ denotes the embedding of node i, and W_{y_i} is the classifier weight corresponding to its ground-truth class label y_i. After injecting a virtual class, we conservatively estimate the new margin:

$$\tilde{m}_i := \min\{m_i, l_{vir} - \max_{j \neq y_i} W_j^\top F_\theta(X_i)\}, \tag{2}$$

where $l_{vir} = \|W_{y_i}\| \cdot \|F_\theta(X_i)\|$ is the logit gained with the virtual class. By the Cauchy-Schwarz inequality, we have:

$$\begin{aligned} \|W_{y_i}\| \cdot \|F_\theta(X_i)\| &\geq W_{y_i}^\top F_\theta(X_i), \\ i.e., l_{vir} &\geq W_{y_i}^\top F_\theta X_i, \\ i.e., l_{vir} - \max_{j \neq y_i} W_j^\top F_\theta(X_i) &\geq \\ W_{y_i}^\top F_\theta(X_i) - \max_{j \neq y_i} W_j^\top F_\theta(X_i), & \end{aligned} \tag{3}$$

it follows that $\tilde{m}_i \geq m_i$. Let $\Delta m := \tilde{m}_i - m_i$, then by conservative estimation, $0 \leq \Delta m$. If we have $\tilde{m}_i = l_{vir} - \max_{j \neq y_i} W_j^\top F_\theta(X_i)$, the upper bound of Δm is:

$$\begin{aligned} \Delta m &\leq l_{vir} - W_{y_i}^\top F_\theta(X_i) \\ &= \|W_{y_i}\| \cdot \|F_\theta(X_i)\| \\ &\quad - \|W_{y_i}\| \cdot \|F_\theta(X_i)\| \cdot \cos\Theta \\ &= \|W_{y_i}\| \cdot \|F_\theta(X_i)\| \, (1 - \cos\Theta), \end{aligned} \tag{4}$$

where Θ is the angle between the classifier weight vector and the node embedding.

Assuming the classifier and the embedding function satisfy Lipschitz continuity, i.e., $\|W_c\| \leq L_w$, and $\|F_\theta(X_i)\| \leq L_x$, then:

$$\Delta m \leq L_w \cdot L_x \cdot (1 - \cos\Theta). \tag{5}$$

Therefore, we conclude:

$$0 \leq \Delta m_i \leq L_w \cdot L_x \cdot (1 - \cos\Theta). \quad \blacksquare$$

This theorem demonstrates that injecting a virtual class into the source domain is guaranteed to preserve or enlarge the classification margin. When the virtual logit is incorporated into the cross-entropy loss, the model can approach the upper bound of this margin improvement. From the expression of the margin gain, we observe that: If the node embedding and its corresponding classifier anchor vector are already aligned (i.e., small Θ), then the gain Δm is small, which is acceptable since the classification is already accurate. Conversely, if the angle Θ is large (i.e., the embedding and classifier vectors are poorly aligned), then the margin gain becomes significant. In such cases, the virtual class acts as an auxiliary force, encouraging the node to align with its correct class anchor, shrinking the decision boundary and increasing the inter-class separation.

A.2 Proof of Theorem 2

Following the same setting as in Theorem 1, for a certain instance i belonging to a specific class, the virtual logit after injecting a virtual class is given by $l_{vir} = \|W_{y_i}\| \cdot \|F_\theta(X_i)\|$. Let $\|W_c\| = \mho$, then the variance of virtual logit is:

$$\begin{aligned} \mathcal{V}(l_{vir}) &= \mathcal{V}(\|W_{y_i}\| \cdot \|F_\theta(X_i)\|) \\ &= \|W_c\|^2 \cdot \mathcal{V}(\|F_\theta(X_i)\|) \\ &= \mho^2 \cdot \mathcal{V}(\|F_\theta(X_i)\|), \end{aligned} \tag{6}$$

since the embedding of a node satisfies the Lipschitz condition, i.e., $||W_c|| \leq L_w$, which implies $\mho \leq L_w$, we have:

$$\mathcal{V}(l_{vir}) \leq L_w^2 \cdot \mathcal{V}(\|F_\theta(X_i)\|). \quad \blacksquare \tag{7}$$

This theorem indicates that if the classifier weight of the corresponding class remains approximately constant during training (a condition that is easily satisfied in practice), then the virtual logit obtained after injecting a virtual class is solely controlled by the norm of the embedding and exhibits a linear scaling behavior. Consequently, the virtual logits computed via the injected virtual class demonstrate strong stability and intra-class consistency, making them effective for identifying high-confidence known-class nodes.

