Unleashing the Power of Graph Data Augmentation on Covariate Distribution Shift

Yongduo Sui[1†], Qitian Wu[2], Jiancan Wu[1], Qing Cui[3], Longfei Li[3], Jun Zhou[3*], Xiang Wang[1*], Xiangnan He[1*]

[1]University of Science and Technology of China,
[2]Shanghai Jiao Tong University, [3]Ant Group
syd2019@mail.ustc.edu.cn, echo740@sjtu.edu.cn,
{wujcan,xiangwang1223,xiangnanhe}@gmail.com,
{cuiqing.cq,longyao.llf,jun.zhoujun}@antgroup.com

Abstract

The issue of distribution shifts is emerging as a critical concern in graph representation learning. From the perspective of invariant learning and stable learning, a recently well-established paradigm for out-of-distribution generalization, stable features of the graph are assumed to causally determine labels, while environmental features tend to be unstable and can lead to the two primary types of distribution shifts. The correlation shift is often caused by the spurious correlation between environmental features and labels that differs between the training and test data; the covariate shift often stems from the presence of new environmental features in test data. However, most strategies, such as invariant learning or graph augmentation, typically struggle with limited training environments or perturbed stable features, thus exposing limitations in handling the problem of covariate shift. To address this challenge, we propose a simple-yet-effective data augmentation strategy, Adversarial Invariant Augmentation (AIA), to handle the covariate shift on graphs. Specifically, given the training data, AIA aims to extrapolate and generate new environments, while concurrently preserving the original stable features during the augmentation process. Such a design equips the graph classification model with an enhanced capability to identify stable features in new environments, thereby effectively tackling the covariate shift in data. Extensive experiments with in-depth empirical analysis demonstrate the superiority of our approach. The implementation codes are publicly available at https://github.com/yongduosui/AIA.

1 Introduction

While recent advances have made solid progress in learning effective representations for graph-structured data, most of the existing approaches operate under the assumption that training and test graphs are independently drawn from an identical distribution [1, 2]. However, this assumption often falls short in real-world scenarios due to the out-of-distribution (OOD) data that potentially exists during the testing phase [3], which results in distribution shifts between training and test graphs. As a result, there is increasing research interest in OOD generalization on graphs or learning with distribution shifts on graphs [4]. Some of the typical recent works attempt to build effective methods for handling general distribution shifts on graphs, from (causal) invariant learning [5, 3], model architecture designs [6], and data augmentation [7].

[†]This work was done during author's internship at Ant Group.
[*]Corresponding authors. Xiang Wang and Xiangnan He are also affiliated with Institute of Artificial Intelligence, Institute of Dataspace, Hefei Comprehensive National Science Center.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

With more specific views, other attempts focus on designing generalizable models for tackling distribution shifts in particular domains with certain data formats, *e.g.,* molecular graphs [8], recommender systems [9, 10], and anomaly detection [11].

However, the majority of existing studies primarily focus on the correlation shift, one type of distribution shift concerning OOD generalization [12, 13], leaving another equally important type of distribution shift, *i.e.,* the covariate shift, largely under-explored in graph representation learning. From the perspective of invariant learning and stable learning [14, 15, 16], covariate shift is in stark contrast to correlation shift *w.r.t.* stable and environmental features of graph data. Specifically, according to the commonly-used graph generation hypothesis in prior studies [17, 18, 5, 8], there often exist stable features, which are informative features of the entire graphs and can reflect the predictive patterns in data. Based on this, the relationship between stable features and labels is assumed to be invariant across environments. The remaining features could be unstable and varying across different environments, which mainly causes the following two distribution shifts: (1) correlation shift indicates that environments and labels establish inconsistent statistical correlations in training and test data, under the assumption that test environments are covered in the training data; whereas, (2) covariate shift characterizes that the environmental features in test data are unseen in training data [2, 12].

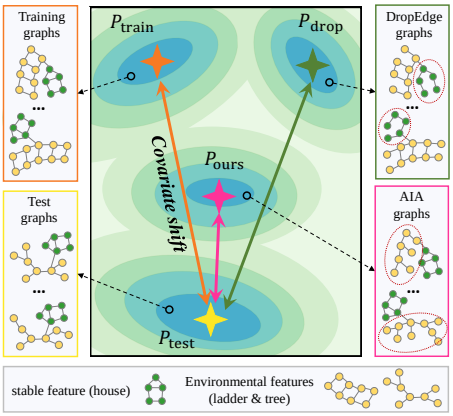

Figure 1: P_{train} and P_{test} denote the training and test distributions. P_{drop} and P_{ours} represent the distributions of augmented data via DropEdge and AIA.

Considering a toy example in Figure 1, the environmental features *ladder* and *tree* are different in training and test data, which forms the covariate shift. Taking molecular property predictions as another example, functional groups (*e.g.,* nitrogen dioxide (NO_2)) are stable to determine the predictive property of molecules [5, 8]. Whereas, scaffolds (*e.g.,* carbon rings) are usually patterns irrelevant to the molecule properties, which can be seen as environmental features [2, 19]. In practice, we often need to use molecules collected in the past to train models, expecting that the models can predict the properties of molecules with new scaffolds in the future [20, 2, 19].

Considering the differences between correlation and covariate shifts, we take a close look into the existing efforts on graph generalization. They mainly fall into the following research lines, each of which has inherent limitations to solving the covariate shift. *i) Invariant Graph Learning* [17, 18, 5, 21] recently becomes a prevalent paradigm for OOD generalization. The basic idea is to capture stable features by minimizing the empirical risks in different environments. Unfortunately, it implicitly makes a prior assumption that test environments are available during training. This assumption is unrealistic owing to the obstacle of training data covering all possible test environments. Learning in limited environments can alleviate the spurious correlations that are hidden in the training data, but fail to extrapolate the test data with unseen environments. *ii) Graph Data Augmentation* [22, 23] perturbs graph features to enrich the distribution seen during training for better generalization. It can be roughly divided into node-level [24], edge-level [25], and graph-level [26, 7] with random [27] or adversarial strategies [28]. However, blindly augmenting the graphs can presumably destroys the stable features, and makes the augmented distributions out of control. For example, in Figure 1, the random strategy of DropEdge [25] will inevitably perturb the stable features (highlighted by red circles). As such, it may not sufficiently address the covariate shift and could potentially affect the generalization ability. Hence, we naturally ask a question: "*Compared to the training data, can we generate new data that satisfy two conditions: 1) having new environments; 2) keeping the original stable features unchanged?*"

Towards this end, we introduce two intuitive principles for graph augmentation: environmental feature discrepancy and stable feature consistency. The discrepancy principle promotes the exploration of new environments beyond the scope of training data, while the consistency principle seeks to maintain the integrity of stable features during augmentation. In order to achieve these principles, we devise a simple yet effective graph augmentation strategy: Adversarial Invariant Augmentation (AIA). Specifically, we employ an adversarial augmenter, a network that augments graphs by adversarially generating masks on them, thereby facilitating OOD exploration to enhance environmental discrep-

ancy. To foster stable feature consistency, we use another network, *i.e.,* stable feature generator, which constructs masks that encapsulate stable features. We then delicately merge these masks and apply them to the graph data. As depicted in Figure 1, AIA primarily augments environmental features while leaving the stable elements unchanged. Our approach equips the graph classifier model with an enhanced ability to identify stable features in new environments and effectively mitigate the covariate shift issue. We also conduct extensive experiments and in-depth analyses. The experimental results highlight the limitations of several previous methods and underscore the superiority of our method in addressing the covariate shift issue, providing empirical support for our claims.

2 Preliminaries

We define the uppercase letters (*e.g.,* G) as random variables. The lower-case letters (*e.g.,* g) are samples of variables, and the blackboard bold typefaces (*e.g.,* $\mathbb{G}$) denote the sample spaces. Let $g = (A; X) \in \mathbb{G}$ denote a graph, where A and X are its adjacency matrix and node features, respectively. It is assigned with a label $y \in \mathbb{Y}$ with a fixed labeling rule $\mathbb{G} \to \mathbb{Y}$. Let $\mathbb{D} = \{(g_i; y_i)\}$ denote a dataset that is divided into a training set $\mathbb{D}_{tr} = \{(g_i^e; y_i^e)\}_{e \in \mathbb{E}_{tr}}$ and a test set $\mathbb{D}_{te} = \{(g_i^e; y_i^e)\}_{e \in \mathbb{E}_{te}}$. $\mathbb{E}_{tr}$ and $\mathbb{E}_{te}$ are the index sets of training and test environments, respectively.

2.1 Definitions and Problem Formations

In this work, we focus on the graph classification scenario, which aims to train models with $\mathbb{D}_{tr}$ and infer the labels in $\mathbb{D}_{te}$, such as molecular property prediction. From the viewpoints of invariant learning and stable learning, the inner mechanism of the labeling rule $\mathbb{G} \to \mathbb{Y}$ is usually assumed to depend on the stable features [17, 18, 5, 8, 21], which are informative substructures of the entire graph. The relationship between the stable features and labels is assumed to be invariant across different environments, which makes OOD generalization possible [12]. Environmental features in graph data are assumed to have no causal-effect on labels. For instance, the chemical properties of molecules are mainly determined by specific functional groups, which can be regarded as stable features [17, 5, 19, 8]. Conversely, their scaffold structures, often irrelevant to their properties, can be seen as environmental features [20, 8].

Due to the instability of environmental features and the limitations in the data collection process, the training and test distributions are often inconsistent in real-world scenarios, *i.e.,* $P_{tr}(G; Y) \neq P_{te}(G; Y)$, which leads to two main types of distribution shifts [2, 12]: (1) Correlation shift (*aka.* spurious correlation or concept shift [2]) refers to $P_{tr}(G|Y) \neq P_{te}(G|Y); P_{tr}(G) = P_{te}(G)$. It indicates that there exist spurious statistical correlations in the training data, while these correlations might not hold in the test data. (2) Covariate shift denotes $P_{tr}(G|Y) = P_{te}(G|Y); P_{tr}(G) \neq P_{te}(G)$, which means that there exist new features, *e.g.,* environmental features, in the test data. It may be ascribed to either insufficient quantity or diversity of data in the training set, as well as the unknown characteristics of the test environments. We provide formal definitions and examples of these two distribution shifts in Appendix A. Here, inspired by the prior study [12], we offer a formal definition to measure the graph covariate shift.

Definition 2.1 (Graph Covariate Shift) *Let P_{tr} and P_{te} denote the probability functions of the training and test distributions. We measure the covariate shift between distributions P_{tr} and P_{te} as*

$$\mathrm{GCS}(P_{tr}; P_{te}) = \frac{1}{2} \int_{\mathbb{S}} |P_{tr}(g) - P_{te}(g)| dg; \tag{1}$$

where $\mathbb{S} = \{g \in \mathbb{G} \mid P_{tr}(g) \cdot P_{te}(g) = 0\}$, which covers the features (e.g., environmental features) that do not overlap between the two distributions.

$\mathrm{GCS}(\cdot; \cdot)$ is always bounded in $[0; 1]$. The issue of graph covariate shift is very common in practice. For example, we often need to train models on past molecular graphs, and hope that the model can predict the chemical properties of future molecules with new features, *e.g.,* new scaffolds [20]. In addressing the graph OOD issue, a majority of the strategies [17, 5, 18, 29, 21] grounded in invariant graph learning aim to pinpoint stable or invariant features. This is mainly accomplished by minimizing empirical risks across an array of training environments. Nonetheless, these methodologies frequently operate under the assumption of a shared input space across training and test data. This assumption, however, often fails on covariate shift. It presents substantial challenges for these models when it

comes to accurately identifying stable features within new testing environments. Consequently, while these methods typically exhibit satisfactory performance in managing correlation shifts, they often underperform in the face of covariate shifts. In this work, we focus on the covariate shift issue in graph classification, and we also give a formal definition of this problem as follows.

