
Out-of-Distribution Generalized Graph Anomaly Detection with Homophily-aware Environment Mixup

Sibo Tian¹, Xin Wang^{1*}, Zeyang Zhang¹, Haibo Chen¹, Wenwu Zhu^{1*}

¹Tsinghua University

Abstract

Graph anomaly detection (GAD) is widely prevalent in scenarios such as financial fraud detection, anti-money laundering and social bot detection. However, structural distribution shifts are commonly observed in real-world GAD data due to selection bias, resulting in reduced homophily. Existing GAD methods tend to rely on homophilic shortcuts when trained on high-homophily structures, limiting their ability to generalize well to data with low homophily under structural distribution shifts. In this study, we propose to handle structural distribution shifts by generating novel environments characterized by diverse homophilic structures and utilizing invariant patterns, *i.e.*, features and structures with the capability of stable prediction across structural distribution shifts, which face two challenges: (1) How to discover invariant patterns from entangled features and structures, as structures are sensitive to varying homophilic distributions. (2) How to systematically construct new environments with diverse homophilic structures. To address these challenges, we propose Ego-Neighborhood Disentangled Encoder with **H**omophily-aware **E**nvironment **M**ixup (**HEM**), which effectively handles structural distribution shifts in GAD by discovering invariant patterns. Specifically, we first propose an ego-neighborhood disentangled encoder to decouple the learning of feature and structural embeddings, which facilitates subsequent improvements in the invariance of structural embeddings for prediction. Next, we introduce a homophily-aware environment mixup that dynamically adjusts edge weights through adversarial learning, effectively generating environments with diverse structural distributions. Finally, we iteratively train the classifier and environment mixup via adversarial training, simultaneously improving the diversity of constructed environments and discovering invariant patterns under structural distribution shifts. Extensive experiments on real-world datasets demonstrate that our method outperforms existing baselines and achieves state-of-the-art performance under structural distribution shift.

1 Introduction

Graph anomaly detection (GAD) [1] represents a classical task in graph machine learning, with extensive applications in financial fraud detection[2], anti-money laundering[3], and social bot detection[4, 5]. Typically, GAD can be formulated as a semi-supervised node classification problem[6], where the objective is to identify anomalous nodes in the input graph. Graph neural networks (GNNs) have demonstrated superior predictive capabilities in GAD tasks due to their ability to handle complex topological relationships and model sophisticated node representations[7].

However, structural distribution shifts[8–11] are prevalent in real-world graph anomaly detection scenarios due to factors such as selection bias[12]. For example, in anti-money laundering contexts[13], coordinated money laundering accounts are more likely to be identified and labeled compared to

*Corresponding authors. {xin_wang, wwzhu}@tsinghua.edu.cn

isolated ones, while large-scale coordinated bot comments on social platforms are more easily detected by automated systems. This results in a structural distribution shift between training and test data, as nodes with stronger homophily and more connections to similar nodes are more likely to be labeled. Existing GNNs for GAD methods trained on high-homophily structures tend to rely on the homophilic shortcut, limiting their ability to generalize effectively to test data with low homophily under structural distribution shifts.

In this paper, we study the problem of handling structural distribution shifts in graph anomaly tasks by generating novel environments characterized by diverse homophilic structures and utilizing invariant patterns, *i.e.*, features and structures with the capability of stable prediction across structural distribution shifts, which remains largely unexplored in the literature. However, this problem is highly nontrivial with the following challenges:

- How to discover invariant patterns from entangled features and structures, as structures are sensitive to varying homophilic distributions.
- How to systematically construct new environments with diverse homophilic structures.

To address these challenges, we propose a novel framework named Ego-Neighborhood Disentangled Encoder with **Homophily-aware Environment Mixup (HEM)** to discover invariant patterns with stable predictive abilities under structural distribution shifts. Specifically, we first propose an ego-neighborhood disentangled encoder to separately model the feature and structure representations. By this design, we can (1) disentangle feature information and structure information, enabling the following modules to strengthen the invariance of the prediction patterns. (2) model the interaction patterns between ego-node and neighborhood to better predict the anomaly probability. Then, we propose a homophily-aware environment mixup, which dynamically adjusts edge weights within the graph to implicitly modify the structure of ego graphs. This mechanism generates diverse training environments and facilitates the learning of invariant patterns in a memory-efficient manner. Finally, the two modules are trained iteratively in an adversarial manner, in order to improve the diversity of the constructed environment and discover invariant patterns under structural distribution shifts. Extensive experiments on real-world datasets demonstrate that our proposed method achieves state-of-the-art performance in GAD under structural distribution shift scenarios.

The contributions of our work can be summarized as follows:

- We propose Ego-Neighborhood Disentangled Encoder with **Homophily-aware Environment Mixup (HEM)**, which can handle structural distribution shift in graph anomaly detection, which is largely unexplored in literature.
- We propose an ego-neighborhood disentangled encoder to disentangle feature and structure representations for learning invariant patterns, which is a general framework that can be utilized to improve model performance for many GNNs. Besides, we propose a homophily-aware environment mixup to efficiently generate training environments with diverse local structures to improve the model’s generalization capability.
- Experiments on real-world datasets demonstrate that our method achieves state-of-the-art performance compared to existing baselines.

2 Problem Formulation

In this section, we formulate the problem of graph anomaly detection under distribution shift.

Node-Attribute Graph A node-attribute graph G consists of a node set V , an edge set E , and node attributes $X \in R^{n \times d}$, where each row of X represents the feature vector of the corresponding node. The edges between nodes are represented by the adjacency matrix $A \in R^{n \times n}$, where n is the number of nodes and d is the dimension of the input features. The node labels are defined as $Y \in R^{n \times 1}$. Thus, a node-attribute graph can be summarized as $G = \{A, X, Y\}$. In node-level graph anomaly detection, the labels are binary variables indicating whether a node is normal or anomalous.

Node-Level Graph Anomaly Detection (GAD) Node-level graph anomaly detection is an imbalanced binary classification task. The goal of the proposed model is to predict whether a node is normal or anomalous. In this paper, we focus on the transductive setting, while our proposed

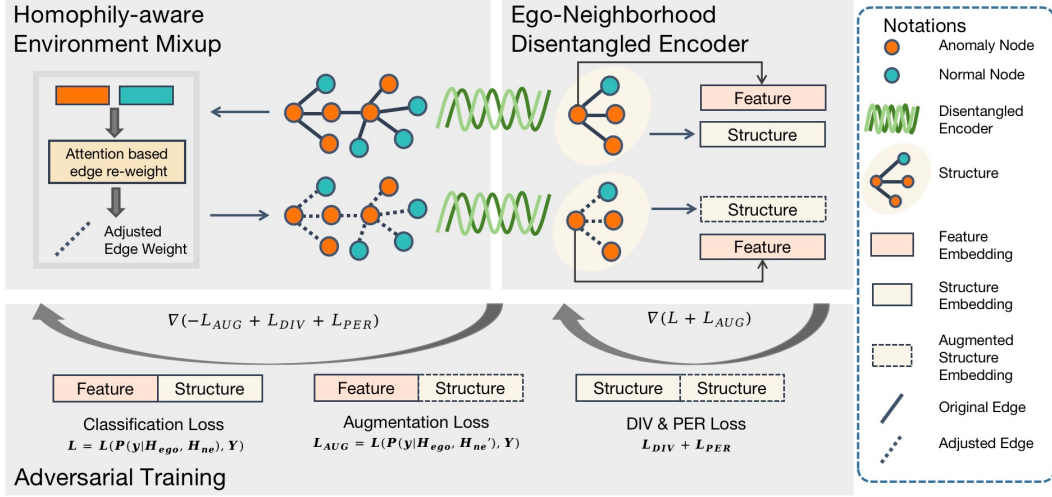


Figure 1: The overall framework of our proposed **HEM**. The framework transforms an original graph into an augmented graph by dynamically adjusting edge weights with homophily-aware environment muxup. Both graphs are processed by the ego-neighborhood disentangled encoder to extract feature and structural representations. The original loss L and augmented loss L_{AUG} are computed using cross-entropy with the disentangled representations, while the diversity loss L_{DIV} measures the difference between original and augmented structural embeddings, and the PER loss L_{PER} constrains the proportion of perturbed edges to avoid trivial solutions. These losses jointly optimize the encoder and environment muxup, enhancing generalization across varying structural distributions.

method can be simply applied in inductive scenarios. Following [14] [15] [16] [17], we formulate this problem as predicting the probability to be an anomaly of a node given its induced ego-graph, i.e., $p(Y_v|G_v)$. A k -order ego-graph G_v induced from v is a subgraph containing node v with all of its k -hop neighbors, k is an arbitrary integer. The optimization objective is to learn an optimal predictor to model the node’s anomaly probability

$$\min_{\theta} \mathbb{E}_{(y_v, G_v) \sim p(y_v, G_v)} L(f_{\theta}(G_v), y_v) \quad (1)$$

where f_{θ} is a graph neural network parameterized by θ . Denote random variable of ego-graph and node label as G_v, y_v , and respective instances as G_v, y_v .