B Related Work

Graph Domain Adaptation (GDA) aims to transfer knowledge from a source domain to a related but distinct target domain (Fang et al., 2026b; Chen et al., 2026), can generally be categorized into two types: methods based on predefined statistical metrics such as Maximum Mean Discrepancy (Gretton et al., 2012; Shen et al., 2020), and adversarial-based methods (Yang et al., 2026; Fang et al., 2026a). Open-Set Domain Adaptation (OSDA) is designed to tackle the challenge that target domain contains classes which are absent in source domain (Fang et al., 2020), and several methods have recently been proposed in the image domain (Saito et al., 2018; You et al., 2019; Saito et al., 2020; Lifshitz & Wolf, 2021). Nevertheless, they often overlook the unique topological properties and structural dependencies of graph data when applied to graph-structured domains, resulting in limited feature alignment across domains and reduced transfer effectiveness. OSGDA was recently proposed and has attracted increasing attention (Wang et al., 2024; Shen et al., 2025). Different from existing OSGDA methods that neglect the robustness of decision boundary in the source domain, we optimize the representation structure of the source model by compressing its decision space to enable more robust separation between known classes and unknown classes in the target domain.

Virtual Class Learning was first introduced by (Chen et al., 2018) through a simple yet effective Virtual Softmax. By incorporating dynamic virtual negatives, their method improves recognition by enlarging inter-class margins and reducing intra-class variance. Current approaches to virtual class learning typically follow two directions: one treats virtual classes as regularization terms to enhance model robustness (Zhang et al., 2024; Fu et al., 2025; Tong et al., 2025; Chen et al., 2025a; Zhang et al., 2026), while the other employs them as transitional classes between source and target domains to facilitate cross-domain knowledge transfer (Mao et al., 2019; Yin et al., 2024). However, few of them explore this idea in OSGDA tasks, where virtual classes are inherently suited for modeling unknown classes. In DEAR, we exploit the semantic information encoded in virtual classes and utilize it to mine intra- and inter-class relationships between known and unknown categories, facilitating subsequent cross-domain knowledge alignment and improving the robustness of the model.

C Time Complexity

In DEAR, assume we use an L-layer GCN as the feature extractor, then the time complexity for extracting graph features into embeddings is $O(l|E^s|D^2 + l|V^s|D^2 + l|E^t|D^2 + l|V^t|D^2)$, which can be simplified to $O(|E^s| + |E^t|)$ since $|E| \gg |V|$. For the boundary structure learning, the time complexity is $O(|V^s|)$. During the mining class relations for separation phase, the time complexity is $O(|V^t|)$. For the cross domain learning via perturbing, the time complexity for modifying the graph and performing cross-domain contrastive learning is $O(|V^s| + 2|V^t| \times |E^t|)$ and $O(|V^s| \times |V^t|)$, so the time complexity for this phase is $O(|V^s| + 2|V^t| \times |E^t| + |V^s| \times |V^t|)$. Finally, the time

complexity for DEAR is $O(|E^s| + |E^t| + |V^s| + |V^t| + 2|V^t| \times |E^t| + |V^s| \times |V^t|)$, which can be simplified to $O(|E^s| + |E^t| + |V^t| \times |E^t| + |V^s| \times |V^t|)$.

D ALGORITHM FOR DEAR

Our proposed DEAR can be summarized in Algorithm 1. And our code for DEAR is available at: https://anonymous.4open.science/r/DEAR-0553.

Algorithm 1 The overall training process of DEAR

Input: Labeled source graph $G^s = \{V^s, E^s, X^s, A^s, Y^s\}$ and unlabeled target graph $G^t = \{V^t, E^t, X^t, A^t\}$ with $C_s \subset C_t$.

Output: Parameters of graph feature extractor, and virtual-class oriented open-set classifier.

while not converged or epoch ≤ 200 **do**
  Extract graph feature to get $\boldsymbol{Z}^*, * \in \{s, t\}$; `// Eq. 1`
  Inject virtual class to open-set classifier ; `// Eq. 2`
  `/* Virtual-class-oriented Boundary Structure Learning */`
  Distinguish high-confidence unknown-class and known-class nodes with critical threshold and virtual logit ; `// Eq. 5-12`
  `/* Mining Class Relations for Separation */`
  Calculate $\mathcal{L}_t$ with high-confidence nodes ; `// Eq. 13`
  Modify the source graph and target graph with perturbations ; `// Eq. 14-15`
  Consistency learning between domains using modified graphs for domain alignment ; `// Eq. 16`
  `/* Cross Domain Learning via Perturbing */`
end while
Obtain graph feature extractor and virtual-class oriented open-set classifier for OSGDA.