Problem 2.2 (Graph Classification under Covariate Shift) *Given the training and test sets with environment sets* E_{tr} *and* E_{te}*, they follow distributions* P_{tr} *and* P_{te}*, and satisfy:* $\text{GCS}(P_{\text{tr}}; P_{\text{te}}) > \ $*, where* $\in (0; 1)$ *represents the degree of covariate shift. We aim to use the data collected from training environments* E_{tr}*, and learn a powerful graph classifier* $f^{\square} \in \mathsf{G} \to \mathsf{Y}$ *that performs well in all possible test environments* E_{te}*:*

$$f^{\square} = \arg\min_{f} \sup_{e \in \mathsf{E}_{\text{te}}} \mathsf{E}^{e}[\ell(f(g); y)]; \tag{2}$$

where $\mathsf{E}^{e}[\ell(f(g); y)]$ *is the empirical risk on the environment* e*, and* $\ell(\cdot; \cdot)$ *is the loss function.*

3 Methodology

To solve Problem 2.2, our idea is to generate new graphs through data augmentation. In this section, we first propose two principles for graph augmentation. Guided by these principles, we design a novel graph augmentation method, AIA, which can effectively address the covariate shift issue.

3.1 Two Principles for Graph Augmentation

We can observe that covariate shift is mainly caused by the scarcity of training environments. Hence, we first propose the discrepancy principle for graph augmentation.

Principle 3.1 (Environmental Feature Discrepancy) *Given a graph set* $\{g\}$ *with distribution function* P*, let* $T(\cdot)$ *denote an augmentation function that augments graphs* $\{T(g)\}$ *to distribution* $\tilde{P}$*. Then* $T(\cdot)$ *should meet* $\text{GCS}(P; \tilde{P}) \to 1$*.*

From the perspective of data distribution, it requires that $\tilde{P}$ should keep away from the original distribution P. From the perspective of data instances, it emphasizes the discrepancy in the environments between the generated graphs and the original graphs. Since it does not give constraints on stable features, we here propose the second principle for graph augmentation.

Principle 3.2 (Stable Feature Consistency) *Given a set of graphs* $\{g\}$ *with a corresponding stable feature set* $\{g_{\text{sta}} = (\mathsf{A}_{\text{sta}}; \mathsf{X}_{\text{sta}})\}$*. Let* $T(\cdot)$ *denote an augmentation function that augments graphs* $\{T(g)\}$ *with a corresponding stable feature set* $\{\tilde{g}_{\text{sta}} = (\tilde{\mathsf{A}}_{\text{sta}}; \tilde{\mathsf{X}}_{\text{sta}})\}$*. Then* $T(\cdot)$ *should meet* $\mathsf{E}[\square \mathsf{A}_{\text{sta}} - \tilde{\mathsf{A}}_{\text{sta}} \square_F^2] \to 0$ *and* $\mathsf{E}[\square \mathsf{X}_{\text{sta}} - \tilde{\mathsf{X}}_{\text{sta}} \square_F^2] \to 0$*, where* $\square \cdot \square_F$ *is the Frobenius norm.*

It necessitates that the stable features of the generated graphs should maintain consistency with those of the original graphs. This principle ensures the preservation of these stable features within the original training data, thereby safeguarding sufficient information pertaining to the labels. Consequently, this principle enhances the potential for generalization.

3.2 Out-of-distribution Exploration

Given a GNN model $f(\cdot)$ with parameters $\ $, we decompose $f = \ \circ h$, where $h(\cdot) \in \mathsf{G} \to \mathsf{R}^d$ is a graph encoder to yield d-dimensional representations, and $(\cdot) \in \mathsf{R}^d \to \mathsf{Y}$ is a classifier. To comply with Principle 3.1, we need to do OOD exploration. Inspired by distributionally robust optimization [30, 31], we consider the following optimization objective:

$$\min \{ \sup_{\tilde{P}} \{ \mathsf{E}_{\tilde{P}}[\ell(f(g); y)] \mid D(\tilde{P}; P) \leq \ \} \}; \tag{3}$$

where P and $\tilde{P}$ are the original and explored data distributions, respectively. $D(\cdot; \cdot)$ is a distance metric between two probability distributions. The solution to Equation (3) guarantees the generalization within a robust radius $\ $ of the distribution P. To better measure the distance between distributions,

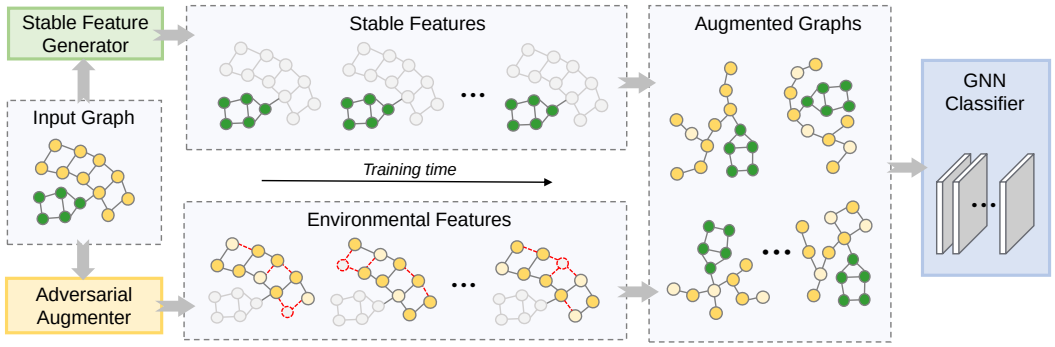

Figure 2: The overview of Adversarial Invariant Augmentation (AIA) Framework.

as suggested by [32], we adopt the Wasserstein distance [33, 34] as the distance metric. The distance metric function can be defined as $D(\tilde{P};P) = \inf_{\Pi(\tilde{P};P)} \mathbb{E}[c(\tilde{g};g)]$, where $\Pi(\tilde{P};P)$ is the set of all couplings of $\tilde{P}$ and P; $c(\cdot;\cdot)$ is the cost function. Studies [35, 34] also suggest that the distances in representation space typically correspond to semantic distances. Hence, we define the cost function in the representation space and give the transportation cost as $c(\tilde{g};g) = \|h(\tilde{g}) - h(g)\|_2^2$. It denotes the "cost" of augmenting the graph g to $\tilde{g}$. We can observe that it is difficult to set a proper γ. Instead, we consider the Lagrangian relaxation for a fixed penalty coefficient γ. Inspired by [32], we can reformulate Equation (3) as follows:

$$\min\{\sup_{\tilde{P}}\{\mathbb{E}_{\tilde{P}}[\ell(f(g);y)] - \gamma D(\tilde{P};P)\} = \mathbb{E}_P[\phi_\gamma(f(g);y)]\}; \tag{4}$$

where $\phi_\gamma(f(g);y) := \sup_{\tilde{g}\in\mathcal{G}}\{\ell(f(\tilde{g});y) - \gamma c(\tilde{g};g)\}$. And we define $\phi_\gamma(f(g);y)$ as the robust surrogate loss. If we conduct gradient descent on the robust surrogate loss, we will have:

$$\nabla_\theta \phi_\gamma(f(g);y) = \nabla_\theta \ell(f(\tilde{g}^\gamma);y), \text{ where } \tilde{g}^\gamma = \arg\max_{\tilde{g}\in\mathcal{G}}\{\ell(f(\tilde{g});y) - \gamma c(\tilde{g};g)\}. \tag{5}$$

$\tilde{g}^\gamma$ is an augmented view of the original data g. Hence, to achieve OOD exploration, we need to perform graph data augmentation via Equation (5) on the original data g.

3.3 Implementations of AIA

Equation (5) endows the ability of OOD exploration to data augmentation, which makes the augmented data meet the discrepancy principle. To achieve the consistency principle, we also need to capture stable features. Hence, we design a graph augmentation strategy: Adversarial Invariant Augmentation (AIA). The overview of the proposed framework is depicted in Figure 2, which mainly consists of two components: adversarial augmenter and stable feature generator. Adversarial augmenter achieves OOD exploration through adversarial data augmentation; meanwhile, the stable feature generator keeps stable feature consistency by identifying stable features from data. Below we elaborate on the implementation details.

Adversarial Augmenter & Stable Feature Generator. We design two networks, adversarial augmenter $T_{\phi_1}(\cdot)$ and stable feature generator $T_{\phi_2}(\cdot)$, which generate masks for nodes and edges of graphs. They have the same structure and are parameterized by ϕ_1 and ϕ_2, respectively. Given an input graph $g = (\mathbf{A}; \mathbf{X})$ with n nodes, mask generation network first obtains the node representations via a GNN encoder $\tilde{h}(\cdot)$. To judge the importance of nodes and edges, it adopts two MLP networks $\mathrm{MLP}_1(\cdot)$ and $\mathrm{MLP}_2(\cdot)$ to generate the soft node mask matrix $\mathbf{M}^x \in \mathbb{R}^{n\times 1}$ and edge mask matrix $\mathbf{M}^a \in \mathbb{R}^{n\times n}$ for graph data, respectively. In summary, the mask generation network can be decomposed as:

$$\mathbf{Z} = \tilde{h}(g); \quad \mathbf{M}^x_i = \sigma(\mathrm{MLP}_1(\mathbf{h}_i)); \quad \mathbf{M}^a_{ij} = \sigma(\mathrm{MLP}_2([\mathbf{z}_i;\mathbf{z}_j])); \tag{6}$$

where $\mathbf{Z} \in \mathbb{R}^{n\times d}$ is node representation matrix, whose i-th row $\mathbf{z}_i = \mathbf{Z}[i;]$ denotes the representation of node i, and $\sigma(\cdot)$ is the sigmoid function that maps the mask values $\mathbf{M}^x_i$ and $\mathbf{M}^a_{ij}$ to $[0;1]$.

Adversarial Invariant Augmentation. To estimate $\tilde{g}^{\square}$ in Equation (5), we define the following adversarial learning objective:

$$\max_{1} \{\mathsf{L}_{\text{adv}} = \mathsf{E}_{P_{\text{tr}}}[\ell(f(T_{1}(g)), y) - c(T_{1}(g), g)]\}; \tag{7}$$

Then we can augments the graph by $T_{1}(g) = (\mathsf{A} \square \mathsf{M}^{a}_{\text{adv}}, \mathsf{X} \square \mathsf{M}^{x}_{\text{adv}})$, where $\square$ is the broadcasted element-wise product. Although adversarially augmented graphs guarantee environmental discrepancy, it might destroys the stable parts. Therefore, we utilize the stable feature generator $T_{2}(\square)$ to capture stable features and combine them with different environmental features. Following the sufficiency and invariance conditions [17, 3, 18, 5, 8, 29], we define the stable feature learning objective as:

$$\min_{;\,2} \{\mathsf{L}_{\text{sta}} = \mathsf{E}_{P_{\text{tr}}}[\ell(f(T_{2}(g)), y) + \ell(f(\tilde{g}), y)]\}; \tag{8}$$

where $\tilde{g} = (\mathsf{A} \square \tilde{\mathsf{M}}^{a}, \mathsf{X} \square \tilde{\mathsf{M}}^{x})$ is the augmented graph. It adopts the mask combination strategy: $\tilde{\mathsf{M}}^{a} = (1^{a} - \mathsf{M}^{a}_{\text{sta}}) \square \mathsf{M}^{a}_{\text{adv}} + \mathsf{M}^{a}_{\text{sta}}$ and $\tilde{\mathsf{M}}^{x} = (1^{x} - \mathsf{M}^{x}_{\text{sta}}) \square \mathsf{M}^{x}_{\text{adv}} + \mathsf{M}^{x}_{\text{sta}}$, where $\mathsf{M}^{a}_{\text{sta}}$ and $\mathsf{M}^{x}_{\text{sta}}$ are generated by $T_{2}(\square)$, 1^{a} and 1^{x} are all-one matrices, and if there is no edge between node i and node j, then we set 1^{a}_{ij} to 0. Now we explain this combination strategy. Taking $\tilde{\mathsf{M}}^{x}$ as an example, since $\mathsf{M}^{x}_{\text{sta}}$ denotes the captured stable regions via $T_{2}(\square)$, $1^{x} - \mathsf{M}^{x}_{\text{sta}}$ represents the complementary parts, which are environmental regions. $\mathsf{M}^{x}_{\text{adv}}$ represents the adversarial perturbation, so $(1^{x} - \mathsf{M}^{x}_{\text{sta}}) \square \mathsf{M}^{x}_{\text{adv}}$ is equivalent to applying the adversarial perturbation on environmental features, meanwhile, sheltering the stable features. Finally, $+\mathsf{M}_{\text{sta}}$ signifies that the augmented data should preserve the original stable features. Consequently, it satisfies both principles. Upon analysis of Equation (8), the first term implies that the stable features are sufficient for making right predictions. The second term promotes right and invariant predictions under generated environments utilizing stable features.