Structural Distribution Shift Structural distribution shift in node-level tasks can be treated as a covariate shift, which means the ego-graph distribution is different across training and test data, i.e. $p_{train}(G_v) \neq p_{test}(G_v)$, leading to different local structures. Therefore, we formulate this problem as an out-of-distribution (OOD) problem,

$$\min_{\theta} \max_{e \in \mathcal{E}} \mathbb{E}_{(y_v, G_v) \sim p(y_v, G_v | e=e)} [L(f_{\theta}(G_v), y_v) | e]. \quad (2)$$

In GAD tasks, structural distribution shift is a common problem[12], due to factors like sampling bias. For example, in e-commerce networks, accounts that transact with anomalous accounts are more likely to be anomalous themselves, potentially indicating participation in money laundering transactions. This will cause a structural distribution shift that ego-graphs in the training set show a higher homophily ratio, i.e. the proportion of neighbors sharing the same label as the central node is much higher than that in the test set. The homophily ratio is defined as $\text{Homo}(v) = \frac{1}{|N(v)|} |u \in N(v) | Y_u = Y_v|$. The ego-graphs with a high homophily ratio in the training stage may lead GNNs to rely on the homophilic shortcut, a spurious pattern that predicts nodes’ labels by the majority label in their neighborhood, and finally to fail in the test stage ego-graphs with low homophily ratio.

3 Method

In this section, we propose the **HEM**, a novel framework for graph anomaly detection under structural distribution shift by introducing two key modules, ego-neighborhood disentangled encoder and homophily-aware environment mixup.

3.1 Ego-Neighborhood Disentangled Encoder

It’s critical to utilize both knowledge from ego-node and neighborhood to extract invariant patterns in graph anomaly detection for two reasons. (1) The interaction between the ego node and neighborhood indicates the discrepancy, which is important for predicting anomalies. (2) Excessive reliance on structural information results in unstable predictions under structural distribution shifts. Encoding both feature and structure information with one encoder makes the variant and invariant patterns entangled, leading to a performance drop under distribution shift. Most existing works simply employ a single GNN as an encoder, leading to fail in test data with distribution shift. While CoLA[15] models anomaly probability via local discrepancy, its method faces a trade-off between sampling size and speed, resulting in suboptimal performance.

To overcome these limits, we propose an ego-neighborhood disentangled encoder to efficiently and effectively extract feature and structural patterns from ego graphs in a decoupled manner.

Neighborhood Encoder We define the k -hop neighborhood as the ego-graph for a given node. The neighborhood encoder can be implemented using any commonly used message-passing-based GNN architecture. In our framework, we adopt the BWGNN[18], a strong backbone in GAD tasks, as the neighborhood encoder.

Given the initial node features X and the adjacency matrix A , we first employ an Multi-Layer Perceptron (MLP) projector to project it into the embedding space.

$$H_0 = \text{Proj}(X) \quad (3)$$

we can generally compute the neighborhood embeddings as follows

$$H_{ne} = GNN(H_0, A) \quad (4)$$

Specifically, with BWGNN applied, the neighborhood embeddings can be modeled as

$$H_{ne} = \sum_{k=0}^K w_k \hat{A}^k H_0 \quad (5)$$

where $\hat{A}$ is the normalized adjacency matrix, i.e. $\hat{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$, D is the diagonal matrix. w_k is the k th coefficient from a Beta polynomial.

Ego-Node Encoder The relationship between ego-node’s feature and label is stable under structural distribution shift since ego-nodes only contain information about a single instance. We disentangle them from structural patterns so that we can model the anomaly probability by the interaction patterns. Specifically, we employ an MLP to encode the ego-node’s representation. To align the neighborhood embedding space and ego-node embedding space, we utilize the same projector as the neighborhood embedding projector

$$H_{ego} = H_0 = \text{Proj}(X). \quad (6)$$

The sharing projector ensures that the embeddings generated by the two encoders are aligned within the same embedding space, facilitating effective interaction between the ego node and its neighborhood embeddings.

Contrastive Prediction To capture the discrepancy between the ego-node embedding and structural embedding, we employ a bilinear function to predict anomaly probability. This design is commonly employed by contrastive learning methods [15, 19].

$$P(y | h_{ego}, h_{ne}) = \sigma(h_{ego}^T W h_{ne}), \quad (7)$$

where σ denotes the sigmoid function, and W is a learnable weight matrix that models the interaction between the ego node and structural embeddings.

The classification loss can be computed as

$$L = \mathcal{L}(P(y \mid H_{\text{ego}}, H_{\text{ne}}), Y) \quad (8)$$

where $\mathcal{L}$ is cross-entropy loss.

3.2 Homophily-aware Environment Mixup

Existing out-of-distribution graph anomaly detection methods suffer from issues like reliance on labeled and diverse environments[20, 21], high memory cost[14], or limited performance due to lack of supervision[22].

To overcome these limitations, we propose a homophily-aware environment mixup, which generates diverse environments exhibiting effective structural distribution shifts by dynamically re-weighting edges based on the homophily observed among neighboring nodes. This process is both computationally and memory efficient.

Homophily-aware environment mixup We propose a homophily-aware environment mixup, to generate environments with structural distribution shift and improve the generalization capability of the classifier across environments with different local structures. Our objective for this module is to generate new environments with diverse local structures which is challenging enough to the classifier. This can be formulated as

$$\max_{e \in \mathcal{E}} \mathbb{E}_{(y_v, G_v) \sim P(\mathbf{y}_v, \mathbf{G}_v | e=e)} [L(f_\theta(G_v), y_v) | e]. \quad (9)$$

As mentioned above, homophily is an important metric to describe local structure. An obvious shift between training and test environments is the significant drop in homophily ratio. Motivated by this, we propose an attention mechanism that dynamically adjusts edge weights according to the features of the nodes connected by the edge. Specifically, following [23], we employ a node-wise attention mechanism to model the connection strength between the nodes connected by the same edge as follows

$$w_{(u,v)} = \text{sigmoid}(\text{LeakyReLU}(\text{concat}(h_{\text{ego}|u}, h_{\text{ego}|v})W)), \quad (10)$$

where W is a learnable attention weight matrix. $w_{(u,v)}$ is the adjusted weight, normalized to $[0, 1]$. Then we apply the disentangled encoder on the generated graph with the adjusted edge weights in the message passing process, resulting in ego-node and neighborhood embeddings H'_{ego} and H'_{ne} . The augmented loss is defined as:

$$L_{AUG} = \mathcal{L}(P(y \mid H'_{\text{ego}}, H'_{\text{ne}}), Y) \quad (11)$$

Our design offers three key advantages. First, in contrast to methods that directly add or drop edges, our approach enables continuous modulation of edge weights, allowing for a more fine-grained perturbation of local structures. This continuous adjustment effectively mixes the discrete states of connected and disconnected edges, providing a richer spectrum of graph augmentations. Second, by optimizing the environment generator rather than directly manipulating several fully connected adjacency matrices, our method achieves significant memory efficiency. This approach circumvents the substantial memory overhead associated with storing and processing dense adjacency representations in EERM[14]. Third, our method leverages the features of both endpoints of an edge to perform homophily-aware edge weight adjustments, as opposed to random or uninformed modifications, leading to generating environments with diverse local structures with respect to homophily.