E ADDITIONAL DETAILS OF EXPERIMENTS

In this section, we provide a detailed supplementary explanation of the datasets used and the analysis performed in our experiments to further validate the effectiveness and superiority of our DEAR.

E.1 PERFORMANCE EXPERIMENT

To demonstrate the superiority of our method, we conduct a series of OSGDA performance evaluation experiments across multiple datasets, comparing DEAR against various baselines. Here, we provide the complete experimental results in citation datasets, including the values for all four evaluation metrics, as shown in Table 1 and Table 2. From the overall experimental data, it can be observed that despite the challenging settings where each experiment represents a novel OSGDA task, DEAR consistently achieves stable and promising results. Notably, the ACC_k and ACC_{uk} are both high and balanced across all tasks. This balance significantly contributes to the improvement in the H-score, where DEAR outperforms existing state-of-the-art OSGDA approaches. This finding further validates that DEAR by enhancing model robustness through source domain decision boundary optimization enables the learning of stable representations for known and unknown classes.

E.2 MORE DISCUSSION OF ABLATION STUDY

To verify the contribution of each component in DEAR, we conduct ablation studies by systematically removing five different parts of the model. The detailed results are presented in Figure1. As consistent with previous analysis, the removal of any single component results in noticeable performance degradation, which strongly confirms the effectiveness of each part of DEAR.

E.3 MORE EXPLOITATION OF VIRTUAL CLASS

We theoretically explored the properties of virtual classes before. In the experimental section, we further conduct empirical studies to validate these properties. The results on five additional dataset

Table 1: Classification ACC and HS of 10 performance experiments with different unknown classes on six OSGDA tasks in the citation datasets (%).

C_{uk}	A⇒D		D⇒A		A⇒C		C⇒A		C⇒D		D⇒C	
	ACC	HS	ACC	HS	ACC	HS	ACC	HS	ACC	HS	ACC	HS
$\{0,1\}$	74.69	74.14	60.44	59.15	75.49	74.95	69.29	67.81	75.73	73.22	69.75	66.64
$\{0,2\}$	63.71	63.05	56.27	56.16	70.52	69.68	58.23	58.44	61.49	61.11	69.88	66.19
$\{0,3\}$	53.87	55.92	55.30	57.79	58.82	58.14	58.06	59.33	58.97	58.41	65.18	67.17
$\{0,4\}$	55.85	54.63	59.83	60.50	64.82	64.02	61.73	62.18	63.68	64.64	66.69	63.91
$\{1,2\}$	72.56	67.80	69.84	57.45	74.37	72.73	72.24	64.72	79.74	77.85	76.07	72.61
$\{1,3\}$	68.95	67.74	57.64	49.41	72.56	69.61	65.41	64.50	72.06	70.02	66.63	60.29
$\{1,4\}$	72.87	72.76	59.42	57.82	73.25	72.47	64.88	62.15	64.97	62.27	64.94	64.87
$\{2,3\}$	50.05	51.75	64.75	66.87	59.88	59.75	61.25	63.23	57.04	58.32	70.11	71.28
$\{2,4\}$	55.11	55.74	57.84	58.28	58.23	58.68	58.59	59.02	61.11	60.56	62.29	60.92
$\{3,4\}$	57.99	62.52	61.24	64.46	62.98	66.99	57.69	60.92	58.92	62.05	63.63	67.53
Average	62.57	62.61	60.26	58.79	67.09	66.70	62.74	62.23	65.37	64.85	67.52	66.14

Table 2: Classification ACC of known nodes (ACC_k) and unknown nodes (ACC_u) in the 10 performance experiments with different unknown classes in the citation datasets (%).