Regularization. For Equation (7), the adversarial optimization tends to remove more nodes and edges, so we should also constrain the perturbations. Although Equation (8) satisfies the sufficiency and invariance conditions, it is necessary to impose constraints on the ratio of the stable features to prevent trivial solutions. Hence, we first define the regularization function $r(\mathsf{M}; k; \ ) = (\sum_{ij} \mathsf{M}_{ij}/k - \ ) + (\sum_{ij} \mathsf{I}[\mathsf{M}_{ij} > 0]/k - \ )$, where k is the total number of elements to be constrained, $\mathsf{I} \ \{0; 1\}$ is an indicator function. The first term penalizes the average ratio close to , while the second term encourages an uneven distribution. Given a graph with n nodes and m edges, we define the regularization term for adversarial augmentation and stable feature learning as:

$$\mathsf{L}_{\text{reg}_1} = \mathsf{E}_{P_{\text{tr}}}[r(\mathsf{M}^{x}_{\text{adv}}; n; \ _{a}) + r(\mathsf{M}^{a}_{\text{adv}}; m; \ _{a})]; \tag{9}$$

$$\mathsf{L}_{\text{reg}_2} = \mathsf{E}_{P_{\text{tr}}}[r(\mathsf{M}^{x}_{\text{sta}}; n; \ _{s}) + r(\mathsf{M}^{a}_{\text{sta}}; m; \ _{s})]; \tag{10}$$

where $_{s} \ (0; 1)$ is the ratio of stable features, we usually set $_{a} = 1$ for adversarial learning, which can alleviate excessive perturbations. The algorithm is provided in Appendix D.1.

4 Theoretical Discussions

In this section, we engage in theoretical discussions to elucidate our learning objective and its connections with the covariate shift. We first explore the relationship between our optimization objective and the discrepancy principle. Recalling the optimization objective of Equation 3, we aspire to identify a distribution $\tilde{P}$ that can manifest within a Wasserstein ball [36], which is centered on distribution P, with distance serving as the radius. Under appropriate conditions, we find that our learning optimization objective can establish a close connection with OOD exploration.

Proposition 4.1 *Consider a probability distribution P defined over a measurable space (; F), where denotes the sample space and F is a -algebra on . We construct two Wasserstein balls with P at their center and radii $_{1}$ and $_{2}$ respectively. Utilizing Equation 3, we generate two distinct distributions, $\tilde{P}_1$ and $\tilde{P}_2$, within the space (; F). If (i) P is an isotropic distribution; (ii) $\square x_1; x_2 \square$ such that $P(x) = \tilde{P}_1(x - x_1) = \tilde{P}_2(x - x_2)$; (iii) $_{1} < \ _{2}$, then we have $\text{GCS}(P; \tilde{P}_1) \le \text{GCS}(P; \tilde{P}_2)$.*

This suggests that by appropriately increasing the robustness radius in AIA, we can effectively amplify the covariate shift between the training and generated distributions. This in turn underscores the reliability of our discrepancy principle, to a certain degree. Comprehensive proofs and detailed discussions supporting these conclusions can be found in Appendix B.

Table 1: Performance on synthetic and real-world datasets. Numbers in **bold** indicate the best performance, while the underlined numbers indicate the second best performance.

Type	Method	Motif		CMNIST	Molbbbp		Molhiv	
		base	size	color	scaffold	size	scaffold	size
General Generalization	ERM	68.66±4.25	51.74±2.88	28.60±1.87	68.10±1.68	78.29±3.76	69.58±2.51	59.94±2.37
	IRM	70.65±4.17	51.41±3.78	27.83±2.13	67.22±1.15	77.56±2.48	67.97±1.84	59.00±2.92
	GroupDRO	68.24±8.92	51.95±5.86	29.07±3.14	66.47±2.39	79.27±2.43	70.64±2.57	58.98±2.16
	VREx	71.47±6.69	52.67±5.54	28.48±2.87	68.74±1.03	78.76±2.37	70.77±2.84	58.53±2.88
Graph Generalization	DIR	62.07±8.75	52.27±4.56	33.20±6.17	66.86±2.25	76.40±4.43	68.07±2.29	58.08±2.31
	CAL	65.63±4.29	51.18±5.60	27.99±3.24	68.06±2.60	79.50±4.81	67.37±3.61	57.95±2.24
	GSAT	62.80±11.41	53.20±8.35	28.17±1.26	66.78±1.45	75.63±3.83	68.66±1.35	58.06±1.98
	OOD-GNN	61.10±7.87	52.61±4.67	26.49±2.94	66.72±1.23	79.48±4.19	70.46±1.97	60.60±3.77
	StableGNN	57.07±14.10	46.93±8.85	28.38±3.49	66.74±1.30	77.47±4.69	68.44±1.33	56.71±2.79
	CIGA	66.43±11.31	49.14±8.34	32.22±2.67	64.92±2.09	65.98±3.31	69.40±2.39	59.55±2.56
	DisC	51.08±3.08	50.39±1.15	24.99±1.78	67.12±2.11	56.59±10.09	68.07±1.75	58.76±0.91
Graph Augmentation	DropEdge	45.08±4.46	45.63±4.61	22.65±2.90	66.49±1.55	78.32±3.44	70.78±1.38	58.53±1.26
	GREA	56.74±9.23	54.13±10.02	29.02±3.26	69.72±1.66	77.34±3.52	67.79±2.56	60.71±2.20
	FLAG	61.12±5.39	51.66±4.14	32.30±2.69	67.69±2.36	79.26±2.26	68.45±2.30	60.59±2.95
	M-Mixup	70.08±3.82	51.48±4.91	26.47±3.45	68.75±0.34	78.92±2.43	68.88±2.63	59.03±3.11
	G-Mixup	59.66±7.03	52.81±6.73	31.85±5.82	67.44±1.62	78.55±4.16	70.01±2.52	59.34±2.43
	AIA (ours)	**73.64±5.15**	**55.85±7.98**	**36.37±4.44**	**70.79±1.53**	**81.03±5.15**	**71.15±1.81**	**61.64±3.37**

5 Experiments

In this section, we conduct extensive experiments to answer the following **R**esearch **Q**uestions:

- **RQ1:** Compared to existing efforts, how does AIA perform under covariate shift?
- **RQ2:** Can the proposed AIA achieve the principles of environmental feature discrepancy and stable feature consistency, thereby alleviating the graph covariate shift?
- **RQ3:** How do the different components and hyperparameters of AIA affect the performance?

5.1 Experimental Settings

Datasets. We use graph OOD datasets [2] and OGB datasets [20], which include Motif, CMNIST, Molbbbp, and Molhiv. Following [2], we adopt the base, color, size, and scaffold data splitting to create various covariate shifts. The details of the datasets, metrics, implementations, and other settings are provided in Appendix D.2 and D.4. More experiments are provided in Appendix E.

Baselines. We adopt 16 baselines, which can be divided into the following three specific categories:

- **General Generalization Algorithms:** ERM, IRM [14], GroupDRO [31], VREx [37].
- **Graph Generalization Algorithms:** DIR [17], CAL [5], GSAT [38], OOD-GNN [39], StableGNN [40], CIGA [41], DisC [21].
- **Graph Augmentation:** DropEdge [25], GREA [18], FLAG [24], M-Mixup [26], G-Mixup [7].

5.2 Main Results (RQ1)

We first make comparisons with various baselines in Table 1, and have the following observations:

Most generalization and augmentation methods fail under covariate shift. VREx achieves a 2.81% improvement on Motif (base). For two shifts of Molhiv, data augmentation methods GREA and DropEdge obtain 1.20% and 0.77% improvements. The invariant learning methods, *i.e.,* DIR and CAL also obtain 4.60% and 1.53% improvements on CMNIST and Molbbbp (size). Unfortunately, none of the methods consistently outperform ERM. For example, GREA and DropEdge perform poorly on Motif (base), ↓11.92% and ↓23.58%. DIR and CAL also fail on Molhiv. These show that both invariant learning and data augmentation methods have their own weaknesses, which lead to unstable performance when facing complex and diverse covariate shifts from different datasets.

AIA consistently outperforms most baseline methods. Compared with ERM, AIA can obtain significant improvements. For two types of covariate shifts on Motif, AIA surpasses ERM by 4.98% and 4.11%, respectively. In contrast to the large performance variances on different datasets achieved by

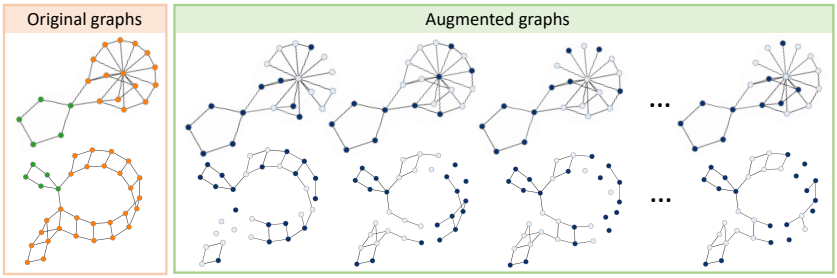

Figure 3: Visualizations of the augmented graphs via AIA.

Table 2: Covariate shift comparisons with different augmentation strategies.

Method	Motif (base)		Motif (size)		CMNIST (color)		Molbbbp (scaffold)	
	Aug-Train	Aug-Test	Aug-Train	Aug-Test	Aug-Train	Aug-Test	Aug-Train	Aug-Test
Original	0	0.557±0.141	0	0.522±0.421	0	0.490±0.226	0	0.419±0.079
DropEdge	0.772±0.213	0.515±0.033	0.851±0.138	0.161±0.271	0.627±0.186	0.539±0.260	0.758±0.192	0.737±0.211
FLAG	0.001±0.001	0.533±0.016	0.002±0.018	0.507±0.121	0.003±0.002	0.442±0.062	0.001±0.001	0.413±0.088
G-Mixup	0.690±0.186	0.472±0.043	0.816±0.154	0.299±0.343	0.408±0.228	0.351±0.318	0.551±0.258	0.545±0.231
AIA (ours)	0.369±0.169	0.462±0.063	0.649±0.143	0.098±0.070	0.516±0.106	0.307±0.108	0.422±0.049	0.393±0.028

baselines, AIA consistently obtains the leading performance across the board. For CMNIST, AIA achieves a performance improvement of 3.17% compared to the best baseline DIR. For Motif, the performance is improved by 2.17% and 1.72% compared to VREx and GREA. These results illustrate that AIA can overcome the shortcomings of invariant learning and data augmentation. Armed with the principles of environmental feature diversity and stable feature invariance, AIA achieves stable and consistent improvements on different datasets with various covariate shifts. In addition, although we focus on covariate shift in this work, we also carefully check the performance of AIA under correlation shift, and the results are presented in Appendix E.