Diversity Loss To ensure the module generates diverse local structures, we introduce a diversity loss that maximizes the dissimilarity between the original neighborhood embedding and the generated neighborhood embedding:

$$L_{DIV} = \text{Cosine Similarity}(H_{ne}, H'_{ne}), \quad (12)$$

where H_{subgraph} and H'_{subgraph} represent the embeddings of the original and augmented subgraphs, respectively.

Preserved Edge Ratio(PER) Loss To prevent trivial solutions, such as adjusting all edge weights to zero, we regularize the environment generator to perturb only a small subset of edges. This is achieved by minimizing the following PER loss:

$$L_{PER} = \left\| \frac{\sum_{e \in E} w(e)}{|E|} - \rho \right\|_2, \quad (13)$$

where ρ is the target proportion of edges to preserve from perturbation.

3.3 Overall Adversarial Training

To jointly optimize the disentangled encoder and the environment mixup, we employ an iterative adversarial training framework. This framework consists of an outer loop for training the encoder and an inner loop for training the environment mixup, ensuring that both components are optimized in a coordinated manner.

Outer Loop: Training the Disentangled Encoder In the outer loop, we train the encoder using a composite loss function that mixes the standard classification loss L and the augmented loss L_{AUG} . The augmented loss L_{AUG} ensures that the model generalizes well to environments with structural distribution shifts. The overall loss function for the outer loop is defined as:

$$L_{outer} = L + L_{AUG}. \quad (14)$$

Inner Loop: Training the Environment Mixup In the inner loop, we optimize the environment mixup to generate diverse and challenging augmented environments. The corresponding loss function combines the augmented loss $-L_{AUG}$, the diversity loss L_{DIV} , and the PER loss L_{PER} . Specifically:

$$L_{inner} = -L_{AUG} + L_{DIV} + L_{PER}. \quad (15)$$

Here, $-L_{AUG}$ encourages to generate more challenging environments, L_{DIV} maximizes the dissimilarity between original and generated environments to enhance diversity, and L_{PER} ensures that only a small subset of edges is perturbed, avoiding trivial solutions.

Training Dynamics The training process alternates between the outer and inner loops, as shown in the Algorithm. 1. In each iteration of the outer loop, the encoder is updated to minimize L_{outer} , while in the inner loop, the environment mixup is updated to minimize L_{inner} . This adversarial training strategy encourages to generation of new distributions with diverse local structures and to improve the generalization capability across different distributions.

Algorithm 1 Training pipeline for **HEM**

Require: Training epochs L , edge preservation ratio ρ

- 1: **for** $l = 1, \dots, L$ **do**
 - 2: Obtain H_{ego}, H_{ne} for each node as described in Section 3.1
 - 3: Calculate classification loss L as Eq. 8
 - 4: Generate new environment and calculate classification loss in generated environment L_{AUG} as Eq. 11
 - 5: Calculate diversity loss L_{DIV} and PER loss L_{PER} as Eq. 12 and Eq. 13
 - 6: Calculate inner loss L_{inner} and outer loss L_{outer} as Eq. 15 and Eq. 14
 - 7: Update the homophily-aware environment mixup by minimizing inner loss
 - 8: Update the disentangled encoder by minimizing outer loss
 - 9: **end for**
-

3.4 Theoretical Explanation

To theoretically justify the proposed **HEM** framework, we analyze how it promotes prediction stability and invariant learning across varying homophily ratios through a simplified example.

Variable Definitions. Let h denote the homophily ratio and $Y \in \{0, 1\}$ denote the node label, where $Y = 1$ indicates an anomalous node and $Y = 0$ a normal node. Let $\pi = P(Y = 1)$ be the anomaly rate, and denote by r_v the proportion of anomalous neighbors for node v :

$$r_v = \frac{1}{d_v} \sum_{u \in \mathcal{N}(v)} Y_u, \quad (16)$$

where $d_v = |\mathcal{N}(v)|$ is the node degree. We assume a constant node degree d . Let $\mu_1 = \mathbb{E}[X | Y = 1]$ and $\mu_0 = \mathbb{E}[X | Y = 0]$, and denote $\Delta\mu = \mu_1 - \mu_0 \neq 0$. For a one-layer message-passing encoder followed by a linear classifier, the node embedding can be expressed as

$$Z_v = r_v \mu_1 + (1 - r_v) \mu_0 = \mu_0 + r_v \Delta\mu. \quad (17)$$

Lemma 1 (Predictor Dependence on Homophily). The optimal coefficient that minimizes the Mean Squared Error (MSE) between predictions and labels is

$$\alpha^*(h) = \frac{\text{Cov}(Y, r)}{\text{Var}(r)} = \frac{\pi(1 - \pi)(2h - 1)}{\frac{h(1-h)}{d} + \pi(1 - \pi)(2h - 1)^2}. \quad (18)$$

Hence, the learned predictor α^* depends explicitly on the homophily ratio h ; a model trained under homophily h_{tr} will be suboptimal for a test distribution with a different h_{te} .

Lemma 2 (Bounded Deviation on the Test Distribution). If there exists an ϵ such that $|h_{\text{te}} - h_{\text{tr}}| \leq \epsilon$, with the proposed environment mixup and adversarial training, the gap between the expected test loss of the training-optimal parameters and the optimal test loss is bounded. Specifically,

$$L(h_{\text{te}}, \alpha_{\text{tr}}^*) - L(h_{\text{te}}, \alpha_{\text{te}}^*) \leq V_{\max} L_f \epsilon^2 \quad (19)$$

, where L_f denotes the Lipschitz constant of $f(h) = \alpha_h^*$, and $V_{\max}$ represents the maximum of $\text{Var}(r)$ for h satisfying $|h - h_{\text{tr}}| \leq \epsilon$.

4 Experiments

In this section, we conduct various experiments to verify that our proposed method can handle structural distribution shifts in graph anomaly detection tasks by utilizing the invariant patterns in the disentangled subgraph representations.

4.1 Datasets

We choose 3 commonly used GAD datasets, including Amazon, Yelp, and T-finance. We provide the details of the datasets in Appendix A.

4.2 Baselines

To demonstrate the effectiveness of our proposed method in addressing structural distribution shifts in graph anomaly detection tasks through a complete comparison, we have selected four types of baselines for evaluation, including (1) General GNNs: GCN[24], GAT[23], GraphSAGE[25], (2) GNNs Specialized for Graph Anomaly Detection: BernNet[26], BWGNN[18], GHRN[27], PCGNN[28], BAT[29], (3) General Out-of-Distribution Methods: GroupDRO[21], V-REx[20], (4) Graph Out-of-Distribution Methods: SRGNN[22], EERM[14], GDN[12]. Details of these baseline models are provided in Appendix B.

4.3 Settings

We adopt the above three real-world datasets as node-level graph anomaly detection tasks. We transform Amazon and Yelp into homogeneous graphs by simply merging all types of edges into one type to make a comparison between homogeneous GNNs and heterogeneous GNNs. We divide each dataset into 2 domains, with/without structural distribution shift. For brevity, we denote domains with/without distribution shift as 'w/ DS' and 'w/o DS' separately. Specifically, we sample nodes by their homophily ratio, which means nodes with low homophily are more likely to be divided

Table 1: Results of different methods on real-world graph anomaly detection datasets. The best results are in bold and the second-best results are underlined. The results are reported in AUPRC.