C_{uk}	A⇒D		D⇒A		A⇒C		C⇒A		C⇒D		D⇒C	
	ACC_k	ACC_u	ACC_k	ACC_u	ACC_k	ACC_u	ACC_k	ACC_u	ACC_k	ACC_u	ACC_k	ACC_u
$\{0,1\}$	70.50	78.17	51.61	69.25	69.91	80.79	59.18	79.39	64.07	85.41	55.59	83.18
$\{0,2\}$	54.53	74.17	48.71	66.29	61.45	80.45	56.43	60.61	53.83	70.66	52.86	88.51
$\{0,3\}$	41.27	86.71	45.41	79.44	43.54	91.84	54.69	66.27	59.64	57.24	56.84	82.08
$\{0,4\}$	41.40	80.31	53.52	69.57	53.90	78.82	59.70	64.86	55.06	78.25	50.27	87.72
$\{1,2\}$	56.48	84.78	40.91	96.43	62.43	87.08	49.61	93.05	70.70	86.61	59.09	94.15
$\{1,3\}$	72.28	63.74	69.55	38.32	77.24	63.36	67.82	61.48	77.18	64.07	74.84	50.48
$\{1,4\}$	74.64	70.98	48.97	70.56	77.30	68.21	76.79	52.20	51.08	79.75	58.03	73.54
$\{2,3\}$	46.58	58.21	59.59	76.18	60.07	59.43	56.49	71.78	54.77	62.36	68.08	74.80
$\{2,4\}$	50.90	61.60	53.70	63.71	52.64	66.28	55.10	63.53	48.53	80.52	48.30	82.46
$\{3,4\}$	52.96	76.29	54.84	78.18	53.92	88.41	48.05	83.21	55.98	69.60	56.00	85.04

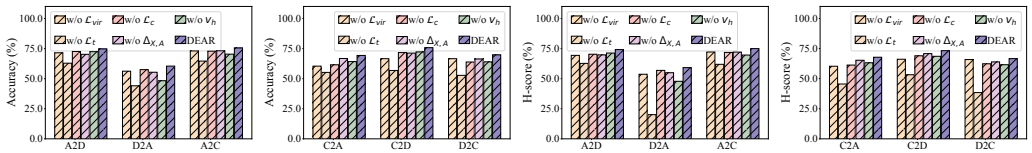
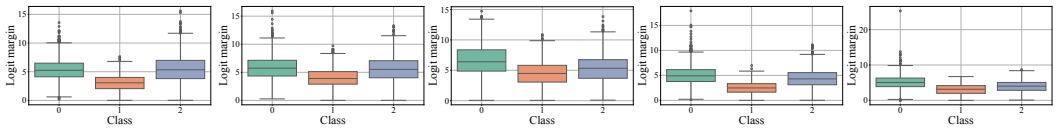


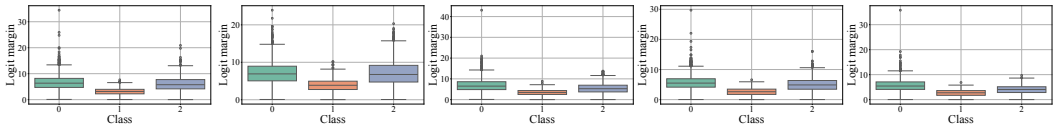
Figure 1: Visualization of the ablation study results across six OSGDA tasks on citation datasets. The horizontal axis represents the tasks, while the vertical axis denotes the performance metrics (Accuracy and H-score). Each column corresponds to a specific ablation variant.

pairs are presented in Figure 2, Figure 3, Figure 4 and Figure 5. The first experiment focuses on decision boundary visualization. As shown in the Figure 2 and Figure 3, across nearly all domain adaptation tasks, the decision boundaries with virtual class injection are consistently larger than those without. This empirically supports Theorem 1, demonstrating that introducing virtual classes compresses the decision space of known source classes, thereby enlarging inter-class margins and leaving space for recognizing unknown classes. The second experiment investigates the intra-class consistency of virtual classes. The results in Figure 4 and Figure 5 show that correctly classified nodes (denoted as v_c^t) tend to have higher corresponding virtual logits, whereas misclassified nodes (denoted as v_w^t) exhibit lower virtual logits. This aligns with Theorem 2, confirming that virtual classes capture stable intra-class information, which can help identify high-confidence nodes. These two experiments provide strong empirical support for our theoretical findings on virtual classes, further validating the rationality and effectiveness of our proposed DEAR.