5.3 In-depth Analyses (RQ2)

In this section, we conduct qualitative and quantitative experiments to support our two principles. Firstly, we utilize $\mathsf{GCS}(\mathbb{I};\mathbb{I})$ as the measurement to quantify the degree of covariate shift. The detailed estimation procedure is provided in Appendix C. We select four different domains, *i.e.,* base, size, color and scaffold, to create covariate shifts. The experimental results are shown in Table 2. We calculated covariate shifts between the augmentation distribution P_{aug} with the training P_{tr} or test distribution P_{te}. "Aug-Train" and "Aug-Test" represent $\mathsf{GCS}(P_{\mathsf{aug}};P_{\mathsf{tr}})$ and $\mathsf{GCS}(P_{\mathsf{aug}};P_{\mathsf{te}})$, respectively. From the results in Table 2, we make these observations.

Discrepancy Principle. The term "Original" denotes the training distribution prior to augmentation. It's observed that substantial covariate shifts exist between the training and test distributions, ranging from 0.419 to 0.557. The DropEdge technique notably expands *Aug-Train*, with a range of 0.627 to 0.851, while concurrently increasing *Aug-Test*, as evidenced by CMNIST (0.490 to 0.539) and Molbbbp (0.419 to 0.737). However, a distribution that deviates excessively from the test distribution may not effectively address the issue of covariate shift. FLAG, which perturbs only the node features, yields minor values in both *Aug-Train* and *Aug-Test*. G-Mixup notably augments *Aug-Train* by generating OOD samples, but doesn't necessarily limit *Aug-Test*. Finally, AIA extends the disparity with the training distribution by augmenting environmental features, signifying that AIA can adeptly implement the principle of environmental feature discrepancy. Simultaneously, the imposed consistency constraint on stable features restricts the generated distribution from straying too far from the test distribution, as observed in Motif-base (0.557 to 0.462), Motif-size (0.522 to 0.098), and CMNIST (0.490 to 0.307).

Augmentation Diversity. We further delve into the diversity of data augmentation. The concept of augmentation diversity stems from the intuition that augmentations with greater degrees of freedom yield better performance [42]. Accordingly, we propose conditional entropy to

Table 3: Augmentation Diversity.

Method	Full Graph	Env. Feature	Sta. Feature
DropEdge	0.999±0.065	0.933±0.029	0.971±0.067
AIA (ours)	0.561±0.223	0.508±0.136	0.259±0.106

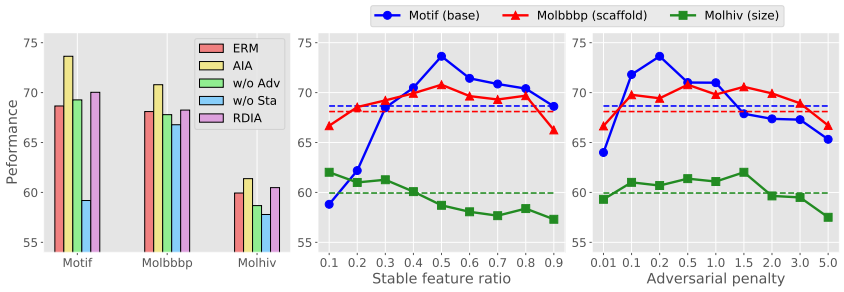

Figure 4: (*Left*): Different components in AIA. (*Middle*): Different ratios $_s$ of stable features. (*Right*): Different penalties . Dashed lines denote the ERM.

measure the diversity of generated data: $H(G|G) = -E_G[\sum_g p(g|G)\log(p(g|G))]$, where G and G represent the generated graph and original graph, respectively. To substantiate our ability to manage perturbed environmental features while preserving stable features, we examine diversity at the feature level, *i.e.,* stable and environmental features. We employ the Motif dataset for validation due to its inclusion of ground-truth labels for stable and environmental features. The results in Table 3 reveal that our approach can guarantee the diversity of environmental features while constraining the variation of stable features.

To substantiate two principles of AIA, we present a selection of augmented graphs in Figure 3. These graphs are randomly sampled during the training phase. In the original Motif, the stable feature, is represented by the green portion and its type determines the label, while the yellow portion signifies the base-graph, or the environmental feature. Figure 3 (*Right*) exhibits the augmented samples generated during training. Nodes depicted in darker colors and edges with broader lines indicate higher soft-mask values. These results lead us to several noteworthy observations.

Visualization Analyses. AIA primarily perturbs the environmental features, while leaving the stable components undisturbed. In the Motif dataset, the base-graph represents a *ladder* and the motif-graph signifies a *house*. Following augmentation, the nodes and edges of the *ladder* graph undergo perturbations. However, the *house* component remains consistently stable throughout the training process. It shows that AIA successfully adheres to the proposed two principles, thus providing empirical support for our claims. Furthermore, under covariate shift, we also depict the stable features identified by AIA in comparison to other baseline methods (refer to Appendix E.4). It further underscores the limitations of alternative methods and highlights the superior performance of AIA.

5.4 Ablation Study (RQ3)

Adversarial Augmentation & Stable Feature Learning. As illustrated in Figure 4 (*Left*), "w/o Adv" and "w/o Sta" denote AIA without adversarial augmentation and without stable feature learning, respectively. RDIA is a variant that replaces adversarial augmentation in AIA with random augmentation (*i.e.,* random masks). The performance degrades when either component is used independently, compared to their combined application in AIA. The removal of adversarial perturbations results in a loss of the invariance condition inherent in stable feature learning [17, 29], leading to suboptimal outcomes. Conversely, the sole use of adversarial augmentation disrupts the stable features, thereby diminishing the performance. RDIA surpasses ERM, yet falls short of AIA, indicating that although randomness can foster discrepancy, it is less effective than the adversarial strategy.

Sensitivity Analysis. We conduct experiments to explore the sensitivities of ratio $_s$ and penalty coefficient . The results are displayed in Figure 4 (*Middle*) and (*Right*). $_s$ with 0.3□0.8 performs well on Motif and Molbbbp, while Molhiv is better in 0.1□0.3. It indicates that $_s$ is a dataset-sensitive hyper-parameter that needs careful tuning. For , the appropriate values range from 0.1□1.5.

6 Related Work

Graph Data Augmentation [22, 23, 43] enlarges the training distribution by perturbing features in graphs. Recent studies [44, 13] observe that it often outperforms other generalization efforts [14, 31]. DropEdge [25] randomly removes edges, while FLAG [24] augments node features with an adversarial strategy. M-Mixup [26] interpolates graphs in semantic space. However, studies [14, 45]

point out that stable features are the key to OOD generalization. These augmentation efforts are prone to perturb the stable features, which easily lose control of the augmented data distribution.

Invariant Graph Learning has been widely adopted by recent works as a paradigm for handling distribution shifts on graphs. The pioneering works [3, 17] leverage the causal invariance principle to model the invariant predictive patterns in data for the OOD generalization purpose. With the similar spirit, GREA [18] and CAL [5] aim to learn stable features by considering different environments. Some other works also utilize invariant learning to develop generalizable models and algorithms for improving the generalization *w.r.t.* molecular graphs [8] and recommender systems [9].

Out-of-Distribution Learning on Graphs has aroused wide research interest in the graph learning community. One line of research is centered around the goal of improving the OOD generalization capabilities of models when encountered with test data from new unseen distributions [46, 3, 47, 39, 40, 17, 38, 48, 5]. Another line of research, differently, aims to identify the OOD data in the testing set and improve the reliablity of models against OOD data for which the model should reject for prediction [49, 50, 51]. The latter task is called Out-of-Distribution Detection in the literature and serves as another under-explored area that has different technical aspect from the present work. Due to space constraints, we put more discussions of other related studies in Appendix G.

7 Conclusion

In this study, we address the pervasive yet largely unexplored issue of covariate shift in graph learning. We introduce a novel graph augmentation method, AIA, grounded in two principles: environmental feature discrepancy and stable feature consistency. The discrepancy principle enables the model to explore new environments, thereby facilitating better generalization to potentially unseen environments. Meanwhile, the consistency principle maintains the integrity of stable features. We conduct extensive comparisons with various baseline models and perform thorough analyses.

8 Limitations and Broader Impacts

This paper presents a graph augmentation method, AIA, designed to bolster the academic community's application of data augmentation methodologies. We do not foresee any immediate, direct, or adverse societal implications resulting from our study's findings. We also present additional discussions regarding AIA's limitations and potential future work in Appendix H.

Acknowledgments and Disclosure of Funding

This research is supported by the National Natural Science Foundation of China (9227010114, U19A2079, 62302321) and the University Synergy Innovation Program of Anhui Province (GXXT-2022-040). This work is also sponsored by Ant Group through CCF-Ant Research Fund and CCF-AFSG Research Fund.

References

[1] Zheyan Shen, Jiashuo Liu, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. Towards out-of-distribution generalization: A survey. *arXiv preprint arXiv:2108.13624*, 2021.

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

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

[4] Haoyang Li, Xin Wang, Ziwei Zhang, and Wenwu Zhu. Out-of-distribution generalization on graphs: A survey. *arXiv preprint arXiv:2202.07987*, 2022.

[5] Yongduo Sui, Xiang Wang, Jiancan Wu, Min Lin, Xiangnan He, and Tat-Seng Chua. Causal attention for interpretable and generalizable graph classification. In *KDD*, 2022.

[6] Chenxiao Yang, Qitian Wu, Jiahua Wang, and Junchi Yan. Graph neural networks are inherently good generalizers: Insights by bridging GNNs and MLPs. In *ICLR*, 2023.

[7] Xiaotian Han, Zhimeng Jiang, Ninghao Liu, and Xia Hu. G-mixup: Graph data augmentation for graph classification. In *ICML*, pages 8230–8248, 2022.

[8] Nianzu Yang, Kaipeng Zeng, Qitian Wu, Xiaosong Jia, and Junchi Yan. Learning substructure invariance for out-of-distribution molecular representations. In *NeurIPS*, 2022.

[9] Chenxiao Yang, Qitian Wu, Qingsong Wen, Zhiqiang Zhou, Liang Sun, and Junchi Yan. Towards out-of-distribution sequential event prediction: A causal treatment. In *NeurIPS*, 2022.

[10] An Zhang, Jingnan Zheng, Xiang Wang, Yancheng Yuan, and Tat-Seng Chua. Invariant collaborative filtering to popularity distribution shift. In *WWW*, pages 1240–1251, 2023.

[11] Yuan Gao, Xiang Wang, Xiangnan He, Zhenguang Liu, Huamin Feng, and Yongdong Zhang. Alleviating structural distribution shift in graph anomaly detection. In *WSDM*, pages 357–365, 2023.

[12] Nanyang Ye, Kaican Li, Haoyue Bai, Runpeng Yu, Lanqing Hong, Fengwei Zhou, Zhenguo Li, and Jun Zhu. Ood-bench: Quantifying and understanding two dimensions of out-of-distribution generalization. In *CVPR*, pages 7947–7958, 2022.

[13] Olivia Wiles, Sven Gowal, Florian Stimberg, Sylvestre Alvise-Rebuffi, Ira Ktena, Taylan Cemgil, et al. A fine-grained analysis on distribution shift. In *ICLR*, 2022.