Dataset Model	Amazon		Yelp		T-Finance	
	w/o DS	w/ DS	w/o DS	w/ DS	w/o DS	w/ DS
GCN	94.45$\pm$0.97	69.90 $\pm$ 2.13	53.63 $\pm$ 0.94	40.79 $\pm$ 1.73	93.66 $\pm$ 0.10	60.33 $\pm$ 0.41
GAT	92.84 $\pm$ 0.00	<u>72.44$\pm$0.00</u>	54.74 $\pm$ 0.00	42.83 $\pm$ 0.00	87.38 $\pm$ 0.00	53.93 $\pm$ 0.00
GraphSAGE	90.57 $\pm$ 1.03	69.71 $\pm$ 1.00	63.05$\pm$0.24	53.25 $\pm$ 0.76	73.61 $\pm$ 0.47	44.21 $\pm$ 0.98
BernNet	92.83 $\pm$ 0.02	71.34 $\pm$ 0.21	52.99 $\pm$ 0.20	48.53 $\pm$ 0.32	89.25 $\pm$ 0.71	65.21 $\pm$ 0.58
PCGNN	90.95 $\pm$ 0.01	72.02 $\pm$ 0.03	54.05 $\pm$ 0.78	44.52 $\pm$ 0.32	69.37 $\pm$ 1.54	36.84 $\pm$ 2.07
GHRN	92.01 $\pm$ 0.25	69.39 $\pm$ 0.14	59.27 $\pm$ 0.45	54.68 $\pm$ 0.61	93.62 $\pm$ 0.15	62.43 $\pm$ 0.36
BWGNN	91.59 $\pm$ 0.20	69.26 $\pm$ 0.38	58.21 $\pm$ 0.32	52.97 $\pm$ 0.51	93.44 $\pm$ 0.10	62.34 $\pm$ 0.72
BAT	<u>94.03$\pm$2.04</u>	72.26 $\pm$ 1.16	<u>62.19$\pm$2.61</u>	<u>54.93$\pm$0.49</u>	92.77 $\pm$ 0.67	56.35 $\pm$ 1.75
V-REx	92.54 $\pm$ 0.14	68.84 $\pm$ 0.23	59.16 $\pm$ 0.70	53.58 $\pm$ 0.65	93.04 $\pm$ 0.12	60.66 $\pm$ 0.78
GroupDRO	92.58 $\pm$ 0.04	68.87 $\pm$ 0.29	58.73 $\pm$ 0.66	52.13 $\pm$ 0.31	93.86 $\pm$ 0.13	60.81 $\pm$ 0.28
SRGNN	90.48 $\pm$ 0.36	70.36 $\pm$ 0.56	58.73 $\pm$ 0.21	52.84 $\pm$ 0.40	91.11 $\pm$ 0.24	60.36 $\pm$ 0.14
EERM	92.57 $\pm$ 0.01	70.94 $\pm$ 0.51	OOM	OOM	OOM	OOM
GDN	91.97 $\pm$ 0.50	68.93 $\pm$ 0.43	48.20 $\pm$ 1.13	48.43 $\pm$ 0.92	92.51 $\pm$ 0.07	59.84 $\pm$ 0.05
HEM(+GCN)	93.70 $\pm$ 0.11	70.59 $\pm$ 0.09	55.09 $\pm$ 0.47	52.05 $\pm$ 1.00	95.69$\pm$0.08	64.97 $\pm$ 0.11
HEM(+BWGNN)	91.97 $\pm$ 0.74	74.91$\pm$0.49	61.42 $\pm$ 0.56	58.45$\pm$1.01	<u>94.76$\pm$0.06</u>	66.04$\pm$0.29

Table 2: Results of different methods on real-world graph anomaly detection datasets. The best results are in bold and the second-best results are underlined. The results are reported in Recall@K.

Model Model	Amazon		Yelp		T-Finance	
	w/o DS	w/ DS	w/o DS	w/ DS	w/o DS	w/ DS
GCN	90.18 $\pm$ 1.01	65.37 $\pm$ 0.69	51.95 $\pm$ 0.24	44.04 $\pm$ 0.27	89.57 $\pm$ 0.22	57.50 $\pm$ 0.89
GAT	90.22 $\pm$ 2.18	<u>66.99$\pm$0.83</u>	56.42 $\pm$ 0.00	48.11 $\pm$ 0.00	82.59 $\pm$ 0.00	51.88 $\pm$ 0.00
GraphSAGE	88.18 $\pm$ 0.64	65.20 $\pm$ 0.61	58.92$\pm$0.23	51.01 $\pm$ 0.49	65.88 $\pm$ 0.72	43.68 $\pm$ 1.22
BernNet	89.74 $\pm$ 1.65	65.53 $\pm$ 1.84	52.33 $\pm$ 0.21	47.89 $\pm$ 0.25	86.59 $\pm$ 0.77	<u>62.31$\pm$0.18</u>
PCGNN	89.55 $\pm$ 0.64	66.67 $\pm$ 0.46	50.89 $\pm$ 0.12	43.14 $\pm$ 0.08	80.08 $\pm$ 1.44	<u>53.36$\pm$0.69</u>
GHRN	89.55 $\pm$ 0.00	62.93 $\pm$ 0.00	54.96 $\pm$ 0.54	51.13 $\pm$ 0.57	87.53 $\pm$ 0.69	60.16 $\pm$ 0.28
BWGNN	90.00 $\pm$ 0.00	63.90 $\pm$ 0.00	54.98 $\pm$ 0.83	50.33 $\pm$ 0.37	87.76 $\pm$ 0.84	60.75 $\pm$ 0.72
BAT	90.18 $\pm$ 1.21	<u>66.99$\pm$0.83</u>	<u>56.80$\pm$2.64</u>	<u>51.37$\pm$2.14</u>	88.83 $\pm$ 1.54	56.54 $\pm$ 0.72
V-REx	89.24 $\pm$ 0.21	62.76 $\pm$ 0.92	54.94 $\pm$ 0.25	<u>51.37$\pm$0.54</u>	86.82 $\pm$ 0.19	59.72 $\pm$ 0.28
GroupDRO	89.55 $\pm$ 0.00	61.63 $\pm$ 0.61	55.63 $\pm$ 0.54	51.09 $\pm$ 0.40	88.63 $\pm$ 0.29	60.53 $\pm$ 0.31
SRGNN	90.00 $\pm$ 0.37	63.41 $\pm$ 0.40	54.31 $\pm$ 0.58	50.79 $\pm$ 0.59	87.92 $\pm$ 0.73	59.20 $\pm$ 1.10
GDN	89.39 $\pm$ 0.21	64.88 $\pm$ 1.05	49.34 $\pm$ 1.66	48.21 $\pm$ 1.64	85.73 $\pm$ 0.22	58.02 $\pm$ 0.91
HEM(+GCN)	90.45$\pm$0.00	65.85 $\pm$ 0.00	52.83 $\pm$ 0.49	49.47 $\pm$ 0.20	<u>90.27$\pm$0.44</u>	62.23 $\pm$ 0.58
HEM(+BWGNN)	90.15 $\pm$ 0.21	67.80$\pm$0.80	56.10 $\pm$ 0.68	52.63$\pm$0.28	90.43$\pm$0.22	63.12$\pm$0.10

into test data('w/ DS'). Furthermore, we divide the 'w/o DS' domain into training, validation, and test('w/o DS') data. Since graph anomaly detection datasets are usually with severe data imbalance, we use AUPRC (Area Under the Precision-Recall Curve) and Recall@K (the recall among the top-K highest-confidence predictions) as the evaluation metrics.

4.4 Results

Based on the results in Table 1, we can get the following observations:

- Baselines all fail significantly under structural distribution shift: (1) Although baseline methods performs well on test data without distribution shift, they all experience a significant performance drop on test data with structural distribution shift. This phenomenon indicates that these methods may rely on spurious patterns that can not be generalized to the test ('w/ DS') environment.

Table 3: Ablation studies on ego-neighborhood disentangled encoder and homophily-aware environment mixup, where 'w/o END' denotes removing the ego-neighborhood disentangled encoder, 'w/o DIV' denotes removing diversity loss and 'w/o PER' denotes removing PER loss. The best results are in bold and the second-best results are underlined. The results are reported in AUPRC.