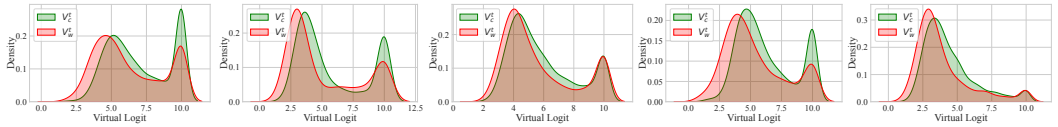
(a) C2A (b) A2D (c) D2A (d) C2D (e) D2C

Figure 2: Decision margins on each known class for the remaining OSGDA tasks without virtual class injection, where class 0/1 are treated as unknown classes. The vertical distance represents the logits margin, which corresponds to the width of the decision boundary.



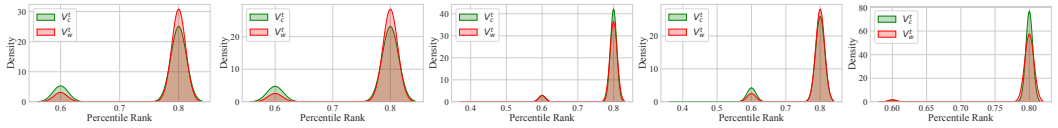
(a) C2A (b) A2D (c) D2A (d) C2D (e) D2C

Figure 3: Decision margins on each known class for the remaining OSGDA tasks with virtual class injection, where class 0/1 are treated as unknown classes.



(a) C2A (b) A2D (c) D2A (d) C2D (e) D2C

Figure 4: Distribution of virtual logit values for correctly classified and misclassified target domain nodes, where class 0/1 are treated as unknown classes.



(a) C2A (b) A2D (c) D2A (d) C2D (e) D2C

Figure 5: Distribution of virtual logit percentile rank for correctly classified and misclassified target domain nodes, where class 0/1 are treated as unknown classes.

F Datasets and Baselines

Datasets. We evaluate DEAR on three groups of graph benchmarks. The first comprises the citation networks ACMv9, Citationv1, and DBLPv7 (Tang et al., 2008), while the additional groups are ogbn-arxiv (Hu et al., 2020) and WebKB (Pei et al., 2020). Together, they cover transfer between citation networks, temporal shifts on a larger citation graph, and transfer between heterophilous webpage graphs. For each task, we remove nodes of designated unknown classes from the source graph and retain them in the target graph under a single unknown label. On ACMv9, Citationv1, and DBLPv7, each graph has five classes, following SDA (Wang et al., 2024), we designate two classes as unknown and evaluate all 10 possible class combinations for each of the six transfer directions. For ogbn-arxiv, we designate the first 10 of the 40 classes as unknown; for WebKB, we designate two of the five classes as unknown. We train for 200 epochs per run and report mean performance across the 10 configurations for the citation networks. For each ogbn-arxiv and WebKB task, we report the mean and standard deviation over five runs. Table 3 summarizes the node counts, edge counts, feature dimensions, and numbers of classes for all datasets. The descriptions of datasets are as below.

- **Citation graphs.** ACMv9 (A) is a dataset derived from the Association for Computing Machinery (ACM), containing papers on artificial intelligence, software engineering, machine learning, and

Table 3: Statistics of three groups of benchmark datasets.

Dataset	Nodes	Edges	Feats.	Classes
ACMv9	9,360	15,602	5,571	5
Citationv1	8,935	15,113	5,379	5
DBLPv7	5,484	8,130	4,412	5
ogbn-arxiv	169,343	1,166,243	128	40
Cornell	183	298	1,703	5
Texas	183	325	1,703	5
Wisconsin	251	515	1,703	5

data mining. Citationv1 (C) is a subset of the Microsoft Academic Graph (MAG), comprising academic papers across several fields, including artificial intelligence. DBLPv7 (D) is a subset of the DBLP (Digital Bibliography & Library Project), a major bibliographic database covering diverse research areas. It provides publication metadata, including authors, journals, and conferences. In these benchmark graphs, nodes represent papers, and edges represent citation relationships treated as undirected links. Node features are bag-of-words representations of paper titles, with each dimension corresponding to a vocabulary term, and we unify them in the same dimensions. Each graph contains five research topic classes, and node labels indicate the research field of each paper.