[14] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[15] Shiyu Chang, Yang Zhang, Mo Yu, and Tommi Jaakkola. Invariant rationalization. In *ICML*, pages 1448–1458, 2020.

[16] Xingxuan Zhang, Peng Cui, Renzhe Xu, Linjun Zhou, Yue He, and Zheyan Shen. Deep stable learning for out-of-distribution generalization. In *CVPR*, pages 5372–5382, 2021.

[17] Yingxin Wu, Xiang Wang, An Zhang, Xiangnan He, and Tat-Seng Chua. Discovering invariant rationales for graph neural networks. In *ICLR*, 2022.

[18] Gang Liu, Tong Zhao, Jiaxin Xu, Tengfei Luo, and Meng Jiang. Graph rationalization with environment-based augmentations. In *KDD*, pages 1069–1078, 2022.

[19] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

[20] 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. In *NeurIPS*, 2020.

[21] Shaohua Fan, Xiao Wang, Yanhu Mo, Chuan Shi, and Jian Tang. Debiasing graph neural networks via learning disentangled causal substructure. In *NeurIPS*, 2022.

[22] Kaize Ding, Zhe Xu, Hanghang Tong, and Huan Liu. Data augmentation for deep graph learning: A survey. *arXiv preprint arXiv:2202.08235*, 2022.

[23] Tong Zhao, Gang Liu, Stephan Günnemann, and Meng Jiang. Graph data augmentation for graph machine learning: A survey. *arXiv preprint arXiv:2202.08871*, 2022.

[24] Kezhi Kong, Guohao Li, Mucong Ding, Zuxuan Wu, Chen Zhu, Bernard Ghanem, Gavin Taylor, and Tom Goldstein. Robust optimization as data augmentation for large-scale graphs. In *CVPR*, pages 60–69, 2022.

[25] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. In *ICLR*, 2020.

[26] Yiwei Wang, Wei Wang, Yuxuan Liang, Yujun Cai, and Bryan Hooi. Mixup for node and graph classification. In *WWW*, pages 3663–3674, 2021.

[27] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *NeurIPS*, 2020.

[28] Susheel Suresh, Pan Li, Cong Hao, and Jennifer Neville. Adversarial graph augmentation to improve graph contrastive learning. In *NeurIPS*, pages 15920–15933, 2021.

[29] Haoyang Li, Ziwei Zhang, Xin Wang, and Wenwu Zhu. Learning invariant graph representations for out-of-distribution generalization. In *NeurIPS*, 2022.

[30] Hamed Rahimian and Sanjay Mehrotra. Distributionally robust optimization: A review. *arXiv preprint arXiv:1908.05659*, 2019.

[31] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *ICLR*, 2020.

[32] Aman Sinha, Hongseok Namkoong, and John Duchi. Certifying some distributional robustness with principled adversarial training. In *ICLR*, 2018.

[33] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, pages 214–223, 2017.

[34] Riccardo Volpi, Hongseok Namkoong, Ozan Sener, John C Duchi, Vittorio Murino, and Silvio Savarese. Generalizing to unseen domains via adversarial data augmentation. *NeurIPS*, 31, 2018.

[35] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. *NIPS*, 2016.

[36] Jaeho Lee and Maxim Raginsky. Minimax statistical learning with wasserstein distances. *NeurIPS*, 31, 2018.

[37] David Krueger, Ethan Caballero, Joern-Henrik Jacobsen, Amy Zhang, Jonathan Binas, Dinghuai Zhang, Remi Le Priol, and Aaron Courville. Out-of-distribution generalization via risk extrapolation (rex). In *ICML*, pages 5815–5826, 2021.

[38] Siqi Miao, Miaoyuan Liu, and Pan Li. Interpretable and generalizable graph learning via stochastic attention mechanism. In *ICML*, pages 15524–15543, 2022.

[39] Haoyang Li, Xin Wang, Ziwei Zhang, and Wenwu Zhu. Ood-gnn: Out-of-distribution generalized graph neural network. *IEEE TKDE*, 2022.

[40] Shaohua Fan, Xiao Wang, Chuan Shi, Peng Cui, and Bai Wang. Generalizing graph neural networks on out-of-distribution graphs. *arXiv preprint arXiv:2111.10657*, 2021.

[41] Yongqiang Chen, Yonggang Zhang, Yatao Bian, Han Yang, Kaili Ma, Binghui Xie, Tongliang Liu, Bo Han, and James Cheng. Learning causally invariant representations for out-of-distribution generalization on graphs. In *NeurIPS*, 2022.

[42] Raphael Gontijo-Lopes, Sylvia Smullin, Ekin Dogus Cubuk, and Ethan Dyer. Tradeoffs in data augmentation: An empirical study. In *ICLR*, 2021.

[43] Jaemin Yoo, Sooyeon Shim, and U Kang. Model-agnostic augmentation for accurate graph classification. In *WWW*, pages 1281–1291, 2022.

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

[45] Chaochao Lu, Yuhuai Wu, José Miguel Hernández-Lobato, and Bernhard Schölkopf. Invariant causal representation learning for out-of-distribution generalization. In *ICLR*, 2021.

[46] Qi Zhu, Natalia Ponomareva, Jiawei Han, and Bryan Perozzi. Shift-robust gnns: Overcoming the limitations of localized graph training data. *NeurIPS*, 34:27965–27977, 2021.

[47] Wei Jin, Tong Zhao, Jiayuan Ding, Yozen Liu, Jiliang Tang, and Neil Shah. Empowering graph representation learning with test-time graph transformation. In *ICLR*, 2023.

[48] Junchi Yu, Jian Liang, and Ran He. Mind the label shift of augmentation-based graph ood generalization. In *CVPR*, 2023.

[49] Zenan Li, Qitian Wu, Fan Nie, and Junchi Yan. Graphde: A generative framework for debiased learning and out-of-distribution detection on graphs. In *NeurIPS*, 2022.

[50] Qitian Wu, Yiting Chen, Chenxiao Yang, and Junchi Yan. Energy-based out-of-distribution detection for graph neural networks. In *ICLR*, 2023.

[51] Yuxin Guo, Cheng Yang, Yuluo Chen, Jixi Liu, Chuan Shi, and Junping Du. A data-centric framework to endow graph neural networks with out-of-distribution detection ability. In *KDD*, 2023.

[52] Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8(5), 2007.

[53] Zheyan Shen, Peng Cui, Kun Kuang, Bo Li, and Peixuan Chen. Causally regularized learning with agnostic data selection bias. In *ACM MM*, pages 411–419, 2018.

[54] Renzhe Xu, Xingxuan Zhang, Zheyan Shen, Tong Zhang, and Peng Cui. A theoretical analysis on independence-driven importance weighting for covariate-shift generalization. In *ICML*, pages 24803–24829. PMLR, 2022.

[55] Tongtong Fang, Nan Lu, Gang Niu, and Masashi Sugiyama. Rethinking importance weighting for deep learning under distribution shift. *NeurIPS*, 33:11996–12007, 2020.

[56] Yue He, Xinwei Shen, Renzhe Xu, Tong Zhang, Yong Jiang, Wenchao Zou, and Peng Cui. Covariate-shift generalization via random sample weighting. *AAAI*, 2023.

[57] Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

[58] Kurt Binder, Dieter Heermann, Lyle Roelofs, A John Mallinckrodt, and Susan McKay. Monte carlo simulation in statistical physics. *Computers in Physics*, 7(2):156–157, 1993.

[59] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *CVPR*, pages 5115–5124, 2017.

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

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

[62] Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *ICML*, pages 1725–1735. PMLR, 2020.

[63] Kiran K Thekumparampil, Chong Wang, Sewoong Oh, and Li-Jia Li. Attention-based graph neural network for semi-supervised learning. *arXiv preprint arXiv:1803.03735*, 2018.

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

[65] Jiashuo Liu, Zheyuan Hu, Peng Cui, Bo Li, and Zheyan Shen. Heterogeneous risk minimization. In *ICML*, pages 6804–6814. PMLR, 2021.

[66] Masanori Koyama and Shoichiro Yamaguchi. When is invariance useful in an out-of-distribution generalization problem? *arXiv preprint arXiv:2008.01883*, 2020.

[67] Kartik Ahuja, Karthikeyan Shanmugam, Kush Varshney, and Amit Dhurandhar. Invariant risk minimization games. In *ICML*, pages 145–155. PMLR, 2020.

[68] Haohan Wang, Zeyi Huang, Xindi Wu, and Eric Xing. Toward learning robust and invariant representations with alignment regularization and data augmentation. In *KDD*, pages 1846–1856, 2022.

[69] Pritish Kamath, Akilesh Tangella, Danica Sutherland, and Nathan Srebro. Does invariant risk minimization capture invariance? In *International Conference on Artificial Intelligence and Statistics*, pages 4069–4077. PMLR, 2021.

[70] Mateo Rojas-Carulla, Bernhard Schölkopf, Richard Turner, and Jonas Peters. Invariant models for causal transfer learning. *The Journal of Machine Learning Research*, 19(1):1309–1342, 2018.

[71] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

[72] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE TKDE*, 2022.

[73] Jinheon Baek, Dong Bok Lee, and Sung Ju Hwang. Learning to extrapolate knowledge: Transductive few-shot out-of-graph link prediction. *NeurIPS*, 33:546–560, 2020.

[74] Divyat Mahajan, Shruti Tople, and Amit Sharma. Domain generalization using causal matching. In *ICML*, pages 7313–7324. PMLR, 2021.

[75] Massimiliano Mancini, Zeynep Akata, Elisa Ricci, and Barbara Caputo. Towards recognizing unseen categories in unseen domains. In *ECCV*, pages 466–483. Springer, 2020.

[76] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *ICML*, pages 10–18. PMLR, 2013.

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

[78] Davide Buffelli, Pietro Lio, and Fabio Vandin. Sizeshiftreg: a regularization method for improving size-generalization in graph neural networks. In *NeurIPS*, 2022.

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

[80] Zeyang Zhang, Xin Wang, Ziwei Zhang, Haoyang Li, Zhou Qin, and Wenwu Zhu. Dynamic graph neural networks under spatio-temporal distribution shift. In *NeurIPS*, 2022.

[81] Junfeng Fang, Xiang Wang, An Zhang, Zemin Liu, Xiangnan He, and Tat-Seng Chua. Cooperative explanations of graph neural networks. In *WSDM*, pages 616–624, 2023.

[82] Junfeng Fang, Wei Liu, An Zhang, Xiang Wang, Xiangnan He, Kun Wang, and Tat-Seng Chua. On regularization for explaining graph neural networks: An information theory perspective. 2022.

[83] Yongduo Sui, Xiang Wang, Tianlong Chen, Meng Wang, Xiangnan He, and Tat-Seng Chua. Inductive lottery ticket learning for graph neural networks. *Journal of Computer Science and Technology*, 2023.

[84] Tianlong Chen, Yongduo Sui, Xuxi Chen, Aston Zhang, and Zhangyang Wang. A unified lottery ticket hypothesis for graph neural networks. In *ICML*, pages 1695–1706. PMLR, 2021.

[85] Yanfang Wang, Yongduo Sui, Xiang Wang, Zhenguang Liu, and Xiangnan He. Exploring lottery ticket hypothesis in media recommender systems. *International Journal of Intelligent Systems*, 37(5):3006–3024, 2022.

[86] Yuan Gao, Xiang Wang, Xiangnan He, Zhenguang Liu, Huamin Feng, and Yongdong Zhang. Addressing heterophily in graph anomaly detection: A perspective of graph spectrum. In *WWW*, pages 1528–1538, 2023.