Dataset Model	Amazon		Yelp		T-Finance	
	w/o DS	w/ DS	w/o DS	w/ DS	w/o DS	w/ DS
w/o END encoder	91.41 $\pm$ 0.43	66.59 $\pm$ 1.63	58.33 $\pm$ 1.09	56.53 $\pm$ 0.90	94.52 $\pm$ 0.23	62.50 $\pm$ 0.50
w/o DIV	93.34$\pm$0.64	70.96 $\pm$ 1.64	<u>61.08$\pm$0.89</u>	<u>57.52$\pm$1.18</u>	94.08 $\pm$ 0.71	64.73 $\pm$ 1.37
w/o PER	<u>92.45$\pm$0.04</u>	<u>72.06$\pm$0.16</u>	60.07 $\pm$ 0.88	56.00 $\pm$ 0.55	94.88$\pm$0.15	<u>65.50$\pm$0.51</u>
HEM(+BWGNN)	91.97 $\pm$ 0.74	74.91$\pm$0.49	61.42$\pm$0.56	58.45$\pm$1.01	<u>94.76$\pm$0.06</u>	66.04$\pm$0.29

For example, a strong baseline GHRN, experiences performance drop of nearly 25%, 8%, 33%. (2) Out-of-distribution baselines all fail to achieve consistent improvement across all datasets. Compared with BWGNN, the backbone classifier used in these OOD methods, some of the OOD baselines can achieve improvement on specific datasets, like EERM in Amazon and V-REx in Yelp. However, none of them achieves consistent generalization improvement across all datasets, indicating these OOD methods rely on diverse training environments with ground-truth environment labels, e.g. V-REx, GroupDRO, SRGNN. EERM employs the REINFORCE algorithm to create diverse training environments by directly optimizing multiple edge masks, which costs too much memory resources.

- Our proposed method achieves consistent performance improvement across all datasets. Compared with the best baselines, our method achieves relative improvements of 3.4%, 6.9%, and 1.3% on test ('w/ DS') data, and outperforms all general and graph-specialized OOD baselines. This implies that our method can capture the invariant patterns consistently, and create new environments in a memory-efficient manner.

4.5 Ablation Study

In this section, we conduct ablation studies to verify the effectiveness of our proposed ego-neighborhood disentangled encoder and homophily-aware environment mixup.

Ego-neighborhood disentangled encoder We remove the disentangled encoder and simply apply a GNN encoder. From Table 3, we can find that without ego-neighborhood disentangled encoder, the model's performance drops significantly in 'w/o DS' and 'w/ DS' test data across all datasets, which implies the disentanglement design is important in modeling feature and structural information in GAD tasks.

Homophily-aware environment mixup We remove the diversity loss ('w/o DIV') and preserved edge ratio loss ('w/o PER') separately, and both of them show significant performance drops in Table 3. Specifically, removing diversity loss leads to consistent performance drop across all datasets, but removing PER loss increases the performance on T-Finance, implying that deleting a large proportion of edges in T-Finance can increase the generalization capability while preserving the performance in the training stage. This can be verified in the hyperparameter sensitivity analysis. Moreover, our proposed method reduced overfitting in the training stage by adding the diversity loss and preserved edge ratio loss. For example, in the Amazon dataset, the models removing diversity loss or removing PER loss perform better in the test('w/o DS') but worse in the test('w/ DS').

4.6 Complexity Analysis

In this section, we analyze the computation complexity of **HEM**. Denote $|V|$ and $|E|$ as the total number of nodes and edges in the graph, and denote d as the dimension of hidden representation. The ego-neighborhood disentangled encoder has a time complexity of $\mathcal{O}(|V|d^2 + |E|d)$. The homophily-aware environment mixup has a time complexity of $\mathcal{O}(|E|d + |E|d + |E|) = \mathcal{O}(|E|d)$, including the edge reweighting and computation of diversity loss and PER loss. Therefore, the total computation complexity for **HEM** is $\mathcal{O}(|V|d^2 + |E|d)$. **HEM** has a linear computation complexity with respect to the number of nodes and edges, comparable to existing GNN baselines.

5 Related Work

In this section, we review the existing works about graph anomaly detection and out-of-distribution generalization.

Graph Anomaly Detection The common challenges in graph anomaly detection are data imbalance and heterophily[1], and extensive works are proposed to solve these problems from spatial and spectral perspectives[30–34]. For spatial methods, PCGNN[28] proposes a balanced sampler for the neighborhood aggregation process in GNNs to reduce data imbalance. BAT[29] further proposes a class-rebalancing-free data augmentation framework based on a topological paradigm, which can mitigate the class-imbalance bias and achieve consistent performance boosting across general class imbalance node classification tasks. GAS[35] utilizes structural preprocessing to handle severe heterophily before applying the GNN model. CARE-GNN[36], PMP[37], GraphConsis[38] propose to design new message passing and aggregation mechanisms to mitigate the influence of heterophily on prediction. Another type of method like H2-FDetector[39] decouples the information aggregation for homophilic and heterophilic patterns. Since traditional graph convolution is known as low-frequency filters[40], extensive spectral GNN methods are proposed to handle high-frequency signals in heterophilic graphs[41][42]. BWGNN[18] propose a polynomial spectral GNN with Beta kernel as prior, which is proved to fit the spectral energy distribution in GAD scenario and achieves obvious performance improvement compared with BernNet[26], which has no prior and tries to learn a kernel from all possible polynomial functions. AMNet[43] tries to learn high-frequency and low-frequency signals separately and combines them for prediction. While these GAD-specific methods have achieved substantial progress, there is still room for further enhancement. They fail to adequately model the interaction patterns between the ego-node and its surrounding neighborhood, which could potentially reveal discrepancies that are crucial for more accurate anomaly identification.

Out-of-Distribution in Graph (OOD) The structural distribution shift is a prevalent issue in graph data. For example, variations in disciplines and time periods may result in different citation patterns in citation networks, and variations in geographic locations and relationship types result in different interaction types in social networks. Existing OOD methods can be divided into general methods and graph specialized methods[44–51]. General OOD approaches, such as V-REx[20] and GroupDRO[21], commonly depend on diverse training environments characterized by explicit environment labels, a challenging requirement in graph anomaly detection where inherent homophily obscures environment distinctions. While graph-specific OOD methods have emerged, limitations persist. For instance, EERM[14] constructs novel training environments using learnable graph editors and trains with V-REx to learn invariant representations, yet incurs significant memory overhead due to maintaining multiple full adjacency matrices. SRGNN[22] utilizes the kernel mean matching method to regularize the discrepancy between embeddings of training and test data, while the lack of supervised information leads to its limited performance. GDN[12, 8] is the first work to address structural distribution shift in graph anomaly detection. They handle the problem in an invariant learning way by identifying the critical anomaly features with gradients, while neglecting the importance of varying local structures in distribution shift. In conclusion, prevailing methodologies still have limitations prominently including memory inefficiency, insufficient availability of supervised information, neglect of inherent local structural patterns, and undue reliance on specific training environment assumptions.

6 Conclusion

In this paper, we proposed the Ego-Neighborhood Disentangled Encoder with **H**omophily-aware **E**nvironment **M**ixup (**HEM**), for improving the OOD generalization performance for GAD problems. First, we proposed an ego-neighborhood disentangled encoder to disentangle feature and structural representations, in order to model discrepancy patterns and discover invariant patterns. Then, we proposed the homophily-aware environment mixup to generate training environments with diverse local structures to improve the generalization capability of the model. Our model achieves consistent improvement across all datasets.

Acknowledgments and Disclosure of Funding

This work was supported by the National Key Research and Development Program of China (No. 2022ZD0115903, 2023YFF1205001), National Natural Science Foundation of China No. 62222209, Beijing National Research Center for Information Science and Technology under Grant No. BNR2023TD03006, and Beijing Key Lab of Networked Multimedia.