- **Ogbn-arxiv.** It is a part of the Open Graph Benchmark, contains computer science papers published on arXiv and indexed by the Microsoft Academic Graph (Hu et al., 2020). To construct temporal transfer tasks, we partition these papers by publication time into three domains: Arxiv I, II, and III (Arx.I, Arx.II, and Arx.III). Each node represents a paper, and each directed edge indicates that one paper cites another. Node features are 128-dimensional vectors obtained by averaging the word embeddings in each paper's title and abstract. And the label denotes one of 40 arXiv subject areas.
- **WebKB.** It contains webpages collected from computer science departments at different universities. We use its Cornell (Cor), Texas (Tex), and Wisconsin (Wis) datasets, each corresponding to webpages from the respective university (Pei et al., 2020). The nodes represent individual webpages, and edges indicate hyperlinks connecting them. Node features are 1,703-dimensional bag-of-words vectors derived from webpage text, with each dimension corresponding to a vocabulary term. Each webpage has one of five labels describing its role: student, project, course, staff, or faculty.

Baselines. We compare DEAR with 12 baselines covering five methodological categories: $\star$ classical GNNs (GCN, GraphSAGE); $\star$ unsupervised domain adaptation (UDA) methods (DANN, CDAN); $\star$ open-set domain adaptation (OSDA) methods (OSBP, DANCE); $\star$ closed-set graph domain adaptation (CSGDA) methods (UDAGCN, ASN); and $\star$ OSGDA methods (SDA, UAGA, ETA, GraphRTA). A brief description of each method is shown as follows:

- **GCN** (Kipf & Welling, 2017): It uses a localized first-order graph convolution to aggregate neighboring node features. By combining local graph structure with node attributes, it learns representations for semi-supervised node classification from a limited set of labeled nodes.
- **GraphSAGE** (Hamilton et al., 2017): This inductive framework learns an embedding function by sampling neighbors and aggregating their features. Because it does not assign a separate learned embedding to every node, it can generate representations for unseen nodes and graphs.
- **DANN** (Ganin et al., 2016): It trains a label predictor and a domain discriminator using labeled source data and unlabeled target data. A gradient reversal layer encourages features that remain discriminative for the source task while reducing the distinction between domains.
- **CDAN** (Long et al., 2018): It conditions adversarial adaptation on feature representations and classifier predictions. Multilinear conditioning captures their interactions, while entropy conditioning gives greater weight to confident predictions during cross-domain alignment.
- **OSBP** (Saito et al., 2018): This open-set domain adaptation method uses adversarial training to handle target classes absent from the source. Its feature generator can align target samples with known source classes or separate them as unknown, avoiding indiscriminate alignment.

- **DANCE** (Saito et al., 2020): It handles arbitrary category overlap through self-supervised neighborhood clustering in the target domain. Entropy-based alignment and rejection associate target features with source classes or identify them as unknown.
- **UDAGCN** (Wu et al., 2020): It combines local and global graph convolutions through attention to represent nodes across graphs. Its domain adaptation module jointly optimizes source classification, domain discrimination, and target classification objectives for cross-graph node prediction.
- **ASN** (Zhang et al., 2021): It separates domain-shared from domain-private node information using a shared encoder and graph-specific encoders. Each encoder combines local and global structure, while adversarial adaptation reduces discrepancies between the two networks.
- **SDA** (Wang et al., 2024): It divides target nodes into certain and uncertain groups according to prediction entropy. The method aligns certain target nodes with the source adversarially and applies neighbor-center clustering to the uncertain group to separate known from unknown classes.
- **UAGA** (Shen et al., 2025): It separates known and unknown target nodes with an adversarial encoder and neighborhood-aggregation classifier. Unknown-excluded alignment uses positive and negative coefficients to align known nodes with the source and push unknown nodes away.
- **ETA** (Ju et al., 2025a): It combines a dual-branch graph encoder with evidential deep learning to estimate class probabilities and uncertainty. The method uses cross-domain neighborhoods and virtual samples to guide domain alignment while identifying unknown target nodes.
- **GraphRTA** (Zhang & He, 2025): It reprograms target graph features and structure while pruning domain-specific model parameters. It also adds an unknown-class output to the classifier, enabling open-set recognition without a manually specified rejection threshold.

Evaluation Metrics. We report overall accuracy (ACC) and H-score (HS) results for our OSGDA tasks. ACC measures the overall prediction accuracy on the target domain, while HS is defined as the harmonic mean of the known-class accuracy (ACC_k) and unknown-class accuracy (ACC_u) (Fu et al., 2020), which is defined as:

$$\text{HS} = \frac{2 \times \text{ACC}_k \times \text{ACC}_u}{\text{ACC}_k + \text{ACC}_u}.$$

Compared with ACC alone, HS provides a more balanced evaluation of the model's ability to correctly classify known classes while identifying unknown classes, preventing performance on either group from dominating the overall assessment.