[87] Yuan Gao, Xiang Wang, Xiangnan He, Huamin Feng, and Yong-Dong Zhang. Rumor detection with self-supervised learning on texts and social graph. *Frontiers Comput. Sci.*, 17(4):174611, 2023.

[88] Yanqiao Zhu, Yuanqi Du, Yinkai Wang, Yichen Xu, Jieyu Zhang, Qiang Liu, and Shu Wu. A survey on deep graph generation: Methods and applications. *arXiv preprint arXiv:2203.06714*, 2022.

A Correlation Shift and Covariate Shift

Following graph data generation process [17, 8, 3, 29], we can observe that environmental features easily change outside the training distribution, owing to their unstable nature. Hence, distribution shifts are mainly caused by the environmental features [12, 13, 2]. Specifically, we define the joint distribution of training and test data as $P_{\text{tr}}(G; Y)$ and $P_{\text{te}}(G; Y)$, respectively. Since their joint distribution can be rewritten as $P_{\text{tr}}(G; Y) = P_{\text{tr}}(Y|G)P_{\text{tr}}(G)$ and $P_{\text{te}}(G; Y) = P_{\text{te}}(Y|G)P_{\text{te}}(G)$, we can find that there exist two main reasons that lead to the distribution shift $P_{\text{tr}}(G; Y) \neq P_{\text{te}}(G; Y)$. We give intuitive examples in Figure 5 and formal definitions of these two distribution shifts.

- **Correlation shift** $P_{\text{tr}}(Y|G) \neq P_{\text{te}}(Y|G); P_{\text{tr}}(G) = P_{\text{te}}(G)$**.** If the statistical correlation of environmental features and labels is inconsistent in training and test data, a well-fitted model in training data may fail in test data, which is also known as spurious correlation [14], correlation shift [12] or concept shift [2]. Formally, correlation shift describes the conditional distribution $P_{\text{tr}}(Y|G) \neq P_{\text{te}}(Y|G)$.
- **Covariate shift** $P_{\text{tr}}(G) \neq P_{\text{te}}(G); P_{\text{tr}}(Y|G) = P_{\text{te}}(Y|G)$**.** If there exist environmental features in the test distribution that the model has not seen during training, it will also result in a large performance drop. This unseen distribution shift is well known as covariate shift [2] or diversity shift [12]. It means that the environmental features in test data are unseen in training data, which leads to $P_{\text{tr}}(G) \neq P_{\text{te}}(G)$. In Definition 2.1, we also quantitatively measure the covariate shift between $P_{\text{tr}}(G)$ and $P_{\text{te}}(G)$.

It is worth noting that within the computer vision domain, the general covariate shift is frequently synonymous with sample selection bias [52, 53, 54, 55, 56]. Various factors contribute to covariate shift, such as heterogeneous category distribution or domain-specific variances. In the context of our investigation into graph-based models, we adhere to the assumptions outlined in previous literature [2, 12], which mainly ignore the influence of label shifts. And we posit that covariate shifts are principally attributed to the environmental features. The exploration of more comprehensive scenarios involving covariate shifts will be undertaken in our future work.

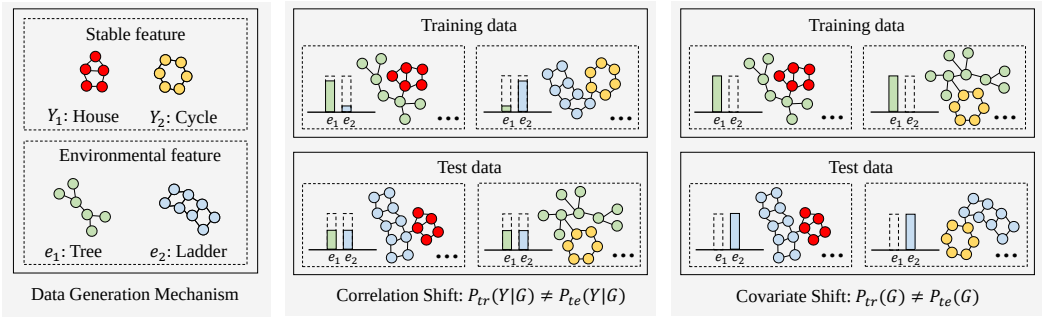

Figure 5: Intuitive examples of correlation shift and covariate shift

B Proofs

In this section, we provide the detailed proofs to our proposition. We start with the definition of the 1-Wasserstein distance, $D_W(P; P')$, between two distributions P and P':

$$D_W(P; P') = \inf_{\gamma \in \Gamma(P;P')} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\| \, d\gamma(x; y);$$

where $\Gamma(P; P')$ represents the set of all joint distributions $\gamma(x; y)$ that have P and P' as their respective marginals. Under our conditions, due to $P'(x - x') = P(x)$, we can precisely determine the manner in which the mass from P was transferred to create P'. This process involves moving each point x under P to $x + x'$ under P'. Therefore, the infimum is attained by the coupling that deterministically transitions each point x to $x + x'$. Consequently, the Wasserstein distance simplifies to:

$$D_W(P; P') = \int_{\mathbb{R}^d} \|x - (x + x')\| \, dP(x) = \int_{\mathbb{R}^d} \|x'\| \, dP(x) = \|x'\|;$$

This result establishes that the Wasserstein distance between the distributions P and P' is equivalent to the magnitude of x'. Therefore, we obtain:

$$D_W(P;\tilde{P}_1) = \int \|x - (x + x_1)\| \, d\tilde{P}_1(x) = \int \|x_1\| \, dP_1(x) = \|x_1\|.$$

Similarly, $D_W(P;\tilde{P}_2) = \|x_2\|$. Consequently, if $D_W(P;\tilde{P}_1) < D_W(P;\tilde{P}_2)$, we can deduce that $\|x_1\| < \|x_2\|$.

Next, we examine the GCS measure. The covariate shift between P and $\tilde{P}_1$ is given by:

$$\mathrm{GCS}(P;\tilde{P}_1) = \frac{1}{2}\int_{S_1} |P(x) - P(x + x_1)| \, dx.$$

Similarly, we have:

$$\mathrm{GCS}(P;\tilde{P}_2) = \frac{1}{2}\int_{S_2} |P(x) - P(x + x_2)| \, dx;$$

where S_1 and S_2 denote the non-overlapping regions between P and $\tilde{P}_1$, and between P and $\tilde{P}_2$, respectively. Now we define another $\tilde{P}_1'$ that satisfies $\tilde{P}_1'(x) = P(x + \|x_1\| \frac{x_2}{\|x_2\|})$ and let S_1' denote the non-overlapping regions between P and $\tilde{P}_1'$. Given the isotropy of $\tilde{P}_1$, the integral over S_1 is equal to the integral over S_1'. Therefore, we can proceed to deduce:

$$\begin{aligned}
\mathrm{GCS}(P;\tilde{P}_1) &= \frac{1}{2}\int_{S_1} |P(x) - P(x + x_1)| \, dx \\
&= \frac{1}{2}\int_{S_1'} |P(x) - P(x + x_1)| \, dx \\
&= \mathrm{GCS}(P;\tilde{P}_1')
\end{aligned}$$

Let $S_2' = \{x + (\|x_2\| - \|x_1\|) \frac{x_2}{\|x_2\|} \mid x \in S_1'\}$, we can easily know that $|S_1'| = |S_2'|$ and $S_2' \cap S_2$. Thus we have:

$$\begin{aligned}
\mathrm{GCS}(P;\tilde{P}_2) - \mathrm{GCS}(P;\tilde{P}_1) &= \frac{1}{2}\int_{S_2} |P(x) - P(x + x_1)| \, dg - \frac{1}{2}\int_{S_1} |P(x) - P(x + x_1)| \, dx \\
&= \frac{1}{2}\int_{S_2} |P(x) - P(x + x_2)| \, dx - \frac{1}{2}\int_{S_1'} |P(x) - P(x + x_1)| \, dx \\
&= \frac{1}{2}\int_{S_2} |P(x) - P(x + x_2)| \, dx - \frac{1}{2}\int_{S_2'} |P(x) - P(x + x_2)| \, dx \\
&= \frac{1}{2}\int_{S_2 \setminus S_2'} |P(x) - P(x + x_2)| \, dx \geq 0.
\end{aligned}$$

The integral over S_2 could not be less than the integral over S_1, leading to $\mathrm{GCS}(P;\tilde{P}_1) \leq \mathrm{GCS}(P;\tilde{P}_2)$. Hence, we complete the proof.

C Estimation of Graph Covariate Shift

In this section, we elaborate on the implementation details of estimating the graph covariate shift. Without loss of generality, we start with the example of estimating the graph covariate shift between the training and test distributions. Given the training set and test set D_{tr} and D_{te}, they follow probability distribution functions P_{tr} and P_{te}. The process of estimating $\mathrm{GCS}(P_{\mathrm{tr}};P_{\mathrm{te}})$ is summarized in the following two steps:

- Firstly, it is intractable to directly estimate the distribution in graph space G. Inspired by [12], we can obtain the graph features and estimate the distribution in feature space F. Specifically, given a sample, we train a binary GNN classifier f to distinguish which distribution it comes from, where $f(\cdot) = \circ h$, $h(\cdot) \in \mathsf{G} \to \mathsf{F}$ is a graph encoder, and $(\cdot) \in \mathsf{F} \to \{0;1\}$ is a binary classifier. Then we can adopt the pre-trained GNN encoder h to extract graph features.
- Secondly, we prepare the features and estimate the distribution of the data via Kernel Density Estimation (KDE) [57]. Finally, we adopt the Monte Carlo Integration under importance sampling [58] to approximate the integrals in Definition 2.1.

We summarize these implementations in Algorithm 1. In lines 4 and 5, to avoid the label shift [12], we adopt sample reweighting to ensure the balance of each class.

Algorithm 1: Estimation of Graph Covariate shift

Require: Training dataset D_{tr} and test dataset D_{te}; Batch size N; Loss function $\grave{}$; GNN $f = \circ h$; Importance sampling size M; Threshold .

Ensure: Estimated covariate shift $\mathsf{GCS}(P_{\mathsf{tr}}; P_{\mathsf{te}})$.