References

- [1] X. Ma, J. Wu, S. Xue, J. Yang, C. Zhou, Q. Z. Sheng, H. Xiong, and L. Akoglu, “A comprehensive survey on graph anomaly detection with deep learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 12, pp. 12 012–12 038, 2021.
- [2] X. Huang, Y. Yang, Y. Wang, C. Wang, Z. Zhang, J. Xu, L. Chen, and M. Vazirgiannis, “Dgraph: A large-scale financial dataset for graph anomaly detection,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 22 765–22 777, 2022.
- [3] M. Weber, G. Domeniconi, J. Chen, D. K. I. Weidele, C. Bellei, T. Robinson, and C. E. Leiserson, “Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics,” *arXiv preprint arXiv:1908.02591*, 2019.
- [4] L. Cheng, R. Guo, K. Shu, and H. Liu, “Causal understanding of fake news dissemination on social media,” in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 148–157.
- [5] X. Yuan, N. Zhou, S. Yu, H. Huang, Z. Chen, and F. Xia, “Higher-order structure based anomaly detection on attributed networks,” in *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, 2021, pp. 2691–2700.
- [6] J. Tang, F. Hua, Z. Gao, P. Zhao, and J. Li, “Gadbench: Revisiting and benchmarking supervised graph anomaly detection,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 29 628–29 653, 2023.
- [7] J. Chen, G. Zhu, C. Yuan, and Y. Huang, “Boosting graph anomaly detection with adaptive message passing,” in *The Twelfth International Conference on Learning Representations*.
- [8] Y. Gao, J. Li, X. Wang, X. He, H. Feng, and Y. Zhang, “Revisiting attack-caused structural distribution shift in graph anomaly detection,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 9, pp. 4849–4861, 2024.
- [9] M. Wu, X. Zheng, Q. Zhang, X. Shen, X. Luo, X. Zhu, and S. Pan, “Graph learning under distribution shifts: A comprehensive survey on domain adaptation, out-of-distribution, and continual learning,” *arXiv preprint arXiv:2402.16374*, 2024.
- [10] W. Bao, Z. Zeng, Z. Liu, H. Tong, and J. He, “Adarc: Mitigating graph structure shifts during test-time,” *arXiv preprint arXiv:2410.06976*, 2024.
- [11] S. Liu, T. Li, Y. Feng, N. Tran, H. Zhao, Q. Qiu, and P. Li, “Structural re-weighting improves graph domain adaptation,” in *International Conference on Machine Learning*. PMLR, 2023, pp. 21 778–21 793.
- [12] Y. Gao, X. Wang, X. He, Z. Liu, H. Feng, and Y. Zhang, “Alleviating structural distribution shift in graph anomaly detection,” in *Proceedings of the sixteenth ACM international conference on web search and data mining*, 2023, pp. 357–365.
- [13] E. Altman, J. Blanuša, L. Von Niederhäusern, B. Egressy, A. Anghel, and K. Atasu, “Realistic synthetic financial transactions for anti-money laundering models,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [14] Q. Wu, H. Zhang, J. Yan, and D. Wipf, “Handling distribution shifts on graphs: An invariance perspective,” *arXiv preprint arXiv:2202.02466*, 2022.

- [15] Y. Liu, Z. Li, S. Pan, C. Gong, C. Zhou, and G. Karypis, “Anomaly detection on attributed networks via contrastive self-supervised learning,” *IEEE transactions on neural networks and learning systems*, vol. 33, no. 6, pp. 2378–2392, 2021.
- [16] Z. Zhang, X. Wang, Z. Zhang, H. Li, Z. Qin, and W. Zhu, “Dynamic graph neural networks under spatio-temporal distribution shift,” *Advances in neural information processing systems*, vol. 35, pp. 6074–6089, 2022.
- [17] Z. Zhang, X. Wang, Z. Zhang, Z. Qin, W. Wen, H. Xue, H. Li, and W. Zhu, “Spectral invariant learning for dynamic graphs under distribution shifts,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [18] J. Tang, J. Li, Z. Gao, and J. Li, “Rethinking graph neural networks for anomaly detection,” in *International Conference on Machine Learning*. PMLR, 2022, pp. 21 076–21 089.
- [19] A. v. d. Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” *arXiv preprint arXiv:1807.03748*, 2018.
- [20] D. Krueger, E. Caballero, J.-H. Jacobsen, A. Zhang, J. Binas, D. Zhang, R. Le Priol, and A. Courville, “Out-of-distribution generalization via risk extrapolation (rex),” in *International conference on machine learning*. PMLR, 2021, pp. 5815–5826.
- [21] S. Sagawa, P. W. Koh, T. B. Hashimoto, and P. Liang, “Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization,” *arXiv preprint arXiv:1911.08731*, 2019.
- [22] Q. Zhu, N. Ponomareva, J. Han, and B. Perozzi, “Shift-robust gnns: Overcoming the limitations of localized graph training data,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 27 965–27 977, 2021.
- [23] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio *et al.*, “Graph attention networks,” *stat*, vol. 1050, no. 20, pp. 10–48 550, 2017.
- [24] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *arXiv preprint arXiv:1609.02907*, 2016.
- [25] W. Hamilton, Z. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” *Advances in neural information processing systems*, vol. 30, 2017.
- [26] M. He, Z. Wei, H. Xu *et al.*, “Bernnet: Learning arbitrary graph spectral filters via bernstein approximation,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 14 239–14 251, 2021.
- [27] Y. Gao, X. Wang, X. He, Z. Liu, H. Feng, and Y. Zhang, “Addressing heterophily in graph anomaly detection: A perspective of graph spectrum,” in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 1528–1538.
- [28] Y. Liu, X. Ao, Z. Qin, J. Chi, J. Feng, H. Yang, and Q. He, “Pick and choose: a gnn-based imbalanced learning approach for fraud detection,” in *Proceedings of the web conference 2021*, 2021, pp. 3168–3177.
- [29] Z. Liu, R. Qiu, Z. Zeng, H. Yoo, D. Zhou, Z. Xu, Y. Zhu, K. Weldemariam, J. He, and H. Tong, “Class-imbalanced graph learning without class rebalancing,” 2024.
- [30] Y. Lin, J. Tang, C. Zi, H. V. Zhao, Y. Yao, and J. Li, “Unigad: Unifying multi-level graph anomaly detection,” *arXiv preprint arXiv:2411.06427*, 2024.
- [31] Q. Wang, G. Pang, M. Salehi, X. Xia, and C. Leckie, “Open-set graph anomaly detection via normal structure regularisation,” *arXiv preprint arXiv:2311.06835*, 2023.
- [32] J. Zhang, S. Wang, and S. Chen, “Reconstruction enhanced multi-view contrastive learning for anomaly detection on attributed networks,” *arXiv preprint arXiv:2205.04816*, 2022.