1: Initialize parameters of f
2: # Train a graph classifier
3: **while** not converge **do**
4: Sample a batch $\mathsf{B}_{\mathsf{tr}} \leftarrow \{(g_i; y_i)\}_{i=1}^{N} \sqsubset \mathsf{D}_{\mathsf{tr}}$ and relabel all $y_i \leftarrow 0$
5: Sample a batch $\mathsf{B}_{\mathsf{te}} \leftarrow \{(g_i; y_i)\}_{i=1}^{N} \sqsubset \mathsf{D}_{\mathsf{te}}$ and relabel all $y_i \leftarrow 1$
6: $\mathsf{B} \leftarrow \mathsf{B}_{\mathsf{tr}} \sqcup \mathsf{B}_{\mathsf{te}}$
7: **for** each $(g_i; y_i) \sqsubset \mathsf{B}$ **do**
8: Compute loss $\grave{}(f(g_i); y_i)$ and back-propagate gradients
9: **end for**
10: Update the parameters of f via gradient descent and reset the gradients
11: **end while**
12: # Prepare the features for the estimation
13: Extract training and test feature sets F_{tr} and F_{te} via encoder h
14: $\mathsf{F} \leftarrow \mathsf{F}_{\mathsf{tr}} \sqcup \mathsf{F}_{\mathsf{te}}$
15: Scale F to zero mean and unit variance
16: $\hat{\mathsf{P}} \leftarrow$ fit by KDE the distribution of F
17: Split F to recover the original partition $\mathsf{F}'_{\mathsf{tr}}; \mathsf{F}'_{\mathsf{te}}$
18: $\hat{P}_{\mathsf{tr}}; \hat{P}_{\mathsf{te}} \leftarrow$ fit by KDE the distributions of $\mathsf{F}'_{\mathsf{tr}}; \mathsf{F}'_{\mathsf{te}}$
19: # Estimate the covariate shift
20: Initialize $\mathsf{GCS}(P_{\mathsf{tr}}; P_{\mathsf{te}}) \leftarrow 0$
21: **for** $t \leftarrow \{1; \ldots; M\}$ **do**
22: $z \leftarrow$ sample from $\hat{\mathsf{P}}$
23: **if** $\hat{P}_{\mathsf{tr}}(z) <$ or $\hat{P}_{\mathsf{te}}(z) <$ **then**
24: $\mathsf{GCS}(P_{\mathsf{tr}}; P_{\mathsf{te}}) \leftarrow \mathsf{GCS}(P_{\mathsf{tr}}; P_{\mathsf{te}}) + |\hat{P}_{\mathsf{tr}}(z) - \hat{P}_{\mathsf{te}}(z)| / \hat{\mathsf{P}}(z)$
25: **end if**
26: **end for**
27: $\mathsf{GCS}(P_{\mathsf{tr}}; P_{\mathsf{te}}) \leftarrow \mathsf{GCS}(P_{\mathsf{tr}}; P_{\mathsf{te}})/2M$

Algorithm 2: Adversarial Invariant Augmentation

Require: Training set D_{tr}; Adversarial augmenter $T_1(\cdot)$; Stable feature generator $T_2(\cdot)$; GNN classifier $f(\cdot)$ with parameters ; Learning rates ; ; Batch size N; Stable feature ratio $_S$; Penalty .

1: Randomly initilize , $_1$, $_2$
2: **while** not converge **do**
3: Sample a batch $\mathsf{B}_{\mathsf{tr}} \leftarrow \{(g_i; y_i)\}_{i=1}^{N} \sqsubset \mathsf{D}_{\mathsf{tr}}$
4: **for** each $(g_i; y_i) \sqsubset \mathsf{B}_{\mathsf{tr}}$ **do**
5: $\mathsf{M}^a_{\mathsf{adv}}; \mathsf{M}^x_{\mathsf{adv}} \leftarrow T_1(g_i)$ `// adversarial perturbations`
6: $\mathsf{M}^a_{\mathsf{sta}}; \mathsf{M}^x_{\mathsf{sta}} \leftarrow T_2(g_i)$ `// regions of stable features`
7: $\tilde{\mathsf{M}}^a \leftarrow (1 - \mathsf{M}^a_{\mathsf{sta}}) \sqcup \mathsf{M}^a_{\mathsf{adv}} + \mathsf{M}^a_{\mathsf{sta}}$ `// augment edges`
8: $\tilde{\mathsf{M}}^x \leftarrow (1 - \mathsf{M}^x_{\mathsf{sta}}) \sqcup \mathsf{M}^x_{\mathsf{adv}} + \mathsf{M}^x_{\mathsf{sta}}$ `// augment nodes`
9: $\tilde{g}_i \leftarrow (\mathsf{A}_i \sqcup \tilde{\mathsf{M}}^a; \mathsf{X}_i \sqcup \tilde{\mathsf{M}}^x)$ `// augmented graph`
10: **end for**
11: Compute $\mathsf{L}_{\mathsf{adv}} - \mathsf{L}_{\mathsf{reg}_1}$ via Equation (7) and (9)
12: Compute $\mathsf{L}_{\mathsf{sta}} + \mathsf{L}_{\mathsf{reg}_2}$ via Equation (8) and (10)
13: Update parameters of adversarial augmenter via gradient ascent:
$_1 \leftarrow _1 + \sqcup _1(\mathsf{L}_{\mathsf{adv}} - \mathsf{L}_{\mathsf{reg}_1})$
14: Update parameters of GNN and stable feature generator via gradient descent:
$\leftarrow - \sqcup (\mathsf{L}_{\mathsf{sta}} + \mathsf{L}_{\mathsf{reg}_2})$; $_2 \leftarrow _2 - \sqcup _2(\mathsf{L}_{\mathsf{sta}} + \mathsf{L}_{\mathsf{reg}_2})$
15: **end while**

Table 4: Statistics of graph classification datasets.

Dataset		Motif		CMNIST	Molbbbp		Molhiv	
Covariate shift		base	size	color	scaffold	size	scaffold	size
Train	Graph#	18000	18000	42000	1631	1633	24682	26169
	Avg. node#	17.07	16.93	75.00	22.49	27.02	26.25	27.87
	Avg. edge#	48.89	43.57	1392.76	48.43	58.71	56.68	60.20
Val	Graph#	3000	3000	7000	204	203	4113	2773
	Avg. node#	15.82	39.22	75.00	33.20	12.06	24.95	15.55
	Avg. edge#	33.00	107.03	1393.73	71.84	24.27	54.53	32.77
Test	Graph#	3000	3000	7000	204	203	4108	3961
	Avg. node#	14.97	87.18	75.00	27.51	12.26	19.76	12.09
	Avg. edge#	31.54	239.65	1393.60	59.75	24.87	40.58	24.87
Class#		3	3	10	2	2	2	2

Table 5: Hyper-parameter details of AIA.

Dataset	Motif		CMNIST	Molbbbp		Molhiv	
Covariate shift	base	size	color	scaffold	size	scaffold	size
Backbone (layer-hidden)	4-300	4-300	4-300	4-64	4-32	4-128	4-128
Augmenter (layer-hidden)	2-300	2-300	2-300	2-64	2-32	2-128	2-128
Generator (layer-hidden)	2-300	2-300	2-300	2-64	2-32	2-128	2-128
Optimizer	Adam	Adam	Adam	Adam	Adam	Adam	Adam
Learning rate	1e-3	1e-3	1e-3	1e-3	5e-3	1e-3	1e-2
Learning rate	5e-3	5e-3	5e-3	1e-3	5e-3	1e-2	1e-2
Stable feature ratio $_s$	0.5	0.5	0.5	0.5	0.5	0.1	0.1
Adversarial penalty	0.2	0.2	0.2	0.5	0.5	0.5	0.5

D Implementation Details

D.1 Algorithm

We summarize the detailed implementations of AIA in Algorithm 2. We alternately optimize the adversarial augmenter and stable feature generator with the backbone model, in lines 13 and 14. We adopt the learned stable features for predictions in the inference stage.

D.2 Datasets

In this paper, we conduct experiments on graph OOD datasets [2] and OGB datasets [20], which include Motif, CMNIST, Molbbbp and Molhiv. We follow [2] to create various covariate shifts, according to base, color, size and scaffold splitting. Base, color, size and scaffold are features of the graph data and do not determine the labels of the data, so they can be regarded as environmental features. The statistics of the datasets are summarized in Table 4. Below we give a brief introduction to each dataset.

- **Motif:** It is a synthetic dataset from Spurious-Motif [17, 5]. As shown in original graphs in Figure 5, each graph is composed of a base-graph (*wheel, tree, ladder, star, path*) and a motif (*house, cycle, crane*). The label is only determined by the type of motif. We create covariate shift according to the base-graph type and the graph size (*i.e.,* node number). For base covariate shift, we adopt graphs with *wheel, tree, ladder* base-graphs for training, *star* for validation and *path* for testing. For size covariate shift, we use small-size of graphs for training, while the validation and the test sets include the middle- and the large-size graphs, respectively.
- **CMNIST:** Color MNIST dataset contains graphs transformed from MNIST via superpixel techniques [59]. We define color as the environmental features to create the covariate shift. Specifically, we color digits with 7 different colors, where five of them are adopted for training while the remaining two are used for validation and testing.
- **Molbbbp & Molhiv:** These are molecular datasets collected from MoleculeNet [19]. We define the scaffold and graph size (*i.e.,* node number) as the environmental features to create two types of covariate shifts. For scaffold shift, we follow [2] and use scaffold split to create training, validation

Table 6: Performance over diverse backbones.

Backbone	Method	Motif		Molbbbp	
		base	size	scaffold	size
GCN	EEM	67.41±4.47	50.76±2.92	66.44±1.91	77.79±4.04
	M-Mixup	68.83±4.04	51.50±4.95	67.09±0.57	78.42±2.71
	FLAG	59.87±5.61	50.99±4.18	66.03±2.59	78.76±2.54
	AIA (ours)	**72.39±5.37**	**54.87±5.02**	**69.13±1.76**	**80.53±4.43**
GCNII	EEM	67.84±4.74	51.74±3.15	67.64±1.84	77.96±4.02
	M-Mixup	67.53±4.31	52.31±5.18	66.64±0.50	76.67±2.69
	FLAG	58.67±5.88	50.18±4.41	66.72±2.52	78.55±2.52
	AIA (ours)	**72.70±4.14**	**53.23±6.25**	**67.93±1.69**	**79.72±3.37**
GAT	EEM	66.53±4.42	51.16±3.22	66.51±1.77	77.41±3.81
	M-Mixup	69.25±3.99	51.37±5.25	66.95±0.43	77.21±2.48
	FLAG	59.53±5.56	51.32±4.48	67.36±2.45	77.87±2.31
	AIA (ours)	**71.95±3.39**	**54.38±6.32**	**69.21±1.62**	**78.49±3.20**

and test sets. For size shift, we adopt the large-size of graphs for training and the smaller ones for validation and testing.

D.3 Metrics

We adopt classification accuracy as the metric for Motif and CMNIST. As suggested by [20], we use ROC-AUC for Molhiv and Molbbbp datasets. In addition, we use $\mathsf{GCS}(P;Q)$ to measure the covariate shift between distributions P and Q. For all experimental results, we perform 10 random runs and report the mean and standard derivations. For augmentation diversity, we use conditional entropy to measure the diversity of generated data. We normalize it to $[0;1]$ for better comparison. Specifically, for a given graph data, we compute the conditional entropy by collecting augmented data generated during the training process. We conduct experiments by collecting 1000 graph data, and report the mean and standard deviation.

D.4 Training Settings

We use the NVIDIA GeForce RTX 3090 (24GB GPU) to conduct all our experiments. To make a fair comparison, we adopt GIN [60] as the default architecture to conduct all experiments. We tune the hyper-parameters in the following ranges: and $\in \{0{:}01; 0{:}005; 0{:}001\}$; $_2 \in \{0{:}1; ::: ; 0{:}9\}$; $\in \{0{:}01; 0{:}1; 0{:}2; 0{:}5; 1{:}0; 1{:}5; 2{:}0; 3{:}0; 5{:}0\}$. The hyper-parameters are summarized in Table 5.

D.5 Baseline Settings

For a more comprehensive comparison, we selected 16 baselines. In this section, we give a detailed introduction to the settings of these methods.

- For ERM, IRM [14], GroupDRO [31], VREx [37], and M-Mixup [26], we report the results from the study [2] by default and reproduce the missing results on Molbbbp.
- For DIR [17], CAL [5], GSAT [38], DropEdge [25], GREA [18], FLAG [24], G-Mixup [7], CIGA [41] and DisC [21], they provide source codes for the implementations. We adopt default settings from their source codes and detailed hyper-parameters from their original papers.
- For OOD-GNN [39] and StableGNN [40], their source codes are not publicly available. We reproduce them based on the codes of StableNet [16].
- For RDIA in Section 5.4, it is a variant that replaces the adversarial augmentation in AIA with random augmentation. In our implementation, we use all-one matrices to create the initial node and edges masks. Then we randomly set 20% of nonzero elements to zero in these masks. Finally, we apply these masks to the graphs for random data augmentation. The process of stable feature learning is consistent with AIA.