- [33] J. Duan, B. Xiao, S. Wang, H. Zhou, and X. Liu, “Arise: Graph anomaly detection on attributed networks via substructure awareness,” *IEEE transactions on neural networks and learning systems*, 2023.
- [34] R. Guo, M. Zou, S. Zhang, X. Zhang, Z. Yu, and Z. Feng, “Graph local homophily network for anomaly detection,” in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 2024, pp. 706–716.
- [35] A. Li, Z. Qin, R. Liu, Y. Yang, and D. Li, “Spam review detection with graph convolutional networks,” *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019. [Online]. Available: <https://api.semanticscholar.org/CorpusID:201660501>
- [36] Y. Dou, Z. Liu, L. Sun, Y. Deng, H. Peng, and P. S. Yu, “Enhancing graph neural network-based fraud detectors against camouflaged fraudsters,” in *Proceedings of the 29th ACM international conference on information & knowledge management*, 2020, pp. 315–324.
- [37] W. Zhuo, Z. Liu, B. Hooi, B. He, G. Tan, R. Fathony, and J. Chen, “Partitioning message passing for graph fraud detection,” in *The Twelfth International Conference on Learning Representations*, 2024.
- [38] Z. Liu, Y. Dou, P. S. Yu, Y. Deng, and H. Peng, “Alleviating the inconsistency problem of applying graph neural network to fraud detection,” in *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, 2020, pp. 1569–1572.
- [39] F. Shi, Y. Cao, Y. Shang, Y. Zhou, C. Zhou, and J. Wu, “H2-fdetector: A gnn-based fraud detector with homophilic and heterophilic connections,” in *Proceedings of the ACM web conference 2022*, 2022, pp. 1486–1494.
- [40] D. Bo, X. Wang, C. Shi, and H. Shen, “Beyond low-frequency information in graph convolutional networks,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 5, 2021, pp. 3950–3957.
- [41] E. Chien, J. Peng, P. Li, and O. Milenkovic, “Adaptive universal generalized pagerank graph neural network,” *arXiv preprint arXiv:2006.07988*, 2020.
- [42] Y. Gao, J. Fang, Y. Sui, Y. Li, X. Wang, H. Feng, and Y. Zhang, “Graph anomaly detection with bi-level optimization,” in *Proceedings of the ACM Web Conference 2024*, ser. WWW ’24. New York, NY, USA: Association for Computing Machinery, 2024, p. 4383–4394. [Online]. Available: <https://doi.org/10.1145/3589334.3645673>
- [43] Z. Chai, S. You, Y. Yang, S. Pu, J. Xu, H. Cai, and W. Jiang, “Can abnormality be detected by graph neural networks?” in *IJCAI*, 2022, pp. 1945–1951.
- [44] X. Han, Z. Jiang, N. Liu, and X. Hu, “G-mixup: Graph data augmentation for graph classification,” in *International Conference on Machine Learning*. PMLR, 2022, pp. 8230–8248.
- [45] G. Liu, T. Zhao, J. Xu, T. Luo, and M. Jiang, “Graph rationalization with environment-based augmentations,” in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 1069–1078.
- [46] K. Kong, G. Li, M. Ding, Z. Wu, C. Zhu, B. Ghanem, G. Taylor, and T. Goldstein, “Robust optimization as data augmentation for large-scale graphs,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 60–69.
- [47] S. Fan, X. Wang, C. Shi, P. Cui, and B. Wang, “Generalizing graph neural networks on out-of-distribution graphs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.
- [48] Y.-X. Wu, X. Wang, A. Zhang, X. He, and T.-S. Chua, “Discovering invariant rationales for graph neural networks,” *arXiv preprint arXiv:2201.12872*, 2022.
- [49] Q. Zhu, C. Zhang, C. Park, C. Yang, and J. Han, “Shift-robust node classification via graph adversarial clustering,” *arXiv preprint arXiv:2203.15802*, 2022.

- [50] J. Ma, J. Deng, and Q. Mei, “Subgroup generalization and fairness of graph neural networks,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 1048–1061, 2021.
- [51] Y. Qin, X. Wang, Z. Zhang, P. Xie, and W. Zhu, “Graph neural architecture search under distribution shifts,” in *International Conference on Machine Learning*. PMLR, 2022, pp. 18 083–18 095.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: The main claims can be verified in the Experiments Section.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[NA\]](#)

Justification: This paper proposes a new GAD OOD method in a structural distribution shift setting and shows no obvious limitations.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: The assumptions and proofs can be found in the Appendix E.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: The details can be found in the Experiments Section, especially the settings subsection.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [NA]

Justification: The data used in this paper are all public datasets, and it requires some time to get the code fully prepared to be released.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: This can be found in the Experiments Section.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: This can be found in Table 1

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: The details can be found in Appendix F.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: This research is with the NeurIPS Code of Ethics in every respect.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: This paper is focusing on improving the OOD algorithm in GAD problems, with little impact on society.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This paper has no such risks since it's an improved OOD method without releasing any data or models that have a high risk for misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: This can be found in the Experiments and Related Work Sections.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset’s creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: This paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This paper does not involve such experiments.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

A Datasets

A.1 Dataset Details

- **Amazon.** The task of the Amazon dataset is to identify anomalous users who wrote fake reviews for products under the Musical Instrument category on the Amazon website. Users are nodes in the graph and there are three types of edges: U-P-U (connects users reviewing at least one same product), U-S-U (connects users giving at least one same star rating within one week), U-V-U (connects users with top 5% mutual review text similarities among all users). The node features are handcrafted features about user information.
- **Yelp.** Yelp dataset includes hotel and restaurant reviews, and the task is to filter spam reviews. The reviews are nodes in the graph and there are three types of edges: R-U-R (connects reviews posted by the same person), R-S-R (connects reviews about the same product with the same rating), R-T-R (connects reviews about the same product while posted in the same month). The node features are handcrafted features from raw texts.
- **T-Finance.** T-Finance dataset is about a transaction network and the target is to find anomaly accounts. The nodes are accounts and the edges are transactions between accounts. The node features are user profile details like registration days.

A.2 Dataset Split Method

The dataset is divided by the homophily ratio of nodes, split into 50%/10%/20%/20% for training, validation, in-distribution test, and out-of-distribution (OOD) test sets, respectively.

The bottom 20% of nodes with the lowest homophily ratios are selected to form the OOD test set.

B Baseline Models

- **General GNNs:** GCN[24] utilizes convolution operation to aggregate messages from the neighborhood and achieves excellent performance in semi-supervised graph tasks. GAT[23] adds an attention mechanism to the message passing process, assigning different weights to edges in order to aggregate information from important neighbors. GraphSAGE[25] utilizes neighborhood sampling methods to fit inductive learning.
- **GNNs Specialized for Graph Anomaly Detection:** BernNet[26] utilizes a K-order Bernstein polynomial approximation to estimate arbitrary graph spectral filters. BWGNN[18] uses Beta distribution as prior for spectrum distribution of graph anomaly detection data. GHRN[27] drops inter-class edges to delineate high-frequency components and increases the homophily of heterophilic dataset. PCGNN[28] designs a label-balanced sampler to sample nodes and edges within neighborhoods to increase homophily. BAT[29] proposed a plug-and-play framework that dynamically augments the graph topology via uncertainty-based node risk estimation and virtual connections.
- **General Out-of-Distribution Methods:** GroupDRO[21] puts more weights on worst-case across different training domains. V-REx[20] enhances out-of-distribution generalization by extrapolating risks and minimizing the variance of losses across different environments.
- **Graph Out-of-Distribution Methods:** SRGNN[22] uses kernel mean matching to regularize the difference in the distributions of training and test nodes' embeddings, in order to avoid learning spurious embedding distribution. EERM[14] utilizes REINFORCE to create diverse environments at the training stage to enhance generalization capability. GDN[12] designs invariance losses and uses gradient descent on input features to filter invariant features.

C Hyperparameter Sensitivity Analysis

C.1 Edge Preservation Ratio

We analyze the sensitivity of the hyperparameter ρ , the edge preservation ratio in PER loss from Equation 13. From Figure 2, we can find a proper ρ is neither too small nor too big. A small ρ means preserving only a small proportion of edges, which may destroy the invariant patterns, and a big ρ lead to a structural distribution without much shift. A perfect choice of ρ is a balance of both aspects.

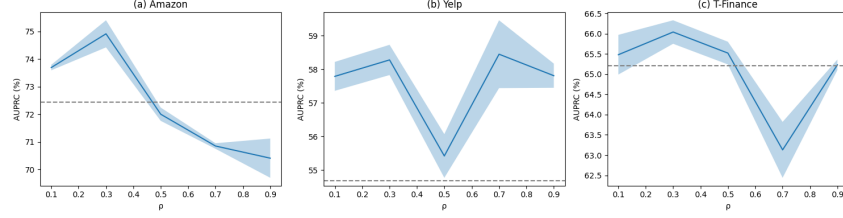


Figure 2: Sensitivity of hyperparameter ρ . The area shows the average AUPRC and standard deviations in the test('w/ DS') data. The dashed line represents the average AUPRC of the best-performed baseline.