E More Experimental Results

E.1 Results on Correlation Shift

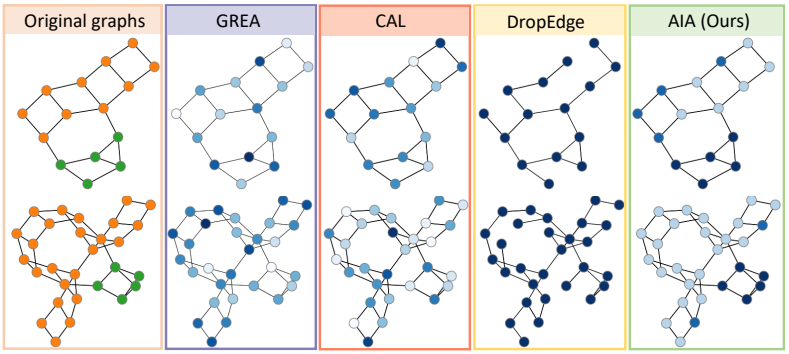

Figure 6: Visualizations of captured stable features.

Table 7: Performance on correlation shift.

Method	Motif	CMNIST	Molhiv
ERM	81.44±2.54	42.87±1.37	63.26±1.25
IRM	80.71±2.81	42.80±1.62	59.90±1.17
VREx	81.56±2.14	43.31±1.03	60.23±1.60
DIR	73.25±6.37	38.78±1.45	66.78±1.50
CAL	81.94±1.20	41.82±0.85	62.36±1.42
OOD-GNN	80.22±2.28	39.03±1.24	57.49±1.08
DropEdge	78.97±3.41	38.43±1.94	54.92±1.73
FLAG	80.91±1.04	43.41±1.38	66.44±2.32
M-Mixup	77.63±1.12	40.96±1.21	64.87±1.36
AIA (ours)	**82.51**±2.81	**49.73**±1.70	**68.11**±1.82

Although this work focuses on the OOD issue of covariate shift, for completeness, we also evaluate the performance of AIA under correlation shift. Following [2], we choose three graph OOD datasets (*i.e.,* Motif, CMNIST, Molhiv) with three different graph features (*i.e.,* base, color, size) to create correlation shifts. For baselines, we choose three generalization algorithms (*i.e.,* ERM, IRM [14], VREx [37]), three graph generalization methods (*i.e.,* DIR [17], CAL [5], OOD-GNN [39]) and three data augmentation methods (*i.e.,* DropEdge [25], FLAG [24], M-Mixup [26]). The experimental results are shown in Table 7. We can observe that AIA can also effectively alleviate the correlation shift. These results demonstrate that AIA learns better stable features by encouraging environmental discrepancy, which can effectively break spurious correlations that are hidden in the training data.

E.2 Results on Diverse Backbones

We select three different GNN backbone models (GCN [61], GCNII [62] and GAT [63]) for experiments. From the results in Table 6, our observations and conclusions remain the same with the diverse backbones.

E.3 Results on More Real-world Datasets

To demonstrate the effectiveness of the proposed AIA, we also conduct experiments on commonly used TU datasets [64], which include MUTAG, NCI1, PROTEINS, COLLAB, IMDB-B, IMDB-M. For training settings, we follow CAL [5] and adopt GIN [60] as our backbone model. The experimental results are shown in Table 8. For the results, we can observe that our method can achieve the best performance over different datasets.

E.4 More Visualizations

To demonstrate the superiority of our method, we also visualize the captured stable features by AIA and compare them with other baselines. The results are displayed in Figure 6. From the results, we can easily observe that our method can find stable parts more accurately than other baseline methods.

F Complexity Analyses

Firstly, we define the average numbers of nodes and edges per graph in the dataset to be n and m, respectively. Let N denote the batch size, l, l_a and l_c denote the numbers of layers in the GNN backbone, adversarial augmenter and stable feature generator, respectively. d, d_a and d_c are the

Table 8: Performance comparisons on TU datasets.

Method	MUTAG	NCI1	PROTEINS	COLLAB	IMDB-B	IMDB-M
ERM	89.42±7.40	82.71±1.52	76.21±3.83	82.08±1.51	73.40±3.78	51.53±2.97
CAL	89.91±8.34	83.89±1.93	76.92±3.31	82.68±1.25	74.13±5.21	52.60±2.36
DropEdge	86.11±9.41	82.35±3.77	74.40±3.10	80.59±2.14	72.34±5.83	51.06±3.04
FLAG	89.45±7.20	82.67±2.12	76.89±3.66	82.48±1.79	73.37±4.94	52.16±2.70
M-Mixup	89.83±7.67	83.89±2.38	76.76±3.40	82.90±1.43	74.07±4.76	52.89±2.84
AIA (ours)	**90.34±7.75**	**84.12±2.64**	**77.92±3.72**	**82.98±1.76**	**74.23±5.10**	**53.02±2.76**

Table 9: Running time, model size and performance improvement.

Dataset	ERM		CAL			DIR			AIA (ours)		
	Running Time	Model Size	Running Time	Model Size	Performance Improvement	Running Time	Model Size	Performance Improvement	Running Time	Model Size	Performance Improvement
Motif	00h 51m 19s	1.515M	01h 37m 15s	2.213M	↓ 2.98%	01h 50m 27s	2.158M	↓ 5.03%	01h 49m 08s	2.366M	↑ 7.55%
CMNIST	01h 56m 32s	1.517M	02h 28m 49s	2.244M	↓ 2.13%	02h 34m 35s	2.161M	↑ 16.08%	02h 45m 30s	2.373M	↑ 27.17%
Molbbbp	00h 11m 58s	1.515M	00h 18m 22s	2.213M	↑ 0.81%	00h 20m 11s	2.158M	↓ 2.12%	00h 19m 37s	2.366M	↑ 3.72%
Molhiv	00h 27m 19s	1.515M	00h 46m 14s	2.213M	↓ 3.23%	00h 58m 40s	2.158M	↓ 2.59%	00h 55m 46s	2.366M	↑ 2.54%

dimensions of hidden layers in the GNN backbone, adversarial augmenter and stable feature generator, respectively.

Time Complexity. The time complexity of the adversarial learning objective is $O(N(l_a m d_a + 2lmd))$. For the stable feature learning objective, the time complexity is $O(N(l_c m d_c + 2lmd))$. For the regularization terms, the time complexity is $O(2N(n+m))$. For simplicity, we assume $l_a = l_c$ and $d_a = d_c$. Hence, the time complexity of a forward propagation is $O(2N(l_a m d_a + 2lmd + n + m))$.

Model Size. In addition to the GNN backbone model, we also introduce two small networks for adversarial augmentation and stable feature learning. In our implementations, the parameters of AIA are around twice as large as those of the original GNN model. The running time and model size are shown in Table 9.

G More Related Studies

OOD Generalization [1, 65, 66] has been widely explored. Inspired by invariant learning and causal theory, a series of general algorithms [67, 68, 69, 66, 70] have recently been proposed to solve the OOD problem. Recent studies [12, 2, 13] point out that OOD falls into two specific categories: correlation shift (*aka.* concept shift) and covariate shift (*aka.* diversity shift). Correlation shift denotes that the environmental features and labels establish a statistical correlation that is inconsistent in training and test data. Thus, the models prefer to learn spurious correlations and rely on shortcut features [71] for predictions, resulting in a large performance drop. In contrast, covariate shift indicates that there exist unseen environmental features in test data. The limited training environment makes this issue intractable. Tasks pertaining to domain generalization [72] often exhibit the covariate shift issue, such as model inference under previously unseen test domains. Consequently, there have been numerous recent efforts [73, 74, 75, 76] to address such challenges. In recent years, OOD generalization on graphs is drawing widespread attention [4]. Given the intricate nature of graph data types and associated tasks [77], issues related to distribution shifts can emerge in various contexts, such as node classification [46, 3, 47], graph classification [39, 40, 17, 38, 48, 5, 18, 78, 79], or dynamic graph data [80], among others. Emerging research has increasingly focused on the identification of stable features or subgraphs within graph data. These substructures are posited to maintain causal relationships with target labels, thereby enhancing the model's capacity for generalization [5, 18], explainability [81, 17, 38, 82], and computational efficiency [83, 84, 85]. Concurrently, efforts are being made to extend these principles to diverse applications, such as molecular graphs [8], recommender systems [9, 10], and anomaly detection [11, 86, 87].

Comprehensive Comparisons with EERM [3]. Although EERM shares similar goals with us, generating several environments through augmentation, there exist many technical and contribution differences. Firstly, EERM ignores the distinction between correlation shift and covariate shift problems, while we distinguish these two shifts in detail and design a framework specifically for covariate shift. Secondly, EERM does not model stable and environmental features, which results in the inability to explicitly distinguish them. In contrast, we explicitly model the environmental and stable features. Hence, we can effectively identify stable and environmental features and explicitly

Table 10: Comparisons with EERM.

		EERM	Our AIA
Scope	Is it specifically designed for covariate shift?	✗	✓
Separability	Can environmental/stable features be separated?	✗	✓
Environmental Feature Discrepancy	Can environmental features be identified explicitly?	✗	✓
	How to model environmental features?	-	Mask model $T_1(\cdot)$
	Metric for environmental Discrepancy	-	$\mathrm{GCS}(\tilde{P};P)$
	Generation principle for environmental features	"Blindly" maximize $\mathrm{V}_e[R(e)]$	Maximize $\mathrm{GCS}(\tilde{P};P)$
Stable Feature Consistency	Can stable features be identified explicitly?	✗	✓
	How to model stable features?	-	Mask model $T_2(\cdot)$
	Learning principles for stable features	$\min\ \mathrm{V}_e[R(e)]$	Sufficiency/Independence
Generalization	Theoretical basis	IRM	DRO
	Generalization scope	K environments	Robust radius $D(\tilde{P};P) \leq$

separate them from data. Thirdly, we also design a metric, $\mathrm{GCS}(\tilde{P};P)$, which can effectively measure the discrepancy of the environmental features for our augmented data. And we directly encourage the environmental discrepancy of the augmented samples by maximizing $\mathrm{GCS}(\tilde{P};P)$. However, EERM does not provide any evaluation metric for environmental discrepancy. To encourage the discrepancy, they "blindly" maximize the variance of the empirical risk in K environments. Finally, for generalization scope, EERM is based on the IRM [14] by minimizing the empirical risk in K environments. In contrast, inspired by DRO [31], we can guarantee the generalization within the robust radius . We summarize the above detailed discussions in Table 10.

H Limitation & Future Work

Although AIA outperforms numerous baselines and can achieve outstanding performance under various covariate shifts, we also prudently introspect the following limitations of our method. And we leave the improvements of these limitations as our future work.

- AIA performs OOD exploration through an adversarial data augmentation strategy to achieve environmental discrepancy. However, it only perturbs the existing graph data in a given training set, such as perturbing original graph node features or graph structures. Hence, it is possible that there still exist some overlaps between the augmented distribution and training distribution, so discrepancy principle cannot be thoroughly achieved. In future work, we will attempt to design more advanced data augmentation methods, such as graph generation-based strategies [88], to generate more unseen and novel graph data, for pursuing the discrepancy principle.
- For model training, we adopt adversarial training and stable feature learning to alternately optimize the adversarial augmenter, stable feature generator and backbone GNN. This training strategy may make the training process unstable, so the performance of AIA may experience a large variance. In addition, these two networks also involve additional parameters. Optimizing these parameters separately will also increase the time complexity, as shown in Appendix F. Hence, in future work, we will explore how to utilize more advanced optimization methods and lightweight models to achieve the principles of environmental feature discrepancy and stable feature consistency.