C.2 Advserarial Training

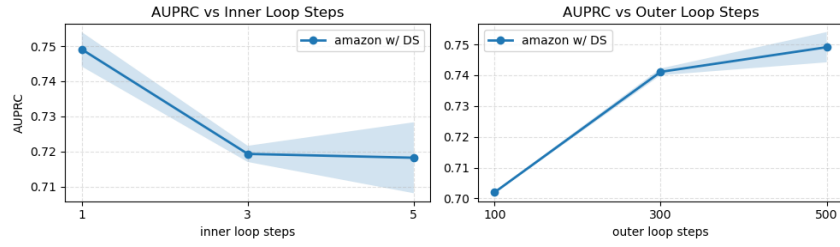


Figure 3: Sensitivity of inner/outer training steps in the adversarial training. The area shows the average AUPRC and standard deviations in the amazon test('w/DS'). By default we choose inner loop steps = 1 and outer loop steps = 500.

We conducted an extensive analysis on the Amazon dataset to evaluate the sensitivity of HEM to the number of inner loop and outer loop steps. Our findings indicate that for the outer loop steps, the model demonstrates robust performance as long as the training steps are not excessively limited. Conversely, the model exhibits relatively higher sensitivity to the inner loop steps. Utilizing a smaller number of inner loop training steps allows for the generation of augmented graphs with an appropriate level of difficulty, thereby facilitating easier model convergence.

D Case Study

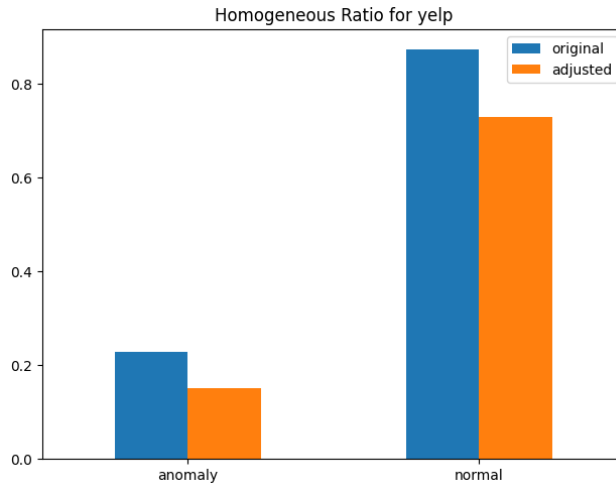


Figure 4: Case Study of Homophily Ratio in Yelp Dataset.

In the case study, we analyze the homophily ratio of anomaly and normal nodes in Yelp dataset. During the data preprocessing, we construct structural distribution shift by sampling nodes with high homophily ratio as training data, leading to a shortcut to predict the label of central node by its neighbors. From Figure 4, the model has learnt to adjust edge weight to decrease the homophily ratio of the total graph and discovered patterns based on the augmented graph.

E Theoretical Proofs

E.1 Lemma 1 (Predictor Dependence on Homophily)

The optimal coefficient that minimizes the Mean Squared Error (MSE) between predictions and labels is

$$\alpha^*(h) = \frac{\text{Cov}(Y, r)}{\text{Var}(r)} = \frac{\pi(1 - \pi)(2h - 1)}{\frac{h(1-h)}{d} + \pi(1 - \pi)(2h - 1)^2}. \quad (20)$$

Firstly, the value of some key statistical properties are:

- $\text{Cov}(Y, r) = \pi(1 - \pi)(2h - 1)$,
- $\text{Var}(Y) = \pi(1 - \pi)$,
- $\text{Var}(r | Y=y) = \frac{h(1-h)}{d}$,
- $\text{Var}(r) = \frac{h(1-h)}{d} + \pi(1 - \pi)(2h - 1)^2$.

For a one-layer message-passing GNN followed by a linear classifier, the node embedding Z_v is:

$$Z_v = (AX)_v = r_v \mu_1 + (1 - r_v) \mu_0 = \mu_0 + r_v \Delta \mu. \quad (21)$$

The final prediction $\hat{Y}_v$ is then:

$$\hat{Y}_v = w^\top \mu_0 + \underbrace{(w^\top \Delta \mu)}_{\alpha} r_v. \quad (22)$$

The optimal coefficient α^* that minimizes the Mean Squared Error (MSE) is given by linear regression:

$$\alpha^* = \frac{\text{Cov}(Y, r)}{\text{Var}(r)} = \frac{\pi(1 - \pi)(2h - 1)}{\frac{h(1-h)}{d} + \pi(1 - \pi)(2h - 1)^2}. \quad (23)$$

Lemma 2 (Bounded Deviation on the Test Distribution). If there exists an ϵ such that $|h_{\text{te}} - h_{\text{tr}}| \leq \epsilon$, with the proposed environment mixup and adversarial training, the gap between the expected test loss of the training-optimal parameters and the optimal test loss is bounded. Specifically,

$$L(h_{\text{te}}, \alpha_{\text{tr}}^*) - L(h_{\text{te}}, \alpha_{\text{te}}^*) \leq V_{\max} L_f \epsilon^2 \quad (24)$$

, where L_f denotes the Lipschitz constant of $f(h) = \alpha_h^*$, and $V_{\max}$ represents the maximum of $\text{Var}(r)$ for h satisfying $|h - h_{\text{tr}}| \leq \epsilon$.

For simplicity, we define:

- $C(h) = \text{Cov}(Y, r) = \pi(1 - \pi)(2h - 1)$,
- $V(h) = \text{Var}(r) = \frac{h(1-h)}{d} + \pi(1 - \pi)(2h - 1)^2$.

Then the MSE Loss is:

$$L(h; \alpha) = \mathbb{E}[(Y - \hat{Y})^2 | h] = V(h) - 2\alpha C(h) + \alpha^2 V(h). \quad (25)$$

Define the optimal α on h as $\alpha^* = f(h) = \frac{C(h)}{V(h)}$, and the optimal loss as:

$$L^* = V(h) - \frac{C(h)^2}{V(h)}. \quad (26)$$

We assume HEM trains on $\mathcal{H}_\epsilon = [h_{\text{tr}} - \epsilon, h_{\text{tr}} + \epsilon]$ by applying environment mixup, which covers h_{te} .

Define:

- $f_{\min} = \inf_{h \in \mathcal{H}_\epsilon} f(h)$,
- $f_{\max} = \sup_{h \in \mathcal{H}_\epsilon} f(h)$,
- $V_{\max} = \sup_{h \in \mathcal{H}_\epsilon} V(h)$.

Then

$$L(h; \alpha) = L^*(h) + V(h)(\alpha - f(h))^2, \quad (27)$$

therefore we can get the upper bound of the loss difference:

$$L(h; \alpha) - L^*(h) \leq \sup_{h \in \mathcal{H}_\epsilon} V(h)(\alpha - f(h))^2 \leq V_{\max} \left(\sup_{h \in \mathcal{H}_\epsilon} (\alpha - f(h))^2 \right).$$

Since the goal of adversarial training in HEM is to minimize the loss in the worst case, we further assume HEM can minimize the upper bound, then:

$$\alpha = \frac{f_{\min} + f_{\max}}{2}. \quad (28)$$

Thus,

$$L(h; \alpha) - L^*(h) \leq V_{\max} \left(\sup_{h \in \mathcal{H}_\epsilon} \left(\frac{f_{\max} - f_{\min}}{2} \right)^2 \right). \quad (29)$$

If f follows Lipschitz continuity, then

$$f_{\max} - f_{\min} \leq L_f((h_{\text{tr}} + \epsilon) - (h_{\text{tr}} - \epsilon)) = 2L_f\epsilon. \quad (30)$$

Finally,

$$L(h; \alpha) - L^*(h) \leq V_{\max} L_f^2 \epsilon^2. \quad (31)$$

F Configurations

All experiments are conducted with:

- Operating System: Ubuntu 20.04.6 LTS
- CPU: Intel(R) Xeon(R) Gold 6348 CPU @ 2.60GHz
- GPU: NVIDIA GeForce RTX 4090 with 24 GB of memory
- Software: Python 3.12.4; CUDA 12.2; PyTorch 2.4.1